

Генеза

НОВА УКРАЇНЬСЬКА
ШКОЛА

ІНФОРМАТИКА

9
КЛАС



ЯК ЗМЕНШИТИ ЗАГРОЗИ ПІД

ЧАС РОБОТИ З КОМП'ЮТЕРОМ?



Використовуйте ліцензійне програмне забезпечення

Застосовуйте антивірусне програмне забезпечення



Проводьте резервне копіювання даних

Не відкривайте вкладення до листів невідомих кореспондентів



Не розміщуйте в інтернеті персональних даних



Не розміщуйте в інтернеті кодів доступу до фінансових документів



Дотримуйтеся етичних і правових норм, не робіть дій, що можуть нашкодити іншим користувачам



Повідомляйте дорослих про намагання отримати ваші персональні дані, дані про ваших рідних, про випадки залякування, приниження, переслідування вас або ваших знайомих



ІНФОРМАТИКА

Підручник для 9 класу
закладів загальної середньої освіти

*Рекомендовано
Міністерством освіти і науки України*



Київ
«Генеза»
2026

УДК 004(075.3)

I-74

Авторський колектив:

Йосиф Ривкінд, Тетяна Лисенко, Людмила Чернікова, Віктор Шакотько

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 27.01.2026 р. № 110)

Видано за рахунок державних коштів.

Продаж заборонено

Відповідно до модельної навчальної програми
«Інформатика. 7–9 класи» для закладів загальної середньої освіти
(автори: Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А., Шакотько В. В.)

Інтерактивний електронний додаток до підручника

<https://cutt.ly/xtS7I52>



Інтерактивні методичні матеріали до підручника

<https://faino.school/>



I-74 **Інформатика** : підруч. для 9-го кл. закл. заг. серед.
освіти / Й. Я. Ривкінд та ін. — Київ : Генеза, 2026. —
280 с. : іл.

ISBN 978-617-8468-42-2.

УДК 004(075.3)

ISBN 978-617-8468-42-2

© Ривкінд Й. Я., Лисенко Т. І.,
Чернікова Л. А., Шакотько В. В., 2026
© «Генеза», оригінал-макет, 2026

Дорогі дев'ятикласники та дев'ятикласниці! Шановні вчителі та вчительки!

Учні та учениці дев'ятого класу продовжуватимуть вивчення найсучаснішого шкільного предмета – **інформатики**.

Інформатика – це наука про інформаційні процеси, інформаційні системи та інформаційні ресурси, про комп'ютери та їх використання. У сучасному світі інформаційно-комунікаційні технології є важливим інструментом для досягнення успіху. Знання інформатики відкривають можливості для інновацій, для створення нових продуктів і послуг, що змінюють світ. Уміння доцільно й ефективно використовувати цифрові технології в навчальній діяльності та в повсякденному житті – одна з основних вимог до компетентностей, які мають бути сформовані в учнів та учениць **Нової української школи**.

Підручник, який підготував наш авторський колектив, сприятиме успішному вивченню інформатики. Він допоможе навчитися логічно мислити, обґрунтовувати власні дії та свій вибір цифрових засобів для розв'язування задач із різних галузей з використанням інформаційних технологій. А також бути креативними, розвивати навички вирішення проблем, генерувати власні ідеї та доброчесно використовувати чужі для створення нових інформаційних об'єктів.

На уроках інформатики в 9 класі ви дізнаєтеся або поглибите свої знання про інформаційні системи, інформаційні процеси в них, програмне забезпечення персонального комп'ютера, його інсталяцію і деінсталяцію, ліцензії на програмне забезпечення та інформаційну безпеку, особливості роботи із цифровими технологіями, створеними для користувачів з особливими освітніми потребами, зокрема осіб з інвалідністю. Удосконалите і розширите свої знання та навички опрацювання текстових документів, продовжите створювати проекти однією із сучасних мов програмування – **Python**. Ви дізнаєтеся, що таке *бази даних* і для чого їх використовують. Навчитесь створювати 3D-моделі в редакторі тривимірної графіки.

Набуті знання та вміння, сформовані компетентності ви застосовуватимете під час виконання навчальних проєктів на уроках інформатики та інших предметів.

Матеріал підручника поділено на сім розділів. Розділи складаються з пунктів, а кожен пункт – із підпунктів. Корисними та цікавими для вас будуть такі рубрики:



Пригадайте



Працюємо
з комп'ютером



Інтеграція
(поєднання)
наук



Для тих, хто
працює
з Android



Поміркуйте



Практична
робота



Працюємо
у групах /
у парах



Цікаві факти
з історії



Обговоріть
і зробіть
висновки



Найважливіше
в цьому
пункті



Дослідіть



Виконайте
завдання



Для тих, хто
хоче знати
більше



Дайте
відповіді на
запитання



Готуємось до
вивчення но-
вого матеріалу



Оцініть свої
знання
та вміння

Значну увагу в підручнику приділено практичній діяльності, у процесі якої й формуються відповідні інформатичні компетентності. Учні та учениці буде запропоновано обдумати й обговорити в парах або в невеликих групах відповіді на деякі запитання, виконати завдання зі створення чи вдосконалення певного інформаційного продукту, який може бути корисним у навчанні або побуті.

Детальніше зрозуміти, наскільки справдилися сподівання учнів та учениць щодо опанування навчального матеріалу, яких результатів досягнуто, вони зможуть, відповівши на запитання рубрики **Оцініть свої знання та вміння**, яку розміщено в кінці кожного розділу підручника. Ця рубрика містить запитання для самооцінювання досягнення обов'язкових результатів навчання відповідно до вимог Державного стандарту базової середньої освіти і модельної програми.

У підручнику описано дії з програмним забезпеченням на базі операційної системи **Windows 11**, якщо не зазначено іншу операційну систему.

Матеріали для виконання завдань, вправ, зокрема файли-заготовки, розміщено на сайті **Інформатика для всіх** за адресою <https://cutt.ly/YtSyxakK> або за QR-кодом.



Підручник також має електронний додаток, адреса якого так само розміщена на зазначеному сайті.

Бажаємо плідного та захопливого навчання!

Авторський колектив





ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

У цьому розділі ви отримаєте нові, а також поглибите і розширите свої знання й удосконалисте навички з таких тем:

- ▶ інформаційні системи, інформаційні процеси в них;
- ▶ програмне забезпечення персонального комп'ютера;
- ▶ ліцензії на програмне забезпечення, їхні типи;
- ▶ операційні системи;
- ▶ сумісність програмного забезпечення;
- ▶ драйвери;
- ▶ інсталяція та деінсталяція програмного забезпечення;
- ▶ особливості роботи із цифровими технологіями, створеними для користувачів з особливими освітніми потребами, зокрема осіб з інвалідністю.

1.1. ІНФОРМАЦІЙНІ СИСТЕМИ, ІНФОРМАЦІЙНІ ПРОЦЕСИ В НИХ. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА

У цьому пункті йтиметься про:

- ▶ інформаційні системи, інформаційні процеси в них;
- ▶ програмне забезпечення персонального комп'ютера;
- ▶ ліцензії на програмне забезпечення, їхні типи.

ІНФОРМАЦІЙНІ СИСТЕМИ



Пригадайте

● Що таке *інформаційний процес*? ● Які ви знаєте інформаційні процеси? ● Які засоби використовують для здійснення інформаційних процесів?

Усе життя людини пов'язано з інформаційними процесами. Для передавання та опрацювання повідомлень, їх зберігання використовують різноманітні технічні засоби та програмне забезпечення (ПЗ). Для підвищення ефективності здійснення інформаційних процесів засоби здійснення цих процесів об'єднують у системи.

Інформаційна система (ІС) – це організована сукупність технічних, програмних, інформаційних, організаційних і людських ресурсів, призначена для здійснення інформаційних процесів. Зазвичай інформаційні системи використовують для збирання, зберігання, аналізу та розповсюдження даних з метою надання допомоги під час прийняття рішень.

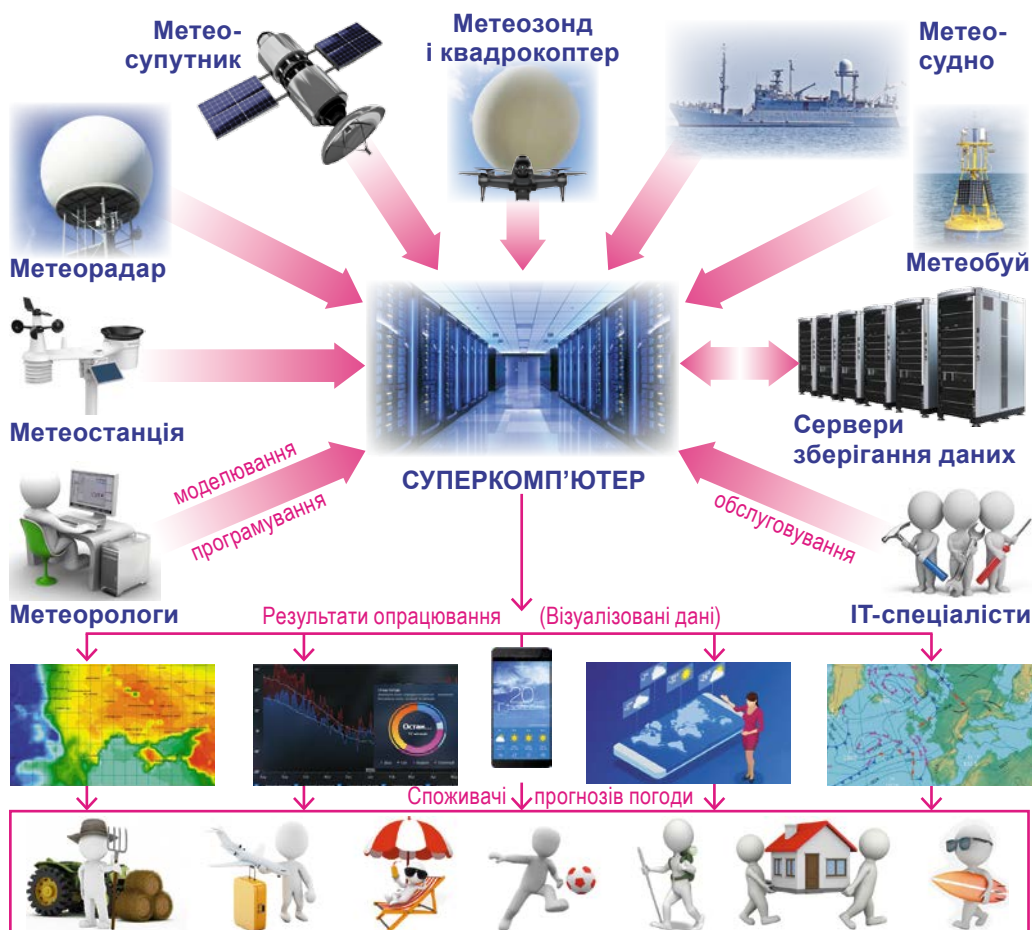
Розглянемо структуру та основні складові інформаційної системи на прикладі *Системи прогнозування погоди*.

Метою інформаційної системи прогнозування погоди є збирання, опрацювання (аналіз) метеорологічних даних для створення прогнозів погоди, попереджень про небезпечні явища та надання цих даних користувачам (населенню, авіаційним службам, працівникам сільського господарства та енергетики, рятувальним службам тощо).

Компонентами інформаційної системи прогнозування погоди є (мал. 1.1):

- **Дані (Information / Data):**

- **Вхідні дані.** Величезні обсяги первинних метеорологічних даних, що надходять у реальному часі. Дані збирають різними способами:
 - *наземні спостереження* з великої кількості метеостанцій, розміщених по всьому світу (значення температури, атмосферного тиску, вологості, швидкості та напрямку вітру, кількості опадів, дистанція видимості тощо);
 - *спостереження в атмосфері* – дані з метеозондів, що запускаються в атмосферу й вимірюють значення температури, тиску, вологості, швидкості та напрямку вітру на різних висотах над поверхнею Землі тощо;
 - *дані зі штучних метеорологічних супутників Землі* – про хмарність, температуру поверхні, рух повітряних мас тощо;



Мал. 1.1. Схема компонентів інформаційної системи прогнозування погоди

- *дані з інших джерел* – метеорологічних радарів, суден і метеорологічних буїв, літаків тощо (дані про опади та переміщення циклонів, температуру води, течії, хвилі, стан атмосфери на різних висотах).
- *Історичні та нормативно-довідкові дані* – дані за роки попередніх спостережень, моделі рельєфів, карти, стандарти в галузі метеорології тощо.
- **Апаратне забезпечення (Hardware):**
 - *суперкомп'ютери* – для виконання складних обчислень різноманітних моделей прогнозу погоди, які вимагають виконання трильйонів операцій за секунду;
 - *сервери зберігання даних* – для зберігання величезних обсягів вхідних, проміжних і вихідних даних;
 - *робочі станції* – високопродуктивні комп'ютери на робочих місцях метеорологів, оснащені спеціалізованим програмним забезпеченням для візуалізації та аналізу даних;
 - *мережне обладнання* – для швидкої передачі даних між станціями спостережень, обчислювальними центрами та користувачами;
 - *технічне обладнання для автоматичного збирання даних* – метеостанції, супутники, радари, метеозонди, буї тощо.
- **Програмне забезпечення (Software):**
 - *програмне забезпечення пристроїв фіксації даних*;
 - *числові (комп'ютерні) моделі* прогнозу погоди – програми, що моделюють атмосферні процеси на основі системи гідродинамічних рівнянь з великою кількістю невідомих;
 - *програмне забезпечення для візуалізації даних* – для створення карт, діаграм, анімації, ділової графіки;
 - *системи керування базами даних (СКБД)* – для упорядкованого зберігання і забезпечення зручного та швидкого доступу до метеорологічних даних;
 - *комп'ютерні програми для забезпечення комунікації та розповсюдження даних* про погодні умови (спеціальні складові операційних систем, вебсервіси, соціальні мережі, програми для мобільних пристроїв, системи оповіщення тощо).
- **Працівники та працівниці метеорологічних установ / користувачі та користувачки (People):**
 - *науковці-метеорологи*, які створюють і вдосконалюють моделі та методи прогнозування погоди;
 - *фахівці (прогнозисти)*, які аналізують результати опрацювання даних за відповідними моделями і за потреби коригують прогнози, надані комп'ютерними системами, а також формують тексти повідомлень про погоду, попереджень про стихійні лиха;
 - *ІТ-спеціалісти*, які здійснюють обслуговування апаратного та програмного забезпечення інформаційної системи;
 - *кінцеві користувачі*: пересічні громадяни, фермери, авіадиспетчери, моряки, рятувальники, енергетики, будівельники та інші, які використовують наданий прогноз погоди для особистих потреб.

Система прогнозування погоди є прикладом складної інформаційної системи, що демонструє взаємодію всіх її компонентів для досягнення конкретної та важливої мети.



Інформаційними системами, але вже з іншими завданнями, є, наприклад, система керування польотами літаків, служба продажу квитків на транспорті (залізниця, автобусне та авіасполучення), бібліотека, служба новин на телебаченні, довідкова система Верховної Ради, система онлайн-замовлень, інформаційна система банку тощо. Слід зауважити, що інформаційні системи існували давно, тільки інформаційні процеси в них здійснювалися або вручну, або з використанням механічних пристроїв. Застосування комп'ютерної техніки, і особливо комп'ютерних мереж, значно пришвидшило інформаційні процеси в інформаційних системах.

Спрощену узагальнену схему інформаційної системи наведено на малюнку 1.2.



Мал. 1.2. Узагальнена схема інформаційної системи

Одним з перших у світі впроваджувати комп'ютеризовані інформаційні системи почав Інститут кібернетики Академії наук України. Під керівництвом його директора, академіка Віктора Михайловича Глушкова (1923–1982) (мал. 1.3), було розроблено автоматичні системи управління виробництвом і впроваджено їх на ряді металургійних комбінатів України, на Львівському телевізійному заводі «Електрон» та інших підприємствах.



Мал. 1.3. Віктор Михайлович Глушков

Глушков В. М. говорив: «Навряд чи можна сумніватися, що в майбутньому значна частина закономірностей навколишнього світу буде пізнаватися та використовуватися автоматичними помічниками людини. Але настільки ж безсумнівно й те, що все найважливіше у процесах мислення та пізнання завжди належатиме людині».

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА



Пригадайте

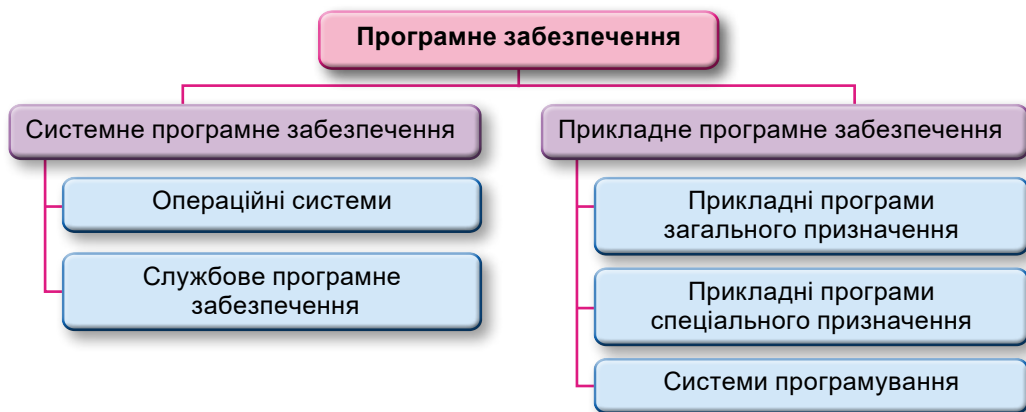
- Що таке комп'ютерна програма? Хто їх створює?
- Для яких цілей ви використовували комп'ютерні програми?

Ви вже знаєте, що є різноманітні програми для комп'ютерів. Є програми для опрацювання текстових, графічних, мультимедійних даних, є програми для керування роботою комп'ютера, його окремими

пристроями, такими як принтер, сканер, монітор. Є програми для забезпечення комунікацій і пошуку даних, для навчання, розваг і розвитку та інші.

Сукупність програм для комп'ютерів називають **програмним забезпеченням** (ПЗ). Тільки під керуванням програмного забезпечення комп'ютер здатен опрацювати, зберігати і передавати різноманітні дані.

Сучасне програмне забезпечення можна поділити на види (мал. 1.4).



Мал. 1.4. Класифікація програмного забезпечення

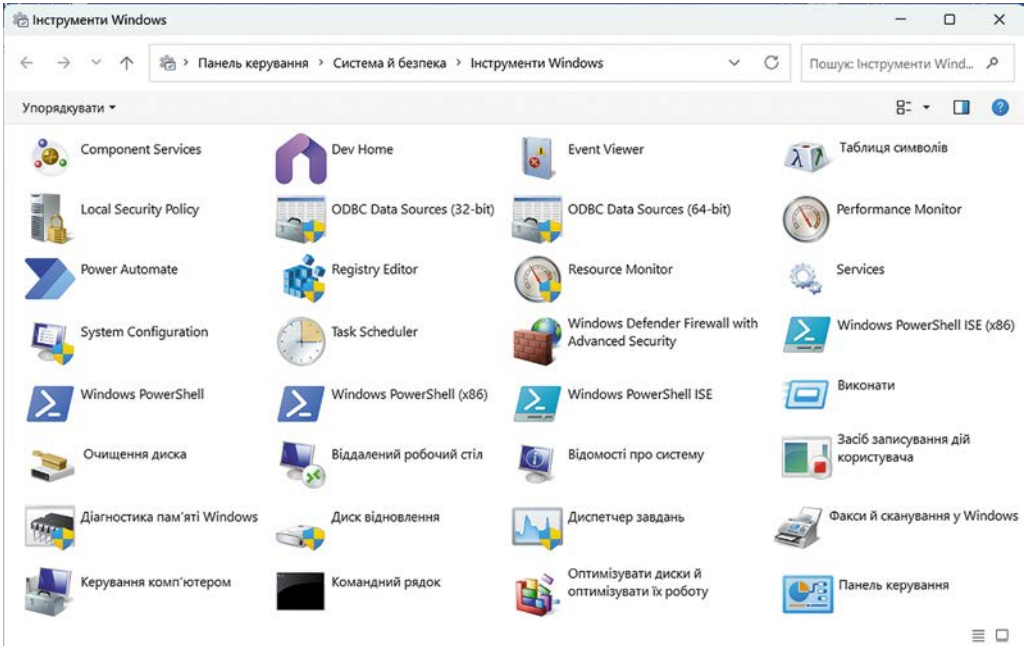
Системне програмне забезпечення призначено для:

- керування роботою складових комп'ютера;
- забезпечення обміну даними між пристроями комп'ютера;
- діагностування та усунення недоліків у роботі комп'ютера;
- автоматизації процесу опрацювання даних;
- організації обміну даними між користувачем і комп'ютером тощо.

До складу системного програмного забезпечення належать **операційні системи** та **службові (сервісні) програми**. Прикладами службових програм є програми, які ви вивчали в попередніх класах: програми резервного копіювання, програми створення й опрацювання архівів даних, антивірусні програми. Доступ до значної кількості службових програм у **Windows 11** можна отримати, виконавши **Пуск** ⇒ **Усі** ⇒ **Інструменти Windows**. Перелік цих програм можна побачити у вікні **Інструменти Windows** (мал. 1.5).

Прикладне програмне забезпечення призначено для реалізації конкретних завдань опрацювання даних, які користувач здійснює під час своєї професійної (навчальної) та побутової діяльності. Його поділяють на прикладні програми **загального** і **спеціального** призначення та **системи програмування**.

До прикладних програм **загального призначення** належать програми, які можуть застосовуватися в різних галузях людської діяльності для опрацювання текстів, графічних зображень, мультимедійних даних, числових даних в електронних таблицях, створення презентацій, баз даних тощо.



Мал. 1.5. Вікно Інструменти Windows

Прикладні програми **спеціального призначення** використовуються для реалізації специфічних завдань опрацювання даних у певній галузі діяльності, на конкретному підприємстві, в організації, фірмі або їхньому підрозділі. До такого типу програм належать програми для фінансових і бухгалтерських обчислень, створення відеоефектів для кінофільмів, креслень машин та механізмів у конструкторських і проектних бюро, діагностування захворювань у медичних закладах, складання шкільного розкладу уроків тощо (мал. 1.6).

Системи програмування призначено для створення нових програм з використанням мов програмування. Такими програмами, наприклад, є: **Scratch, Lazarus, IDLE, DEV-C++, Microsoft Visual Studio Professional, Android Studio** та інші. У деяких класифікаціях *системи програмування* виділяють в окремий вид програмного забезпечення поряд із *системним* і *прикладним* програмним забезпеченням.



Мал. 1.6. Використання прикладних програм у різних галузях

ЛІЦЕНЗІЇ НА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ЇХНІ ТИПИ



Пригадайте

- Що таке *авторське право*? Як захистити авторське право розробників програмного забезпечення?
- Які ліцензії та на які вироби чи види діяльності ви використовували?

На комп'ютерні програми поширюється дія Закону України «Про авторське право і суміжні права». Їх використання можливе лише за умов дотримання вимог цього закону, а також вимог **ліцензії**, з якою користувач погоджується, установлюючи програму на свій комп'ютер.

Використання ПЗ є законним тільки тоді, коли на це є дозвіл власника авторських прав (компанії-виробника або розробника програми). Такий дозвіл оформлюється у вигляді **ліцензії** (ліцензійної угоди) (лат. *licentia* – свобода, влада, право здійснювати щось). Ліцензія визначає взаємовідносини між користувачем і правовласником програми.

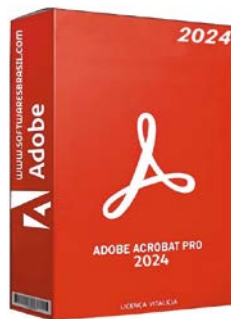
Є кілька видів ліцензій на комп'ютерні програми:

- **комерційна (пропрієтарна) ліцензія** – передбачає, що користувач оплачує вартість використання певної програми на одному чи кількох зазначених у ліцензії комп'ютерах.

Пропрієтарний – від лат. *Proprius* – власний.

Купівля такої ліцензії надає користувачеві право використовувати програму, але обмежує його права на модифікацію, розповсюдження та часто навіть на копіювання. Серед комерційних ліцензій розрізняють:

- «**коробкові**» *версії* ліцензії продаються зазвичай для одного екземпляра програми, можуть містити носій, на якому записано програму, та інструкцію щодо її використання разом із ключем для встановлення (мал. 1.7). Може бути перенесена на інший комп'ютер, але потрібно видалити цю програму на попередньому. Наприклад: **Adobe Acrobat 2024, 3D CAD5;**
- **OEM** (англ. *Original Equipment Manufacturer* – оригінальний виробник обладнання) – ліцензія, що надається разом з новим обладнанням, наприклад з новим ноутбуком. Прив'язана до обладнання, не може бути перенесена на інший пристрій. Підтвердженням ліцензії є спеціальна наклейка (мал. 1.8). Наприклад: **MS Windows 11 OEM, OEM NORTON SECURITY DELUXE 3.0;**



Мал. 1.7. Коробкова версія програми **Adobe Acrobat**



Мал. 1.8. Наклейка OEM-версії ліцензії операційної системи

- *корпоративна ліцензія* – ліцензія на кілька копій програми для використання, наприклад на комп'ютерах закладів освіти. До таких ліцензій можна віднести ліцензії на програми **MS Office 365**, **Антивірус для Бізнесу Zillya!** та інші;
- *підписка (на програмне забезпечення)* – надання права використання програмного забезпечення протягом певного періоду, зазвичай протягом місяця або року, за періодичну плату. Наприклад: **Adobe Creative Cloud**, **AWAX**.

- *Пробна*, або *trial* (англ. *trial* – випробування), – ліцензія на комерційне ПЗ, яка надається для пробного використання програми протягом певного часу (зазвичай приблизно на місяць) або на певну кількість запусків програм. Іншим варіантом програм, що розповсюджуються за цією ліцензією, є програми, що не мають обмежень за часом використання, але в них заблоковані або відсутні певні операції, наприклад операції зі збереження створених інформаційних продуктів у файли певних форматів;
- *вільного використання*, або *freeware* (англ. *freeware* – вільний товар), – ліцензія передбачає вільне використання програм без виплат винагороди автору, але не передбачає можливості внесення змін у програму, наприклад **CCleaner**, **Opera**;
- *з відкритим кодом*, або *free* (англ. *free* – вільне) *software*, або *libre* (ісп. *libre* – вільне) *software*, – ліцензія на вільне програмне забезпечення, що передбачає не тільки безкоштовне використання програм, а й право на їх модифікацію, внесення змін у код програми. Це зазвичай ліцензія **GNU General Public License**. Прикладами таких програм є **GIMP**, **OpenOffice.org**;
- *з рекламою* певних продуктів або послуг – *ad-supported* (англ. *advertisement* – оголошення, реклама; *support* – підтримка) ліцензія передбачає вільне використання програм без виплат винагороди автору, але з уключенням реклами. Версія програми з відключеною рекламою передбачає додаткову оплату. Прикладом такої програми є програма комунікації **Viber**, яка містить рекламу й для її відключення потрібно сплатити певну суму.



Для тих, хто хоче знати більше

Фонд вільного програмного забезпечення (англ. **FSF – Free Software Foundation**) – це об'єднання розробників вільного програмного забезпечення. Його створено 1985 року, і він займається юридичним захистом розробників ПЗ, а також розробкою та підтримкою нормативних актів, що визначають рух вільного програмного забезпечення, дотримання ліцензії **GNU General Public License** та інших. FSF також підтримує низку проєктів зі спільної розробки програмних продуктів. Фондом також сформульовані чотири фундаментальні свободи вільного програмного забезпечення:

1. Свобода запускати програму з будь-якою метою (свобода 0).
2. Свобода вивчати, як працює програма, й адаптувати її до своїх потреб (свобода 1). Доступ до вихідного коду є обов'язковою умовою.
3. Свобода розповсюджувати копії, щоб допомогти своєму ближньому (свобода 2).
4. Свобода покращувати програму та випускати покращення публічно, щоб уся спільнота могла скористатися перевагами (свобода 3). Доступ до вихідного коду є обов'язковою умовою.



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання 1. Дослідіть інформаційну систему установи.

1. Оберіть одну з установ, наприклад:

-  Школа;
-  Магазин;
-  Поліклініка;
-  Автовокзал;
-  Банк;
-  Транспортна компанія.

2. Уявіть, що ви – фахівець, який має пояснити, як у цій організації працює інформаційна система. Підготуйте відповіді на запитання:

- а) *Які дані зберігаються в системі?* (Наприклад, розклад уроків, список учнів, оцінки або товари, ціни, кількість на складі.)
- б) *Хто користується цією системою?* (Учні, вчителі, касири, покупці тощо.)
- в) *Для яких цілей створена інформаційна система цієї установи?* (Щоб зберігати інформацію, швидко знаходити дані, оформлювати покупки, рахувати оцінки тощо.)

Для запису відповідей на запитання створіть у текстовому процесорі документ, у який уведіть запитання та відповіді на них.

3. Збережіть створений документ у вашій папці у файлі з іменем **вправа 1.1.**

4. Складіть у створеному текстовому документі таблицю, у яку введіть відповідні дані.

Компонент ІС	Приклади компонентів у обраній установі
Апаратне забезпечення	
Програмне забезпечення	
Інформаційне забезпечення (дані)	
Користувачі	

5. Збережіть змінений документ у тому самому файлі.

6. Створіть у цьому документі схему обміну даними між користувачами й інформаційною системою установи. Для побудови схеми використовуйте засоби створення графічних зображень текстового процесора.

7. Збережіть змінений документ у тому самому файлі.

Завдання 2. Дослідіть програмне забезпечення комп'ютера (у комп'ютерному класі або вдома).

1. Відкрийте меню **Пуск** на комп'ютері.
2. Перегляньте перелік установлених програм.
3. За наведеним зразком створіть у текстовому процесорі таблицю, у яку введіть дані про вісім програм різного призначення.

Назва програми	Тип програмного забезпечення	Для чого використовується
Windows 11	Операційна система	Для забезпечення роботи комп'ютера
Google Chrome		
		Для створення графічних зображень
Безпека у Windows		
	Службове	
		Для роботи з архівами даних

4. Збережіть створений документ у вашій папці у файлі з іменем **вправа 1.2.**

Завдання 3. Визначте тип ліцензії.

- Відкрийте перелік програм, установлених на вашому комп'ютері, виконавши **Пуск** $\Rightarrow$ **Усі**.
- Виберіть у цьому переліку чотири програми, що мають різні ліцензії. Про тип ліцензії можна дізнатися, виконавши відповідний запит в одній із пошукових систем або зробивши запит одному із сервісів ІІІ, наприклад, такий: «Яку ліцензію має програма 7Z?»
- За результатами пошуку заповніть таблицю.

Програма	Тип ліцензії	Чому ти так вважаєш?
Microsoft Windows	ОЕМ комерційна	Є відповідна наклейка на корпусі ноутбука

4. Збережіть створений документ у вашій папці у файлі з іменем **вправа 1.3.**



Найважливіше в цьому пункті

Інформаційна система – це організована сукупність технічних, програмних, інформаційних, організаційних і людських ресурсів, призначена для здійснення інформаційних процесів. Зазвичай інформаційні системи використовують для збирання, зберігання, аналізу та розповсюдження даних з метою надання допомоги під час прийняття рішень.

Основними складовими комп'ютеризованих інформаційних систем є технічні пристрої, що використовуються для передавання, зберігання та опрацювання повідомлень, та комп'ютерні програми, під керуванням яких здійснюються всі названі інформаційні процеси.

Одним з перших у світі впроваджувати комп'ютеризовані інформаційні системи почав Інститут кібернетики Академії наук України. Під керівництвом його директора, академіка Віктора Михайловича Глушкова, було розроблено автоматичні системи управління виробництвом і впроваджено їх на низці підприємств України.

Сукупність програм для комп'ютерів називають **програмним забезпеченням** (скорочено **ПЗ**). Його можна розподілити на такі види: *системне програмне забезпечення* та *прикладне програмне забезпечення*. Останній вид поділяють на прикладні програми **загального** і **спеціального** призначення та **системи програмування**.

До складу **системного програмного забезпечення**, крім операційних систем, входять й інші програми, їх ще називають *утилітами*. **Утиліти** – це програми, які призначені для діагностування апаратної та програмної складових комп'ютера, розширення можливостей ОС.

Є кілька видів ліцензій на комп'ютерні програми: *комерційна ліцензія* («коробкова», OEM, корпоративна, підписка на програмне забезпечення), *пробна* (trial), *вільного використання* (freeware), *з відкритим кодом* (free software), *з рекламою* (ad-supported).



Дайте відповіді на запитання

1. Що таке *інформаційна система*? Назвіть складові інформаційної системи.
2. Які службові програми ви використовували в роботі з комп'ютером? З якою метою?
3. Які програми належать до систем програмування? Які з них ви використовували? Наведіть приклади.
4. Які з вивчених вами раніше програм можна віднести до прикладних програм? Які з них є прикладними програмами загального призначення?
5. Для яких цілей використовують прикладні програми спеціального призначення? Наведіть приклади таких програм.
6. Які існують види ліцензій для комп'ютерних програм? Чим вони відрізняються?



Обговоріть і зробіть висновки

1. Які моральні та етичні питання можуть виникнути під час застосування такої інформаційної системи, яка використовує штучний інтелект для прийняття важливих рішень (наприклад, здійснює оцінювання навчальних досягнень учнів / учениць та оцінювання результатів роботи учителів / учительок)?
2. Навіщо потрібно системне програмне забезпечення?
3. Уявіть, що ви обираєте програмне забезпечення для нового комп'ютера. Які п'ять програм ви встановите найперше і чому?
4. Які три прикладні програми ви використовуєте найчастіше? Які завдання ви виконуєте з використанням кожної з них?
5. Який вид ліцензії операційної системи використовує ваш заклад освіти для комп'ютерів у комп'ютерному класі?



Виконайте завдання

1. Уявіть себе в ролі конструктора «розумного будинку», який повинен мати власну інформаційну систему. Створіть у текстовому процесорі новий документ і запишіть у нього відповіді на запитання:
 - а) Які пристрої (апаратне забезпечення) будуть збирати інформацію (наприклад, про температуру, світло, рух)?
 - б) Які рішення будуть прийматися системою на основі отриманих даних?
 - в) Як ця система може покращити життя мешканців, зробити його безпечнішим та економнішим?

Збережіть створений документ у вашій папці у файлі з іменем **завдання 1.1.**

2. Порівняйте інформаційну систему людини (мозок + органи чуття) і сучасний комп'ютер як інформаційну систему. У чому вони схожі, а в чому – різні? За результатами порівняння підготуйте комп'ютерну презентацію та збережіть її у файлі з іменем **завдання 1.2.**

3. Визначте засоби, що використовуються в компонентах інформаційної системи сучасної шкільної бібліотеки. Результати вашої роботи введіть у таблицю, яку створіть у текстовому документі за зразком:

<i>Компоненти</i>	<i>Приклади засобів</i>
Апаратне забезпечення	
Програмне забезпечення	
Інформаційне забезпечення (дані)	
Персонал, який обслуговує ІС	
Користувачі	

Збережіть створений документ у вашій папці у файлі з іменем **завдання 1.3.**

4. Дослідіть, до якого типу належать наведені програми, та заповніть таблицю, яку створіть у текстовому документі.

<i>Назва програми</i>	<i>Вид програмного забезпечення</i>	<i>Призначення програми</i>
Android		
Google Chrome		
Microsoft PowerPoint		
Антивірус Zillya!		
Linux Ubuntu		
Paint		
Media Player Classic		

Збережіть створений документ у вашій папці у файлі з іменем **завдання 1.4.**

5. Укажіть приклади програм, які використовують для виконання відповідних завдань. Також укажіть, до якого виду ПЗ вони належать (системне чи прикладне). Для виконання завдання створіть у текстовому документі таблицю за зразком:

<i>Завдання, яке потрібно виконати</i>	<i>Приклад програми</i>	<i>Пояснення, чому ця програма підходить</i>	<i>Вид програмного забезпечення</i>
Написати шкільне есе	Microsoft Word	У цій програмі можна створювати текстові документи	Прикладне загального призначення

Продовження таблиці

Завдання, яке потрібно виконати	Приклад програми	Пояснення, чому ця програма підходить	Вид програмного забезпечення
Знайти в інтернеті інформацію для проєкту			
Деінсталювати програму			
Подивитися улюблений мультфільм або фільм			
Створити презентацію до уроку			
Поспілкуватися з друзями онлайн			
Перевірити на наявність вірусів флешносій			

Збережіть створений документ у вашій папці у файлі з іменем **завдання 1.5**.

6. Підготуйте в текстовому процесорі схему класифікації видів ліцензій програмного забезпечення. Збережіть її у файлі **завдання 1.6** у папці **Документи** і надішліть на електронну пошту вчителю / учительці.



Готуємось до вивчення нового матеріалу

1. Що таке *операційна система*?
2. Що таке *драйвер*? Яка його роль у забезпеченні роботи комп'ютера?
3. Як установити певну програму на ваш комп'ютер?



1.2. ОПЕРАЦІЙНІ СИСТЕМИ. СУМІСНІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. ДРАЙВЕРИ

У цьому пункті йтиметься про:

- ▶ операційні системи та їхні класифікації;
- ▶ інсталяцію та деінсталяцію програмного забезпечення;
- ▶ сумісність програмного забезпечення;
- ▶ драйвери.

КЛАСИФІКАЦІЯ ТА ОСНОВНІ ФУНКЦІЇ ОПЕРАЦІЙНИХ СИСТЕМ



Пригадайте

- Що таке *операційна система*? До якого типу програмного забезпечення вона належить?
- Які операційні системи використовують у мобільних пристроях?

Як ви вже знаєте, існує багато різноманітних *операційних систем* (ОС). Операційні системи призначені для:

- керування роботою пристроїв комп'ютера;
- забезпечення обміну даними між пристроями комп'ютера;

- зберігання даних в оперативній пам'яті та на зовнішніх носіях;
- забезпечення виконання інших програм;
- організації обміну даними між користувачем і комп'ютером.

ОС організовує функціонування всього службового програмного забезпечення. Основними складовими операційної системи є:

- **базова система введення / виведення – BIOS** – незалежний від конкретної версії операційної системи набір базових команд, які використовуються для забезпечення обміну даними між пристроями і зберігаються в мікросхемі постійної пам'яті;
- **ядро операційної системи** – набір програм, з використанням яких виконуються команди, розподіляються ресурси пам'яті та каналів зв'язку між пристроями й програмами, надаються розширені функції з керування пристроями комп'ютера та інше;
- **файлова система** – визначає структуру збереження даних на зовнішніх носіях та сукупність програм, які забезпечують роботу із цією структурою. Зазвичай операційна система може працювати з кількома файловими системами: **NTFS, FAT16** та іншими.
- **драйвери пристроїв** (англ. *driver* – водій, той, хто керує пристроєм) – програми, які забезпечують обмін даними між операційною системою та конкретною моделлю пристрою;
- **інтерфейс користувача** (англ. *interface* – засоби узгодження) – сукупність засобів, які забезпечують обмін даними між користувачем та ОС.

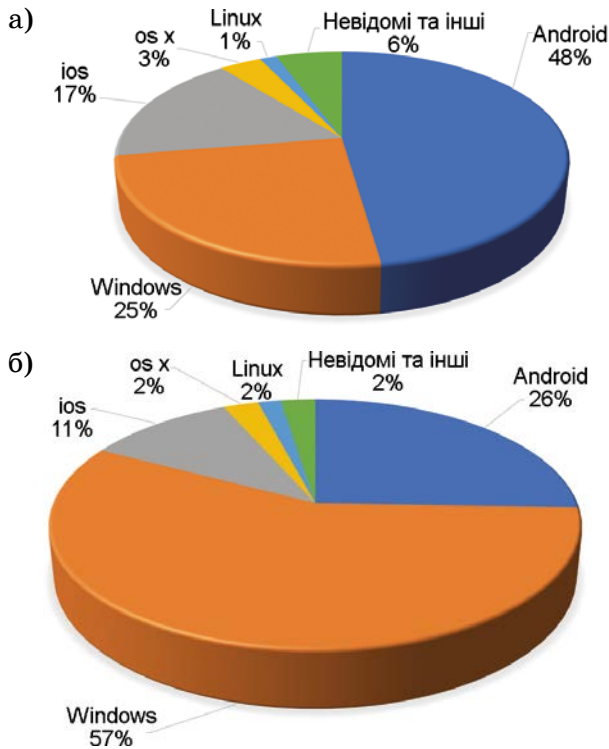
Залежно від призначення їх поділяють на операційні системи для:

- **стаціонарних персональних комп'ютерів (Windows 11, macOS (OS X), Linux, Chrome OS та ін.);**
- **мобільних комп'ютерів, у тому числі смартфонів (Android, iOS, KaiOS, Samsung, Windows, Tizen та ін.);**
- **серверів і суперкомп'ютерів (Unix, Linux, Windows Server, Solaris та ін.);**
- **різноманітних вбудованих комп'ютерів у автомобілі, літаки, пристрої побутової техніки тощо (FreeRTOS, ADAS, Embedded Linux, OpenWrt VxWorks та ін.).**

Статистичні дані щодо використання операційних систем для персональних комп'ютерів (уключаючи мобільні пристрої) у світі та в Україні станом на червень 2025 року, за даними сайту Statcounter (<https://gs.statcounter.com>), наведено на малюнку 1.9. Найбільш розповсюдженими операційними системами у світі є різні версії ОС **Android**, що зумовлено інтенсивним використанням мобільних пристроїв. В Україні перше місце серед операційних систем залишилося за різними версіями ОС **Windows**.

Також операційні системи можна поділити за типом ліцензії, що використовується під час їх розповсюдження. Так, операційні системи **Windows** мають комерційну ліцензію, проте окремі з них можуть використовувати ліцензію *trial*. До операційних систем, що розповсюджуються за ліцензією *free software*, у першу чергу належать різні версії операційних систем **Android** та **Linux**.

Операційні системи розрізняють також за розрядністю, тобто за тим, яка розрядність центральних процесорів у тих комп'ютерах, на які буде встановлена ця ОС. На сьогодні розрізняють ОС для



Мал. 1.9. Розподіл використання операційних систем у персональних комп'ютерах у світі (а) та в Україні (б)

32-розрядних і 64-розрядних комп'ютерних систем. Центральні процесори із 32-розрядною архітектурою в останні роки майже не випускаються, тому й відповідні операційні системи на нові комп'ютери практично не встановлюються.

ПОНЯТТЯ ІНСТАЛЯЦІЇ ТА ДЕІНСТАЛЯЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Пригадайте

- Як додати нову програму до програмного забезпечення комп'ютера?
- Як завантажити певну програму з інтернету?

Для встановлення ОС на комп'ютері потрібно виконати спеціальну операцію, яка називається **інсталяцією операційної системи** (англ. *installation* – установлення, розміщення).

Для інсталяції операційної системи потрібно мати інсталяційний пакет програм – **дистрибутив** (англ. *distribute* – розподіляти). Його можна завантажити з інтернету або скопіювати з використанням флешносія. Після запуску файлу встановлення операційної системи зазвичай буде зроблено запит на дозвіл для інсталяції. На наступних кроках пропонується ввести ряд додаткових даних: диск, на який буде встановлено ОС, мову і набір регіональних стандартів тощо. Існують також інші варіанти здійснення інсталяції, наприклад з використанням локальної мережі або інтернету.

Під час інсталяції відбувається розміщення складових ОС на вибраному диску, налагодження її взаємодії з апаратною складовою комп'ютера. Диск, на який встановлено операційну систему, називають **системним**.

32-розрядні операційні системи, можуть встановлюватися як на 32-розрядні, так і на 64-розрядні комп'ютерні системи. У той самий час 64-розрядна ОС не може бути встановлена на комп'ютер із 32-розрядним процесором.

Будь-яке програмне забезпечення, що використовується в сучасних комп'ютерах, працює під управлінням операційної системи. Кожна з програм створюється для певної операційної системи і не може працювати з іншою. Для того щоб якась із програм могла працювати з іншою ОС, потрібно її код адаптувати під іншу операційну систему.

Процес встановлення певної програми в певній операційній системі також називається інсталяцією, тільки **інсталяцією програми**. Під час інсталяції програми відбувається процес її узгодження з операційною системою, занесення даних про програму до службових програм (наприклад, у ОС **Windows** у файл *реєстру* ОС) та розміщення складових програми в певних папках.

Цей процес для різних операційних систем та різних програм може суттєво відрізнятись. Зазвичай він здійснюється відповідно до алгоритму, записаного у спеціальній програмі, наприклад, *setup.exe*, *install.exe* (для **Windows**) або *INSTALL.sh* чи файлів з розширенням *deb* (для **Linux**).

Складові програми, що інстальються, можуть надаватися користувачеві на відповідному носіїві даних разом із програмами інсталяції, а можуть копіюватися із сайтів розробників у процесі інсталяції. У ході інсталяції зазвичай здійснюється обмін даними між програмою інсталяції та користувачем для уточнення окремих значень властивостей встановлення програми (диск і папка для розміщення програми, встановлення окремих компонентів, мова інтерфейсу тощо).

Слід зауважити, що для окремих програм у певних операційних системах (наприклад, **macOS 1-9**) інсталяція не здійснюється. Програма запускається безпосередньо із зовнішнього носія даних.

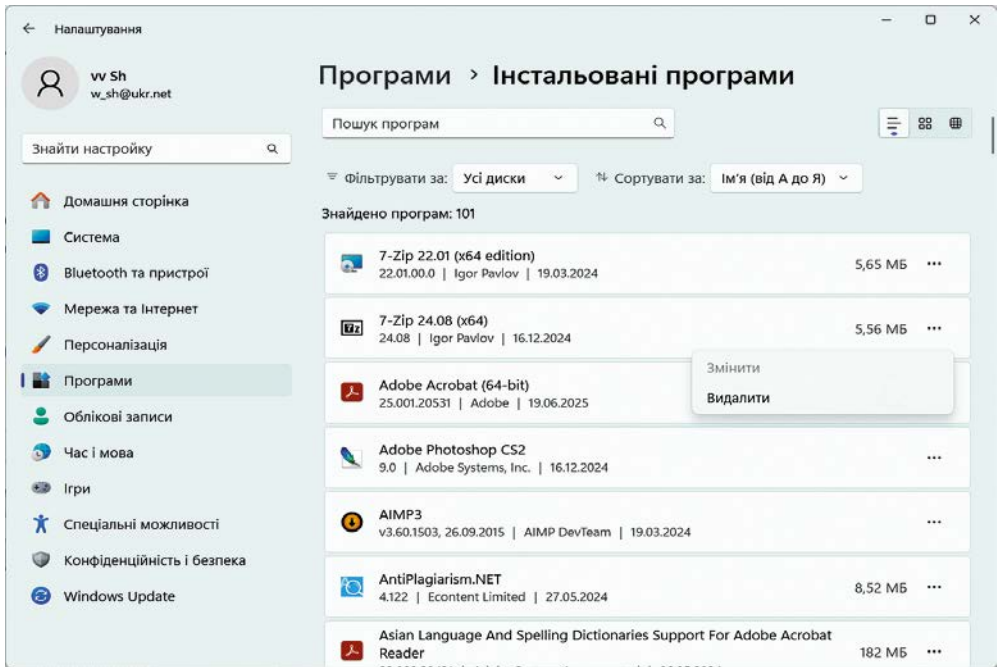
Якщо інсталяція програми відбулася успішно, то команда для запуску програми буде додана:

- для **Windows** – до меню **Пуск** і до списку вже встановлених програм;
- для **Linux Ubuntu** – до **Панелі запуску** та до **Центру програмного забезпечення**.

Якщо користувач не планує надалі використовувати встановлену програму, то варто провести спеціальну операцію – **деінсталяцію** програми. Бо якщо просто видалити папку з файлами програми або видалити ярлик програми з меню **Пуск**, то це призведе до сповільнення роботи операційної системи та засмічення диска файлами, які не використовуються.

Деінсталяція програм в ОС **Windows** здійснюється з використанням інструментів вкладки **Програми та функції** вікна **Налаштування** (**Пуск** ⇒ **Налаштування** ⇒ **Програми** ⇒ **Інстальовані програми**).

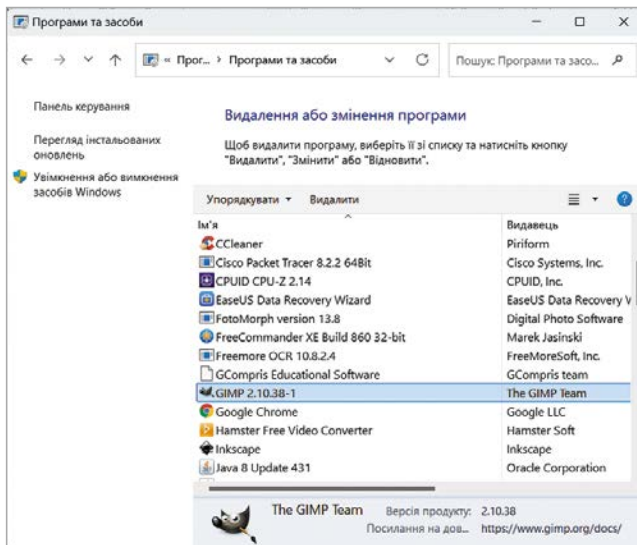
Потрібно у списку встановлених програм вибрати програму, яку слід видалити, після чого – кнопку **Додаткові параметри ...** та команду **Видалити** (мал. 1.10).



Мал. 1.10. Вікно Програми > Інстальовані програми

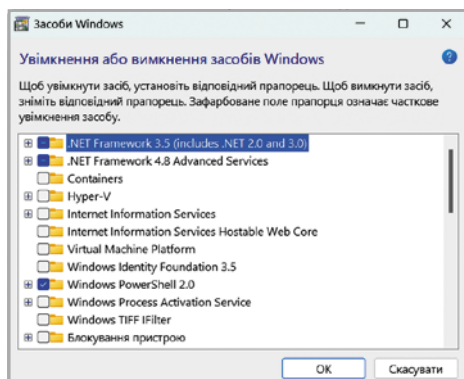
Вибір операції видалення запускає програму, яка коректно видаляє всі компоненти програми з комп'ютера та видаляє записи про цю програму у відповідних службових програмах **Windows**.

Деінсталяцію програм можна також виконати, використовуючи компонент **Програми та засоби**. Для його відкриття слід виконати **Пуск** ⇒ **Усі** ⇒ **Інструменти Windows** ⇒ **Панель керування** ⇒ **Програми**. Зображення вікна **Програми та засоби** подано на малюнку 1.11.



Мал. 1.11. Вікно Програми та засоби

Для деінсталяції програми слід вибрати потрібну програму в списку та кнопку **Видалити**.



Мал. 1.12. Вікно **Засоби Windows**

У вікні **Програми та засоби** користувач також може переглянути перелік установлених компонентів **Windows** та інстальювати / деінстальювати певні компоненти операційної системи. Для цього слід вибрати посилання  **Увімкнення або вимкнення засобів Windows**.

Увімкнені інстальовані компоненти операційної системи можна розрізнити за встановленою позначкою прапорця (мал. 1.12). Для встановлення певного компонента слід поставити відповідну позначку прапорця та вибрати кнопку **ОК**.

СУМІСНІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



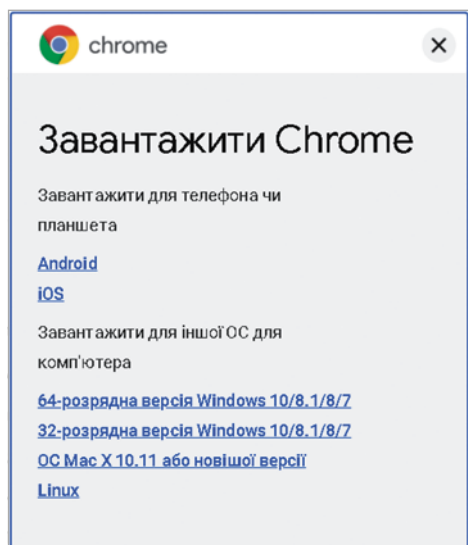
Пригадайте

• Чому програми для **Android** не працюють у **Windows**? • Що таке *інсталяція / деінсталяція* програми в операційній системі? Для чого їх використовують?

Як ви вже знаєте, комп'ютерні програми створюються для певної операційної системи відповідно до стандартів, які встановлюють розробники ОС. Програми, розроблені для одного типу операційних систем, зазвичай є несумісними з іншими типами або версіями операційних систем і не можуть бути інстальованими. Так програми, розроблені для ОС **Windows**, не можуть працювати в ОС сімейства **Linux** або **Android**.

Для кожної з операційних систем створюється своя версія програми. Так існують, наприклад, браузери **Chrome** для **Windows**, **Linux**, **Android** тощо (мал. 1.13).

Певне програмне забезпечення не завжди коректно працює навіть під управлінням операційних систем одного сімейства, але різних його версій. Так більшість програм для **Windows 10** не зможуть працювати з **Windows XP**. Установлюючи певне програмне забезпечення, потрібно правильно обирати відповідну ОС, наприклад із малюнка 1.13 зрозуміло, що є версія **Chrome**, яка буде коректно працювати на всіх версіях **Windows**, починаючи з **Windows 7**.



Мал. 1.13. Фрагмент вебсторінки завантаження браузера **Google Chrome**

СЛУЖБОВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ. ДРАЙВЕРИ



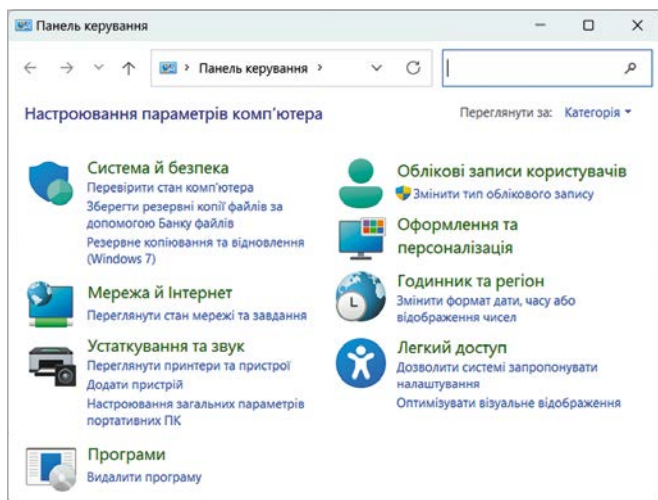
Пригадайте

- Які програми належать до службового програмного забезпечення?
- Що таке *драйвер принтера*?

Як було вже зазначено, до складу **службового програмного забезпечення** входять й інші програми, їх ще називають **утилітами** (англ. *utility* – корисність). **Утиліти** – це програми, які призначені для діагностування апаратної та програмної складових комп'ютера, розширення можливостей ОС. За потреби вони усувають недоліки та оптимізують роботу комп'ютера. Частина таких програм включається до складу операційної системи під час інсталяції ОС. Наприклад, до складу операційної системи **Windows 11** входять службові програми: **Диспетчер завдань** ,

Оптимізувати диски і оптимізувати їх роботу , **Registry Editor** (англ. редактор реєстра) ,

System Configuration (англ. конфігурація системи)  та інші. Базові утиліти для налаштування ОС **Windows** під потреби користувача доступні у вікні **Панель керування** (**Пуск** ⇒ **Усі** ⇒ **Інструменти Windows** ⇒ **Панель керування**) (мал. 1.14).



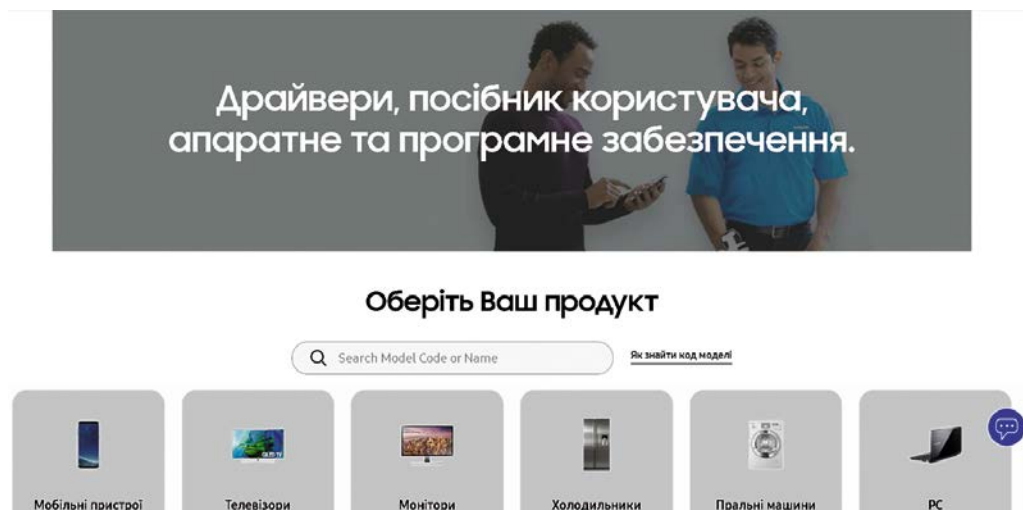
Мал. 1.14. Вікно **Панель керування**

Аналогічні програми є і в інших операційних системах. Так у **Linux Ubuntu** такими програмами, наприклад, є **Менеджер архівів**, **Журнал системи**, **Аналізатор використання диска**, **Системний монітор**, **System Testing**, **Диски**, а в ОС **Android** – **Диспетчер файлів**, **Календар**, **Диск**, **Android Auto**, **Налаштування** та інші.

Додаткові службові програми розробників певних операційних систем, а також сторонніх розробників можна придбати або вільно скопіювати із сайтів та інсталювати для подальшого використання.

До службового програмного забезпечення також належать **драйвери** (англ. *driver* – водій) – програми для узгодження роботи операційної системи та певного комп’ютерного пристрою (принтера, сканера, монітора, вебкамери, клавіатури та ін.). Операційні системи мають у своєму складі набори драйверів для узгодження роботи (забезпечення обміну даними) з найбільш поширеними моделями пристроїв. На сайтах розробників операційних систем створено й оновлюється склад більш повних бібліотек драйверів для нових пристроїв. Операційні системи зазвичай мають програми автоматичного пошуку потрібних драйверів під час підключення до комп’ютерів нових пристроїв. Пошук здійснюється спочатку в локальній базі певного комп’ютера, а потім на сайті розробників операційних систем.

Зазвичай драйвери постачаються з відповідними пристроями та інструкцією з їх встановлення. Бази драйверів також створюються на сайтах виробників комп’ютерних пристроїв (мал. 1.15).



Мал. 1.15. Фрагмент сайту для завантаження драйверів пристроїв корпорації **Samsung**



Для тих, хто хоче знати більше...

Сучасні операційні системи автоматично здійснюють перевірку стану пристроїв зовнішньої пам'яті. Разом з тим доволі часто виникає потреба в перевірці логічної структури та фізичного стану носіїв даних. Для цього користувач може скористатися стандартною службовою програмою перевірки носіїв, вибравши в контекстному меню пристрою збереження даних команду **Властивості** та на вкладці **Знаряддя** – кнопку **Перевірити** . Під час перевірки програма намагається відшукати частини носія даних, які можна віднести до проблемних – з них погано читаються дані або є ділянки, які не записані в логічну структуру диска і до них відсутній доступ. Програма помічає проблемні ділянки носія, переносить з них дані до надійних ділянок, а логічну структуру оновлює. На цій же вкладці розміщено кнопку запуску програми оптимізації розміщення даних на носії. Подібні перевірки носіїв даних прискорюють роботу з ними.



Працюємо з комп'ютером

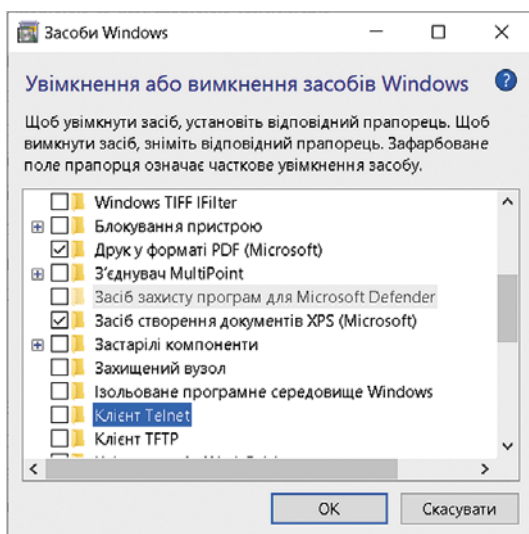
Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання 1. Установлення й видалення програм.

1. Відкрийте вікно **Інсталювані програми**. Для цього виконайте (**Пуск** ⇒ **Усі Інструменти Windows** ⇒ **Панель керування** ⇒ **Програми** ⇒ **Видалити програми**).
2. Визначте, які програми встановлено на вашому комп'ютері. Скільки їх і який обсяг пам'яті вони займають?
3. Виберіть посилання **Системні компоненти**.
4. Виберіть посилання **Увімкнення або вимкнення засобів Windows**.
5. Проведіть інсталяцію таких компонентів **Windows**:

- **Клієнт TFTP;**
- **Прості служби TCP/IP**

Для цього у вікні **Засоби Windows** у списку компонентів установіть позначки прапорців біля назв об'єктів, які потрібно інстальювати (мал. 1).

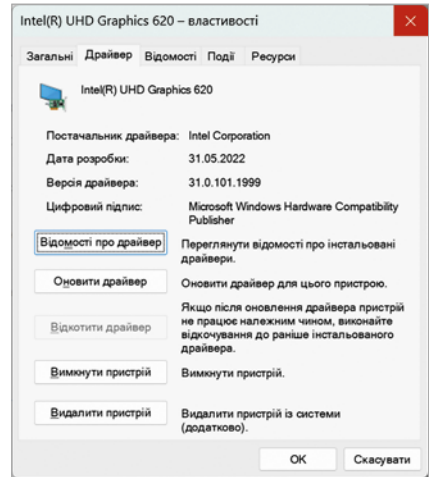


Мал. 1. Вікно **Засоби Windows**

6. Деінсталюйте компонент **Засіб створення документів XPS (Microsoft)**, знявши позначку прапорця біля назви цього об'єкта.
7. Виберіть кнопку **ОК**.
8. Перезавантажте з дозволу вчителя / учительки операційну систему.
9. Відновіть попередній стан компонентів операційної системи.
10. Проведіть інсталяцію програми, указанної вчителем / учителькою, наприклад програми для відновлення файлів, випадково видалених або втрачених у результаті програмного збою або помилки – **Recuva** (Розділ 1\Завдання 1.1\ccsetup637.exe або із сайту виробника за адресою <https://www.ccleaner.com/download>).
11. Запустіть проінсталювану програму, ознайомтеся з її інтерфейсом і призначенням.
12. Закрийте вікно програми **Recuva**.

Завдання 2. Пристрої та їхні драйвери.

1. Відкрийте **Диспетчер пристроїв**. Для цього відкрийте контекстне меню кнопки **Пуск** і виберіть у ньому команду **Диспетчер пристроїв**.
2. У вікні, що відкрилося, знайдіть назви трьох різних пристроїв комп'ютера, наприклад із груп: **Відеоадаптери, Клавіатури, Миші та інші вказівні пристрої**). Для кожного із цих пристроїв визначте відомості про драйвер. Для цього:
 1. Двічі клацніть на назві пристрою та у вікні, що відкрилося (мал. 2), виберіть вкладку **Драйвер**.
 2. Створіть текстовий документ і запишіть у нього відомості про пристрій, розробника драйвера, дату його розробки та версію.
3. Повторіть ці дії для драйверів наступних двох пристроїв.
4. Поясніть, чому кожен із цих пристроїв потребує драйвера.



Мал. 2. Вкладка **Драйвер**



Найважливіше в цьому пункті

Існує багато різноманітних **операційних систем** (ОС). Операційні системи призначені для: керування роботою пристроїв комп'ютера; забезпечення обміну даними між пристроями комп'ютера; зберігання даних в оперативній пам'яті та на зовнішніх носіях; забезпечення виконання інших програм; організації обміну даними між користувачем і комп'ютером.

ОС забезпечує функціонування всього службового програмного забезпечення. Основними складовими операційної системи є: **базова система введення / виведення – BIOS, ядро операційної системи, файлова система, драйвери пристроїв, інтерфейс користувача**.

Залежно від призначення операційні системи поділяють на: **операційні системи для стаціонарних персональних комп'ютерів, для мобільних комп'ютерів, для серверів і суперкомп'ютерів та для вбудованих комп'ютерів**. Також ОС поділяють залежно від типу ліцензії, що використовується під час їх розповсюдження.

Для встановлення програм (включаючи операційні системи) потрібно виконати спеціальну операцію, яка називається **інсталяцією**. Програми, розроблені для одного типу операційних систем, зазвичай є несумісними з іншими операційними системами і не можуть бути інсталюваними.

До складу **системного програмного забезпечення**, крім операційних систем, входять й інші програми, їх ще називають **утилітами**. Утиліти – це програми, які призначені для діагностування апаратної та програмної складових комп'ютера, розширення можливостей ОС.

Драйвери також належать до службового програмного забезпечення. **Драйвер** – у комп'ютерних системах програма для узгодження роботи операційної системи та певного комп'ютерного пристрою (принтера, сканера, монітора, вебкамери, клавіатури та ін.).



Дайте відповіді на запитання

1. За значенням яких властивостей класифікують операційні системи? Наведіть приклади.
2. Що таке **дистрибутив**?

3. Як здійснити інсталяцію компонентів, що входять до складу ОС **Windows**? Як деінсталювати програму в цій операційній системі?

4. Яке призначення драйверів?

5. Як установити драйвер для одного з нових пристроїв, які ви підключаєте до комп'ютера, наприклад принтера?



Обговоріть і зробіть висновки

1. Чому, на вашу думку, існує така велика кількість різних операційних систем? Чи це добре для користувачів?

2. Чи можна вважати операційну систему «мозком» комп'ютера? Наведіть ваші аргументи.

3. Як ОС взаємодіє з апаратним забезпеченням? Чи можна працювати без драйверів?

4. Чи варто користувачам мати можливість вносити зміни в код ОС? Чи може це бути небезпечним?

5. Які проблеми можуть виникнути, якщо однакові операційні системи будуть використовуватися для всіх видів комп'ютерів? Відповідь обґрунтуйте.



Виконайте завдання

1. Відкрийте вікно службової програми **Відомості про систему** (**Пуск** ⇒ **Усі** ⇒ **Інструменти Windows** ⇒ **Відомості про систему**). Дослідіть, які відомості про апаратне та програмне забезпечення містить ця програма. Результати дослідження впишіть у таблицю, яку створіть за наведеним зразком у текстовому процесорі.

№	Властивість ПЗ комп'ютера на вашому робочому місці	Значення властивості
1.	Назва операційної системи	
2.	Версія операційної системи	
3.	Виробник операційної системи	
4.	Розрядність операційної системи	
5.	Програми, що автоматично завантажуються разом з ОС (вказати 1–2 програми)	
6.	Ім'я папки для драйверів пристроїв комп'ютера	

Збережіть таблицю у вашій папці у файлі з іменем завдання **1.2.1**.

2. Відкрийте вікно програми **Очищення диска** (**Пуск** ⇒ **Усі** ⇒ **Інструменти Windows** ⇒ **Очищення диска**). Виберіть диск для очищення. Визначте, використовуючи довідку **Windows**, у чому полягає операція очищення дисків, у яких групах об'єктів відбувається пошук файлів, які можуть бути зайвими. Який загальний розмір файлів, від яких програма пропонує очистити диски? Відповіді на ці запитання збережіть у файлі текстового процесора **завдання 1.2.1** у папці **Документи** і надішліть на електронну пошту вчителю / учительці.

3. Використовуючи матеріал сайту **Statcounter** (адреса ресурсу: <https://gs.statcounter.com>), дослідіть:

а) Які операційні системи для смартфонів є найпопулярнішими у світі та в Україні?

б) Як дані про операційні системи для смартфонів змінилися за останні два роки?

За підсумками дослідження побудуйте діаграми за зразком діаграм, наведених у цьому пункті, та включіть їх до презентації. Презентацію збережіть у вашій папці у файлі з іменем **завдання 1.2.3**.

4. Складіть у текстовому процесорі схему класифікації операційних систем за наведеним у підручнику описом. Використовуючи матеріали інтернету, доповніть цю схему класифікацією за значенням ще однієї властивості операційних систем, наприклад за типом інтерфейсу. Збережіть її у вашій папці у файлі з іменем **завдання 1.2.4** та надішліть на електронну пошту вчителю / учительці.

5. Створіть, використовуючи один із графічних редакторів, схему, яка ілюструє потоки даних між одним із пристроїв введення даних (наприклад, клавіатура) і, після опрацювання процесором, одним із пристроїв виведення даних (наприклад, принтером). Укажіть, на яких етапах обміну даних будуть задіяні драйвери. Збережіть схему у вигляді зображення у вашій папці у файлі з іменем **завдання 1.2.5**.



6. Проведіть опитування серед друзів / подруг чи однокласників / однокласниць: якими операційними системами вони користуються на комп'ютерах і смартфонах. Створіть у табличному процесорі таблицю з результатами опитування. Визначте, яка ОС використовується частіше і на яких пристроях. Побудуйте діаграму, що відображає результати опитування. Збережіть таблицю у вашій папці у файлі з іменем **завдання 1.2.6**.



Готуємось до вивчення нового матеріалу

1. Чи можна встановити нову програму на мобільний пристрій?
2. Як потрібно змінити налаштування інтерфейсу операційної системи для людей зі слабким зором?



1.3. ОСОБЛИВОСТІ РОБОТИ ІЗ ЦИФРОВИМИ ТЕХНОЛОГІЯМИ, СТВОРЕНИМИ ДЛЯ КОРИСТУВАЧІВ З ОСОБЛИВИМИ ОСВІТНІМИ ПОТРЕБАМИ, ЗОКРЕМА ДЛЯ ОСІБ З ІНВАЛІДНІСТЮ

У цьому пункті йтиметься про:

- ▶ сучасні технології, які полегшують життя осіб з інвалідністю;
- ▶ напрямки розвитку цифрових технологій для забезпечення потреб людей з інвалідністю;
- ▶ налаштування інтерфейсу операційних систем для забезпечення нужд людей з особливими освітніми потребами;
- ▶ налаштування інтерфейсу прикладних програмних засобів з метою покращення інклюзивності та доступності.

СУЧАСНІ ТЕХНОЛОГІЇ, ЯКІ ПОЛЕГШУЮТЬ ЖИТТЯ ОСОБАМ З ІНВАЛІДНІСТЮ



Пригадайте

- Які є засоби для змінення масштабу перегляду в офісних програмах? У браузерях?
- Яким способом можна ввести повідомлення без використання клавіатури?

Сьогодні інформаційні технології стали невід’ємною частиною життя кожної людини. Для людей з інвалідністю – це не просто зручність, а часто єдина можливість спілкуватися, навчатися, працювати, подорожувати, отримувати інформацію та реалізовувати свої таланти.

Є також люди, які не мають фізичних вад, але мають особливості інтелектуального розвитку, що призводять до труднощів у навчанні, особи з аутизмом, дислексією, дисграфією, із порушеннями емоційно-вольової сфери тощо. І для них завдяки сучасним інформаційним технологіям світ стає більш доступним та інклюзивним.

В освіті **інклюзія** – це залучення осіб (дітей і дорослих) з особливими освітніми потребами до навчання в загальноосвітніх закладах разом з іншими учнями із забезпеченням необхідної підтримки. У загальному плані інклюзія розглядається під кутом зору створення можливостей для людей з інвалідністю та особливими освітніми потребами реалізовувати свої здібності й досягати успіху, зокрема з використанням інформаційно-комунікаційних технологій.

Розглянемо приклади діяльності відомих людей, у житті яких інформаційно-комунікаційні технології відіграли значну роль.

Унікальним прикладом є життя і наукова діяльність видатного англійського вченого в галузі фізики та космології – **Стівена Вільяма Гокінга** (1942–2018). 1963 року Гокінгу, коли йому виповнився 21 рік, поставили діагноз повільно прогресуючої форми захворювання, яке поступово, протягом десятиліть, паралізувало його.

Проте Стів Гокінг не зупинив своєї наукової діяльності, спілкування з рідними та науковою спільнотою. З кінця 1960-х років у віці 25–28 років він втратив можливість переміщатися без сторонньої допомоги та не міг писати, а рухів руками ледве вистачало для керування інвалідним візком. У середині 1980-х років він повністю втратив можливість говорити. У цей час він починає для спілкування використовувати комп’ютерну програму під назвою «**Еквалайзер**» від компанії **Words+ Inc.**, яку заснував **Волтер Волтош** (нар. 1946), яка допомагала Гокінгу вибирати слова на екрані з використанням одного перемикача, що активувався рухом м’яза щокі. Курсор рухався по сітці літер, і Гокінг смикав щокі, щоб вибрати потрібний символ або слово (мал. 1.16).

Аутизм (грец. autos – «сам») – розлад розвитку, що характеризується патологічною замкнутістю, відгородженістю від реальної дійсності з втратою інтересу до спілкування.

Дислексія (грец. δυσ – порушення, λέξις – слово, мовлення) – труднощі в оволодінні навичками читання.

Дисграфія (грец. δυσ – порушення, γράφω – пишу) – труднощі в оволодінні письмовим мовленням.

Інклюзія (лат. inclusio – включати, поміщати всередину) – включення когось до складу певної спільноти, системи чи групи.



Мал. 1.16. Стів Гокінг на своєму візку з комп’ютером 2007 року

Щойно речення формувалися, їх озвучував особливий синтезатор мовлення – роботизований голос, який став однією з візитівок Гокінга. Це був пристрій, розроблений компанією **Speech Plus**, який перетворював уведений текст на голос. Так Стів Гокінг міг спілкуватися і писати наукові статті та книги. В останні роки життя він співпрацював з корпорацією Intel, яка створила для нього більш сучасну версію синтезатора мовлення.

До самої своєї смерті у віці 76 років Стівен Гокінг вів активне життя – писав наукові та науково-популярні твори, дискутував з науковцями, брав участь у телевізійних передачах, знімався в кіно, спробував невагомий стан у літаку, збирався полетіти в космос. Його життя є яскравим прикладом того, як технології можуть розширювати межі людських можливостей.

Однак він не єдиний, хто завдяки комп'ютерним інноваціям зміг реалізувати свій потенціал. Ось кілька інших видатних особистостей.

Еймі Маллінз (нар. 1976) – паралімпійська спортсменка, модель, актриса та ораторка. Має інвалідність – уроджену гемімелію (відсутність малогомілкових кісток), що призвело до ампутації обох ніг нижче колін у віці одного року.

Еймі використовує різноманітні протези, розроблені з використанням комп'ютерного моделювання та 3D-друку, які надали їй можливість не лише ходити, а й бігати (має спеціальні «бігові» протези), займатися спортом, бути моделлю та виступати з питань захисту людей з інвалідністю.

Паралімпійський спорт

(грец. парά – поруч, разом, olympos – олімпійський) – змагання, які відбуваються нарівні з Олімпійськими іграми, але для спортсменів з інвалідністю.

Технології забезпечили її протезами для різних цілей – від бігу до високих підборів, що розширило її можливості в усіх сферах життя.



Мал. 1.17. Еймі Маллінз



Мал. 1.18. Джеймс Лебрехт

Джеймс Лебрехт (нар. 1956) – американський режисер, звукорежисер, активіст – борець за права людей з інвалідністю. Має інвалідність – розщеплення хребта, що призвело до використання інвалідного візка.

Джеймс є співрежисером номінованого на «Оскар» документального фільму «Табір інклюзивних можливостей». Він активно використовує комп'ютерні програми для звукорежисури, монтажу відео та

комунікації, що надає йому можливість працювати в кіноіндустрії, попри фізичні обмеження. Його робота є прикладом того, як цифрові інструменти роблять творчі професії доступними.

Оксана Потимко (нар. 1973) – українська науковиця та громадська діячка, докторка філософії, заслужений працівник соціальної сфери України, лауреатка Державної премії в галузі освіти, директорка Ресурсного центру Національного університету «Львівська політехніка» для осіб з особливими освітніми потребами. Має інвалідність по зору (незряча), яку отримала у 23 роки.

Використовує програми – екранні диктори (NVDA, JAWS), брайлівські дисплеї та аудіопідручники для роботи, ведення лекцій, підготовки навчальних курсів. Завдяки цифровим технологіям активно бере участь у міжнародних онлайн-конференціях. Авторка української моделі інклюзивної освіти незрячих школярів. Займається виготовленням кіно та мультфільмів з аудіодискрипцією, організацією інклюзивного навчання незрячих дітей у закладах середньої та вищої освіти, підготовкою педагогів до роботи з аудиторією незрячих учнів / учениць та студентів / студенток.

Ці приклади доводять, що цифрові технології надають можливість людям з інвалідністю не тільки спілкуватися і працювати, а й досягати успіхів у науці, творчості, спорті, журналістиці та суспільному житті.



Мал. 1.19. Оксана Потимко

Аудіодискрипція – додатковий звуковий коментар з описом об'єктів та дій, що відбуваються на екрані. Спеціальна технологія для незрячих.

НАПРЯМКИ РОЗВИТКУ ЦИФРОВИХ ТЕХНОЛОГІЙ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПОТРЕБ ЛЮДЕЙ З ІНВАЛІДНІСТЮ



Пригадайте

- Що таке *цифрові технології*?
- Які пристрої потрібні для голосового введення повідомлень?

Розвиток цифрових технологій для осіб з інвалідністю є не лише питанням соціальної відповідальності, а й важливим напрямком інновацій, що сприяє створенню більш інклюзивного та справедливого суспільства. Залежно від особливостей осіб, що потребують специфічних умов доступу до цифрового контенту, розглядають кілька напрямків інформаційно-комунікаційних технологій для забезпечення їх інклюзії:

- **для людей з порушеннями зору (сліпота, слабкий зір):**
 - **екранні лупи** (англ. *Screen Magnifiers*), які збільшують частину екрана, для розгляду дрібних деталей користувачам зі слабким зором, наприклад **Екранна лупа** (Windows), Zoom (macOS/iOS);

Брайлівський дисплей – дисплей, що може відображати текст азбукою Брайля для незрячих.

- **програми для читання з екрана** (англ. *Screen Readers*), які перетворюють текст на екрані на звукові повідомлення або шрифт Брайля, наприклад **JAWS**, **NVDA** (для **Windows**), **VoiceOver** (для **macOS/iOS**), **TalkBack** (для **Android**). Зазвичай такі програми не тільки відтворюють текст на екрані, а й забезпечують уведення повідомлень голосом, «озвучують» пункти і команди меню під час наведення вказівника на них, натискання клавіш;
- **пристрої, що використовують шрифт Брайля** для виведення (брайлівські дисплеї (мал. 1.20), принтери (мал. 1.21), планшети, смартфони) та введення (брайлівські клавіатури (мал. 1.22 та 1.23)) даних, використовують для перетворення тексту з екрана комп'ютера на тактильний шрифт Брайля і навпаки;



Мал. 1.20. Брайлівський дисплей з клавіатурою



Мал. 1.21. Брайлівський принтер



Мал. 1.22. Стандартна клавіатура з нанесеними на клавіші елементами азбуки Брайля

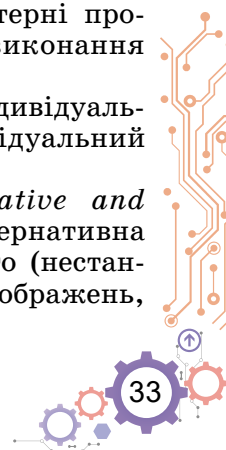


Мал. 1.23. Уведення даних зі спеціальної клавіатури, що використовує азбуку Брайля

- **програми розпізнавання тексту OCR** (англ. *OCR – Optical Character Recognition* – оптичне розпізнавання символів) для перетворення відсканованих документів або тексту на фотографіях чи інших графічних зображеннях на текст, який можна редагувати, а також прочитати з використанням програми для читання з екрана;
- **аудіокниги та подкасти** для звукового відтворення художніх, технічних, наукових та інших текстів;
- **висококонтрастні теми оформлення та великі за розмірами шрифти**, що використовуються в інтерфейсах комп'ютерних програм і вебсайтів (для слабозорих);

- **для людей з порушеннями слуху (глухота, суттєва втрата слуху):**
 - відтворення на екрані **субтитрів** для дублювання голосових повідомлень у відеоконтенті, онлайн-лекціях, вебінарах, наприклад субтитри в **YouTube, Google Meet, Zoom**;
 - **системи перетворення мовлення в текст** (англ. *Speech-to-Text*) для перетворення усного мовлення на текст у реальному часі. Використовуються в комунікаційних додатках;
 - **візуальні сповіщення** для замінення звукових сповіщень візуальними або тактильними, наприклад миготіння екрана, вібрація або світлова індикація;
 - використання **перекладача на жестову мову** як людини-перекладача на спеціальних платформах, наприклад для проведення відеоконференцій (сервіс від Українського товариства глухих (УТОГ), так і комп'ютерних програм – систем перекладу на жестову мову, наприклад **Google Translate, Ukrainian Sign Language (USL) Projects, Mimix3D**;
- **для людей, що мають порушення опорно-рухового апарату (обмежена рухливість рук, параліч):**
 - програми **керування голосом** (англ. *Voice Control Software*) – для введення голосових команд до комп'ютера або мобільного пристрою, наприклад програми **Dragon NaturallySpeaking, Google Voice Typing, Siri, Cortana**;
 - **екранні клавіатури** (англ. *On-Screen Keyboards*) – для введення тексту з використанням миші, джойстика або іншого вказівного пристрою (трекбол, тачпед, сенсорний екран, спеціалізовані миші тощо);
 - **сенсорні перемикачі** (англ. *Switches*) – невеликі кнопки (тумблери), які спрацьовують, реагуючи на рух голови, зміну ритму дихання, моргання тощо. Використовуються для введення команд і текстових повідомлень;
 - **системи керування поглядом** (англ. *Eye-Tracking Devices*) – для відслідковування руху очей і синхронізації з ним руху курсора на екрані, подання команд опусканням вій чи заплюскуванням очей.
- **для людей, що мають потребу в інших особливих умовах комунікації:**
 - **візуальні планувальники та органайзери** – комп'ютерні програми для структурування даних, планування виконання завдань, складання розкладів тощо;
 - **адаптивні навчальні платформи** – для створення індивідуальної траєкторії навчання, підлаштування під індивідуальний темп навчання та стиль сприйняття інформації;
 - **символьні комунікатори** (англ. *AAC – Augmentative and Alternative Communication* – доповнююча та альтернативна комунікація) – для надання способів альтернативного (нестандартного) спілкування з використанням символів, зображень, синтезованого мовлення тощо.

Субтитри (лат. *sub* – під, *titulus* – назва, заголовок) – це текстові рядки, які розміщуються на екрані під час перегляду кіно або відео.



Важливо пам'ятати, що **доступні технології** – це не лише про комфорт, а про рівні права, можливості та гідність кожної людини. Тому розробники комп'ютерних програм намагаються враховувати потреби всіх верств населення, що знайшло своє відображення в кількох правилах для програмних робок:

- правило **універсального дизайну**, що передбачає створення програмних продуктів, які можуть бути використані всіма людьми, наскільки це можливо, без потреби адаптації;
- простота **змінення налаштувань інтерфейсу** – можливість змінювати розмір шрифту, кольорову схему, пристрої введення / виведення даних, швидкість мовлення;
- простота й **інтуїтивність**: зображення об'єктів інтерфейсу та їх розміщення мають бути зрозумілими та легкими у використанні, а їх призначення – інтуїтивно зрозумілим;
- **сумісність** – можливість використання різних допоміжних технологій: пристроїв (наприклад, дисплея **Брайля**) та програм (наприклад, **Transcription**);
- чіткий і зрозумілий **зворотний зв'язок** від системи до користувача (візуальний, звуковий, тактильний), відсутність залежності від одного способу введення / виведення, обмін даними має бути доступний різними сенсорними каналами.

Одним з інструментів для забезпечення доступності комп'ютерних технологій різним верствам населення є дотримання міжнародних стандартів, таких як **WCAG** (англ. *Web Content Accessibility Guidelines* – керівні принципи доступності вебконтенту).

ЗАСОБИ ДЛЯ ЗАБЕЗПЕЧЕННЯ ДОСТУПНОСТІ ЛЮДЯМ З ОСОБЛИВИМИ ОСВІТІНИМИ ПОТРЕБАМИ У WINDOWS

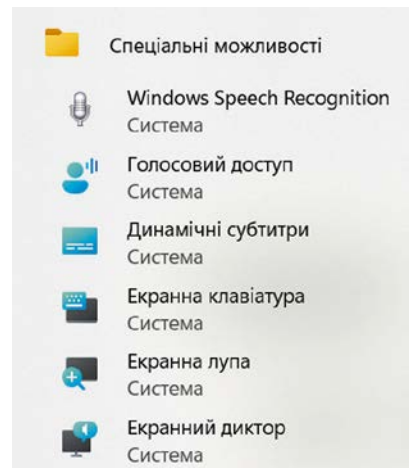


Поміркуйте

- Як змінити налаштування інтерфейсу **Windows**? • Як змінити тему оформлення інтерфейсу в графічному редакторі, наприклад **Krita**?

Розглянемо інструменти операційної системи **Windows** для змінення налаштувань інтерфейсу, які можуть бути використані користувачами з особливими освітніми потребами. Як відомо, значна частина таких інструментів згрупована в меню **Пуск** ⇒ **Усі** ⇒ **Спеціальні можливості**. До цієї групи комп'ютерних програм належать (мал. 1.24):

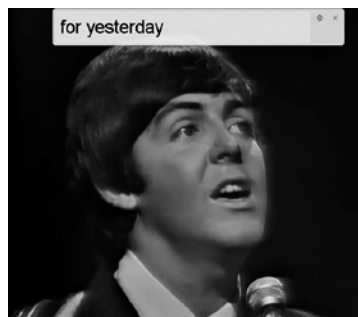
- **Windows Speech Recognition (WSR)** (англ. – розпізнавання мовлення **Windows**) – програма, яка призначена для керування комп'ютером з використанням голосових команд, таких як «Відкрити **Paint**», «Натиснути **Файл**», «Закрити вікно», а також для введення тексту в текстових програмах і браузерах. На жаль,



Мал. 1.24. Список програм групи **Спеціальні можливості**

серед мов, які можуть бути використані у програмі, відсутня українська, тому програма не працює в ОС з українським інтерфейсом. Програма особливо корисна для людей з обмеженою моторикою, користувачів, які не можуть використовувати клавіатуру чи мишу.

- **Голосовий доступ** – це програма, що є удосконаленням попередньої програми **Windows Speech Recognition** в ОС **Windows**, починаючи з 11-ї версії 2024 року. Програма порівняно з **WSR** краще розпізнає мову та контент, має вдосконалену систему навігації інтерфейсом програм, може працювати без підключення до інтернету, використовує новітні технології розпізнавання мовлення. Програма буде корисна для людей з порушеннями опорно-рухового апарату, людям з дислексією або іншими труднощами з письмом, для всіх, хто диктує текст краще, ніж уводить із клавіатури. Поки що не може працювати з українською мовою, але виконує команди англійською (німецькою, іспанською та французькою) навіть в ОС з українським інтерфейсом.
- **Динамічні субтитри** – це програма, яка автоматично перетворює будь-який звук, що відтворюється на комп'ютері, у текст у реальному часі. Вона створена для покращення доступності, особливо для людей з вадами слуху, але також корисна для навчання, перегляду відео без звуку або в шумному середовищі. На сьогодні не працює з українською мовою (мал. 1.25).
- **Екранна клавіатура** – це програма для введення тексту і керування комп'ютером з використанням віртуальної клавіатури, що відображається на екрані (мал. 1.26). Екранна клавіатура призначена для користувачів, які через фізичні обмеження (наприклад: травми рук, параліч, тремор) не можуть користуватися стандартною клавіатурою. Тому виникає потреба в альтернативному пристрої введення тексту з використанням миші, джойстика, пристрою керування поглядом або іншого вказівного пристрою.



Мал. 1.25. Використання субтитрів під час перегляду відеокліпу Beatles



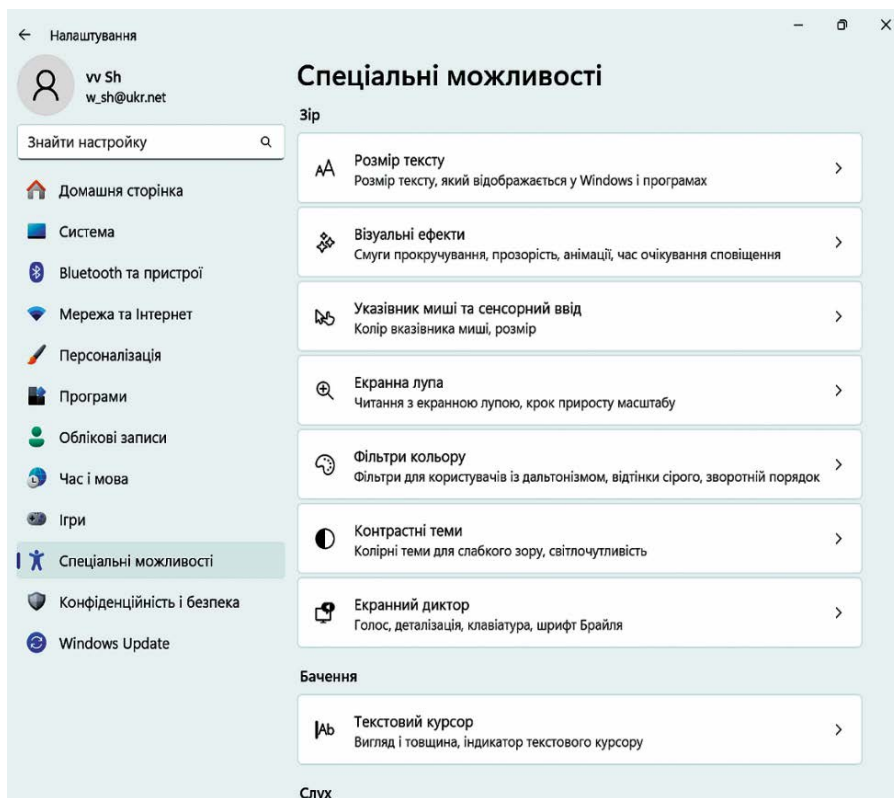
Мал. 1.26. Екранна клавіатура Windows

Також екранну клавіатуру використовують на пристроях із сенсорним екраном (смартфони, планшети, термінали, ноутбуки-трансформери) або користувачі, які тимчасово не мають стандартної

клавіатури (пошкодження, відсутність портів підключення). Ще одним із напрямків використання віртуальної клавіатури є захист від програм-шпигунів, які можуть перехоплювати натискання клавіш на клавіатурі.

- **Екранна лупа** – це програма для збільшення частини екрана для полегшення перегляду тексту, зображень або елементів інтерфейсу.
- **Екранний диктор** – це програма для читання з екрана, розроблена для допомоги незрячим або людям зі слабким зором, може використовуватися також під час вивчення певної мови. Вона озвучує текст, елементи інтерфейсу, дії користувача та інші події на екрані. На жаль, серед мов, які може використовувати **Екранний диктор**, відсутня українська мова.

Налаштувати значення властивостей наведених програм, а також унести інші зміни в інтерфейс операційної системи можна з використанням інструментів вікна **Спеціальні можливості** (*Пуск* ⇒ *Налаштування* ⇒ *Спеціальні можливості*) (мал. 1.27). Користувач може змінити розмір тексту всіх написів на екрані (підписи піктограм, пункти меню, службові повідомлення тощо), налаштувати під свої потреби візуальні ефекти, відображення вказівника та курсора, увімкнути / вимкнути і налаштувати роботу **Екранної лупи** та **Екранного диктора**, субтитрів, голосового вводу, застосувати фільтри кольору (для осіб з дальтонізмом) або контрастні теми оформлення (для осіб зі слабким зором) та інше.



Мал. 1.27. Вікно **Спеціальні можливості**



Працюємо з комп'ютером

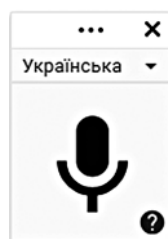
Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання 1. Налаштуйте **Динамічні субтитри** для перегляду відео-контенту.

1. Відкрийте браузер і перейдіть за посиланням (<https://cutt.ly/ttSyQauu>) на сторінку з піснею **Beatles Let It Be** або відкрийте у відеопрогравачі на вашому комп'ютері файл із цією самою піснею, наприклад з папки **Розділ 1\Відеокліп\Let It Be.mp4**.
2. Відкрийте програму **Динамічні субтитри** (**Пуск** ⇒ **Усі** ⇒ **Спеціальні можливості** ⇒ **Динамічні субтитри**).
3. Виберіть мову субтитрів і завантажте мовні файли (за потреби).
4. Змініть розміщення вікна **Динамічні субтитри**. Для цього виберіть кнопку **Параметри**  та виконайте **Положення** ⇒ **Унизу екрана**.
5. Розпочніть відтворення відео у вікні програвача.
6. Відслідкуйте, наскільки коректно програма відтворює текст пісні, чи відбувається корекція тексту під час розпізнавання.
7. Закрийте вікна програвача / браузера та програми **Динамічні субтитри**.

Завдання 2. Використання сервісів **Google** для голосового введення тексту.

1. Відкрийте вікно браузера **Google Chrome**. Уведіть дані вашого облікового запису.
2. Створіть новий **Google Документ**.
3. Увімкніть голосове введення тексту. Для цього виконайте **Інструменти** ⇒ **Голосовий ввід**.
4. В іншому вікні браузера відкрийте вірш **Ліни Костенко** «Красива осінь вишиває клени...» (<https://osvita.ua/school/literature/k/70134/>) або відкрийте цей самий вірш у вікні текстового процесора, наприклад із файлу **Розділ 1\Текст\Костенко.docx**.
5. Розмістіть вікна з **Google Документом** та з віршем поруч.
6. Зробіть поточним вікно з новим **Google Документом**.
7. Уключіть уведення тексту голосом, вибравши кнопку голосового введення у вікні **Голосовий ввід** (мал. 1). За потреби встановіть у списку мов українську мову введення.
8. Прочитайте текст вірша **Ліни Костенко**. Перевірте, чи припустилася програма помилок, чи розставлені розділові знаки.
9. Збережіть створений документ у вашій папці у файлі з іменем **вправа 1.3.1**.



Мал. 1. Вікно **Голосовий ввід**



Найважливіше в цьому пункті

Сьогодні інформаційні технології стали невід'ємною частиною життя кожної людини. Для людей з інвалідністю – це не просто зручність, а часто єдина можливість спілкуватися, навчатися, працювати, подорожувати, отримувати інформацію та реалізовувати свої таланти.

Інклюзія – це процес створення можливостей для людей з інвалідністю та особливими освітніми потребами реалізовувати свої здібності й досягати успіху, зокрема з використанням інформаційно-комунікаційних технологій.

Світ знає не поодинокі приклади життя й діяльності людей, які завдяки комп'ютерним інноваціям змогли реалізувати свій науковий, соціальний і творчий потенціал, наприклад: **Стівен Вільям Гокінг, Еймі Маллінз, Джеймс Лебрехт, Оксана Потимко**. Життєвий шлях цих особистостей доводить, що цифрові технології дають можливість людям з інвалідністю не тільки спілкуватися і працювати, а й досягати успіхів у науці, творчості, спорті, журналістиці та суспільному житті.

Залежно від особливостей осіб, які потребують специфічних умов доступу до цифрового контенту, розглядають кілька напрямків інформаційно-комунікаційних технологій для забезпечення їхньої інклюзії:

- **для людей з порушеннями зору** – екранні лупи, програми для читання з екрана, пристрої, що використовують шрифт Брайля для виведення та введення даних, програми розпізнавання тексту, аудіокнижки та подкасти, спеціальні варіанти оформлення інтерфейсу;
- **для людей з порушеннями слуху** – екранні субтитри, системи перетворення мовлення в текст, візуальні сповіщення, перекладачі на жестову мову;
- **для людей, що мають порушення опорно-рухового апарату** – програми керування голосом, екранні клавіатури, сенсорні перемикачі та системи керування поглядом.

Розробники комп'ютерних програм намагаються враховувати потреби всіх верств населення, що знайшло своє відображення в кількох правилах розробки програм, серед яких виокремлюється правило **універсального дизайну**, що передбачає створення програмних продуктів, які можуть бути використані всіма людьми, наскільки це можливо, без потреби адаптації.

Операційна система **Windows** має ряд інструментів, які можуть бути використані користувачами з особливими освітніми потребами. Частина таких інструментів згрупована в меню **Спеціальні можливості: Windows Speech Recognition, Голосовий доступ, Динамічні субтитри, Екранна клавіатура, Екранна лупа, Екранний диктор**.



Дайте відповіді на запитання

1. Яке головне призначення інформаційних технологій для людей з інвалідністю?
2. Що означає «універсальний дизайн» у контексті розробки програмних продуктів для людей з інвалідністю?
3. Для чого призначена екранна лупа в операційній системі **Windows**?
4. Які програми доступності включені до операційної системи **Windows**?
5. Які програми використовують люди з порушенням зору для читання тексту з екрана?
6. Уявіть, що ваш друг / подруга має порушення слуху. Які цифрові технології ви порадили б йому / їй для комфортного навчання та спілкування? Поясніть свій вибір.



Обговоріть і зробіть висновки

1. Опишіть, як ви уявляєте роботу комп'ютера, яким можна керувати лише поглядом. У яких ситуаціях така технологія була б найкориснішою?
2. Виходячи з прикладів **Стівена Гокінга, Еймі Маллінз та Джеймса Лебрехта**, як ви вважаєте, чи можуть цифрові технології

не просто компенсувати інвалідність, а й стати джерелом унікальних переваг або нових можливостей у певних сферах діяльності? Наведіть аргументи, спираючись на їхні історії.

3. Як саме цифрові технології сприяють підвищенню гідності людей з інвалідністю, виходячи за межі простого «полегшення життя»? Наведіть конкретні приклади з тексту або власні роздуми.

4. Уявіть, що ви є розробником нової соціальної мережі. Які правила універсального дизайну ви застосували б насамперед, щоб зробити вашу платформу максимально інклюзивною (доступною) для людей з різними видами інвалідності? Поясніть, чому саме ці принципи є найважливішими для вашого проекту.

5. Чому, на вашу думку, важливо, щоб розробники комп'ютерних програм враховували потреби людей з інвалідністю? Наведіть приклади.



Виконайте завдання

1. Протестуйте використання програми **Екранна лупа** для перегляду однієї зі сторінок вебсайту. Для виконання завдання відкрийте будь-який вебсайт у браузері; увімкніть «Екранну лупу» та збільште масштаб до 200 %; знайдіть заголовки статті та прочитайте його, використовуючи збільшення. Кому і в яких ситуаціях **Екранна лупа** буде найбільш корисною?



2. Відкрийте в **Microsoft Word** документ із текстом англійською, наприклад **Розділ 1\Текст\Beathles Yesterday.docx**. Прослухайте текст, використовуючи команду контекстного меню **Читати вголос**. Дослідіть роботу із цією функцією та опишіть коротко в новому текстовому документі. Збережіть документ у вашій папці у файлі з іменем **завдання 1.3.2**.



3. Відкрийте в **Microsoft Word** документ із текстом англійською, наприклад **Розділ 1\Текст\Beathles Come Together.docx**. Прослухайте текст, використовуючи програму **Екранний диктор**. Дослідіть роботу із цією програмою та опишіть коротко роботу з нею в новому текстовому документі. Збережіть документ у вашій папці у файлі з іменем **завдання 1.3.3**.



4. Відкрийте програму **Блокнот**. Увімкніть **Екранну клавіатуру**. Уведіть, використовуючи **Екранну клавіатуру**, текст «Привіт, світе! Це екранна клавіатура». Дослідіть, як змінюється розкладка клавіатури, уводяться спеціальні символи (@, \$, &). Додайте у файл послідовність введення цих символів з використанням екранної клавіатури. Збережіть документ з уведеним текстом у вашій папці у файлі з іменем **завдання 1.3.4**.



5. Проекспериментуйте зі зміною оформлення інтерфейсу ОС. Для цього, використовуючи налаштування, розміщені у вікні **Спеціальні можливості (Пуск ⇒ Налаштування ⇒ Спеціальні можливості)**, установіть:

- розмір тексту – 150 %;
- контрастна тема – *Нічне небо*.

Перегляньте, як змінився інтерфейс **Робочого столу**, **Блокнота**. Дайте відповідь на запитання: *Кому і в яких ситуаціях ці налаштування будуть найбільш корисними?* Збережіть відповідь у текстовому документі у вашій папці у файлі з іменем **завдання 1.3.5**.

6. Відкрийте вікно браузера **Google Chrome**. Уведіть дані вашого облікового запису. Створіть новий **Google Документ**. Увімкніть введення тексту з голосу та введіть відповідь на запитання: «Для яких цілей може використовуватися голосовий ввід?» Збережіть створений документ у вашій папці у файлі **завдання 1.3.5**.



Практична робота № 1.

«Використання голосового введення тексту»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Відкрийте вікно браузера **Google Chrome**. Уведіть дані вашого облікового запису. Створіть новий **Google Документ**. Увімкніть голосове введення тексту і введіть відповідь на запитання:

- Яке призначення програми **Динамічні субтитри**?
- Як змінити розміри шрифтів у підписах до піктограм на **Робочому столі**?
- Як уставити крапку під час уведення голосом у **Google Документах**? Збережіть файл з іменем **практична 1**.



Готуємось до вивчення нового матеріалу

- Що таке *текстовий процесор*? Які операції з його використання можна виконувати над текстовими об'єктами?
- Які об'єкти опрацьовуються в текстовому процесорі?



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу «Програмне забезпечення» (початковий, середній, достатній, високий рівень).

- Я можу пояснити, що таке *інформаційні системи*.
- Я розумію, які інформаційні процеси здійснюються в інформаційних системах, і можу пояснити послідовності їх виконання.
- Я можу пояснити узагальнену структуру інформаційної системи та взаємодію її складових.
- Я вмію добирати потрібні програмні продукти для розв'язування певної задачі.
- Я можу навести приклади правової взаємодії та відповідальності за порушення законів і правових норм у галузі цифрових технологій.
- Я можу аргументовано обирати програмні продукти за їхньою ліцензією.
- Я розпізнаю інформаційні продукти з відкритим і закритим кодом.
- Я розумію ліцензійні (правові та етичні) обмеження на використання та редагування власних і чужих інформаційних продуктів.
- Я вмію інсталиювати і деінсталиювати програмні продукти.
- Я розумію особливості сумісності програмних продуктів.
- Я можу пояснити роль драйверів у роботі пристроїв комп'ютера.
- Я можу навести приклади цифрових технологій, створених для користувачів з особливими освітніми потребами, зокрема осіб з інвалідністю.
- Я можу вносити зміни в інтерфейс і зміст інформаційних продуктів з метою покращення інклюзивності та доступності.



ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ

У цьому розділі ви отримаєте нові, а також поглибите і розширите свої знання й удосконалисте навички з таких тем:

- ▶ системи опрацювання текстів, їхні можливості та класифікація;
- ▶ об'єкти текстового документа та їхні властивості;
- ▶ формати файлів текстових документів;
- ▶ режими роботи в середовищі текстового процесора;
- ▶ робота з кількома документами одночасно;
- ▶ створення в текстовому документі багаторівневих списків, колонтитулів, стовпців, розділів;
- ▶ вставлення в текстовий документ символів і формул;
- ▶ стилі та їх використання;
- ▶ створення структури документа й автоматичне створення його змісту;
- ▶ створення гіпертекстових посилань у текстовому документі;
- ▶ шаблони документів та їх використання;
- ▶ рецензування документів.



2.1. СИСТЕМИ ОПРАЦЮВАННЯ ТЕКСТІВ. РОБОТА З ТЕКСТОВИМИ ДОКУМЕНТАМИ

У цьому пункті йтиметься про:

- ▶ системи опрацювання текстів, їхню класифікацію та призначення;
- ▶ об'єкти текстового документа та їхні властивості;
- ▶ формати файлів текстових документів;
- ▶ режими перегляду документа;
- ▶ роботу з кількома документами одночасно.

СИСТЕМИ ОПРАЦЮВАННЯ ТЕКСТІВ, ЇХ ПРИЗНАЧЕННЯ ТА КЛАСИФІКАЦІЯ



Пригадайте

● Що таке *текстовий процесор*? Яке його призначення? ● Які операції над об'єктами текстового документа можна виконувати в текстовому процесорі? Наведіть приклади.

Вивчаючи інформатику в попередніх класах, ви вже створювали текстові документи в текстовому процесорі **Microsoft Word**: вводили текст, редагували та форматували його, вставляли в документ графічні зображення і таблиці, друкували документ.

Крім текстового процесора **Word**, існує багато різноманітних програм, які призначені для роботи з текстами. Такі програми називають **системами опрацювання текстів**. Усі вони належать до прикладного програмного забезпечення.

Системи опрацювання текстів можна використовувати в різних сферах людської діяльності для підготовки різноманітних документів, поліграфічних видань, буклетів, плакатів, макетів книжок тощо. Вибір тієї чи іншої програми залежить від потреб користувача.

Використовуючи ці програми, можна створювати текстові документи, зберігати їх, швидко вносити зміни в раніше підготовлені документи. Можна переставляти фрагменти тексту, копіювати їх у різні місця документа, вставляти в текст частини з інших документів, а за потреби – скасовувати виконані зміни. У більшості систем опрацювання текстів автоматизовано пошук орфографічних і синтаксичних помилок. Використання цих програм значно полегшує оформлення тексту, додавання в текст формул, малюнків, таблиць тощо. Нарешті, створивши електронний текстовий документ один раз, можна в будь-який момент отримати потрібну кількість його копій як електронних, так і паперових.

Класифікація цього виду програмного забезпечення може здійснюватися за різними властивостями цих систем. Так, наприклад, **за призначенням** системи опрацювання текстів розподіляються на такі групи:

- *текстові редактори* – використовують для підготовки простих документів, без форматування та графічних зображень, таблиць тощо. Наприклад, **Блокнот** (в ОС Windows), **Kate** (в ОС Linux). Вони також є частиною HTML-редакторів, призначених для створення вебсторінок, а також середовищ розробки програм і проєктів мовами програмування, зокрема **Python**, **Lazarus**;
- *текстові процесори* – використовують для створення, форматування, макетування документів складної структури, вставлення різноманітних об'єктів тощо. Наприклад, **Microsoft Office Word**, **LibreOffice Writer**, **Google Docs**;
- *редактори наукових документів* – призначені для підготовки документів, що містять математичні, хімічні та інші формули, спеціальні символи, складні діаграми та інші нестандартні для текстових документів об'єкти. Наприклад, **LaTeX**, **MathType**, **Texmaker**;
- *видавничі системи* – призначені для створення макетів поліграфічних видань (від рекламних проспектів до книжок) та підготовки їх до друку. Наприклад, **QuarkXPress**, **Scribus**, **Adobe InDesign**;
- *системи машинного перекладу* – призначені для автоматичного перекладу текстів з однієї мови на іншу, перевірки правопису в тексті різними мовами. Наприклад, **Google Translate**, **DeepL**;
- *системи оптичного розпізнавання текстів* – призначені для перетворення графічного зображення тексту в текстовий формат. Наприклад, **Google Cloud Vision**, **Tesseract**;
- *системи розпізнавання мовлення* (*Speech-to-Text*) – перетворюють усне мовлення на текст. Використовують

Макет текстового документа – це сукупність розміщених на сторінці текстового документа і впорядкованих за певними правилами об'єктів.

Транскрибування – це процес перетворення усного мовлення або аудіозапису в письмовий текст.

для транскрибування аудіо та голосового введення тексту тощо. Наприклад, **Microsoft Azure Speech**, **Amazon Transcribe** та ін.

Також системи опрацювання текстів можна розділити за **типом доступу**: *локальні* – установлюються на комп'ютер і працюють без інтернет-з'єднання, та *хмарні* (доступні онлайн, призначені для спільної роботи з документом) – **Google Docs**, **Microsoft 365**.

Більшість систем опрацювання текстів підтримують технологію **WYSIWYG** (англ. *What You See Is What You Get* – що бачиш, те й отримав), яка забезпечує відображення документа на екрані в тому вигляді, у якому він буде надрукований або збережений у файлі.

ОБ'ЄКТИ ТЕКСТОВОГО ДОКУМЕНТА ТА ЇХНІ ВЛАСТИВОСТІ



Пригадайте

- З яких об'єктів складається текстовий документ? Опишіть їхні властивості.
- Які операції можна виконувати над об'єктами в текстовому документі?

Як ви вже знаєте, основним об'єктом у системах опрацювання текстів є **текстовий документ**. У ньому можуть міститися різні інші об'єкти: символи, слова, речення, абзаци, сторінки, малюнки, таблиці, діаграми та інші. Кожен з об'єктів текстового документа має свої властивості (табл. 2.1).

Таблиця 2.1

Об'єкти текстового документа

Об'єкт	Властивості об'єкта
Символ	Шрифт, розмір, колір, накреслення, видозміна, зсув, інтервал між символами та інші
Абзац	Відступ від краю лівого та правого поля, відступ першого рядка, вирівнювання, міжрядковий інтервал, відстань між абзацами та інші
Сторінка	Розмір аркуша сторінки, її орієнтація, розміри полів, нумерація, колонититули та інші
Малюнок	Тип, розмір, кольори, положення, обтікання текстом та інші
Таблиця	Кількість рядків і стовпців, їх висота та ширина, заливка та інші

Над кожним з об'єктів користувач, використовуючи систему опрацювання текстів, може виконувати певний набір операцій зі створення, редагування та форматування об'єктів текстового документа.

ФОРМАТИ ТЕКСТОВИХ ФАЙЛІВ



Пригадайте

- Що таке *формат файлу*? Що він визначає? • Яким є стандартний формат текстових файлів, створених у текстовому процесорі **Word**?

Створений текстовий документ зберігають у файлі. Існують різні формати файлів, у яких системи опрацювання текстів можуть зберегти текстові документи. Кожна із цих систем використовує певний набір форматів, які користувач може вибрати для запису файлів залежно від потреб.

Один із цих форматів є стандартним для даної системи опрацювання текстів і встановлюється за замовчуванням. Для роботи з файлами інших форматів системи опрацювання текстів мають у своєму складі спеціальні програми – **конвертори**, які перетворюють файли текстового документа з одного формату в інший.

Найбільш поширеними форматами текстових файлів є **TXT, RTF, DOCX, PDF, ODT**. Характеристику цих форматів наведено в таблиці 2.2.

Таблиця 2.2

Характеристика найпоширеніших форматів текстових документів

Формат	Значок	Характеристика формату
TXT		У файлі зберігається тільки текст із розбиттям на абзаци, без форматування. Може опрацьовуватися різними текстовими редакторами, які працюють під різними операційними системами
DOCX		Стандартний формат збереження текстових документів у текстовому процесорі Word . У файлі зберігаються текст, малюнки, вставлені об'єкти, значення їхніх властивостей. Забезпечує значне стиснення файлів порівняно з іншими форматами (до 70 %) та краще відновлення файлів після пошкодження
RTF		У файлі зберігаються текст, малюнки, вставлені об'єкти, значення їхніх властивостей. Є кросплатформним – використовується різними системами опрацювання текстів для різних операційних систем
PDF		У файлі зберігаються форматування, усі вставлені об'єкти. Зберігає вигляд документа незалежно від пристрою, операційної системи чи програмного забезпечення. Є стандартом для обміну електронними документами. Для перегляду файлу використовуються програми Adobe Reader, Foxit Reader та інші або браузер
ODT		Стандартний формат текстових файлів, створених текстовим процесором у LibreOffice та OpenOffice (операційна система Linux)

Текстовий процесор **Word** забезпечує роботу з документами, збереженими в різних форматах, наприклад **RTF, TXT, DOC, PDF, ODT** та ін. Щоб зберегти текстовий документ не в стандартному форматі, слід у вікні **Зберегти як** відкрити список **Тип файлу** і вибрати потрібний формат.

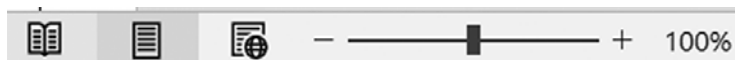
РЕЖИМИ ПЕРЕГЛЯДУ ДОКУМЕНТА



Пригадайте

- Як змінити масштаб перегляду тексту або сторінки, якщо текст погано видно на екрані? Які способи ви знаєте? • Уявіть, що ви готуєте сторінку шкільної газети. Яким чином було б зручно її переглядати?

На початку роботи з документом слід налаштувати його відображення на екрані, установивши зручний для користувача **режим перегляду** документа вибором відповідних кнопок у **Рядку стану** (мал. 2.1).



Мал. 2.1. Кнопки вибору режимів перегляду документа в **Рядку стану**

Для зручності роботи з документом можна встановити необхідний **масштаб** його відображення у вікні. Для цього слід у **Рядку стану** перемістити повзунок установлення масштабу в потрібне місце або збільшити чи зменшити масштаб вибором відповідних кнопок. Також змінити масштаб можна прокручуванням коліщатка миші за натиснутої клавіші **Ctrl**.

Установлення режиму перегляду документа і масштабу його відображення можна здійснити, використовуючи елементи керування відповідних груп вкладки **Подання**. Наприклад, для встановлення масштабу *за шириною сторінки* слід виконати **Подання** ⇒ **Масштаб** ⇒ **За шириною сторінки** ⇄, режиму перегляду *розмітка сторінки* – **Подання** ⇒ **Подання** ⇒ **Розмітка сторінки** [іконка листа з розміткою].

Особливості режимів перегляду документа наведено в таблиці 2.3.

Таблиця 2.3

Особливості режимів перегляду текстового документа

Кнопка	Режим	Особливості режимів
	Макет друку	Відображає текстовий документ так, як він виглядатиме під час друку
	Режим читання	Вигляд тексту оптимізований для читання з екрана: приховано панелі інструментів, адаптовано до розміру екрана
	Вебмакет	Подає документ як вебсторінку: без розбиття на сторінки, з динамічним розташуванням об'єктів
	Структура	Показує ієрархію заголовків, дає змогу легко змінювати порядок розділів та підрозділів
	Чернетка	Подає текст у спрощеному вигляді, без форматування сторінки – для швидкого редагування вмісту

РОБОТА ІЗ ЧАСТИНАМИ АБО КІЛЬКОМА ДОКУМЕНТАМИ ОДНОЧАСНО



Пригадайте

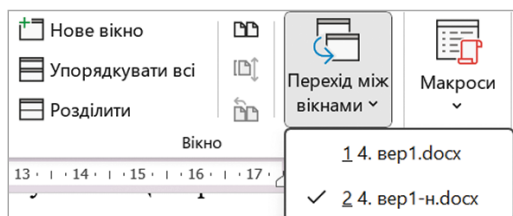
- Які способи перемикання між відкритими вікнами програм ви вже знаєте і використовуєте?
- Чи доводилося вам порівнювати два схожих документи (наприклад, чернетку та остаточну версію)? Як ви це робили?

Під час опрацювання багатосторінкового документа інколи виникає потреба бачити одночасно два різних його фрагменти і переглядати їх незалежно один від одного. Наприклад, коли для написання висновку в кінці тексту твору або реферату потрібно переглянути текст його основної частини на початку документа або коли потрібно порівняти дві різні частини документа для з'ясування відмінностей і виправлень.

Для цього зручно розділити вікно документа на дві частини, виконавши послідовність дій **Подання** ⇒ **Вікно** ⇒ **Розділити** . На екрані з'явиться смуга розділення вікна, яку можна перетягнути по вертикалі в потрібне місце. У кожній частині вікна – зверху та знизу від смуги розділення – є своя смуга прокручування, використовуючи яку можна відобразити на одному екрані дві різні частини документа.

Для повернення до роботи в одному вікні потрібно виконати команду **Зняти поділ** у групі **Вікно** вкладки **Подання** або двічі клацнути на смугі розділення вікна.

У текстовому процесорі **Word** можна відкрити для роботи одночасно кілька документів. Кожен із цих документів буде відкрито в окремому вікні. Щоб зробити вікно одного з відкритих документів поточним, потрібно вибрати кнопку цього вікна на **Панелі задач**.



Мал. 2.2. Група **Вікно**

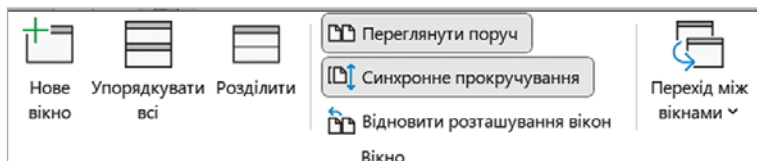
Також для керування відкритими вікнами у програмі існують спеціальні засоби, які розташовані на вкладці **Подання** у групі **Вікно** (мал. 2.2).

Переключення між вікнами у програмі текстового процесора можна здійснити, виконавши такий алгоритм:

1. Вибрати кнопку **Перехід між вікнами**  у групі **Вікно** на вкладці **Подання**.
2. Вибрати у списку відкритих вікон потрібне (поточний документ позначається символом ✓).

Коли робота відбувається з двома версіями одного і того самого документа, що містяться в різних файлах і відкриті в різних вікнах текстового процесора, зручно відобразити їх на екрані поруч. Для виконання цієї операції слід виконати **Подання** ⇒ **Вікно** ⇒ **Переглянути поруч**  і вибрати у списку відкритих документів потрібні. Після цього екран буде розділено вертикально навпіл і в кожній із цих частин буде відображено вікно одного із вказаних документів. Прокручування документів і змінення масштабу перегляду у двох вікнах буде відбуватися синхронно.

Щоб відмінити синхронне прокручування, потрібно вибрати відповідну кнопку в групі **Вікно** (мал. 2.3). Для скасування режиму відображення документів поруч слід повторно вибрати кнопку **Переглянути поруч** у групі **Вікно**.



Мал. 2.3. Команди групи Вікно



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте текстовий процесор **Word**.
2. Відкрийте будь-який файл формату **DOCX** (наприклад, **Розділ 2\Пункт 2.1\вправа 2.1-1.docx**).
3. Перегляньте документ і запишіть у зошит, які об'єкти в ньому є та які значення їхніх властивостей.
4. Збережіть файл у вашій папці з іменем **вправа 2.1-1** у форматі **DOCX, TXT, ODT, RTF, PDF**. Порівняйте значки та розміри цих файлів. Поясніть отриманий результат і запишіть свій висновок у зошит.
5. Відкрийте збережені документи і перегляньте, які зміни відбулися в їх оформленні та змісті під час зміни формату файлу. Запишіть свої спостереження в зошит.
6. Послідовно встановіть для одного з документів на вкладці **Подання** вказані режими роботи з документом і визначте, у якому режимі вам зручніше працювати:

Кнопка	Режим	Кнопка	Режим
	Макет друку		Структура
	Режим читання		Чернетка
	Вебмакет		За шириною сторінки

7. Розділіть вікно документа на дві частини. Для цього:
 1. Виконайте **Подання** ⇒ **Вікно** ⇒ **Розділити**.
 2. Перетягніть лінію розділення на середину екрана.
 3. Перегляньте текст документа у верхній частині екрана, потім – у нижній.
8. Скопіюйте з початку документа перший абзац у кінець документа. Потім останній абзац тексту скопіюйте на його початок.
9. Збережіть текстовий документ у вашій папці у файлі з іменем **вправа 2.1-2**.
10. Відкрийте в текстовому процесорі **Word** два файли (наприклад, із папки **Розділ 2\Пункт 2.1\вправа 2.1-3.docx** і **вправа 2.1-4.docx**). Виконайте переключення перегляду між цими документами. Для цього:

1. Виберіть кнопку **Перехід між вікнами**  у групі **Вікно** на вкладці **Подання**.
 2. Виберіть потрібний документ у списку відкритих вікон.
11. Виконайте синхронний перегляд двох відкритих документів. Для цього:
1. Виконайте **Подання** $\Rightarrow$ **Вікно** $\Rightarrow$ **Переглянути поруч** .
 2. Виберіть у списку відкритих документів потрібний.
 3. Прокрутіть у вікні один з документів. Змініть масштаб перегляду одного з них.
 4. Відмініть синхронне прокручування, вибравши відповідну кнопку на **Стрічці**.
12. Закрийте вікно програми **Word**.



Найважливіше в цьому пункті

Існують різні програми, які призначені для роботи з текстовими документами, їх називають **системами опрацювання текстів**. Використовуючи ці програми, можна створювати текстові документи, зберігати їх, редагувати і формувати їхні об'єкти, здійснювати автоматизовану перевірку правопису, друкувати текстові документи. Використання цих програм значно полегшує оформлення тексту, додавання в текст формул, малюнків, таблиць тощо.

Основним об'єктом опрацювання в цих програмах є **текстовий документ**, у якому можуть розміщуватися різні об'єкти: символи, слова, речення, абзаци, сторінки, малюнки, таблиці, діаграми та інше. Кожен з об'єктів текстового документа має свої властивості.

Існує багато форматів файлів, у яких системи опрацювання текстів зберігають текстові документи. Кожна із цих систем використовує певний набір форматів, які користувач може вибрати для запису файлів залежно від потреб.

Стандартним форматом файлів текстових документів, підготовлених у текстовому процесорі **Word**, є формат **DOCX**. Крім того, **Word** забезпечує роботу з документами, збереженими в інших форматах, наприклад **RTF**, **TXT**, **DOC**, **PDF**, **HTML**, **ODT** тощо.

Для зручності роботи з документом можна встановити необхідний масштаб його відображення у вікні та режим перегляду документа. Для цього призначені такі елементи керування в **Рядку стану**: повзунок установлення масштабу, кнопки **Збільшити** чи **Зменшити** масштаб, кнопки встановлення режиму перегляду.

У текстовому процесорі **Word** можна одночасно працювати з кількома відкритими документами або з різними фрагментами одного документа. Це дає змогу користувачу переглядати документи, порівнювати їхній зміст, уставляти фрагменти одного документа в інший, об'єднувати їх, бачити одночасно у вікні різні фрагменти документа тощо. Для керування відкритими вікнами у програмі існують спеціальні засоби, які розміщені на вкладці **Подання** у групі **Вікно**.



Дайте відповіді на запитання

1. Які програми називають системами опрацювання текстів? Яке їхнє призначення та функції?
2. Які види систем опрацювання текстів ви знаєте? Для чого вони призначені?
3. У яких сферах людської діяльності використовують системи опрацювання текстів? Наведіть приклади.
4. Які об'єкти можна вставляти в текстовий документ та які їхні властивості?

5. У яких форматах може бути збережено файл текстового документа? У чому особливості цих форматів? Який формат є стандартним для текстового процесора **Word**?

6. Які режими перегляду документа можна використати під час роботи в текстовому процесорі? Для яких цілей вони зручніші?

7. Які є способи одночасного відображення вікон відкритих документів на екрані? Як їх установити? Коли зручно це використовувати?



Обговоріть і зробіть висновки

1. Що, на вашу думку, стимулювало розвиток технологій опрацювання тексту і що ви розумієте під терміном «безпаперові технології»?

2. У сучасному світі існує багато платформ, де оформлення тексту не має значення (наприклад, месенджери). Як це впливає на сприйняття змісту повідомлення?

3. Уявіть, що ви створюєте своє власне резюме для участі у проєкті та буклет за результатами проєкту. Які об'єкти доцільно включити в ці документи і з якою метою?

4. Яким чином формат текстового файлу впливає на можливості роботи з документом, на розмір файлу та візуалізацію тексту в документі?

5. Коли доцільніше використати такі режими, як розділення вікна одного документа і перегляд двох окремих документів поруч? Наведіть приклади із власного досвіду.

Резюме – це стислий виклад відомостей про людину (дата й місце народження, освіта, досвід роботи, навички та інше), використовується для працевлаштування та представлення.



Виконайте завдання

1. Відкрийте в текстовому процесорі **Word** файл (наприклад, **Розділ 2\Пункт 2.1\зразок 2.1.docx**). Визначте в **Рядку стану** і запишіть у зошит кількість сторінок, рядків, абзаців, слів, символів у документі. Перегляньте текст документа і запишіть у зошит, які ще об'єкти є в документі та які значення їхніх властивостей. Збережіть документ у форматах: **TXT**, **RTF**, **PDF** та порівняйте розміри цих файлів. Поясніть різницю.

2. Знайдіть на жорсткому диску вашого комп'ютера текстовий файл формату **TXT** і відкрийте його в текстовому процесорі **Word**. Запишіть у зошит послідовність виконаних дій. Перегляньте текстовий документ і запишіть у зошит його особливості.

3. Відкрийте в текстовому процесорі **Word** три текстових файли (наприклад, із папки **Розділ 2\Пункт 2.1** файли **зразок 2.1.1.docx**, **зразок 2.1.2.docx**, **зразок 2.1.3.docx**). Виконайте переключення між вікнами, використовуючи кнопки цих вікон на **Панелі задач** і кнопку **Перехід між вікнами**. Створіть новий текстовий документ і скопіюйте в нього заголовки трьох відкритих документів. Збережіть документ у вашій папці у файлі з іменем **завдання 2.1.3.pdf**.



4. Відкрийте в текстовому процесорі **Word** два файли (наприклад, **Розділ 2\Пункт 2.1** файли **зразок 2.1.1.docx**, **зразок 2.1.2.docx**). Відобразіть вікна цих документів поруч. Виконайте переміщення по документах при включеному та виключеному режимі **Синхронне про-**

кручування. Знайдіть відмінності в першому документі порівняно з другим і виділіть їх кольоровим тлом. Зніміть режим перегляду документів поруч. Збережіть перший документ у вашій папці у файлі з іменем **завдання 2.1.4.rtf**.



5. Об'єднайтеся у кілька груп по 3–4 особи. Знайдіть відомості про інструменти, з використанням яких записували тексти в різні часи. Підготуйте презентацію з п'яти слайдів для представлення своєї відповіді.



6. Об'єднайтеся у кілька груп по 3–4 особи. Знайдіть відомості про різні способи введення тексту в пам'ять комп'ютера (рукописне, голосове, сканування). Яке обладнання та програмне забезпечення для цього потрібно? Підготуйте презентацію з п'яти слайдів для представлення своєї відповіді.



Готуємось до вивчення нового матеріалу

1. Списки яких видів можна використовувати в текстовому документі? Для чого вони призначені?
2. Що таке *приховані (недруковані) символи*? Для чого вони призначені?
3. Чи доводилось вам уставляти в текстовий документ формули? Як ви це виконували?



2.2. СТВОРЕННЯ В ТЕКСТОВОМУ ДОКУМЕНТІ БАГАТОРІВНЕВИХ СПИСКІВ, ВСТАВЛЕННЯ СПЕЦІАЛЬНИХ СИМВОЛІВ І ФОРМУЛ

У цьому пункті йтиметься про:

- ▶ створення в текстовому документі багаторівневих списків;
- ▶ вставлення в текстовий документ спеціальних символів;
- ▶ вставлення формул у текстовий документ.

СТВОРЕННЯ БАГАТОРІВНЕВИХ СПИСКІВ У ТЕКСТОВОМУ ДОКУМЕНТІ



Пригадайте

• Списки яких видів можна створювати в текстовому документі? У яких випадках використовують кожен із них? • Які дії з редагування та форматування однорівневих списків можна виконувати? Яким чином це зробити?

Ви вже знаєте, що одним з особливих видів форматування абзаців текстового документа є оформлення їх у вигляді **списків**, які використовують для наочного подання в тексті переліків об'єктів, описів порядку дій тощо. У текстовому процесорі **Word** можна створювати списки трьох видів: *маркований, нумерований, багаторівневий*.

Марковані та нумеровані списки ви вже створювали раніше, тому ознайомимося зі створенням багаторівневих списків (мал. 2.4).

У багаторівневому списку окремі елементи можуть бути підпорядковані іншим елементам списку, тобто кожен елемент списку може мати свої підсписки. Таке розташування схоже на дерево: є головні гілки – основні елементи списку, і менші гілочки – підсписки, які деталізують головні.

У текстовому процесорі **Word** максимальна кількість вкладень елементів багаторівневого списку – 9 рівнів. І для кожного рівня списку

ЗМІСТ**Розділ 1. КОДУВАННЯ ДАНИХ**

- 1.1. Опрацювання даних як інформаційний процес. Кодування та декодування повідомлень.
- 1.2. Кодування текстових і графічних даних.
- 1.3. Двійкове кодування.

Розділ 2. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРА

- 2.1. Персональний комп'ютер, його основні складові.
- 2.2. Пристрої введення даних.
- 2.3. Пристрої виведення даних.
- 2.4. Історія обчислювальних і комп'ютерних пристроїв.
- 2.5. Види сучасних комп'ютерів та їх застосування.

11. Які правила умовного форматування доступні в Excel?

- а) Колірні шкали.
- б) Гістограми.
- в) Піктограми.
- г) Формули.
- д) Злиття клітинок.

12. Які параметри можна налаштувати перед друком електронної таблиці?

- а) Орієнтацію сторінки
- б) Шрифт тексту
- в) Розмір аркуша
- г) Кількість копій
- д) Поля сторінки

Мал. 2.4. Приклади оформлення багаторівневих списків

можна встановити свій вид маркування чи нумерації, різні відступи. Багаторівневі списки використовують, наприклад, для оформлення розгорнутого плану твору або підручника, списку запитань у тесті.

У текстовому процесорі **Word** є бібліотека шаблонів багаторівневих списків, які користувач може використовувати для оформлення текстового документа.

Для створення багаторівневого списку слід:

1. Установити курсор у тому місці тексту, де буде створюватися багаторівневий список.
2. Відкрити на **Стрічці** на вкладці **Основне** у групі **Абзац** список кнопок **Багаторівневий список**  (мал. 2.5).

3. Вибрати потрібний шаблон списку в розділі **Бібліотека списків**. Після вибору шаблону в тексті з'явиться перший номер (маркер) нового списку.

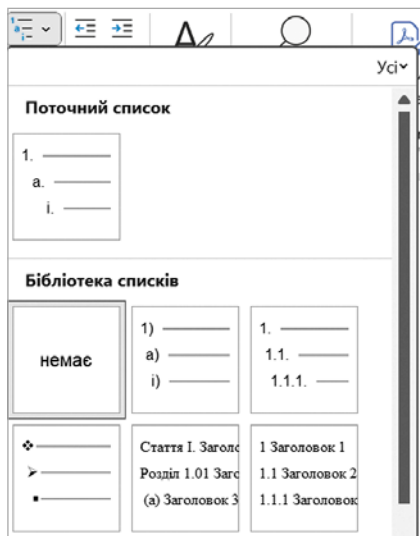
4. Увести елементи списку, змінюючи рівень вкладеності деяких з них, використовуючи кнопки **Зменшити відступ**  (перехід на рівень вище)

або **Збільшити відступ**  (перехід

на рівень нижче), які розташовані на **Стрічці** у групі **Абзац**. Це саме можна зробити, натискаючи клавіші **Tab** і **Shift + Tab**.

Для закінчення введення списку потрібно в наступному за списком рядку тексту повторно натиснути кнопку **Багаторівневий список** .

Змінити рівень вкладеності окремих елементів списку можна в будь-який момент – і під час створення, і в подальшому під час редагування списку.

Мал. 2.5. Відкритий список кнопки **Багаторівневий список**

Операції редагування та форматування багаторівневих списків здійснюються аналогічно до однорівневих.

Іноколи під час оформлення документа виникає потреба перетворити однорівневий список у багаторівневий. Це можна зробити за таким алгоритмом:

1. Виділити створений раніше однорівневий список.
2. Вибрати шаблон багаторівневого списку в списку кнопки **Багаторівневий список** .
3. Змінити рівень вкладеності елементів списку відповідно до потреби, використавши кнопки **Зменшити** / **Збільшити відступ**  .

ВСТАВЛЕННЯ В ТЕКСТОВИЙ ДОКУМЕНТ СПЕЦІАЛЬНИХ СИМВОЛІВ



Пригадайте

● Що таке *символ* у текстовому документі? Які символи ви можете ввести в текстовий документ? ● Що таке *приховані символи*? Як їх уставити в текстовий документ? ● Які ви знаєте таблиці кодів символів? Скільки кодів для символів є в різних таблицях?

Ви вже вмієте вводити в документ текст із клавіатури й оформлювати його певними шрифтами. Але шрифт має більше символів, ніж вміщується на клавіатурі. Такі символи, які не вмістилися на клавіатурі, також можна вставити в документ.

У текстовому процесорі **Word** у документ можна вставляти літери різних алфавітів, математичні символи, умовні позначки та інші. Наприклад, © ≤ β Σ € ∞ ≈  ☺ ♪ §. Це вам знадобиться під час підготовки документів для уроків математики, фізики, хімії та інших. Так само, як у **Word**, ці символи вставляються і в презентації, створені в **PowerPoint**.

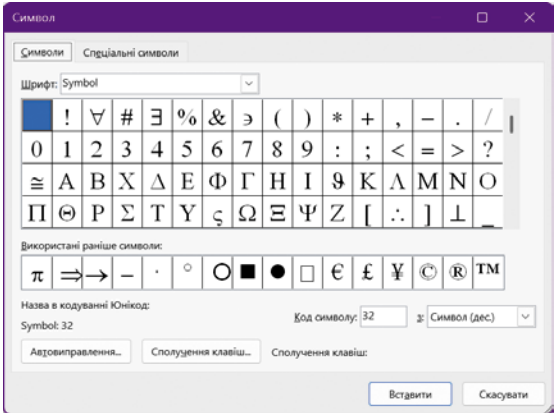
Для вставлення в текст цих символів слід виконати такий алгоритм:

1. Установити курсор у потрібне місце тексту.
2. Вибрати на вкладці **Вставлення** у групі **Символи** кнопку **Символ** .
3. Вибрати потрібний символ у списку, що відкриється (мал. 2.6).
Якщо потрібний символ відсутній у наведеному списку, то слід:
 1. Вибрати команду **Інші символи**.
 2. Переглянути перелік символів у діалоговому вікні **Символ** (мал. 2.7) на вкладках **Символи** та **Спеціальні символи**. За потреби можна у списку поля **Шрифт** або списку **Набір** вибрати інше значення, щоб вивести у вікні інший набір символів.
 3. Вибрати потрібний символ.
 4. Вибрати кнопку **Вставити**.

Тим символам, які будуть використовуватися в тексті часто, можна в цьому вікні **Символ** призначити певні сполучення клавіш (вибрати кнопку **Сполучення клавіш** і заповнити поля вікна **Налаштування клавіатури**). Після натискання вибраного сполучення клавіш потрібний символ вставлятиметься в поточне місце тексту.



Мал. 2.6. Список символів



Мал. 2.7. Вікно Символ

У текстовому процесорі **Word** у документ можна вставляти і символи у вигляді малюнків (піктограм). Наприклад,



Для цього слід вибрати у списку шрифтів один із графічних шрифтів: **MS Outlook**, **Webdings**, **Wingdings** або інший.

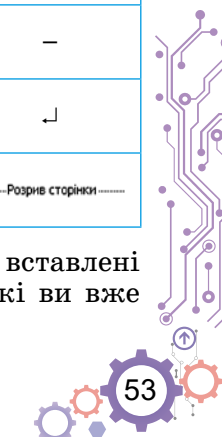
Також до спеціальних символів належать і приховані символи, які ви вже використовували раніше. Нагадаємо, що до **прихованих** (інша назва **недруковані**) належать символи, які вводяться користувачем із клавіатури під час уведення тексту, але зазвичай вони не відображаються на екрані та не виводяться на папері під час друку. Якщо включити **режим відображення прихованих символів** вибором на вкладці стрічки **Основне** у групі **Абзац** кнопки **Відобразити всі знаки ¶**, то у відповідних місцях тексту з'являться спеціальні позначення цих символів (табл. 2.4).

Таблиця 2.4

Приклади прихованих (недрукованих) символів

Символ	Клавіші введення	Позначення	Символ	Клавіші введення	Позначення
Пропуск	Пропуск	.	Нерозривний пропуск	Ctrl + Shift + пропуск	°
Кінець абзацу	Enter	¶	Нерозривний дефіс	Ctrl + Shift + дефіс	—
Табуляція	Tab	→	Розрив рядка	Shift + Enter	↵
М'який перенос	Ctrl + дефіс	~	Розрив сторінки	Ctrl + Enter	Розрив сторінки.....

Редагування тексту і форматування всіх символів, які вставлені в текст будь-яким способом, здійснюються за правилами, які ви вже знаєте.





Для тих, хто хоче знати більше

Деякі послідовності символів під час уведення **Word** автоматично замінює на спеціальні символи. Приклади таких послідовностей наведено в таблиці 2.5.

Таблиця 2.5

Уведення спеціальних символів

Послідовність уведених символів	Спеціальний символ	Послідовність уведених символів	Спеціальний символ
пропуск, дефіс, пропуск	– (тире)	крапка, крапка, крапка	...
(c)	©	-->	→
(r)	®	<--	←
(tm)	™	<==	↔
:)	☺	<=>	↔
:	☹	==>	➔

Режим автоматичної заміни зазначених послідовностей символів на спеціальні символи у програмі **Word** встановлено за замовчуванням. Користувач може за власним бажанням додати до списку автозаміни свої елементи, вибравши кнопку **Автотипування** в діалоговому вікні **Символ**.

ВСТАВЛЕННЯ ФОРМУЛ У ТЕКСТОВИЙ ДОКУМЕНТ



Пригадайте

- Як можна ввести в текст математичні вирази, наприклад, x^3 , x_3 , $3/8$, $\alpha + \beta + \gamma$?
- Який вигляд має формула в табличному процесорі? Які правила їх уведення?

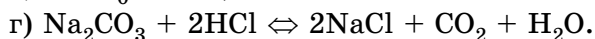
Під час підготовки в текстовому процесорі звітів до практичних і лабораторних робіт, описів виконання проєктів, виступу для учнівської конференції інколи потрібно вставляти формули, рівняння, доведення теорем, що використовують особливу символіку: математичну, фізичну, хімічну та іншу.

Наприклад:

а) $c_2 m_2 (t_2 - \Theta) = c_1 m_1 (\Theta - t_1);$

б) $ax^2 + bx + c \geq 0;$

в) $S = v_0 t + at^2/2;$



Для створення таких виразів, які не мають у своєму складі нелінійних структур, використовують різні спеціальні символи, вставлення яких здійснюється у вікні **Символ (Вставка) ⇒ Символи ⇒ Символ ⇒ Інші символи**). Уведення символів нижнього і верхнього індексу здійснюється з використанням кнопок **Підрядковий символ** x_2 і **Надрядковий символ** x^2 , які розміщено на вкладці **Основне** у групі **Шрифт**.

Але дуже часто в математиці, фізиці трапляються нелінійні, багаторівневі формули та вирази. Наприклад:

$$\text{а) } \frac{h_1}{h_2} = \frac{\rho_2}{\rho_1};$$

$$\text{б) } E_k = \frac{mv^2}{2};$$

$$\text{в) } \eta = \frac{A_1}{A_2} \cdot 100 \%;$$

$$\text{г) } \frac{3x+7}{4} - \frac{x-3}{2} = \frac{5x+2}{8};$$

$$\text{д) } y = \begin{cases} -3x^3 - 2, & \text{якщо } x < 0, \\ 7x - 2, & \text{якщо } x \geq 0; \end{cases}$$

$$\text{е) } \frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a-4b)^2}{9a^4 - b^4}.$$

Уставити в текстовий документ такі формули та вирази можна, скориставшись спеціальним засобом **Конструктор формул**. Формули в ньому конструюються з окремих символів і структур, з використанням відповідних шаблонів.

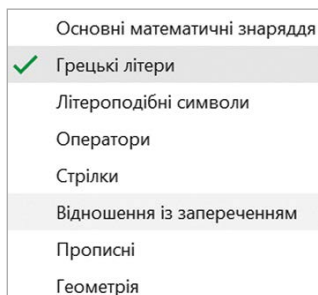
Для відкриття **Конструктора формул** потрібно виконати **Вставлення** $\Rightarrow$ **Символи** $\Rightarrow$ **Рівняння** Π . Після цього в документі з'явиться спеціальна область для введення формул і виразів, а на **Стрічці** тимчасова вкладка **Рівняння** (мал. 2.8).



Мал. 2.8. Вкладка **Рівняння** та область введення формули

Під час введення формули або виразу деякі літери, знаки арифметичних операцій та інші символи можна вводити з клавіатури, а спеціальні – вибирати у відповідних списках групи **Символи**.

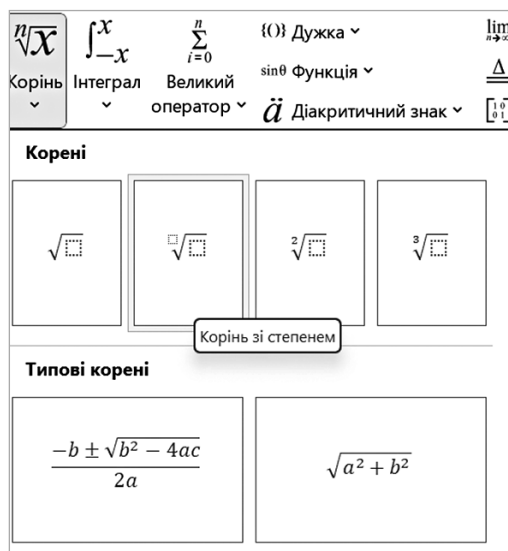
Усі символи, які можна вставити у формулу, розділено на 8 основних наборів (мал. 2.9), які, у свою чергу, можуть бути розділені по окремих блоках.



Мал. 2.9. Набори символів

На **Стрічці** у групі **Символи** відображаються за замовчуванням деякі символи того набору, який використовувався останнім. Їх можна вибрати, і вони вставляються в документ. Для перегляду інших символів слід скористатися кнопкою **Символи для рівнянь** $\approx$ у групі **Символи**. Щоб відкрити перелік назв усіх 8 наборів символів, потрібно в поточному списку вибрати його заголовок.

Крім окремих символів, у виразі можуть використовуватися дробі, функції, дужки, знаки коренів тощо. Вони вставляються у вираз із використанням елементів керування групи **Структури** вкладки **Рівняння**. Усі структури розділені за типами, список шаблонів яких відкривається після вибору відповідних кнопок у групі **Структури** (мал. 2.10). Щоб уставити шаблон у формулу, слід відкрити список шаблонів відповідної кнопки і вибрати потрібний.



Мал. 2.10. Група **Структури** та список шаблонів кнопки **Корені**

Далі слід заповнити шаблон даними. Місця для введення чисел і символів у шаблоні позначені спеціальними полями, які обведено пунктирною рамкою (мал. 2.11). У поля можна не тільки вводити окремі символи, а й вставляти інші шаблони.

Переміщення в межах області введення формули здійснюється клавішами керування курсором або вказівником. Для виходу з режиму створення формул потрібно вибрати точку за межами області введення формули.

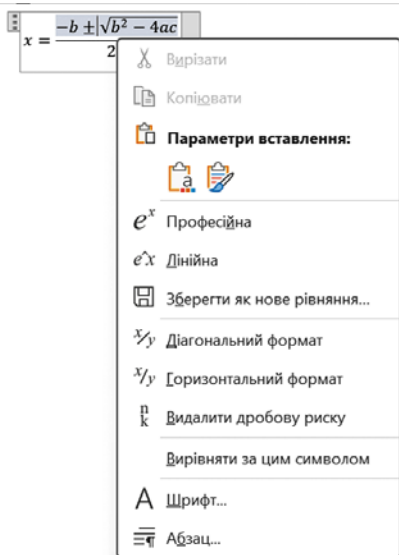
Редагування та форматування створеної **формули чи виразу** або їхніх фрагментів здійснюються стандартними засобами **Word**. Для цього використовують елементи керування вкладки **Основна** та команди контекстного меню формули (мал. 2.12).

Створену формулу користувач може додати до бібліотеки вбудованих формул і використовувати в подальшому за потреби. Для цього слід:

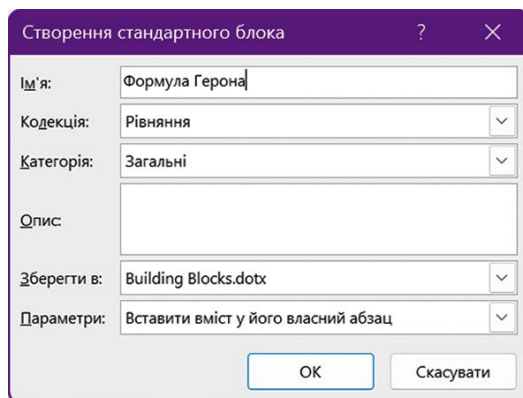
Поля для введення даних



Мал. 2.11. Шаблон для введення даних



Мал. 2.12. Контекстне меню формули



Мал. 2.13. Додавання нової формули до стандартного списку

1. Відкрити контекстне меню формули.
2. Вибрати команду **Зберегти як нове рівняння**.
3. Заповнити поле **Ім'я** діалогового вікна **Створення стандартного блока** (мал. 2.13), увівши назву формули чи рівняння.
4. За потреби вибрати в інших полях потрібні значення із запропонованих списків.
5. Вибрати кнопку **ОК**.



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте вказаний учителем / учителькою файл (наприклад, **Розділ 2\Пункт 2.2\вправа 2.2-1.docx**).
2. Оформте наведений у файлі текст у вигляді багаторівневого списку за зразком (див. зразок). Для цього:
 1. Виділіть увесь наведений текст без рядка заголовка.
 2. Відкрийте список кнопки **Багаторівневий список** , яка розміщена у групі **Абзац** вкладки **Основне**.
 3. Виберіть у **Бібліотеці списків** потрібний вид списку .
 4. Виділіть фрагмент тексту від слова *повені* до слів *сильні зливи*.

Види надзвичайних ситуацій та їх приклади

1. Надзвичайні ситуації природного характеру:
 - 1.1. повені,
 - 1.2. епідемії,
 - 1.3. обвали,
 - 1.4. снігопади,
 - 1.5. землетруси,
 - 1.6. урагани,
 - 1.7. сильні зливи.
2. Надзвичайні ситуації техногенного характеру:
 - 2.1. транспортні аварії (катастрофи),
 - 2.2. пожежі,
 - 2.3. неспровоковані вибухи,
 - 2.4. викид токсичних і радіоактивних речовин.
3. Надзвичайні ситуації соціального характеру:
 - 3.1. терористичні акти,
 - 3.2. напади,
 - 3.3. крадіжки,
 - 3.4. пограбування,
 - 3.5. захоплення заручників,
 - 3.6. установлення вибухових пристроїв у громадських місцях.
4. Надзвичайні ситуації військового характеру:
 - 4.1. застосування зброї масового ураження.

Зразок

5. Збільште рівень для цього фрагмента, вибравши на вкладці

Основне у групі **Абзац** кнопку **Збільшити відступ** .

6. Виконайте аналогічні дії для інших фрагментів списку.

3. Відформатуйте нумерацію на першому рівні створеного списку за такими значеннями властивостей: шрифт – *Verdana*, розмір – 14, колір – червоний; на другому: шрифт – *Arial*, розмір – 12, колір – синій. Для цього:

1. Виберіть будь-який номер першого рівня списку і встановіть потрібні значення властивостей символів на вкладці **Основне**.

2. Виберіть будь-який номер другого рівня списку і встановіть потрібні значення властивостей символів на вкладці **Основне**.

4. Увімкніть режим відображення недрукованих символів. Для цього виконайте **Основне** $\Rightarrow$ **Абзац** $\Rightarrow$ **Відобразити всі знаки** . Проаналізуйте, які приховані символи було введено, де вони розміщені в тексті. Запишіть у зошит, яка роль цих символів у тексті.

5. Збережіть документ у вашій папці у файлі з іменем **вправа 2.2-1**.

6. Створіть новий текстовий документ і введіть кілька символів, які відсутні на клавіатурі, наприклад: $\pi \Sigma \in \int \approx \infty \S \frac{1}{4} \pm \zeta \uparrow \blacktriangleright \text{♪} \heartsuit$. Для цього:

1. Виберіть на вкладці **Вставлення** у групі **Символи** кнопку **Символ** .

2. Виберіть команду **Інші символи**.

3. Перегляньте перелік символів у діалоговому вікні **Символ** на вкладках **Символи** та **Спеціальні символи** і знайдіть потрібні. За потреби змініть шрифт у списку **Шрифт**.

4. Виберіть потрібний символ і кнопку **Вставити** (повторити для кожного символу окремо).

7. Уведіть у наступному рядку послідовності символів, які вказано в таблиці 2.4, та переконайтеся в їх автоматичній заміні на спеціальні символи.

8. Збережіть документ у вашій папці у файлі з іменем **вправа 2.2-2**.

9. Створіть у новому текстовому документі математичну рівність за зразком:

$$\frac{3x + 7}{4} = (x - 3)^2.$$

Для цього:

1. Розгляньте уважно і з'ясуйте, з яких елементів складається математична рівність, які шаблони слід використати і в якій послідовності.

2. Виконайте **Вставлення** $\Rightarrow$ **Символи** $\Rightarrow$ **Рівняння**.

3. Виберіть на вкладці **Рівняння** у групі **Структури** кнопку **Дроби** .

4. Оберіть шаблон уставки дробу .

5. Виберіть вказівником поле в чисельнику і введіть із клавіатури вираз $3x + 7$.

6. Виберіть вказівником поле у знаменнику і введіть число 4.
 7. Перша частина рівності закінчилася. Для переходу до наступної частини натисніть клавішу $\rightarrow$ і введіть знак $=$. *Зверніть увагу*, курсор повинен стояти після риски дробу, висота курсора – як висота дробу.
 8. Виберіть на вкладці **Рівняння** у групі **Структури** кнопку **Індекс e^x** .
 9. Оберіть шаблон для вставлення виразу з верхнім індексом $\frac{\square}{\square}$.
 10. Уведіть в основу степеня вираз $(x - 3)$, у показник степеня – число 2.
 11. Після завершення введення формули виберіть місце поза межами рамки області введення формули в робочій області документа.
 12. Установіть для формули такі значення властивостей символів: розмір – *16 pt*, накреслення – *напівжирний*, колір – *фіолетовий*.
10. Збережіть текстовий документ у вашій папці у файлі з іменем **вправа 2.2-3**.
11. Закрийте вікно програми текстового процесора **Word**.



Найважливіше в цьому пункті

Абзаци тексту можна оформити у вигляді списків, зокрема багаторівневих, які використовують для наочного подання ієрархічного переліку об'єктів. Для цього слід виділити фрагмент тексту, виконати **Оснoвне** $\Rightarrow$ **Абзац** $\Rightarrow$ **Багаторівневий список** і вибрати в **Бібліотеці списків** потрібний вид. Змінення рівня вкладеності елементів списку здійснюється кнопками **Зменшити відступ** (перехід на рівень вище) або **Збільшити відступ** (перехід на рівень нижче), які розташовані на **Стрічці** у групі **Абзац**.

У текстовому процесорі **Word** можна в документ вставляти спеціальні символи, які відсутні на клавіатурі, – літери різних алфавітів, математичні символи, умовні позначки та інші. Вибрати потрібні символи можна у вікні **Символ**, яке відкривається виконанням **Вставлення** $\Rightarrow$ **Символи** $\Rightarrow$ **Символ** $\Rightarrow$ **Інші символи**.

Для створення в текстовому документі нескладних формул лінійного вигляду використовують різні спеціальні символи, уставка яких здійснюється з використанням вікна **Символ**. Уведення символів нижнього і верхнього індексів здійснюється з використанням кнопок **Підрядковий символ** і **Надрядковий символ**, які розміщені на вкладці **Оснoвне** у групі **Шрифт**.

Уведення в текстовий документ складних багаторівневих формул здійснюється спеціальним засобом **Конструктор формул**, який відкривається вибором кнопки **Рівняння** на вкладці **Вставлення**. Формули в ньому конструюються з окремих символів і структур, використовуючи відповідні шаблони, які потрібно заповнити даними.

Літери, знаки арифметичних операцій та інші символи можна вводити з клавіатури, а спеціальні – вибирати у відповідних списках групи **Символи**. Звичайні дробі, функції, дужки, знаки коренів уставляються за допомогою елементів керування групи **Структури** тимчасової вкладки **Рівняння**. Усі структури розділено за типами, список шаблонів яких відкривається під час вибору відповідних кнопок на вкладці.



Дайте відповіді на запитання

1. Який список є багаторівневий? Як його створити? Як змінити рівень вкладеності елементів списку?
2. Які спеціальні символи можна вставити в текстовий документ? Як це зробити? Коли це може вам знадобитися?

3. Які набори математичних символів використовують у формулах? Як уставити потрібний символ?

4. Які структури можна вводити у формули та вирази в **Конструкторі**?

5. Які операції редагування та форматування можна виконувати над формулою в текстовому документі? Як вони здійснюються?



Обговоріть і зробіть висновки

1. У чому полягає доцільність використання багаторівневих списків замість однорівневих? У яких ситуаціях це особливо важливо?

2. Чому доцільно включати відображення прихованих символів під час оформлення та редагування тексту? З якою метою їх використовують?

3. Чи може надмірне використання спеціальних символів або піктограм зробити текст менш зрозумілим? У яких випадках це буде виправдано, а в яких – ні? Врахуйте також спілкування в месенджерах і соціальних мережах.

4. До якого типу об'єктів належить об'єкт *формула*? Які властивості має цей об'єкт?

5. Як оформлення формул у **Word** полегшує підготовку рефератів або лабораторних звітів?



Виконайте завдання

1. Створіть новий документ за зразком (мал. 2.14). Установіть такі відступи елементів списку: відступ першого рівня – 2 см, відступ другого рівня – 3 см від лівого поля документа. Збережіть документ у вашій папці у файлі з іменем **завдання 2.2.1**.

2. Створіть новий документ і оформте в ньому один багаторівневий список за нижченаведеним текстом (мал. 2.15). Оформте список на власний розсуд. Збережіть документ у вашій папці у файлі з іменем **завдання 2.2.2**.



3. Створіть новий текстовий документ і введіть текст, у якому замість деяких слів використайте спеціальні символи – так, як інколи робите в месенджерах. Збережіть документ у вашій папці у файлі з іменем **завдання 2.2.3**.

1. Сніданок

- 1.1. Чай/кава
- 1.2. Бутерброд з маслом

2. Обід

- 2.1. Борщ
- 2.2. Картопляники
- 2.3. Салат з капусти
- 2.4. Компот

3. Вечеря

- 3.1. Кефір
- 3.2. Булочка

Мал. 2.14. Зразок до завдання 1

Кольори умовно поділяють на **теплі** (червоний, оранжевий, жовтий) і **холодні** (блакитний, синій, фіолетовий), а також на **легкі** та **важкі**.

Легкі кольори – прозорі, холодні, світлі, асоціюються з небом і простором.

Важкі кольори – темні, насичені, густі; до таких належать: чорний, темно-сірий, фіолетовий, темно-червоний.

Мал. 2.15. Зразок до завдання 2



4. Створіть новий текстовий документ. Уведіть наведені схеми хімічних реакцій. Збережіть документ у вашій папці у файлі з іменем **завдання 2.2.4**.

- а) $O_2 + 2H_2 \rightarrow 2H_2O$; в) $K + H_2O \rightarrow KOH + H_2\uparrow$;
 б) $AgCl \rightarrow Ag + Cl_2$; г) $KMnO_4 \rightarrow K_2MnO_4 + MnO_2 + O_2\uparrow$.



5. Створіть новий текстовий документ. Уставте в нього наведені фізичні формули. Збережіть документ у вашій папці у файлі з іменем **завдання 2.2.5**.

- а) Період обертання $T = \frac{t}{N}$;
 б) Коефіцієнт корисної дії $\eta = \frac{A_1}{A_2} \cdot 100 \%$;
 в) Формула сполучних посудів $\frac{h_1}{h_2} = \frac{\rho_2}{\rho_1}$;
 г) Кінетична енергія $E_k = \frac{mv^2}{2}$.



6. Створіть новий текстовий документ. Уставте в нього наведені математичні вирази. Збережіть документ у вашій папці у файлі з іменем **завдання 2.2.6**.

- а) $\frac{2}{3}(x-4) + \frac{1}{3} = 5$; в) $\left(\sqrt{a} + \frac{1}{\sqrt{a+1}}\right)^2$;
 б) $\frac{\sqrt{(x+10)^5}}{10x}$; г) $\left(\frac{7}{x}\right)^{-8} = 5x^3$.



7. Створіть новий текстовий документ. Уставте в нього наведені математичні вирази та рівняння. Збережіть документ у вашій папці у файлі з іменем **завдання 2.2.7**.

- а) $y = \begin{cases} x^2 + 3, & \text{якщо } x \leq 0, \\ \frac{x}{2} + 3, & \text{якщо } x > 0; \end{cases}$ в) $y = \begin{cases} -3x^3 - 2, & \text{якщо } x < 0, \\ \frac{7x}{4} - 2, & \text{якщо } x \geq 0; \end{cases}$
 б) $y = \begin{cases} 3x - y = 1, \\ x + y = 3, \\ 4y = 8; \end{cases}$ г) $y = \begin{cases} x - y = 1, \\ x + y = 3, \\ 2x = 6. \end{cases}$



8. Дослідіть, як використовувати рукописне введення формул у текстовий документ (**Вставлення** $\Rightarrow$ **Символи** $\Rightarrow$ **Рівняння** $\Rightarrow$ **Рукописне рівняння**). Створіть за його допомогою 3–4 формули, які наведено в завданнях 6 і 7. Збережіть документ у вашій папці у файлі з іменем **завдання 2.2.8**.



Готуємось до вивчення нового матеріалу

1. Які властивості має сторінка текстового документа? Як можна встановити значення властивостей?
2. Яке призначення маркерів абзацу на горизонтальній лінійці?
3. Як текст ділиться на сторінки?

2.3. СТВОРЕННЯ В ТЕКСТОВОМУ ДОКУМЕНТІ СТОВПЦІВ, РОЗДІЛІВ, КОЛОНТИТУЛІВ

У цьому пункті йтиметься про:

- ▶ створення розділів у текстовому документі;
- ▶ створення стовпців у текстовому документі;
- ▶ вставлення колонтикулів у текстовий документ.

СТВОРЕННЯ РОЗДІЛІВ У ТЕКСТОВОМУ ДОКУМЕНТІ



Пригадайте

- Як текст розбивається на сторінки? Чи можна змінити таку розбивку?
- Як відобразити приховані (недруковані) символи?

Зазвичай у текстовому документі всі сторінки документа відформатовано однаково (розмір аркуша, розміри полів, орієнтація сторінки тощо). Однак під час роботи з багатосторінковими документами може виникнути потреба, щоб деякі сторінки документа мали інші значення властивостей. Наприклад, у тексті з книжковою орієнтацією сторінок потрібно вставити таблицю чи малюнок на сторінці з альбомною орієнтацією або деякий фрагмент документа треба розмістити в кілька стовпців. Таке форматування документа можна зробити, розділивши документ на *розділи*.

Щоб поділити текстовий документ на розділи, у тому місці, де починатиметься інший розділ, потрібно вставити **розрив розділу**. Якщо розділів у документі має бути кілька, то для кожного з розділів уставляється свій розрив. За відсутності розривів розділів текстовий документ розглядається як один розділ.

Створення розділів бажано виконувати в режимі перегляду документа **Розмітка сторінки (Макет друку)** і за включеного режиму відображення прихованих (недрукованих) символів. Потім потрібно виконати **Макет** ⇒ **Параметри сторінки** ⇒ **Розриви**  і вибрати у списку потрібний вид із переліку **Розриви розділів**.

Існує кілька видів розривів розділів:

- *з наступної сторінки* – новий розділ починається з початку наступної сторінки;
- *неперервно* – новий розділ починається в поточному місці документа;
- *з парної сторінки* – новий розділ починається з наступної парної сторінки;
- *з непарної сторінки* – новий розділ починається з наступної непарної сторінки.

У місці розриву розділів з'являється прихований (недрукований) символ **Розрив розділу**, наприклад

—————**Розрив розділу (з наступної сторінки)**—————.

Для видалення розриву розділів потрібно видалити в документі прихований символ **Розрив розділу**. Водночас фрагмент документа над місцем видаленого розриву стає частиною наступного розділу і його форматування автоматично змінюється відповідним чином.

Форматування для поточного розділу здійснюється аналогічно форматуванню сторінки на вкладці **Макет** у групі **Параметри сторінки**.

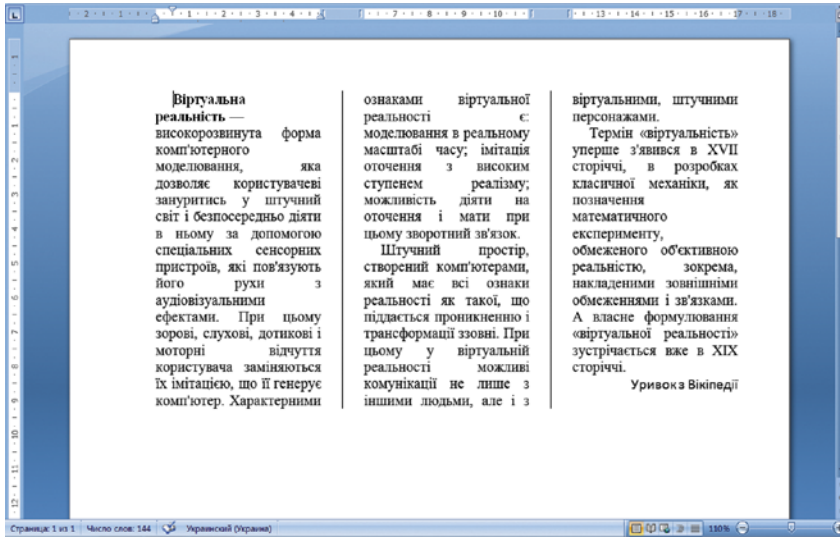
СТВОРЕННЯ СТОВПЦІВ У ТЕКСТОВОМУ ДОКУМЕНТІ



Пригадайте

- Чим відрізняється оформлення тексту в підручнику та в газеті чи журналі від звичайного тексту?
- Яке призначення маркерів абзацу на горизонтальній лінійці?

У текстовому процесорі **Word** можна розмістити текст документа в кількох стовпцях (мал. 2.16).

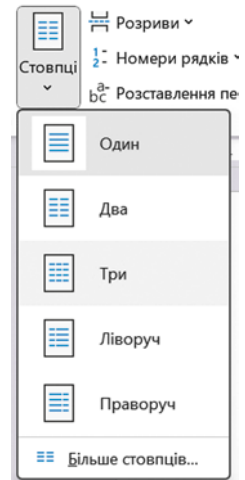


Мал. 2.16. Приклад тексту, який розміщено у трьох стовпцях

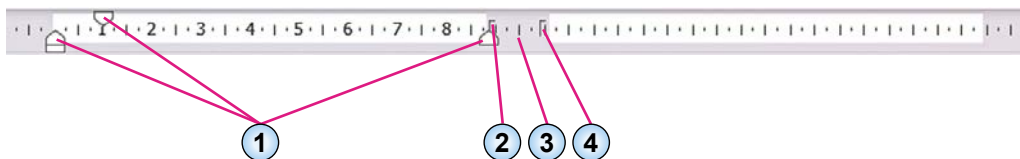
Щоб деякий фрагмент тексту розмістити в кількох стовпцях, його слід виділити і виконати послідовність дій **Макет** ⇒ **Параметри сторінки** ⇒ **Стовпці** . Потім у списку потрібно вибрати необхідну кількість стовпців (мал. 2.17). Текст автоматично буде розміщено відповідно до вибраної кількості стовпців.

Щоб скасувати розміщення фрагмента тексту в стовпцях, цей фрагмент потрібно виділити і встановити для нього режим відображення в один стовець. При цьому в тексті перед першим рядком і після останнього рядка фрагмента тексту залишаються приховані символи **Розрив розділу**, які бажано видалити.

Коли курсор міститься в тексті, розміщеному в стовпцях, на горизонтальній лінійці, крім маркерів абзацу, з'являються **маркери меж** стовпців (мал. 2.18), перетягуючи які можна змінювати ширину стовпців та відстань між ними.

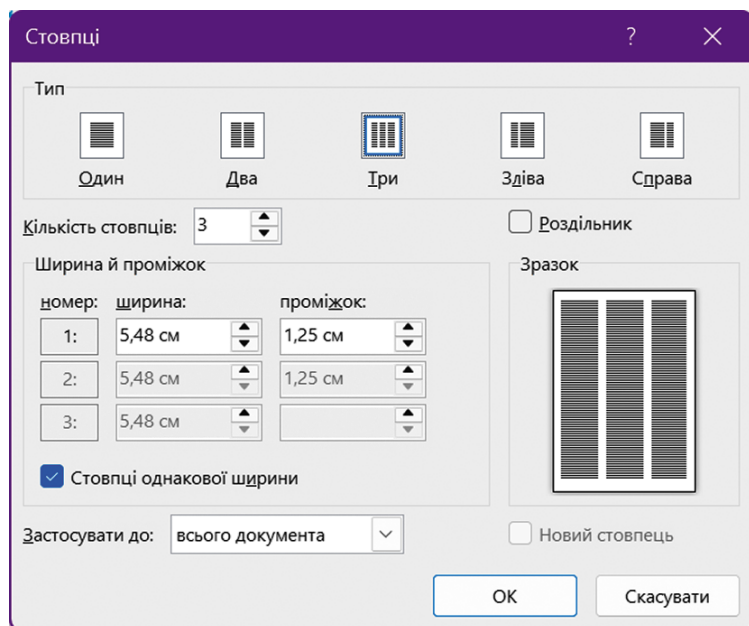


Мал. 2.17. Вибір кількості стовпців у списку кнопки **Стовпці**



Мал. 2.18. Маркери меж стовпців на горизонтальній лінійці:
1 – маркери абзаців; 2 – маркер правої межі першого стовпця;
3 – проміжок між стовпцями; 4 – маркер лівої межі другого стовпця

Форматування стовпців можна виконати й у діалоговому вікні **Стовпці** (мал. 2.19), яке відкривається **Макет** ⇒ **Параметри сторінки** ⇒ **Стовпці** ⇒ **Більше стовпців**. Тут можна встановити кількість стовпців, ширину кожного з них, відстань між окремими стовпцями, наявність роздільвача (на мал. 2.16 роздільник – це вертикальна лінія між стовпцями). Після встановлення значень властивостей стовпців потрібно вибрати кнопку **ОК**.



Мал. 2.19. Вікно **Стовпці**

УСТАНОВЛЕННЯ КОЛОНТИТУЛІВ У ТЕКСТОВИЙ ДОКУМЕНТ



Поміркуйте

- Роздивіться уважно верхню частину сторінки підручника. Які елементи оформлення ви можете назвати? Для чого їх використовують?

У багатосторінкових документах часто використовують **колонти-тули** (франц. *colonne* – стовець, колонка, лат. *titulus* – заголовок) – службові повідомлення, які розміщують на полях сторінки і повторюють на інших сторінках документа. Розрізняють *верхній*, *нижній* та *бічні* колонтидули. Колонтидули можуть містити номери сторінок,

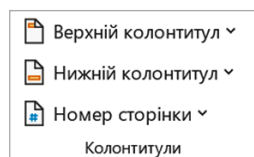
назву документа або поточного розділу, прізвище автора, графічні зображення тощо.

Колонтитули першої сторінки, парних і непарних сторінок можуть розрізнятися. Також можуть бути різними колонтитули різних розділів документа. Наприклад, у цьому підручнику різні розділи мають різні колонтитули.

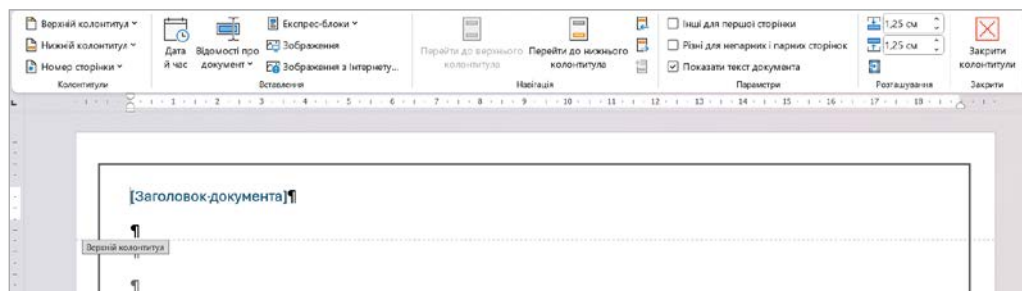
На сторінки текстового документа у **Word** можна вставляти колонтитули, скориставшись готовою колекцією шаблонів колонтитулів або створити власні, які можна зберегти в колекції. Відображаються колонтитули в документі тільки в режимах **Розмітка сторінки (Макет друку)** та **Режим читання**. Опрацювання основної частини документа під час роботи з колонтитулами неможливе.

Для створення на кожній сторінці документа колонтитула з готової колекції шаблонів слід:

1. Вибрати на **Стрічці** вкладку **Вставлення**.
2. Вибрати у групі **Колонтитули** одну з команд **Верхній колонтитул** або **Нижній колонтитул** (мал. 2.20).
3. Вибрати в запропонованому списку шаблонів колонтитулів потрібний варіант.
4. Увести потрібний текст у поля шаблону, за потреби використовуючи елементи керування тимчасової вкладки **Колонтитули**, яка з'являється на **Стрічці** (мал. 2.21).



Мал. 2.20. Група Колонтитули



Мал. 2.21. Шаблон верхнього колонтитула і тимчасова вкладка Колонтитули

5. Закрити вкладку колонтитула, виконавши **Колонтитули** ⇒ **Закрити** ⇒ **Закрити колонтитули** або двічі клацнувши на зоні основного тексту.

Уміст колонтитула вводиться в поле, яке обмежене штриховою лінією. Використовуючи кнопки груп **Колонтитули** та **Вставлення**, у колонтитул можна вставити номер сторінки – кнопка **Номер сторінки** ,

поточну дату та час – кнопка **Дата й час** , малюнок –

кнопка **Зображення** та інше. Розміщення вмісту колонтитула відносно верхнього чи нижнього краю сторінки регулюється відповідними лічильниками 1,25 см у групі **Розташування**.

Для переходу з поля верхнього колонтитула до поля нижнього колонтитула і назад використовують кнопку **Перейти до Верхнього / Нижнього колонтитула**  у групі **Навігація** вкладки **Колонтитули**.

Створений колонтитул можна застосувати до всього документа, до парних чи непарних сторінок або до першої сторінки. Вибрати область застосування створеного колонтитула можна у групі **Параметри** вкладки **Колонтитули**.

Якщо колонтитули для різних частин документа повинні бути різними, то перед їх створенням документ слід поділити на *розділи* і відключити кнопку **Як у попередньому**  в групі **Навігація** на вкладці **Колонтитули**.

Для змінення вставленого колонтитула слід виконати **Вставлення** ⇒ **Колонтитули** ⇒ **Верхній колонтитул (Нижній колонтитул)** ⇒ **Змінити верхній (нижній) колонтитул** і виконати потрібне редагування. Для видалення колонтитула слід виконати **Вставлення** ⇒ **Колонтитули** ⇒ **Верхній колонтитул (Нижній колонтитул)** ⇒ **Видалити верхній (нижній) колонтитул**.



Для тих, хто хоче знати більше

Табуляція та її використання

Ви знаєте, що натискання на клавішу **Tab** переміщує курсор уздовж рядка тексту на задану відстань. За замовчуванням ця відстань становить 1,25 см.

Натисканню клавіші **Tab** відповідає прихований символ →. Використовуючи клавішу **Tab**, можна розташувати об'єкти тексту в певних місцях рядка, установивши для цього на лінійці кілька позначок табуляції в потрібних місцях. Простір між об'єктами в рядку, який утворюється після натискання клавіші **Tab**, можна заповнити послідовністю деяких символів (крапками, тире, підкресленням тощо). Наприклад:

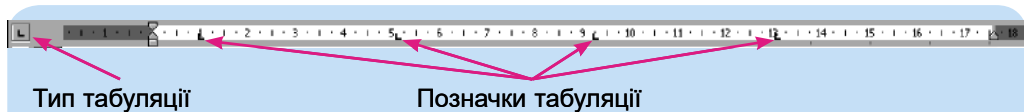
- | | | | | |
|----|-------------|--------------|-------------|--------------|
| 1) | Зима | Весна | Літо | Осінь |
| | Грудень | Березень | Червень | Вересень |
| | Січень | Квітень | Липень | Жовтень |
| | Лютий | Травень | Серпень | Листопад |
- 2) Директор ЗАТ _____ І. В. Сидоренко
- 3) Розділ 1 с. 1
 Розділ 2 с. 8
 Розділ 3 с. 15

У текстовому процесорі **Word** можна використовувати кілька типів табуляції, які вибираються кнопкою **Тип табуляції**, що розміщена ліворуч від горизонтальної лінійки.

Якщо послідовно вибирати цю кнопку, то тип табуляції буде змінюватися і можна встановити:

- | | |
|--|---|
|  – вирівнювання зліва; |  – вирівнювання по центру; |
|  – вирівнювання справа; |  – вирівнювання за розділювачем; |
|  – табуляція з рискою. | |

Після цього на лінійці відмічають позицію табуляції, вибираючи потрібну позицію вказівником миші. У результаті таких дій на лінійці у вибраних місцях з'являються відповідні позначки (мал. 2.22).



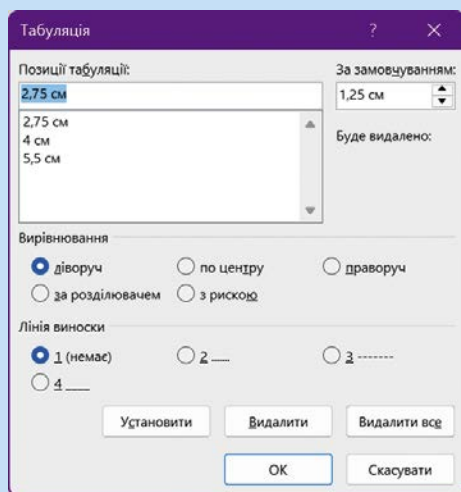
Мал. 2.22. Горизонтальна лінійка з позначками табуляції

Для встановлення або змінення значень властивостей табуляції потрібно відкрити діалогове вікно **Табуляція** (мал. 2.23), двічі клацнувши будь-яку позицію табуляції на лінійці. У цьому вікні для кожної позиції табуляції можна встановити значення таких її властивостей:

- відстань від межі лівого поля (за замовчуванням вимірюється в сантиметрах);
- спосіб вирівнювання тексту відносно позиції табуляції;
- символ – заповнювач простору між об'єктами в рядку, що утворюється після натискання клавіші **Tab**.

Установивши значення властивостей для нової позиції табуляції, слід вибрати кнопку **Установити**, і нова позиція буде внесена до списку **Позиції табуляції**. Щоб змінити значення властивостей уже встановленої позиції табуляції, потрібно вибрати її у списку **Позиції табуляції** та встановити нові значення її властивостей.

Кнопка **Видалити** цього вікна видаляє зі списку вибрану позицію табуляції. Усі встановлені позиції табуляції можна видалити кнопкою **Видалити все**.



Мал. 2.23. Вікно встановлення позицій табуляції



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте запропонований файл (наприклад, **Розділ 2\Пункт 2.3\вправа 2.3.docx**).
2. Уставте розриви розділів з нової сторінки в місцях початку нових пунктів тексту. Для цього для кожного розділу встановіть курсор перед заголовком нового пункту і виконайте **Макет** ⇒ **Параметри сторінки** ⇒ **Розриви** ⇒ **Розриви розділів** ⇒ **Наступна сторінка**.
3. Увімкніть режим відображення прихованих (недрукованих) символів. Перегляньте документ, зверніть увагу на появу в тексті прихованих символів **Розрив розділу**.
4. Установіть у другому розділі документа *альбомну* орієнтацію аркуша та всі поля *по 1 см* у групі **Параметри сторінки** на вкладці **Макет**.
5. Розмістіть текст першого розділу в 2 стовпці. Для цього:
 1. Виділіть увесь текст першого розділу без рядка заголовка.
 2. Виконайте **Макет** ⇒ **Параметри сторінки** ⇒ **Стовпці** ⇒ **Інші стовпці**.

6. Установіть такі значення властивостей у вікні **Стовпці**: кількість стовпців – 2, стовпці різної ширини – 3 см і 14 см, з розділювачем.
7. Перегляньте документ і з'ясуйте, які нові символи з'явилися в тексті.
8. Створіть у документі *верхній* колонтитул з вирівнюванням по *центру* і запишіть у нього назву документа. Для цього:
 1. Виконайте **Вставлення** ⇒ **Колонтитули** ⇒ **Верхній колонтитул**.
 2. Виберіть у списку шаблонів колонтитулів потрібний (наприклад, *Остін*).
 3. Уведіть у змісті колонтитула назву відкритого текстового документа.
9. Створіть у документі *нижній* колонтитул і вставте в нього сьогоденішню дату з вирівнюванням по *лівому* краю. Для цього:
 1. Виберіть кнопку **Перейти до Нижнього колонтитула** у групі **Навігація** вкладки **Колонтитули**.
 2. Виберіть кнопку **Дата й час** у групі **Вставлення** на вкладці **Колонтитули**.
 3. Виберіть короткий формат відображення дати та кнопку **ОК**.
 4. Закрийте вікно колонтитулів, вибравши кнопку **Закрити колонтитули**.
10. Перегляньте документ, зверніть увагу на розміщення в документі верхнього та нижнього колонтитулів.
11. Збережіть документ у вашій папці у файлі з іменем **вправа 2.3**.
12. Закрийте вікно текстового процесора **Word**.



Найважливіше в цьому пункті

У багатосторінкових текстових документах можна змінювати значення властивостей окремих сторінок (орієнтацію, поля тощо), поділивши текст на *розділи*. Для цього вставляють *Розрив розділу* в потрібному місці, виконавши **Макет** ⇒ **Параметри сторінки** ⇒ **Розрив**. Існує кілька типів розривів розділів: з *наступної, непарної, парної сторінки* або *неперервний*. Видалення розриву об'єднує текст у єдиний розділ із загальним форматкуванням.

Також можна розмістити фрагмент тексту в кілька стовпців, так як це роблять у газетах і журналах. Для цього слід виділити потрібний фрагмент тексту, виконати **Макет** ⇒ **Параметри сторінки** ⇒ **Стовпці** ⇒ **Більше стовпців**, установити у вікні **Стовпці** кількість колонок, ширину кожної з них, відстань між окремими колонками, наявність розділювача.

У багатосторінкових документах часто використовують **колонтитули** – це службові повідомлення, які розміщують на полях сторінки і можуть повторювати на інших сторінках документа. У **Word** розрізняють верхній, нижній та бічні колонтитули. Вони можуть містити номери сторінок, назву документа або поточного розділу, прізвище автора, графічні зображення тощо. Колонтитули першої сторінки, парних і непарних сторінок можуть відрізнятися. Також можуть бути різними колонтитули різних розділів документа.

Відображаються колонтитули в документі тільки в режимах **Розмітка сторінки (Макет друку)** та **Режим Читання**. Опрацювання основного тексту документа під час роботи з колонтитулами неможливе. Для створення колонтитулів використовують елементи керування вкладки **Вставлення** групи **Колонтитули**.



Дайте відповіді на запитання

1. Що таке *розділ у текстовому документі*? Для чого їх використовують?
2. Які види розділів можна створити в текстовому документі? Чим вони відрізняються?
3. Як розмістити текст у кілька стовпців? Значення яких властивостей стовпців можна змінити?
4. Як можна встановити ширину стовпців, відступи між ними, їхню кількість?
5. Що таке *колонтикули*? Де можуть бути розташовані колонтикули? Які об'єкти можна вставити в колонтикули?
6. Як установити різні колонтикули для різних сторінок документа?



Обговоріть і зробіть висновки

1. Чим розриви сторінок відрізняються від розривів розділів? Як це можна використати в текстовому документі?
2. Чи завжди стовпці в текстовому документі покращують сприйняття інформації? Наведіть приклади вдалого та невдалого використання.
3. Яку роль відіграють колонтикули у великих документах? Як визначити, яку інформацію варто в них розміщувати?
4. Як оформлення документа (стовпці, колонтикули, розриви) впливає на наочність і виразність тексту?



Виконайте завдання



1. Відкрийте запропонований файл (наприклад, **Розділ 2\Пункт 2.3\зразок 2.3.1.docx**). Кожен пункт текстового документа оформте як окремий розділ, який починається з нової сторінки. Установіть таке форматування для розділів документа:
 - а) непарні розділи: орієнтація сторінки – *книжкова*, поля – *по 2 см*, нумерація сторінок – *зверху по центру*;
 - б) парні розділи: орієнтація сторінки – *альбомна*, поля – *по 1 см*, нумерація сторінок – *знизу зліва*.
 Збережіть документ у вашій папці у файлі з іменем **завдання 2.3.1**.
2. Відкрийте запропонований файл (наприклад, **Розділ 2\Пункт 2.3\зразок 2.3.2.docx**). У відкритому документі розмістіть текст другого абзацу в два стовпці однакової ширини, текст третього абзацу – у три стовпці різної ширини з розділювачем. Збережіть документ у вашій папці у файлі з іменем **завдання 2.3.2**.
3. Відкрийте запропонований файл (наприклад, **Розділ 2\Пункт 2.3\зразок 2.3.3.docx**). Створіть колонтикули: у верхньому запишіть поточну дату з вирівнюванням по центру, у нижньому – ваше прізвище з вирівнюванням по лівому краю. Збережіть документ у вашій папці у файлі з іменем **завдання 2.3.3**.
4. Відкрийте запропонований файл (наприклад, **Розділ 2\Пункт 2.3\зразок 2.3.4.docx**). Уставте в документ колонтикули: верхній на непарних сторінках – *назва текстового документа*, нижній на парних сторінках – *поточна дата і час*. Збережіть документ у вашій папці у файлі з іменем **завдання 2.3.4**.
5. Який зв'язок між нумерацією сторінок і колонтикулами? Які можливості налаштування нумерації сторінок існують? Дослідіть



це питання самостійно. Результати оформте в текстовому документі у вигляді короткого опису ходу дослідження та отриманого висновку. Збережіть документ у вашій папці у файлі з іменем **завдання 2.3.5**.



6. Об'єднайтесь у групи по 3 учні та розробіть інформаційний бюлетень на одну з тем про вашу школу, ваш клас, цікаві події шкільного життя, уже зараз готуючись до вашого випускного вечора. Використайте засоби текстового процесора: багаторівневі списки, стовпці, колонтитули та розриви сторінок, графічні зображення, таблиці. Роздрукуйте документ. Збережіть документ у вашій папці у файлі з іменем **завдання 2.3.6**.



Готуємось до вивчення нового матеріалу

1. Які властивості має сторінка текстового документа? Як установити значення цих властивостей?
2. Які є режими перегляду текстового документа в текстовому процесорі **Word**? Як їх установити?
3. Яким чином структуровано текст вашого підручника з інформатики? Які об'єкти є на його сторінках?



Практична робота № 2.

«Створення текстового документа, що містить об'єкти різних типів»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте вказаний учителем / учителькою файл (наприклад, **Розділ 2\Пункт 2.3\практична 2.docx**).
2. Оформте текст у вигляді багаторівневого списку: абзаци синього кольору – I рівень, абзаци фіолетового кольору – II рівень, абзаци зеленого кольору – III рівень. Установіть нумерацію для списку в такому вигляді.
3. Розмістіть текст останнього абзацу в два стовпці однакової ширини 7 см.
4. Створіть у текстовому документі верхній колонтитул з назвою документа і сьогоднішньою датою; нижній колонтитул – з вашим прізвищем.
5. Збережіть документ у вашій папці у файлі з іменем **Практична робота 2.1**.
6. Створіть новий текстовий документ. Уведіть формули відповідно до наведеного зразка:

1. _____
1) _____
1. _____

а) Густина твердого тіла $\rho = \rho_0 \frac{P}{P - P_1}$;



в) $\frac{3x^2 + 7}{4} - \frac{x^3 - 3}{2} = \frac{5x + 2}{8}$.

7. Збережіть документ у вашій папці у файлі з іменем **Практична робота 2.2**.

2.4. ВИКОРИСТАННЯ СТИЛІВ У ТЕКСТОВИХ ДОКУМЕНТАХ. СТРУКТУРА ТЕКСТОВОГО ДОКУМЕНТА

У цьому пункті йтиметься про:

- ▶ стилі в текстовому документі;
- ▶ теми документа та їх застосування;
- ▶ правила стильового оформлення документів різних типів;
- ▶ структуру документа, її створення та використання.

СТИЛІ В ТЕКСТОВОМУ ДОКУМЕНТІ



Пригадайте

- Що таке *форматування об'єктів*? Які засоби для цього ви використовуєте?
- Як ви розумієте термін «стиль»? У яких ситуаціях ви його використовуєте?

Створивши текстовий документ, його потрібно оформити належним чином: установити потрібні значення властивостей для об'єктів різних типів – тексту, таблиць, малюнків тощо. Якщо документ великий і має велику кількість різних об'єктів, то форматування кожного окремого об'єкта може зайняти багато часу. Тим паче, якщо часто трапляються об'єкти, які потребують однакового форматування, то таке оформлення документа буде досить неефективним.

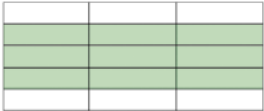
Для пришвидшення оформлення документів і форматування їхніх об'єктів зручно використовувати *стилі*. **Стиль** – це набір значень властивостей об'єктів певного типу. Наприклад, *стиль абзаців* визначає значення властивостей вирівнювання тексту на сторінці, відступів від меж аркуша, міжрядковий інтервал тощо; *стиль таблиць* – колір і стиль ліній меж таблиці, колір заливки клітинок таблиці тощо. Стилi мають власне ім'я, за яким їх можна розрізняти. Наприклад, *Строгий*, *Виокремлення*, *Звичайний* тощо.

Текстовий процесор **Word** за замовчуванням має бібліотеку професійно розроблених стилів, так звані *експрес-стилі*, у яких значення властивостей об'єктів підбрано гармонійно, з урахуванням основних вимог до дизайну. Такі стилі розроблено для об'єктів різних типів текстового документа – текстових об'єктів (абзаців, символів, списків), таблиць, графічних об'єктів (малюнків, фігур, діаграм) тощо (табл. 2.6).

Таблиця 2.6

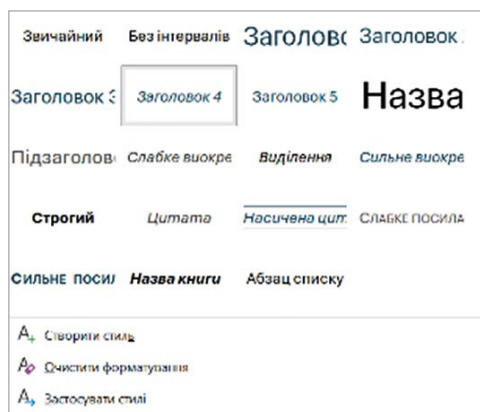
Приклади експрес-стилів об'єктів у Word

Тип об'єкта	Ім'я стилю та зразок оформлення	Значення деяких властивостей
Абзац тексту	Заголовок 2 Для пришвидшення оформлення документів та форматування їх об'єктів зручно використовувати стилі.	Шрифт – Aptos Розмір символів – 16 pt Колір символів – темно-синій Накреслення – звичайне Міжрядковий інтервал – 1,08 рядка Відступ першого рядка – немає Вирівнювання – за лівим краєм Інтервал перед абзацом – 4 pt Інтервал після абзацу – 8 pt

Тип об'єкта	Ім'я стилю та зразок оформлення	Значення деяких властивостей
Таблиця	Світлий список, акцент 3 	Колір ліній – <i>зелений</i> Колір заливки – <i>зелений</i> Штрих лінії – <i>суцільна</i> Товщина лінії – <i>0,5 pt</i>
Фігура	Насичений ефект - Акцент 4 	Колір – <i>фіолетовий</i> Прозорість – <i>0 %</i> Штрих лінії – <i>суцільний</i> Товщина лінії – <i>1 pt</i> Колір контуру – <i>темно-коричневий</i> Тіньові ефекти – <i>знизу справа</i>

Використання стилів дає змогу однією дією відразу змінити значення кількох властивостей об'єкта. Наприклад, щоб у звичайний спосіб відформатувати текст зі значеннями властивостей, які має стиль **Заголовок 2** (табл. 2.6), потрібно виконати 8 окремих дій – вибрати відповідні елементи керування на вкладках або мініпанелі для встановлення значень кожної властивості, а під час форматування із застосуванням стилю **Заголовок 2** – тільки одну дію для вибору стилю з бібліотеки.

Колекцію стилів користувач може доповнювати власними стилями, створюючи їх на основі наявних. Тоді під час збереження документа з ним автоматично зберігаються і застосовані стилі, тобто в подальших відкриттях документа зовнішній вигляд документа не буде змінюватися.



Мал. 2.24. Ескізи експрес-стилів для текстових об'єктів

Ескізи експрес-стилів текстових об'єктів документа (символів, абзаців) відображено на вкладці **Основне** у групі **Стили** (мал. 2.24). У списку завжди відображається кілька ескізів, які використовувалися останніми. Переглянути повний список стилів можна вибором кнопки **Стили**  у групі **Стили**.

Стильове оформлення може бути застосовано не тільки до окремих об'єктів, а й до всього текстового документа як єдиного цілого. Так, на вкладці **Конструктор** у групі **Форматування документа** можна

вибрати стиль оформлення всіх текстових об'єктів документа, де гармонійно підібрані шрифти символів, їхні розміри та колір символів, спосіб вирівнювання абзаців, відступи тощо (мал. 2.25).



Мал. 2.25. Набори стилів оформлення текстового документа

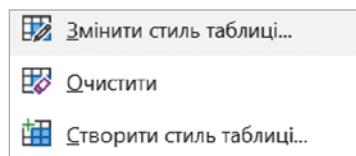
Відповідні групи елементів керування для вибору експрес-стилів інших об'єктів (таблиць, рисунків, графічних зображень тощо) розміщено на тимчасових вкладках **Конструктор таблиць**, **Формат фігури**, **Формат зображень**, які призначені для опрацювання об'єктів певного типу (мал. 2.26).



Мал. 2.26. Ескізи експрес-стилів для зображень, фігур і таблиць

Для застосування експрес-стилю потрібно виділити об'єкт і вибрати у списку потрібний ескіз. *Звертаємо вашу увагу*, що після наведення вказівника на ескіз застосовується попередній динамічний перегляд вибраного формату і можна переглянути вигляд об'єкта із застосуванням вибраного стилю.

Стилі можна перейменувати, змінити, видалити або створити новий, скориставшись командами контекстного меню ескізу стилю (мал. 2.27). Для відмови від застосування експрес-стилю і встановлення значень властивостей об'єкта за замовчуванням слід вибрати команду **Очистити** у вікні стилів.



Мал. 2.27. Команди зміни стилів

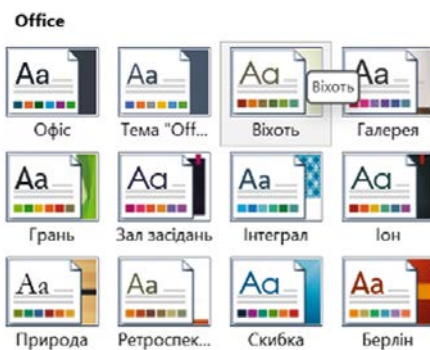
ТЕМИ ДОКУМЕНТА ТА ЇХ ЗАСТОСУВАННЯ



Пригадайте

- Що таке *стиль*? До яких об'єктів текстового документа його можна застосувати?
- Для чого і як використовуються теми оформлення презентацій?

Також для стильового оформлення документа можна застосувати **тему документа** – засіб форматування документа із застосуванням вибраних наборів стилів для текстових об'єктів, малюнків, фігур, таблиць, діаграм цього документа. У **Word** є бібліотека вбудованих тем, список яких відкривається вибором на вкладці **Конструктор** у групі **Форматування документа** однойменної кнопки **Теми** (мал. 2.28).



Мал. 2.28. Бібліотека вбудованих тем документа

Тему можна вибрати на початку створення документа або у процесі його підготовки. Під час перегляду теми відбувається динамічний перегляд вибраного набору стилів. Вибір теми зумовить зміну стилів оформлення всіх об'єктів документа. *Слід пам'ятати*, що змінення теми документа впливає тільки на ті об'єкти, які перед цим було відформатовано з використанням стилів.

ПРАВИЛА СТИЛОВОГО ОФОРМЛЕННЯ ДОКУМЕНТІВ РІЗНИХ ТИПІВ



Пригадайте

- Чому різні документи оформлюють по-різному?
- Які загальні правила оформлення презентацій?

Готуючи офіційні документи різних типів (накази, довідки, звіти, листи тощо), слід дотримуватись єдиних вимог і правил щодо їх оформлення, які встановлюються нормативними актами – *державними стандартами*.

Виконання цих стандартів забезпечує:

- оперативне та якісне створення документів;
- зручне сприймання документа людиною, яка буде його опрацьовувати;
- здійснення комп'ютерного опрацювання ділових паперів.

В Україні використовують кілька основних стандартів оформлення документів, які постійно поповнюються. Наприклад, ДСТУ (Державний стандарт України) 4163-2020 «Уніфікована система організаційно-розпорядчої документації. Вимоги до оформлювання документів» (адреса ресурсу: <https://cutt.ly/4tAlrC1v>). Відповідно до цього стандарту:



- службові документи оформляються на папері формату A4 (210 мм × 297 мм) та A5 (210 мм × 148 мм);
- мінімальні розміри полів становлять: ліве – 30 мм, праве – 10 мм, верхнє та нижнє – по 20 мм;
- шрифт основного тексту – *Times New Roman*, розмір – 12–14 пт, міжрядковий інтервал – 1–1,5 рядка, відступ першого рядка абзацу – 1 см;
- зображення логотипа розміщується біля *лівого* поля на одному рівні з назвою організації, рекомендована висота зображення 1,7 см

тощо.

Під час оформлення документів на двох і більше сторінках друга та наступні сторінки мають бути пронумеровані. Нумери сторінок ставлять посередині верхнього поля аркуша арабськими цифрами без зазначення слова «сторінка» та розділових знаків. Першу сторінку не нумерують. Сторінки основної частини документа і кожного з додатків мають окрему нумерацію. Тексти документів постійного та тривалого (понад 10 років) зберігання друкують на одному боці аркуша. Документи тимчасового терміну (до 10 років включно) зберігання можна друкувати з обох боків аркуша.

Крім того, у стандарті визначено структуру та правила оформлення списків, таблиць, дат, нумерації та інших об'єктів.

Крім державних вимог до оформлення документів, існують *відомчі стандарти* вимог до оформлення наукових праць, тез доповідей на наукових конференціях, шкільних підручників тощо. Ці правила публікуються у відомчих актах і заздалегідь доводяться до відома авторів / авторок документів. Вони можуть дещо відрізнятися від загальноприйнятих і враховувати також поліграфічні особливості видання.



Наприклад, правила оформлення науково-дослідницьких робіт МАН (адреса ресурсу: <https://cutt.ly/2tAU4oPo>):

- робота оформлюється шрифтом *Times New Roman* текстового процесора **Word** (або **Open Office**) **14 nm** на одному боці аркуша *білого* паперу формату **A4** з міжрядковим інтервалом **1,5** (до 30 рядків на сторінці);
- поля: ліве, верхнє та нижнє – *не менше ніж 20 мм*, праве – *не менше ніж 10 мм*;
- обсяг роботи становить **15–20** (для гуманітарних напрямів – **20–25**) друкованих сторінок;
- нумерацію сторінок, розділів, підрозділів, пунктів, рисунків, таблиць, формул подають арабськими цифрами *без знаку «№»*.
- усі сторінки роботи, враховуючи титульну сторінку, анотацію та додатки, нумерують наскрізно; номер на титульній сторінці не ставлять, а на наступних сторінках зазначають у *правому верхньому куті сторінки без крапки в кінці* тощо.

Окремі організації можуть розробляти свій *власний корпоративний або фірмовий стиль* оформлення документів – відповідним чином підібрані кольорова гама та шрифти символів, визначені правила оформлення абзаців документа, вигляд і розміщення логотипа організації тощо. Відповідно до цих вимог розробляються фірмові бланки та шаблони документів. Наприклад, фірмовий стиль компаній наведено на малюнку 2.29.



Мал. 2.29. Приклад фірмового бланка організації

Якщо документ готується для приватних цілей, також бажано, щоб він був оформлений у єдиному стилі. Для цього слід дотримуватися *загальних правил стильового оформлення текстів*:

- для основного тексту документа бажано використовувати однакове форматування, для виділення заголовків, окремих смислових фрагментів можна використовувати інший формат;
- кількість різних кольорів та шрифтів у документі не повинна перевищувати трьох;
- розмір символів та міжрядковий інтервал потрібно підбирати так, щоб текст можна було легко прочитати, без напруження зору;
- кольорова гама повинна відповідати призначенню документа – вітальна листівка може бути оформлена яскравими, насиченими кольорами, а звичайний лист до друга – у більш спокійних тонах;
- однотипну структуровану інформацію бажано подавати в таблицях;
- графічні зображення в документі (малюнки, діаграми, схеми) мають доповнювати зміст тексту, роз'яснювати його окремі моменти;
- у документи, які не планується друкувати, доречно вставляти активні посилання, QR-коди, інтерактивні елементи, якщо це покращує сприйняття змісту;
- обов'язково зазначати джерела текстової інформації, зображень, авторів ідей тощо.

СТРУКТУРА ДОКУМЕНТА ТА ЇЇ СТВОРЕННЯ



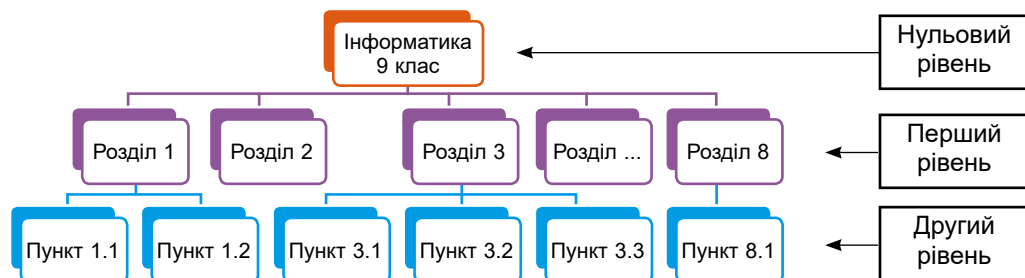
Пригадайте

• Що таке *розділ документа*? Для чого їх використовують? Як вставити розрив розділу? • Як структуровано текст вашого підручника з інформатики? Які об'єкти є на його сторінках? • Які є режими перегляду текстового документа у **Word**? Як їх установити?

Багатосторінкові документи (реферати, курсові роботи, брошури, книжки тощо) для зручності та кращого розуміння ділять на структурні частини – розділи, параграфи, пункти тощо, створюючи таким чином структуру документа. Такі структури документів вам траплялися, наприклад, на уроках української літератури, коли складали план написання твору, або коли складали мапу сайту в 8 класі.

Структура документа – це ієрархічна схема розміщення складових частин документа.

Якщо як приклад розглянути структуру підручника (мал. 2.30), то на *верхньому (нульовому)* рівні міститься назва документа, на *першому рівні* – назви розділів; до другого рівня належать назви пунктів, до третього – назви підпунктів і т. д.



Мал. 2.30. Приклад структури документа

Щоб поділити текстовий документ на розділи в тому місці, де починатиметься інший розділ, потрібно вставити **Розрив розділу**. Якщо розділів у документі має бути кілька, то для кожного з розділів вставляється свій розрив. За відсутності розривів розділів текстовий документ розглядається як один розділ.

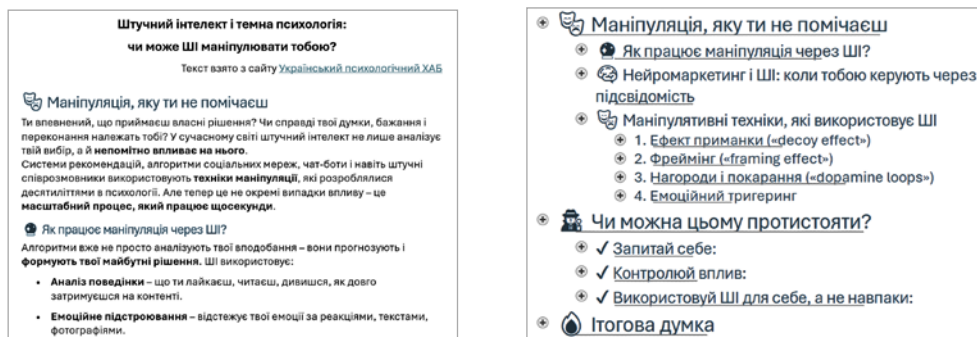
Створення структури документа відбувається із застосуванням стилів **Заголовок 1**, **Заголовок 2**, ... , **Заголовок 9** – назви всіх структурних частин документа слід відформатувати з використанням відповідного стилю. А саме: до назв структурних частин першого рівня (розділів) застосувати стиль **Заголовок 1**, назви частин другого рівня (пунктів) відформатувати із застосуванням стилю **Заголовок 2** і т. д.

Для цього потрібно:

1. Виділити назву певної структурної частини документа (розділу, пункту, підпункту).
2. Вибрати на вкладці **Основне** у групі **Стилі** такий стиль заголовка, який відповідає рівню структурної частини в ієрархічній схемі документа (перший рівень – **Заголовок 1**, другий рівень – **Заголовок 2**, ...).
3. Повторити пп.1–2 цього алгоритму для всіх назв / заголовків структурних частин документа.

Для перегляду структури документа, який відформатовано в такий спосіб, використовується режим перегляду **Структура**, який встановлюється вибором кнопки **Структура**  на вкладці **Подання** у групі **Подання**.

У цьому режимі зовнішній вигляд документа відрізняється від його вигляду в режимі **Розмітка сторінки (Макет друку)**, у якому ви звикли працювати (мал. 2.31).

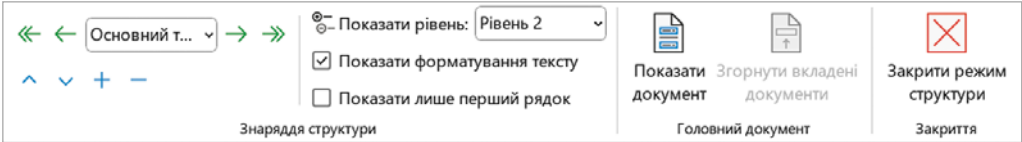


Мал. 2.31. Вигляд тексту документа в режимі **Розмітка сторінки (Макет друку)** та **Структура**

У режимі **Структура** текст структурних частин документа відображається у *згорнутому вигляді* й на екран виводяться тільки заголовки структурних частин. Для кращої візуалізації ієрархічної структури рівні документа мають різний відступ від лівого поля. Біля кожного рівня документа є відповідна позначка:

-  – якщо в рівня є підрівні або текст нижчого рівня;
-  – якщо в рівня відсутній текст нижчого рівня;
-  – якщо цей текст має найнижчий рівень (*основний текст*, до якого не застосовувалися стилі заголовків).

Уключення цього режиму зумовлює появу на **Стрічці** додаткової вкладки **Структура** (мал. 2.32), елементи керування якої використовуються для відображення тексту тільки потрібних рівнів документа, переміщення структурних частин у документі, змінення їхнього рівня, видалення розділів тексту або додавання нових тощо.



Мал. 2.32. Вкладка **Структура**

Основна зручність перегляду документа в режимі **Структура** полягає в тому, що є можливість налаштовувати відображення будь-якого рівня документа, приховуючи при цьому ті рівні, які займають нижчу ієрархію. Для цього слід на вкладці **Структура** у групі **Знаряддя структури** вибрати потрібний рівень зі списку **Показати рівень**. Якщо необхідно розгорнути зміст деякої окремої структурної частини документа, слід двічі клацнути на значку  біля відповідного заголовка. Повторення цих дій приховає вкладений чи текст нижчого рівня.

У режимі **Структура** зручно редагувати ієрархічну схему документа, змінювати рівень текстових фрагментів і послідовність їх розміщення, використовуючи елементи керування групи **Знаряддя структури** (табл. 2.7).

Таблиця 2.7

Призначення елементів керування групи **Знаряддя структури** вкладки **Структура**

Елемент керування	Призначення
Поширити на заголовок 1 	Для підвищення рівня ієрархії поточного фрагмента тексту до найвищого. Автоматично до тексту буде застосовано стиль <i>Заголовок 1</i>
Знизити до звичайного тексту 	Для зниження рівня ієрархії поточного фрагмента тексту до найнижчого. Автоматично до тексту буде застосовано стиль <i>Звичайний</i>
Підвищити рівень  Знизити рівень 	Для переміщення поточного фрагмента тексту на рівень вище або нижче з відповідною зміною стилю
Вгору / Вниз  	Для змінення місця розташування поточного фрагмента тексту без зміни рівня ієрархії
Розгорнути / Згорнути  	Для відображення або приховування вкладеного тексту для поточного заголовка
Рівень 2 	Для встановлення для поточного фрагмента тексту певного рівня, який добирається зі списку з відповідною зміною стилю

Слід пам'ятати, що виділення або переміщення заголовка супроводжується виділенням і переміщенням тексту нижчого рівня.

Упорядкування тексту можна виконувати і перетягуванням структурних елементів документа за позначками ,  та . На екрані відображаються вертикальні чи горизонтальні лінії, які вказують на нове місце фрагмента тексту.

Структуру документа можна надрукувати. На папері вона виглядатиме так само, як і на екрані. Формати абзаців ігноруються, формати символів відтворюються повністю.

Також у режимі **Структура** можна розбити великий документ на кілька окремих вкладених документів, які зберігатимуться в окремих файлах і будуть підпорядковані одному основному документу. Для цього використовуються елементи керування групи **Головний документ** вкладки **Структура**.

Після завершення роботи зі структурою документа цей режим потрібно закрити, вибравши кнопку **Закрити режим структури**  на вкладці **Структура**.



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте запропонований текстовий документ (наприклад, **Розділ 2\Пункт 2.4\вправа 2.4.docx**).
2. Перегляньте документ, знайдіть назви розділів і пунктів, абзаци зі звичайним текстом та інші структурні частини.
3. Увімкніть режим відображення прихованих символів.
4. Відформатуйте об'єкти текстового документа з використанням таких стилів:

- абзаци тексту – стиль *Слабке виокремлення*;
- таблиці – стиль *Таблиця-сітка 1 (світла)*, акцент 3;
- малюнок – стиль *Повернута біла*.

Для цього для кожного об'єкта документа (абзац, таблиця, малюнок):

1. Виділіть потрібний об'єкт документа.
 2. Виберіть на вкладці **Основне (Конструктор таблиць, Формат зображення)** у групі **Стилі** потрібний стиль.
5. Відформатуйте заголовки структурних частин документа, використовуючи такі стилі:
 - назви розділів – стиль *Заголовок 1*;
 - назви пунктів – стиль *Заголовок 2*.

Для цього для кожної назви структурної частини документа:

1. Виділіть назву структурної частини документа (розділу чи пункту).
 2. Виберіть на вкладці **Основне** у групі **Стилі** кнопку **Стилі**.
 3. Виберіть потрібний стиль (*Заголовок 1* або *Заголовок 2*).
6. Увімкніть режим перегляду документа **Структура**. Для цього виберіть кнопку **Структура** на вкладці **Подання** у групі **Подання**.
 7. Перегляньте документ у цьому режимі. Зверніть увагу на умовні позначення, на структуру документа.

8. Налаштуйте відображення у структурі документа *1 рівень тексту, 3 рівень тексту, Усі рівні*. Для цього на вкладці **Структура** у групі **Знаряддя структури** відкрийте список кнопки **Показати рівень** і виберіть потрібний рівень. Перегляньте документ у такому вигляді та з'ясуйте, які зміни відбулися.
9. Установіть і потім зніміть прапорець **Показувати лише перший рядок** у групі **Знаряддя структури**. Перегляньте документ у кожному з цих виглядів і з'ясуйте, які зміни відбулися.
10. Згорніть і розгорніть по черзі різні структурні частини документа. Для цього слід двічі клацнути на значку біля відповідного заголовка.
11. Опрацюйте виконання зі структурою документа команд **Підвищити рівень, Знизити рівень, Вгору, Вниз**. Для цього виділіть структурний елемент документа і виберіть потрібну кнопку на вкладці.
12. Увімкніть режим відображення документа **Розмітка сторінки (Макет друку)**, вибравши відповідну кнопку в **Рядку стану**.
13. Збережіть документ у вашій папці у файлі з іменем **вправа 2.4**.
14. Закрийте вікно програми **Word**.



Найважливіше в цьому пункті

Для прискорення оформлення документів та їхніх об'єктів зручно використовувати **стилі**. **Стиль** – це набір значень властивостей певного типу об'єктів, який має власне ім'я. Текстовий процесор **Word** має власну бібліотеку професійно розроблених стилів, так звані **експрес-стилі**, у яких значення властивостей об'єктів підібрано гармонійно, з урахуванням основних вимог до дизайну. Такі стилі розроблено для різних типів об'єктів текстового документа – тексту, таблиць, графічних об'єктів тощо. Ескізи експрес-стилів текстових об'єктів документа відображено на вкладці **Основне** у групі **Стилі**; для інших об'єктів (таблиць, графічних зображень) – у групі **Стилі** на тимчасових вкладках **Конструктор таблиць, Формат рисунків і Формат зображення**.

Тема документа – це набір стилів оформлення текстового документа, який має ім'я і визначає для його об'єктів значення деяких властивостей (колір, розмір і шрифт символів, спосіб вирівнювання абзаців, міжрядковий інтервал, ефекти графічних зображень тощо). У **Word** є бібліотека вбудованих тем, список яких відкривається вибором на вкладці **Розмітка сторінки** у групі **Теми** однойменної кнопки **Теми**.

Готуючи текстові документи різних типів, слід дотримуватись єдиних вимог і правил щодо їх оформлення, структури та змісту, які встановлюються державними або відомчими стандартами, корпоративними вимогами, правилами ведення ділової документації.

Структура документа – це ієрархічна схема розміщення складових частин документа. Найчастіше структуру створюють для багатосторінкових документів, розділяючи їх на розділи, параграфи, пункти.

Створення структури документа відбувається із застосуванням стилів форматування **Заголовок 1, Заголовок 2, ..., Заголовок 9** для назв структурних частин документа – назви всіх структурних частин документа слід відформатувати з використанням відповідного стилю для певного рівня документа.

Для перегляду структури документа використовують режим перегляду **Структура**, який установлюється вибором кнопки **Структура** на вкладці **Подання**. Зручність перегляду документа в режимі **Структура** полягає в тому, що є можливість налаштувати відображення будь-якого рівня документа, приховуючи при цьому ті рівні, які належать до нижчої ієрархії.

У режимі **Структура** зручно редагувати структуру документа, змінюючи рівень текстових фрагментів і послідовність їх розміщення, використовуючи елементи керування групи **Знаряддя структури**.



Дайте відповіді на запитання

1. Що таке *стилі*? До яких об'єктів можна застосовувати стильове оформлення у **Word**? Які значення властивостей об'єктів документа вони визначають?
2. Що таке *тема документа*? Що вона визначає?
3. Які загальні правила стильового оформлення документів?
4. Що таке *структура документа*? Який максимальний рівень вкладеності структурного елемента документа підтримується текстовим процесором **Word**? Наведіть приклади структурованих документів.
5. Яке призначення стилів заголовків? Скільки таких стилів є у **Word**?



Обговоріть і зробіть висновки

1. Чому, на вашу думку, використання стилів у текстовому документі є не просто зручністю, а необхідністю під час створення багатосторінкового документа? Наведіть приклади.
2. Які переваги, на вашу думку, має використання вбудованих тем документа у **Word**? Чи це може позитивно вплинути на сприйняття змісту користувачем?
3. У яких випадках структурування документа (розділи, підрозділи, пункти) справді покращує його сприйняття, а коли – може ускладнити? Аргументуйте свою думку прикладами.
4. Чи вважаєте ви, що дотримання стандартів оформлення документів (ДСТУ, вимоги МАН та ін.) обмежує творчість? Обґрунтуйте вашу думку.
5. Як ви вважаєте, чи змінює структурування зміст (вміст) документа?



Виконайте завдання



1. Відкрийте запропонований файл (наприклад, **Розділ 2\Пункт 2.4\зразок 2.4.1.docx**). Відформатуйте об'єкти текстового документа з використанням таких стилів:

- Заголовки – стиль *Назва книги*;
- Абзаци тексту – стиль *Строгий*;
- Списки – *маркований* з маркером ❖;
- Таблиці – стиль *Таблиця-список 4, акцент 2*;
- Малюнок – стиль *Металевий овал*.

Збережіть документ у вашій папці у файлі з іменем **завдання 2.4.1**.

2. Відкрийте запропонований файл (наприклад, **Розділ 2\Пункт 2.4\зразок 2.4.2.docx**), у якому до об'єктів 1 і 2 текстового документа було застосовано стильове оформлення. З'ясуйте значення властивостей об'єктів, які встановлюють використані стилі. Запишіть результати в зошит.

№ об'єкта	Тип об'єкта	Стиль об'єкта	Значення властивостей об'єкта
1			
2			

3. Відкрийте запропонований файл (наприклад, **Розділ 2\Пункт 2.4\зразок 2.4.3.docx**). Визначте, яка тема оформлення документа застосована. Змініть тему документа на *Інтеграл*, потім – на *Галерея*. Перегляньте оформлення об'єктів документа в обох випадках. З'ясуйте, які значення властивостей об'єктів документа змінюються під час змінення теми документа. Висновок запишіть у зошит.



4. Знайдіть в інтернеті текст державного стандарту ДСТУ 4163-2020 і дослідіть, які правила оформлення визначено в офіційних документах таблиць і рисунків. Застосуйте ці правила для оформлення таблиць і рисунків у вказаному документі, наприклад із файлу **Розділ 2\Пункт 2.4\зразок 2.4.4.docx**. Збережіть документ у вашій папці у файлі з іменем **завдання 2.4.4**.

5. Відкрийте запропонований документ (наприклад, **Розділ 2\Пункт 2.4\зразок 2.4.5.docx**). Перегляньте документ і визначте, які є помилки в його оформленні за правилами ДСТУ 4163-2020. Виправте їх і збережіть документ у вашій папці у файлі з іменем **завдання 2.4.5**.

6. Відкрийте запропонований файл (наприклад, **Розділ 2\Пункт 2.4\зразок 2.4.6.docx**). Відформатуйте його відповідно до вимог до ділової документації. Збережіть документ у вашій папці у файлі з іменем **завдання 2.4.6**.



7. Розробіть корпоративний стиль оформлення фірмового бланка для громадської організації «Світ чудес». Занотуйте його основні вимоги в зошит. Відформатуйте запропонований текстовий документ (наприклад, **Розділ 2\Пункт 2.4\зразок 2.4.7.docx**) із використанням розробленого стилю. Збережіть документ у вашій папці у файлі з іменем **завдання 2.4.7**.

8. Відкрийте запропонований документ (наприклад, **Розділ 2\Пункт 2.4\зразок 2.4.8.docx**). Відформатуйте документ, використовуючи такі стилі:

- Заголовок 1 рівня – стиль *Заголовок 1*;
- Заголовки 2 рівня – стиль *Заголовок 2*;
- Заголовки 3 рівня – стиль *Заголовок 3*;
- Інші фрагменти тексту – стиль *Звичайний*.

Перегляньте структуру відформатованого документа. Збережіть документ у вашій папці у файлі з іменем **завдання 2.4.8**.

9. Відкрийте запропонований документ (наприклад, **Розділ 2\Пункт 2.4\зразок 2.4.9.docx**). Перегляньте його в режимі **Структура**. Упорядкуйте розділи документа за їхніми номерами. За потреби встановіть потрібні рівні для фрагментів тексту. Збережіть документ у вашій папці у файлі з іменем **завдання 2.4.9**.



Готуємось до вивчення нового матеріалу

1. Що таке *стилі*? Для чого їх використовують?
2. Що таке *структура документа*? Як її переглянути?
3. Що таке *гіперпосилання*? Для чого і де воно використовується?
4. Що ви знаєте про спільну роботу з хмарними документами? Які рівні доступу там використовуються? Які можливості для роботи з документами це надає користувачу?

2.5. ЗМІСТ ДОКУМЕНТА. ГІПЕРПОСИЛАННЯ В ДОКУМЕНТІ. СПІЛЬНА РОБОТА НАД ДОКУМЕНТОМ

У цьому пункті йтиметься про:

- ▶ автоматизоване створення змісту документа;
- ▶ створення гіперпосилань у текстовому документі;
- ▶ спільну роботу над документом та його рецензування.

АВТОМАТИЗОВАНЕ СТВОРЕННЯ ЗМІСТУ ДОКУМЕНТА



Пригадайте

- Що таке *структура документа*? Як її переглянути?
- Що таке *стили*? Для чого їх використовують?

У попередніх пунктах ви ознайомилися з деякими особливостями опрацювання багатосторінкових документів – створення розділів, використання стильового оформлення, перегляд структури документа. Крім цього, у текстовому процесорі **Word** існують й інші засоби для спрощення процесу створення великих і складних документів, швидкого переходу до перегляду окремих частин документа.

Одним із таких засобів є зміст документа – перелік назв структурних частин документа з відповідними номерами сторінок, де ця частина документа розпочинається (мал. 2.33). Зазвичай зміст документа створюється на його початку або в кінці. Кожен елемент змісту є гіперпосиланням на відповідний заголовок у документі.

Зміст	
Програма ЮНЕСКО «Інформація для всіх»	1
1. Преамбула	1
2. Мандат	2
3. Цілі програми	2
4. Основні розділи програми	3
Розділ 1. Розвиток інформаційної політики на міжнародному, регіональному і національному рівнях	3
Розділ 2. Розвиток людських ресурсів, навиків і умінь у століття інформації	3
Розділ 3. Посилення ролі інституцій в забезпеченні доступу до інформації	3
Розділ 4. Розвиток інструментів, способів і систем інформаційного менеджменту	4
Розділ 5. Інформаційні технології для освіти, науки, культури і комунікацій	4
5. Принципи здійснення програми	4
1. Партнерство і співпраця	4
2. Оцінки	5
3. Принципи здійснення проекту	5

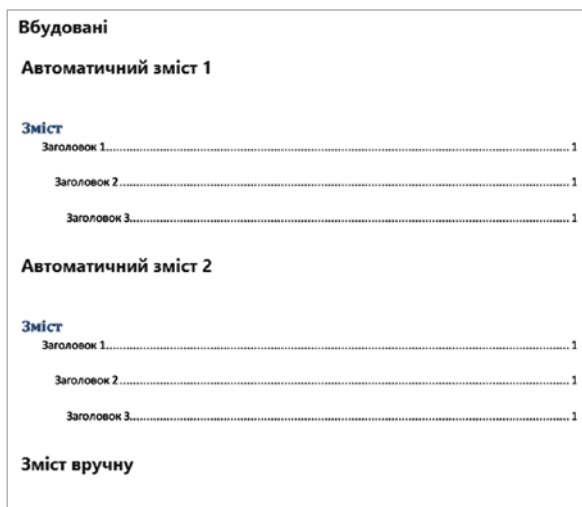
Мал. 2.33. Приклад змісту документа

Розглянемо, як створюється в текстовому документі його зміст. Якщо для кожної зі структурних частин документа використано форматування із застосуванням стилів *заголовків*, то зміст документа можна створити автоматично.

Для цього слід:

1. Установити курсор у тому місці документа, де потрібно розмістити зміст документа.

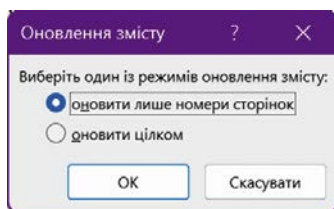
2. Виконати *Посилання* ⇒ *Зміст* ⇒ *Зміст* .
3. Вибрати зі списку вбудованих зразків потрібний варіант оформлення змісту (мал. 2.34).



Мал. 2.34. Список вбудованих зразків оформлення змісту

Створений зміст можна використовувати для швидкого переходу до потрібної частини документа, пошуку потрібних структурних частин – розділів, пунктів тощо. Для цього слід утримувати натиснутою клавішу **Ctrl** і вибрати вказівником  у змісті документа потрібний заголовок.

Створений зміст документа можна редагувати і форматовувати звичайними засобами текстового процесора **Word**. Якщо під час роботи над документом його текст чи структура змінювалися, то зміст документа потрібно оновити. Для цього слід виконати *Посилання* ⇒ *Зміст* ⇒ *Оновити таблицю* .



Мал. 2.35. Вікно Оновлення змісту

Оновлення змісту (мал. 2.35) треба обрати потрібний варіант: оновлювати весь зміст (якщо зміни вносилися в заголовки структурних елементів документа) або оновити лише номери сторінок (якщо змінювалося розташування структурних елементів за рахунок редагування чи форматування документа).

Щоб видалити зміст документа, його слід виділити і натиснути клавішу **Delete**.

СТВОРЕННЯ ГІПЕРПОСИЛАНЬ У ТЕКСТОВОМУ ДОКУМЕНТІ



Пригадайте

- Що таке *гіперпосилання*? Для чого і де воно використовується?
- Як створюється гіперпосилання у презентаціях? На вебсторінках?
- Що таке *внутрішні* та *зовнішні гіперпосилання*?

Зміст документа – зручний об’єкт текстового документа, але він забезпечує перехід усередині документа тільки до його структурних частин. Переходи до інших об’єктів у текстовому документі можна зробити з використанням **гіперпосилань**, з якими ви вже ознайомилися під час роботи з вебсторінками і створювали їх у презентаціях.

Гіперпосилання можуть бути створені для переходу в межах одного документа (на позицію в тексті, зображення, таблицю, схему, бібліографічні джерела або інші об’єкти) – це так зване *внутрішнє* гіперпосилання. Також із використанням гіперпосилань можна зробити перехід із текстового документа на зовнішні ресурси (інший документ, вебсайт, матеріали на **Google Диску**, сторінку в соціальних мережах, папку в локальній мережі, лист в електронній скриньці тощо), створивши *зовнішнє* гіперпосилання.

Наприклад, у текстовому документі зі звітом про виконання вами групового проекту з фізики ви можете не розмішувати всі ваші напрацювання, а вставити у звітний документ гіперпосилання на матеріали проекту, які розміщено на **Google Диска**х учасників / учасниць вашої групи або на інших ресурсах: на електронну таблицю з обчисленнями та діаграмами, на презентацію перебігу проекту, на фотоальбом тощо. Це зробить ваш звіт більш структурованим і компактним та надасть можливість заощадити час на його підготовку.

Гіперпосилання в тексті потрібно пов’язати з якимось об’єктом документа – словом, фрагментом тексту, малюнком тощо. Для переходу за гіперпосиланням потрібно буде вибрати цей об’єкт вказівником за натиснутої клавіші **Ctrl**. Текст гіперпосилання в текстовому документі, як і на вебсторінках, і в презентаціях, виділяється іншим кольором символів та підкреслюється.

Перед створенням внутрішніх гіперпосилань усі об’єкти документа, на які буде здійснюватися перехід, потрібно відформатувати з використанням стилів **Заголовок** або створити відповідні **закладки**.

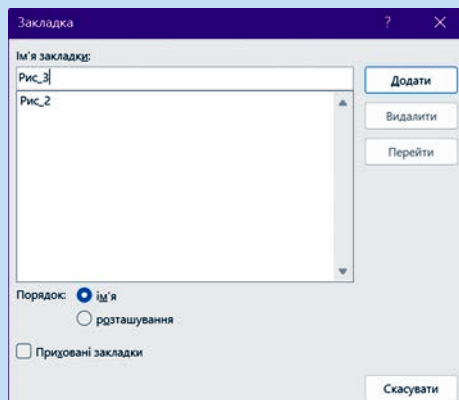


Для тих, хто хоче знати більше

Закладка позначає місце в текстовому документі й має унікальне ім’я. Імена закладок мають починатися з літери і можуть містити літери, цифри та різні символи без пропусків. Закладку можна створити для будь-якого об’єкта текстового документа – фрагмента тексту, графічного зображення, таблиці, змісту тощо. Закладку можна створити не тільки для об’єкта документа, а просто в порожньому місці документа. Використовують закладки для швидкого переходу до потрібних об’єктів документа і для створення гіперпосилань на ці об’єкти.

Для створення закладки слід:

1. Виділити об’єкт текстового документа, для якого створюється закладка.
2. Відкрити вкладку **Вставлення** і в групі **Посилання** вибрати кнопку **Закладка** .
3. Увести ім’я закладки у відповідному полі вікна **Закладка** (мал. 2.36).
4. Вибрати кнопку **Додати**.

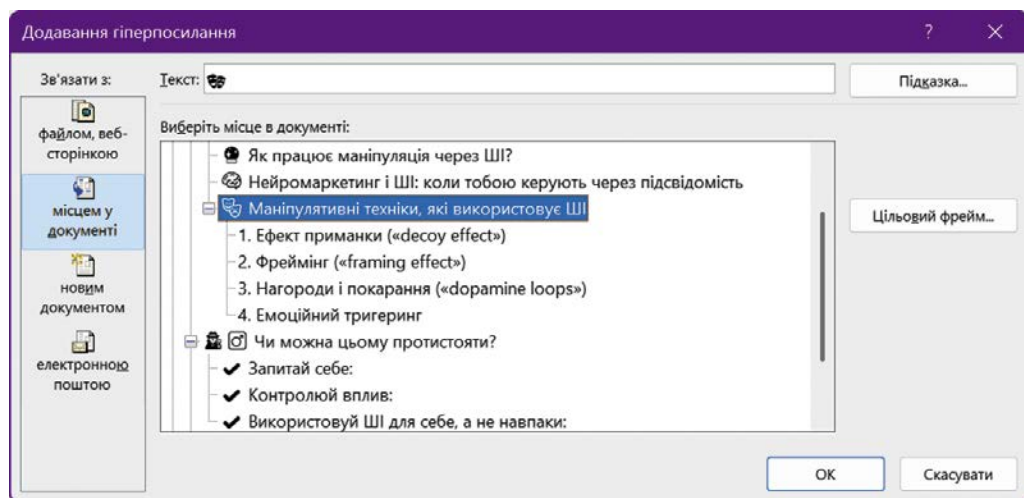


Мал. 2.36. Вікно **Закладка**

Щоб видалити закладку, слід відкрити вікно **Закладка**, вибрати у списку потрібну закладку і натиснути кнопку **Видалити**.

Щоб створити в текстовому документі внутрішнє гіперпосилання, слід:

1. Відформатувати об'єкт документа, на який буде здійснюватися перехід за гіперпосиланням, з використанням будь-якого стилю **Заголовок** або створити для нього **закладку**.
2. Виділити в документі об'єкт, з яким буде пов'язано гіперпосилання (фрагмент тексту, малюнок тощо).
3. Виконати **Вставлення** ⇒ **Посилання** ⇒ **Посилання**.
4. Вибрати у вікні **Додавання гіперпосилання** в області **Зв'язати з:** кнопку **місцем у документі** (мал. 2.37).
5. Вибрати в області **Виберіть місце в документі** у списку **Заголовки** або **Закладки** об'єкт документа, на який має здійснитися перехід.
6. Вибрати кнопку **ОК**.

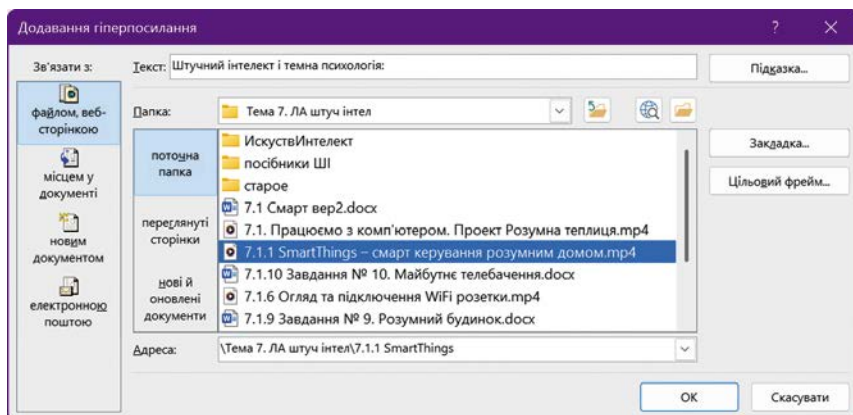


Мал. 2.37. Вікно **Додавання гіперпосилання** для вставлення внутрішнього гіперпосилання

Щоб створити зовнішнє гіперпосилання на файл у локальній мережі чи на носії даних локального комп'ютера, слід:

1. Виділити в документі об'єкт, з яким буде пов'язано гіперпосилання (фрагмент тексту, малюнок тощо).
2. Вибрати у вікні **Додавання гіперпосилання** в області **Зв'язати з:** кнопку **файлом, вебсторінкою**.
3. Відкрити у списку **Папка** потрібну папку та вибрати потрібний файл (мал. 2.38).
4. Вибрати кнопку **ОК**.

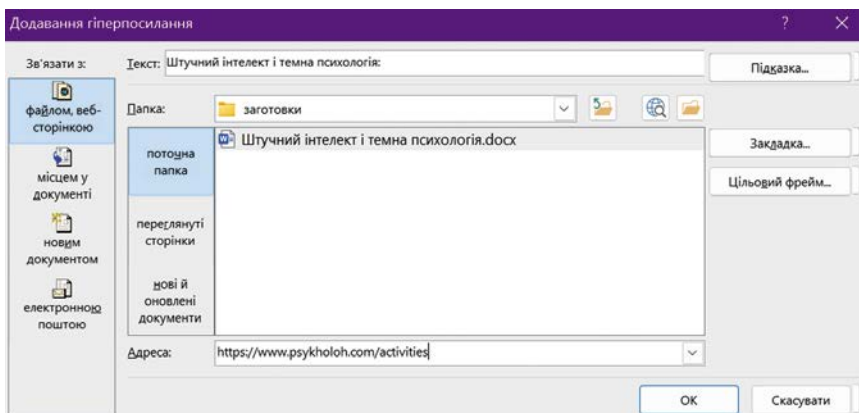
Після вибору такого гіперпосилання буде відкрито вікно програми, яка призначена для роботи з файлами цього типу, і в ньому відкриється вибраний файл. Наприклад, якщо під час створення гіперпосилання було вибрано відеофайл, то після вибору гіперпосилання відкриється вікно програми програвача, і в ній буде відтворено відео.



Мал. 2.38. Вікно **Додавання гіперпосилання** для вставлення гіперпосилання на зовнішній файл

Щоб у текстовому документі вставити гіперпосилання на вебсторінку, слід:

1. Виділити в документі об'єкт, з яким буде пов'язано гіперпосилання (фрагмент тексту, малюнок тощо).
2. Вибрати у вікні **Додавання гіперпосилання** в області **Зв'язати з:** кнопку **файлом, вебсторінкою**.
3. Увести або скопіювати адресу потрібної вебсторінки в поле **Адреса** (мал. 2.39).
4. Вибрати кнопку **OK**.



Мал. 2.39. Вікно **Додавання гіперпосилання** для вставлення гіперпосилання на вебсторінку

Після вибору в текстовому документі такого гіперпосилання буде відкрито вікно браузера, у якому відобразиться вебсторінка, адресу якої було введено в поле **Адреса**.

Уставлені гіперпосилання можна змінювати, копіювати або видаляти, використовуючи команди контекстного меню цього посилання в тексті документа (мал. 2.40).

- Змінити гіперпосилання...
- Відкрити гіперпосилання
- Копіювати гіперпосилання
- Видалити гіперпосилання

Мал. 2.40. Команди контекстного меню гіперпосилання

СПІЛЬНА РОБОТА НАД ДОКУМЕНТАМИ ТА ЇХ РЕЦЕНЗУВАННЯ



Пригадайте

- Що ви знаєте про спільну роботу з хмарними документами? Які рівні доступу в них використовуються? Які можливості для роботи з документами мають користувачі у хмарному середовищі?

Сучасні текстові процесори мають засоби для організації спільної роботи над документами кількома користувачами. Такого роду засоби потрібні для забезпечення можливості різним користувачам, учасникам / учасницям групової роботи співпрацювати під час підготовки різних документів, але в той самий час відстежувати зміни, що вносять у документ співавтори.

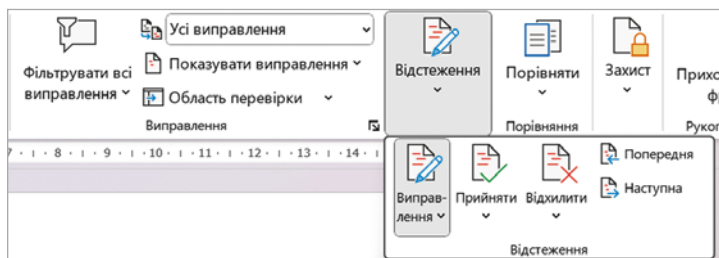
Ви вже знаєте можливості спільної роботи з документами з використанням хмарних сервісів **Google** в онлайн-режимі, де різні користувачі, маючи відповідний доступ, можуть колективно створювати документи, переглядати і редагувати їх, коментувати і пропонувати зміни, обговорювати в чаті спірні питання. Водночас співавтори можуть перебувати територіально в різних місцях і здійснювати цю роботу одночасно або в зручний для них час.

Такі самі засоби для організації спільної роботи з документами має й текстовий процесор **Word**, який входить до пакета програм **Microsoft Office 365**, що є аналогом хмарних сервісів **Google**.

У локальній версії **Word**, яку ми вивчаємо, також існують засоби для колективної роботи. Але здійснюється робота з документом не одночасно, а різні користувачі вносять свої правки в документ по черзі, виконуючи це офлайн на своєму локальному комп'ютері й надсилаючи змінений документ іншим співавторам для ознайомлення та подальшого опрацювання.

Таке опрацювання текстових документів у **Word** здійснюється в режимі *рецензування*. Щоб увімкнути цей режим, потрібно виконати послідовність дій *Рецензування* ⇒ *Відстеження* (мал. 2.41) і ви-

брати кнопку **Виправлення** .



Мал. 2.41. Групи команд вкладки **Рецензування**

Якщо після увімкнення цього режиму здійснювати редагування та форматування документа, то всі внесені зміни будуть відображатися в документі. Дії з редагування документа відображаються в самому тексті документа, а на правому полі – виконані операції форматування.

Наприклад, якщо в режимі рецензування видалити слово і замінити його на інший фрагмент, то воно з тексту не видаляється, а закреслюється, а той текст, що вставляється, виділяється іншим кольором.

Якщо після ввімкнення цього режиму здійснювати редагування та форматування документа, то всі внесені зміни будуть відображатися в документі. Дії з редагування документа відображаються в само́му тексті документа, а на правому полі арку́шу виконані операції форматування.	Людмила Чернікова змінює формат: Шрифт: напівжирний, Шрифт зі складними знаками: напівжирний Людмила Чернікова змінює формат: Шрифт: напівжирний, Шрифт зі складними знаками: напівжирний
--	---

Мал. 2.42. Вигляд фрагмента тексту в режимі рецензування

Так, на малюнку 2.42 видно, що слово *само́му* викреслено, а слово *арку́ша* – додано. Також у тексті змінили формат слів *редагування* та *форматування*. На лівому полі відображаються позначки рядків, де відбулися зміни.

Зміни, унесені в документ різними користувачами, відображатимуться в документі різними кольорами. Під час збереження рецензованого документа всі зміни будуть збережені разом з ним.

Після відкриття такого документа автор документа (або інший користувач) побачить унесені зміни, може ознайомитися з ними та вирішити – прийняти чи відхилити ці правки. За відхилення – унесені правки зникають, а за прийняття – вносяться в документ остаточно.

Для опрацювання окремого виправлення потрібно виділити правку і вибрати на вкладці **Рецензування** у групі **Відстеження** кнопку **Прийняти**  або кнопку **Відхилити** . Для переходу до попереднього або наступного виправлення слід вибрати відповідно кнопки **Попередня**  або **Наступна** .

Якщо автор документа ознайомився з правками і згоден з ними, то він може вибрати у списку кнопки **Прийняти** команду **Прийняти всі зміни** в документі або у списку кнопки **Відхилити** команду **Відхилити всі зміни** в документі, якщо він не згоден з усіма виправленнями.

Після опрацювання правок у документі режим рецензування може бути вимкнено повторним вибором кнопки **Виправлення** , і нові правки вже не будуть відстежуватися.

Для тих, хто хоче знати більше

У документі, який передбачено для рецензування, автор / авторка документа може обмежити права користувачів під час спільного опрацювання документа, а саме: заборонити форматування документа з використанням стилів, установити можливість тільки читання або надання відповідей на запитання форми в документі, визначити конкретних користувачів, які можуть редагувати документ, установити пароль на внесення змін. Усі ці налаштування можна вибрати у вікні **Обмеження форматування і редагування**, яке відкривається кнопкою **Обмежити**

редагування  у групі **Захист** вкладки **Рецензування**.

Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте запропонований текстовий документ (наприклад, **Розділ 2\Пункт 2.5\вправа 2.5.1.docx**).

2. Створіть на початку документа його зміст. Для цього:
 1. Відформатуйте назви структурних частин документа з використанням стилів заголовків відповідного рівня.
 2. Установіть курсор на початку документа, після його заголовка.
 3. Виконайте **Посилання** ⇒ **Зміст** ⇒ **Зміст**.
 4. Виберіть у списку вбудованих зразків варіант оформлення змісту *Автоматизований зміст 2*.
3. Перегляньте створений зміст документа, його структуру.
4. Перейдіть, використовуючи зміст, до першого пункту документа, потім до останнього. Для цього виберіть назву пункту у змісті, утримуючи клавішу **Ctrl**. Для повернення до змісту використовуйте **Ctrl + Home**.
5. Додайте до заголовка «**Цікаві факти про Україну**» гіперпосилання на відповідний сайт проєкту: <https://www.ebooks4ukrkids.org/uk/fun-facts-ukraine-ua>. Для цього:
 1. Виділіть у документі заголовок тексту **Цікаві факти про Україну**, з яким буде пов'язано гіперпосилання.
 2. Виконайте **Вставлення** ⇒ **Посилання** ⇒ **Посилання**. Виберіть у вікні **Додавання гіперпосилання** в області **Зв'язати з:** кнопку **файлом, вебсторінкою**.
 3. Уведіть або скопіюйте в поле **Адреса** потрібну адресу вебсторінки **Цікаві факти про Україну** <https://www.ebooks4ukrkids.org/uk/fun-facts-ukraine-ua>.
 4. Виберіть кнопку **ОК**.
6. Додайте до малюнка острова Хортиця гіперпосилання на файл, який буде вказано (наприклад, **Розділ 2\Пункт 2.5\доп 2.5.1.docx**). Для цього:
 1. Виділіть у документі зображення острова Хортиця.
 2. Виконайте **Вставлення** ⇒ **Посилання** ⇒ **Посилання**. Виберіть у вікні **Додавання гіперпосилання** в області **Зв'язати з:** кнопку **файлом, вебсторінкою**.
 3. Відкрийте в рядку **Папка** папку **Розділ 2\Пункт 2.5** і у вікні нижче виберіть файл **доп 2.5.1.docx**.
 4. Виберіть кнопку **ОК**.
7. Перевірте правильність створених гіперпосилань у завданнях № 5 та № 6, використавши їх у документі. Для цього слід навести вказівник на гіперпосилання і вибрати його за натиснутої клавіші **Ctrl**.
8. Збережіть документ у вашій папці у файлі з іменем **вправа 2.5.1**.
9. Відкрийте запропонований текстовий документ (наприклад, **Розділ 2\Пункт 2.5\вправа 2.5.2.docx**).
10. Увімкніть режим **Рецензування** та відображення виправлень. Для цього виконайте **Рецензування** ⇒ **Відстеження** ⇒ **Виправлення**.
11. Перегляньте документ і з'ясуйте, які виправлення в нього було внесено.
12. Відхиліть усі ці виправлення. Для цього виконайте **Рецензування** ⇒ **Відстеження** ⇒ **Відхилити** ⇒ **Відхилити всі зміни**.
13. Унесіть корективи в текст документа:
 - а) скопіюйте перший абзац тексту в кінець документа;
 - б) видаліть другий абзац тексту;

- в) змініть у третьому абзаці шрифт на **Arial**;
 - г) оформте всі абзаци тексту як маркований список.
- Простежте за відображенням унесених у текст документа змін.
14. Прийміть усі виправлення по черзі. Для цього щодо кожної правки виконайте **Рецензування** ⇒ **Відстеження** ⇒ **Прийняти**. Перехід між правками здійснюйте кнопками **Наступна** та **Попередня** у групі **Відстеження**.
 15. Вимкніть режим **Рецензування** та відображення виправлень. Для цього вимкніть кнопку **Виправлення** на вкладці **Рецензування** у групі **Відстеження**.
 16. Збережіть документ у вашій папці у файлі з іменем **вправа 2.5.2**.
 17. Закрийте вікно програми **Word**.



Найважливіше в цьому пункті

Зміст документа – це перелік назв структурних частин документа з відповідними номерами сторінок, де ця частина тексту розпочинається. Зазвичай зміст документа створюється на його початку або в кінці. Кожен елемент змісту є гіперпосиланням на відповідний заголовок у документі.

Якщо для кожної зі структурних частин документа використано форматування із застосуванням стилів заголовків, то зміст документа можна створити автоматично, виконавши **Посилання** ⇒ **Зміст** ⇒ **Зміст**. Якщо під час роботи з документом його текст чи структура змінювалися, то зміст документа потрібно оновити, виконавши **Посилання** ⇒ **Зміст** ⇒ **Оновити таблицю**.

Переходи до будь-яких об'єктів у текстовому документі можна зробити з використанням **гіперпосилань**. Гіперпосилання можуть бути створені для переходу в межах одного документа (**внутрішнє гіперпосилання**) або на сторонні ресурси (**зовнішнє гіперпосилання**). Гіперпосилання в тексті пов'язується з якимось об'єктом документа – словом, фрагментом тексту, малюнком тощо. Зазвичай текст гіперпосилання виділяється іншим кольором символів і підкреслюється.

Перед створенням внутрішніх гіперпосилань усі об'єкти документа, на які буде здійснюватися перехід, потрібно відформатувати з використанням стилів **Заголовок** або створити відповідні **закладки**. Створення гіперпосилань здійснюється у вікні **Додавання гіперпосилання**, для відкриття якого потрібно виконати **Вставлення** ⇒ **Посилання** ⇒ **Посилання**.

У текстовому процесорі **Word** для організації спільної роботи з документами використовують режим **рецензування**, який установлюється послідовністю дій **Рецензування** ⇒ **Відстеження** ⇒ **Виправлення**. У такому режимі всі внесені зміни відображаються в документі. Під час збереження рецензованого документа всі зміни зберігаються разом з документом.

Після відкриття такого документа користувач побачить внесені зміни і може прийняти чи відхилити ці правки. Опрацювання правок здійснюється елементами керування на вкладці **Рецензування**. Після опрацювання внесених правок слід вимкнути режим рецензування і нові змінення в документі вже не будуть відстежуватися.



Дайте відповіді на запитання

1. Що таке **зміст документа**? Для чого він призначений? Як ним користуватися в електронній версії документа?
2. Що таке **гіперпосилання в текстовому документі**? Навіщо їх використовують? Гіперпосилання яких видів можна створити в текстовому документі?
3. Що таке **режим рецензування документа**? Для чого його використовують?

4. Яким чином в документі відображаються виправлення? Які дії в режимі рецензування можна виконувати?

5. Для чого використовують закладки в текстовому документі і як вони пов'язані з гіперпосиланнями?



Обговоріть і зробіть висновки

1. Як зміст та гіперпосилання сприяють зручності користування великим документом? Наведіть приклади з власного досвіду.

2. Що спільного й відмінного між змістом і структурою документа? У яких випадках краще користуватися цими засобами?

3. Чим внутрішнє гіперпосилання відрізняється від зовнішнього? У яких ситуаціях доцільно використовувати кожне з них? Чи завжди вони роблять документ зручнішим у використанні?

4. Що спільного й відмінного в організації спільної роботи з документом в локальному та хмарному текстовому процесорі?

5. Яким чином можна організувати спільну роботу з документом під час виконання групового проекту, використовуючи текстовий процесор **Word**?



Виконайте завдання

1. Відкрийте запропонований текстовий файл (наприклад, **Розділ 2\Пункт 2.5\зразок 2.5.1.docx**). Відформатуйте назви структурних елементів документа з використанням стилів заголовків відповідних рівнів. Створіть у кінці документа його зміст. Збережіть документ у вашій папці у файлі з іменем **завдання 2.5.1**.



2. Відкрийте запропонований текстовий файл (наприклад, **Розділ 2\Пункт 2.5\зразок 2.5.2.docx**). Додайте до перелічених прізвищ видатних людей гіперпосилання на відповідні статті **Вікіпедії** про цих особистостей. Збережіть документ у вашій папці у файлі з іменем **завдання 2.5.2**.

3. Створіть текстовий документ із переліком ваших друзів, зазначивши прізвище, ім'я та дату народження. Додайте до прізвищ кожного з них гіперпосилання на його сторінку в соціальних мережах. Збережіть документ у вашій папці у файлі з іменем **завдання 2.5.3**.



4. Відкрийте запропонований текстовий файл (наприклад, **Розділ 2\Пункт 2.5\зразок 2.5.4.docx**). До кожної назви фізичного приладу додайте гіперпосилання на відповідний малюнок із папки **Розділ 2\Пункт 2.5.4**. Збережіть документ у вашій папці у файлі з іменем **завдання 2.5.4**.



5. Створіть у текстовому документі таблицю з описом п'яти пам'яток історії вашого регіону, зазначивши назву та місце розташування. Додайте до назв цих об'єктів гіперпосилання на їхні зображення, які знайдіть в інтернеті та збережіть на власному **Google Диску**. Додайте до місця розміщення гіперпосилання на **Google Карту** цього об'єкта. Збережіть документ у вашій папці у файлі з іменем **завдання 2.5.5**.

6. Відкрийте вказаний текстовий файл (наприклад, **Розділ 2\Пункт 2.5\зразок 2.5.6.docx**). Дайте відповіді на запитання в тексті щодо небезпек і ризиків в інтернеті. Перегляньте документ у режимі рецензування. Прийміть усі внесені виправлення. Збережіть документ у вашій папці у файлі з іменем **завдання 2.5.6**.



7. Створіть текстовий документ і сформулюйте в ньому п'ять ваших правил етичного використання сервісів та інструментів зі штучним інтелектом. Відправте документ електронною поштою своєму другові / подрузі на рецензування. Отриманий від друга / подруги документ перегляньте в режимі рецензування. Збережіть документ у вашій папці у файлі з іменем **завдання 2.5.7.docx**.



Практична робота № 3.

«Стилi. Структура документа. Автоматизоване створення змісту. Гіперпосилання. Рецензування документів»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте запропонований текстовий файл (наприклад, **Розділ 2\Пункт 2.5\Практична_3.docx**).
2. Установіть для всіх нумерованих пунктів тексту *рівень 1*, для маркованих – *рівень 2*.
3. У режимі перегляду **Структура** впорядкуйте структурні елементи документа в порядку номерів пунктів.
4. Відформатуйте таблицю та малюнки з використанням стилів.
5. Відформатуйте назви структурних частин текстового документа з використанням стилів заголовків відповідного рівня.
6. Створіть на початку документа його зміст.
7. До всіх малюнкiв додайте гіперпосилання на відповідні інтернет-ресурси, які знайдіть самостійно.
8. Збережіть документ у вашій папці у файлі з іменем **практична робота_3_Прізвище**.
9. Розмістіть документ у спільній папці класу, яку вкаже вчитель / учителька.
10. Відкрийте із цієї папки файл практичної роботи № 3 сусіда / сусідки праворуч, унесіть у нього такі виправлення: видаліть зміст документа і змініть стилі оформлення таблиці та одного з малюнкiв.
11. Збережіть файл з вашими виправленнями у спільній папці з іменем **практична робота_3_Прізвище_ред**.
12. Відкрийте файл вашої практичної роботи № 3 після редагування сусідом / сусідкою.
13. Перегляньте виправлення, відхиліть видалення змісту і прийміть зміну форматування об'єктів.
14. Збережіть файл у вашій папці у файлі з іменем **практична робота_3_Прізвище_фінал**.
15. Закрийте вікно програми **Word**.



Готуємось до вивчення нового матеріалу

1. Як упорядковуються дані в табличному процесорі Excel?
2. Що таке *списки*? Де ви їх використовували?
3. Які типи даних ви вже використовували в **Python**? Наведіть приклади.
4. Чим характеризується змінна? Чи може у неї бути кілька значень?
5. Як працює команда присвоювання?
6. Який вигляд має команда циклу з лічильником у **Python**? Як вона працює?





Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу «Опрацювання текстових даних» (початковий, середній, достатній, високий рівень).

- Я можу пояснити, які інформаційні технології використовують для опрацювання текстових даних.
- Я розумію, чим відрізняються текстові процесори від інших систем опрацювання текстових даних, і можу пояснити їхні можливості.
- Я вмію добирати потрібні ресурси та інструменти для створення та опрацювання текстового документа.
- Я вмію працювати з різними об'єктами текстового документа (зображення, таблиці, формули) і знаю їхні властивості.
- Я знаю, у яких форматах можуть зберігатися текстові файли, та вмію обирати найоптимальніший залежно від ситуації чи потреби користувача.
- Я можу ефективно працювати з кількома документами одночасно.
- Я можу створювати в текстовому документі багаторівневі списки, уставляти спеціальні символи та формули.
- Я вмію оформлювати багатосторінкові текстові документи, використовуючи стовпці, колонтитули, розділи.
- Я вмію використовувати стилі для оформлення текстового документа і створювати структуру документа.
- Я вмію автоматично створювати зміст документа на основі його структури.
- Я розумію, що таке нелінійний текст, і можу створювати гіперпосилання різних типів у текстових документах.
- Я вмію працювати з текстовим документом у режимі рецензування.
- Я розумію, як організувати спільну роботу над текстовим документом.
- Я відчуваю впевненість у використанні текстового процесора для створення складних документів.



АЛГОРИТМИ ТА ПРОГРАМИ

РОЗДІЛ

3

У цьому розділі ви отримаєте нові, а також поглибите і розширите свої знання й удосконалисте навички з таких тем:

- ▶ одновимірні масиви, їхні елементи;
- ▶ списки в **Python** для організації одновимірних масивів;
- ▶ створення й опрацювання списків;
- ▶ рядкові величини та їхні елементи;
- ▶ створення й опрацювання рядкових величин;
- ▶ проекти з модулями та їх створення.

У підручнику цей розділ викладено з використанням мови програмування **Python** і середовища створення та редагування проєктів **IDLE**.

Текст цього розділу з використанням мови програмування **Object Pascal** і середовища створення та редагування проєктів **Lazarus** розміщено за посиланням <https://cutt.ly/JtmBpNfS> або QR-кодом.



3.1. ОДНОВИМІРНІ МАСИВИ. СПИСКИ В PYTHON, ЇХ УВЕДЕННЯ ТА ВИВЕДЕННЯ

У цьому пункті ви дізнаєтеся про:

- ▶ одновимірні масиви, їхні елементи;
- ▶ списки в **Python** для організації одновимірних масивів;
- ▶ введення та виведення значень елементів списків;
- ▶ змінення значень елементів списку.

ПОНЯТТЯ ОДНОВИМІРНОГО МАСИВУ



Пригадайте

- Чи траплялися вам упорядковані набори даних? Якщо так, наведіть приклади. У чому зручність у використанні таких наборів даних?
- Який загальний вигляд і як виконується команда циклу з лічильником у мові програмування **Python**?

У 8 класі ви створювали проєкти, у яких використовувалася невелика кількість змінних. Але існує багато задач, у яких потрібно опрацювати значення досить великої кількості змінних.

Наприклад, нехай потрібно скласти проєкт для знаходження середньої маси ста учнів / учениць 9 класів. Очевидно, потрібно використати 100 змінних, значенням кожної з яких буде маса відповідного дев'ятикласника / дев'ятикласниці, знайти суму значень цих змінних і поділити її на 100.

Якщо використати 100 змінних з іменами, наприклад, $a, b, c, \dots, z, aa, ab, ac, \dots, az, ba, bb, bc, \dots, bz, ca, cb, cc, \dots, cv$, то команда знаходження суми їхніх значень або буде містити у правій своїй частині 100 доданків, або потрібно включити до тексту програми такі 100 команд:

$suma = a; \quad suma = suma + b; \quad suma = suma + c; \quad \dots \quad suma = suma + cv;$

Усе це робить текст програми громіздким і незручним, зважаючи на те, що кількість учнів / учениць може бути ще більшою або взагалі задалегідь невідомою.

Для розв'язування задач, у яких опрацьовується значення багатьох змінних, можна піти іншим шляхом: **розглядати всі ці змінні як елементи деякого впорядкованого за номерами набору змінних.**

Такими наборами змінних, наприклад, є:

- члени будь-якої числової послідовності; наприклад, послідовності парних натуральних чисел: $a_1 = 2$, $a_2 = 4$, $a_3 = 6$, $a_4 = 8$ і т. д.;
 - значення щоденних курсів валют протягом тижня;
 - значення середніх щоденних температур протягом місяця;
 - урожаї пшениці в Україні протягом останніх 10 років
- та інші.

Впорядкований за номерами набір змінних одного типу називається *одновимірним масивом*. Кожна змінна, що входить до одновимірного масиву, називається *елементом одновимірного масиву*. Кожен елемент одновимірного масиву має свій номер (*індекс*) і значення.

Одновимірний масив має ім'я, яке записують за правилами для імен змінних. Ім'я елемента одновимірного масиву складається з імені цього масиву і порядкового номера (індексу) цього елемента в масиві, який узятий у квадратні дужки.

Так, для розглянутого вище прикладу доцільно використати одновимірний масив, що складається зі 100 елементів. Якщо ім'я цього масиву a , то імена змінних – елементів цього масиву будуть такі: $a[1]$, $a[2]$, $a[3]$, ..., $a[100]$. Кожна із цих змінних матиме дійсний тип і певне значення (табл. 3.1).

Таблиця 3.1

Маси учнів / учениць як елементи одновимірного масиву з іменем a

Змінна	$a[1]$	$a[2]$	$a[3]$	...	$a[99]$	$a[100]$
Значення	62,4	54	71,8	...	57,5	49

Номери (індекси) елементів одновимірного масиву мають бути цілими числами. У проєктах номери елементів масиву можуть бути:

- конкретними цілими числами, наприклад $a[0]$, $a[5]$, $a[43]$;
- значенням певної змінної, яка має ціле значення, наприклад $a[i]$;
- значенням виразу, який також має ціле значення, наприклад $a[2*i+3]$.

У різних мовах програмування одновимірні масиви створюються й опрацьовуються по-різному. Зокрема, у мові програмування **Python** номери елементів одновимірного масиву завжди розпочинаються з 0. Тому в мові програмування **Python** вищерозглянутий одновимірний масив буде таким (табл. 3.2):

Таблиця 3.2

Маси учнів /учениць як елементи одновимірного масиву з іменем a , який використовується в мові програмування Python

Змінна	$a[0]$	$a[1]$	$a[2]$	...	$a[98]$	$a[99]$
Значення	62.4	54	71.8	...	57.5	49

Нагадаємо вам, що в мові програмування **Python** у дробових числах дробова частина має відокремлюватися від цілої частини крапкою, а не комою.

Тоді, щоб знайти суму значень усіх цих 100 змінних, достатньо в мові програмування **Python** записати такі команди:

```
suma = 0
for i in range(100):
    suma = suma + a[i]
```

Виконання цього фрагмента проекту для наведеного вище одновимірного масиву відбуватиметься так:

Команди	Результат виконання
<code>suma = 0</code>	<code>suma = 0</code>
<code>range(100)</code>	Згенеровано набір зі 100 чисел: 0, 1, 2, ..., 99
У наборі є невикористані числа?	True
Змінна <code>i</code> набуває початкового значення	<code>i = 0</code>
<code>suma = suma + a[i]</code>	<code>suma = 0 + a[0] = 0 + 62,4 = 62,4</code>
У наборі є невикористані числа?	True
Змінна <code>i</code> набуває наступного значення	<code>i = 1</code>
<code>suma = suma + a[i]</code>	<code>suma = 62,4 + a[1] = 62,4 + 54 = 116,4</code>
У наборі є невикористані числа?	True
Змінна <code>i</code> набуває наступного значення	<code>i = 2</code>
<code>suma = suma + a[i]</code>	<code>suma = 116,4 + a[2] = 116,4 + 71,8 = 187,4</code>
...	...
У наборі є невикористані числа?	True
Змінна <code>i</code> набуває наступного значення	<code>i = 99</code>
<code>suma = suma + a[i]</code>	<code>suma = 187,4 + ... + a[99] = 187,4 + ... + 49</code>
У наборі є невикористані числа?	False

У результаті виконання наведеного фрагмента проекту значення змінної `suma` дорівнюватиме сумі значень усіх 100 змінних – елементів одновимірного масиву, тобто загальній масі 100 учнів / учениць. Після цього можна обчислити шукану середню масу, поділивши знайдену суму на 100.

Очевидно, що використання одновимірних масивів та їх опрацювання в циклі робить програмний код значно коротшим і зрозумілішим. Цього вдалося досягти завдяки тому, що номери (індекси) елементів одновимірного масиву подаються змінною, значення якої змінюється в циклі.

Звертаємо вашу увагу, що за таким алгоритмом знаходять суму значень 100 елементів одновимірного масиву для будь-якої задачі, незалежно від суті елементів масиву (маса, температура, заробітна плата, кількість зроблених деталей тощо).

СПИСКИ В PYTHON. УВЕДЕННЯ, ОПРАЦЮВАННЯ ТА ВИВЕДЕННЯ ЗНАЧЕНЬ ЕЛЕМЕНТІВ СПИСКУ



Пригадайте

• Що таке *списки*? Де вони використовуються? • Які види списків ви знаєте? Чим вони відрізняються один від одного? • У списках яких видів зручніше і швидше виконувати пошук потрібних даних?

У мові програмування **Python** є різні типи даних, які можна використовувати для створення та опрацювання одновимірних масивів. Є тип даних **масив** (англ. **array**). Опрацювання його елементів відбувається досить швидко, але кількість елементів у масиві фіксована і не може змінюватися під час виконання проєкту. В одному масиві можуть бути елементи тільки одного типу: або тільки числа, або тільки текстові величини, або інші.

Більш гнучким є тип даних **список** (англ. **list**). Кількість його елементів може збільшуватися або зменшуватися під час виконання проєкту, і в одному списку можуть бути елементи різних типів.

Будемо в 9 класі використовувати саме списки для створення й опрацювання одновимірних масивів.

Значення елементів списку записують у квадратних дужках, розділяючи їх комою. Наприклад, створити змінну **a** типу **список**, значеннями елементів якого будуть середні температури кожного дня тижня, можна такою командою:

```
a = [4, -3.8, 0, 3.5, -4, 5.1, -2]
```

У цьому списку 7 елементів. Кожен елемент списку має свій **номер** (**індекс**). Номери (індекси) елементів списків розпочинаються з нуля. Так у наведеному вище списку з іменем **a** номерами елементів є цілі числа від 0 до 6. Значенням елемента **a[0]** є число 4, значенням елемента **a[1]** є число -3.8 і т. д. (табл. 3.3).

Таблиця 3.3

Список середніх температур кожного дня протягом тижня

Номер (індекс) елемента списку	0	1	2	3	4	5	6
Ім'я елемента списку	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
Значення елемента списку	4	-3.8	0	3.5	-4	5.1	-2

Номер елемента списку може задаватися цілим невід'ємним числом, а також значенням змінної або виразу, що набувають цілого невід'ємного значення, наприклад **a[i+2]**, але перед цим змінна **i** має набути певного цілого значення.

Установити або змінити значення елемента списку можна командами присвоювання: **a[5] = 3**, **a[i] = 6**, **a[4] = a[4] + 3**.

Вивести у вікно виконання значення всіх елементів списку можна командою **print(a)**, а значення одного елемента списку можна, наприклад, командою **print(a[0])**.

Розглянемо задачу, у якій створимо, опрацюємо і виведемо список.

Задача. Елементами списку є довжини сторін 8 квадратів. Обчислити і вивести значення площ цих квадратів.

Проект для розв'язування цієї задачі може бути такий:

```
a = [4, 5.3, 12, 3, 8.1, 15, 2, 14] # Створити список із наведених чисел
for i in range(8): # Виконати 8 разів команди циклу
    a[i] = a[i]**2 # Замінити значення елементів списку на їхні
                  квадрати
print(a) # Вивести нові значення елементів списку
```

У результаті виконання цього проекту ми отримаємо такий список:

```
a = [16, 28.09, 144, 9, 65.61, 225, 4, 196]
```

Якщо потрібно виконати проект для списку **a** з іншими значеннями його елементів або для списку з іншою кількістю елементів, то потрібно зробити відповідні зміни в тексті проекту.

Уводити в пам'ять комп'ютера значення елементів списку можна й під час виконання проекту. Ці значення можна вводити з клавіатури або по одному, або в один рядок через пропуск.

Щоб створити список зі значень, які будуть вводитися з клавіатури по одному, потрібно у проекті створити порожній список, а потім по одному додавати до цього списку введені з клавіатури значення елементів. Для додавання значення в кінець списку використаємо метод **append**. Відповідний проект буде такий:

```
a = [] # Створити порожній список
for i in range(8): # Виконати 8 разів команди циклу
    x = float(input('Уведіть значення елемента списку > ')) # Зна-
    чення, що введено з клавіатури, перевести в чис-
    ло і присвоїти змінній x
    a.append(x) # Додати в кінець списку з іменем a значення змін-
    ної x
for i in range(8): # Виконати 8 разів команди циклу
    a[i] = a[i]**2 # Замінити значення елементів списку на їх квад-
    рати
print('Площі відповідних квадратів: ', a) # Вивести нові значення
                                           елементів списку
```

Після запуску проекту на виконання потрібно вводити значення елементів списку, натискаючи після введення кожного значення клавішу **Enter** (мал. 3.1).

```
Уведіть значення елемента списку > 4
Уведіть значення елемента списку > 5.3
Уведіть значення елемента списку > 12
Уведіть значення елемента списку > 3
Уведіть значення елемента списку > 8.1
Уведіть значення елемента списку > 15
Уведіть значення елемента списку > 2
Уведіть значення елемента списку > 14
```

Мал. 3.1. Уведення значень елементів списку по одному під час виконання проекту

Значення елементів списку також можна під час виконання проекту вводити з клавіатури в одному рядку, розділяючи їх пропуском.



Для цього у проєкт потрібно включити таку команду:

```
a = list(map(float, input('Уведіть довжини сторін квадратів > ').split()))
```

Після її виконання у вікні виконання проєкту з'явиться текст **Уведіть довжини сторін квадратів >**, біля якого потрібно з клавіатури ввести через пропуски значення і натиснути клавішу **Enter** (мал. 3.2).

```
>>> %Run 3_1-2.py
Уведіть довжини сторін квадратів > 4 5.3 12 3 8.1 15 2 14
Площі відповідних квадратів: 16.0 28.09 144.0 9.0 65.61 225.0 4.0 196.0
>>> |
```

Мал. 3.2. Уведення значень елементів списку з клавіатури в одному рядку через пропуски та виведення результату

Після натиснення клавіші **Enter**:

- спочатку виконається функція **input()**, яка введе в пам'ять комп'ютера рядок тексту, уведений у вікні виконання проєкту з клавіатури: **input('Уведіть довжини сторін квадратів > ');**
- потім виконається метод **split()**, який розділить рядок тексту на окремі частини на місцях пропусків, утворюючи величину типу *список*, значеннями елементів якого є тексти: **input('Уведіть довжини сторін квадратів > ').split();**
- потім виконається функція **map(тип, набір_значень)**, яка перетворює кожне значення у списку на значення вказаного типу, у даному прикладі – типу **float**: **map(float, input('Уведіть довжини сторін квадратів > ').split());**
- потім виконається функція **list(об'єкт)**, яка створює остаточний список чисел: **list(map(float, input('Уведіть довжини сторін квадратів > ').split()));**
- і наостанок створений список чисел буде присвоєний змінній **a**: **a = list(map(float, input('Уведіть довжини сторін квадратів > ').split()));**

Кількість елементів створеного списку **a** можна визначити після його створення функцією **len(a)** (англ. *length* – довжина).

Якщо потрібно вивести отриманий у результаті список без квадратних дужок, розділяючи числа пропуском (мал. 3.2), то потрібно використати команду **print(*a)**.

Нижче наведено повний текст проєкту для розв'язування вищезрозглянутої **Задачі** з уведенням та виведенням значень у рядку через пропуски:

```
a = list(map(float, input('Уведіть довжини сторін квадратів > ').split()))
n = len(a)
for i in range(n):
    a[i] = a[i]**2
print('Площі відповідних квадратів: ', *a)
```

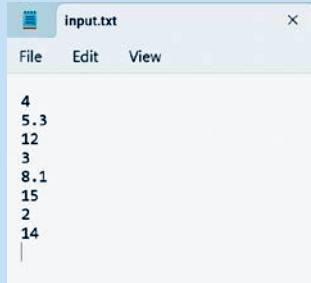


Для тих, хто хоче знати більше

У багатьох задачах, особливо коли потрібно опрацьовувати велику кількість даних, зручніше вводити дані не з клавіатури, а з файлу, і виводити кінцеві результати не у вікно виконання проєкту, а також у файл.

Розв'яжемо **Задачу**, розглянуту вище, використовуючи файли для введення і виведення значень елементів списку.

Створимо текстовий файл, наприклад у текстовому редакторі **Блокнот**, з іменем **input.txt** із 8-ма числами, розмістивши їх по одному в рядку (мал. 3.3). Збережемо цей файл у папці з проєктом.



Мал. 3.3. Уміст файлу **input.txt**

Кінцеві результати виводитимемо у файл з іменем **output.txt** у тій самій папці з проєктом.

Наведемо текст проєкту з уведенням даних з файлу і виведенням результатів у файл.

f1 = open('input.txt', 'r') # Ця команда пов'язує вказаний у ній конкретний файл **input.txt**, який розміщено в одній папці з проєктом, із файловою змінною **f1** для коротшого звертання до цього файлу, і відкриває його для читання даних

a = f1.read().split() # Ця команда читає з файлу, пов'язаного з файловою змінною **f1**, текст, поділяє його на окремі фрагменти тексту в місцях пропусків або переходів до нового рядка й утворює список з іменем **a**, значеннями елементів якого є рядки тексту

a = list(map(float, a)) # Ця команда перетворює список з рядків тексту на список чисел

f1.close() # Ця команда завершує роботу з файлом, пов'язаним з файловою змінною **f1**

n = len(a) # Ця команда визначає кількість елементів у списку

for i in range(n):

a[i] = a[i]2**

f2 = open('output.txt', 'w') # Ця команда відкриває файл, пов'язаний з файловою змінною **f2**, для виведення в нього даних. Якщо такого файлу в папці з проєктом немає, то спочатку він створюється. Якщо цей файл у папці з проєктом існує, то спочатку він очищується

for i in range(n):

f2.write(str(a[i]) + '\n') # Ця команда виконуватиметься в циклі і для кожного елемента списку з іменем **a** виведе його значення в окремий рядок вказаного файлу, пов'язаного з файловою змінною **f2**. Послідовність символів **'\n'** означає, що наступний за нею текст буде виведений у новому рядку файлу

f2.close() # Ця команда завершує роботу з файлом, пов'язаним з файловою змінною **f2**





Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Задача. У списку подано ціни 8 товарів у гривнях. Через місяць вони зросли на 10 грн. Створіть проєкт, у результаті виконання якого утвориться список з новими цінами товарів.

- Відкрийте вікно середовища створення та редагування проєктів.
- Уведіть у вікні редактора коду такий текст:


```
a = [60, 120, 130, 10, 40, 150, 50, 55]
for i in range(8):
    a[i] = a[i] + 10
print(a)
```
- Збережіть проєкт у файлі з іменем **вправа 3.1-1** у вашій папці.
- Запустіть проєкт на виконання.
- Проаналізуйте виведені результати. Чи відповідають вони умові задачі?
- Запустіть ще кілька разів проєкт на виконання, змінивши значення елементів списку.
- Проаналізуйте виведені результати. Чи відповідають вони умові задачі?
- Змініть текст проєкту на такий:


```
a = []
for i in range(8):
    x = float(input('Уведіть значення елемента списку > '))
    a.append(x)
for i in range(8):
    a[i] = a[i] + 10
print('Підвищені ціни: ', *a)
```
- Збережіть проєкт у файлі з іменем **вправа 3.1-2** у вашій папці.
- Запустіть проєкт на виконання.
- Уведіть по черзі числа 60, 120, 130, 10, 40, 150, 50, 55, натискаючи після введення кожного клавішу **Enter**.
- Проаналізуйте виведені результати. Чи відповідають вони умові задачі?
- Запустіть ще кілька разів проєкт на виконання, уводячи щоразу інші числа.
- Проаналізуйте виведені результати. Чи відповідають вони умові задачі?
- Змініть текст проєкту на такий:


```
a = list(map(int, input('Уведіть 8 цілих чисел через пропуск > ').split()))
for i in range(8):
    a[i] = a[i] + 10
print('Підвищені ціни: ', *a)
```
- Збережіть проєкт у файлі з іменем **вправа 3.1-3** у вашій папці.
- Запустіть проєкт на виконання.
- Уведіть через пропуск числа: 60 120 130 10 40 150 50 55.
- Натисніть клавішу **Enter**.

20. Проаналізуйте виведені результати. Чи відповідають вони умові задачі?
21. Запустіть ще кілька разів проєкт на виконання, уводячи щоразу інші числа.
22. Проаналізуйте виведені результати. Чи відповідають вони умові задачі?
23. Закрийте всі відкриті вікна.



Найважливіше в цьому пункті

Упорядкований за номерами набір змінних одного типу називається **одновимірним масивом**. Кожна змінна, що входить до одновимірного масиву, називається **елементом одновимірного масиву**. Кожен елемент одновимірного масиву має свій номер (**індекс**) і **значення**.

Одновимірний масив має **ім'я**, яке записують за правилами для імен змінних. **Ім'я елемента одновимірного масиву** складається з імені цього масиву і **порядкового номера (індексу)** цього елемента в масиві, який узятий у квадратні дужки.

Номери (індекси) елементів одновимірного масиву мають бути цілими числами. У проєктах номери елементів масиву можуть бути:

- конкретними цілими числами, наприклад: **a[0]**, **a[5]**, **a[43]**;
- значенням певної змінної, яка має ціле значення, наприклад **a[i]**;
- значенням виразу, який також має ціле значення, наприклад **a[2*i+3]**.

Значення елементів списку записують у квадратних дужках, розділяючи їх комою. Наприклад, створити змінну **a** типу **список**, значеннями елементів якого будуть середні температури кожного дня тижня, можна такою командою:

```
a = [4, -3.8, 0, 3.5, -4, 5.1, -2]
```

У цьому списку 7 елементів. Кожен елемент списку має свій **номер (індекс)**. Номери (індекси) елементів списків розпочинаються з нуля. Так у вищенаведеному списку з іменем **a** номерами елементів є цілі числа від 0 до 6. Значенням елемента **a[0]** є число 4, значенням елемента **a[1]** є число -3.8 і т. д.

Установити або змінити значення елемента списку можна командами присвоєння: **a[5] = 3**, **a[i] = 6**, **a[4] = a[4] + 3**.

Щоб створити список зі значень, які будуть вводитися з клавіатури по одному, потрібно у проєкті створити порожній список, а потім по одному додавати до цього списку введені з клавіатури значення елементів. Для додавання значення в кінець списку використаємо метод **append**. Відповідний проєкт буде такий:

```
a = [] # Створити порожній список
```

```
for i in range(8): # Виконати 8 разів команди циклу
```

```
    x = float(input('Уведіть значення елемента списку > ')) # Значення, що введено з клавіатури, перевести в число і присвоїти змінній x
```

```
    a.append(x) # Додати в кінець списку з іменем a значення змінної x
```

Значення елементів списку також можна під час виконання проєкту вводити з клавіатури в одному рядку, розділяючи їх пропуском. Для цього у проєкт потрібно включити таку команду:

```
a = list(map(float, input('Уведіть довжини сторін квадратів > ').split()))
```

Після її виконання у вікні виконання проєкту з'явиться текст **Уведіть довжини сторін квадратів >**, біля якого потрібно з клавіатури ввести через пропуски значення і натиснути клавішу **Enter**.

Кількість елементів створеного списку **a** можна визначити після його створення функцією **len(a)**.





Дайте відповіді на запитання

1. Що таке *одновимірний масив*?
2. Із чого складається ім'я елемента одновимірного масиву?
3. Що може використовуватись як номер елемента одновимірного масиву?
4. Якими способами можна ввести значення елементів списку з клавіатури?
5. Якими способами можна вивести значення елементів списку?
6. Яка функція використовується для визначення кількості елементів списку?
7. Який метод використовується для додавання до списку нових елементів?



Обговоріть і зробіть висновки

1. Для розв'язування яких фінансових, екологічних, спортивних задач можна використати одновимірні масиви?
2. Які переваги має використання одновимірного масиву перед використанням окремих змінних?



Виконайте завдання

1. Виконайте в зошиті наведений фрагмент проекту для списку, значеннями елементів якого є числа 2; -2; 3; 0,25; -4:

```
dob = 1
```

```
for i in range(5): dob = dob * a[i]
```

Яке значення змінної **dob** після завершення його виконання? Сформулюйте умову задачі, яку можна розв'язати, використовуючи цей фрагмент проекту.

2. Виконайте в зошиті наведений фрагмент проекту для списку, значеннями елементів якого є числа 0,5; -3; 2; -1,6:

```
suma = 0
```

```
for i in range(4): suma = suma + a[i]*a[i]
```

Яке значення змінної **suma** після завершення його виконання? Сформулюйте умову задачі, яку можна розв'язати, використовуючи цей фрагмент проекту.

3. Виконайте в зошиті наведений фрагмент проекту для списку, значеннями елементів якого є числа 0,5; -3; 2; -1,6:

```
suma = 0
```

```
for i in range(4):
```

```
if a[i] > 0: suma = suma + a[i]
```

Яке значення змінної **suma** після завершення його виконання? Сформулюйте умову задачі, яку можна розв'язати, використовуючи цей фрагмент проекту.

4. Створіть проект, у якому створюється список із вкладів десяти клієнтів у банк. Через рік вклад кожного збільшується на 20 %. Виведіть суми, які можуть забрати ці клієнти через рік. Використайте спосіб уведення початкових сум у список по одному в рядку. Збережіть проект у вашій папці у файлі з іменем **завдання 3.1.4**.

5. Створіть проект, у якому створюється список із вкладів десяти клієнтів у банк. Через рік вклад кожного збільшується на **p** %. Виве-

дідь прибутки, які отримають ці клієнти, якщо заберуть гроші через рік. Використайте спосіб уведення початкових сум у список в одному рядку через пропуски. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.1.5**.

6. Створіть проєкт, у якому створюються два списки з кількостями опадів у місті Житомир за кожен день першої декади двох місяців. Визначте середню кількість опадів у Житомирі за перші 10 днів кожного із цих місяців. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.1.6**.

7. Робітник протягом місяця виготовляв деталі шести видів. Відомо, скільки деталей кожного виду виготовив робітник і скільки гривень йому платять за виготовлення 1 деталі кожного виду. Створіть проєкт, у якому визначається, скільки гривень отримає робітник за виготовлення деталей кожного виду. Використайте три списки: для кількостей деталей кожного із 6 видів, для суми оплати за 1 деталь кожного виду і для сум за всі деталі кожного виду. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.1.7**.

8. Експериментально встановлено, що кількість людей, серед яких поширюється повідомлення в мережі, обчислюється за формулою $n = 5 * i + 2$, де i – номер хвилини поширення. Створіть проєкт для визначення, серед скількох людей поширюється повідомлення за кожну з перших 20 хвилин. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.1.8**.



Готуємось до вивчення нового матеріалу

1. Які функції ви використовували в табличному процесорі Excel?
2. Що таке *функція* у проєкті мовою програмування Python? Для чого їх використовують?



3.2. ОПРАЦЮВАННЯ СПИСКІВ

У цьому пункті ви дізнаєтеся про:

- ▶ операції над списками;
- ▶ функції та методи для опрацювання списків.

ОПЕРАЦІЇ НАД СПИСКАМИ



Пригадайте

- Що таке *список*? Які імена мають елементи списку?
- Які ви знаєте способи створення списку в пам'яті комп'ютера?
- Які ви знаєте способи виведення значень елементів списку?

Ви вже знаєте, що кожен елемент списку має свій порядковий номер (індекс) і що ця нумерація розпочинається з 0. Але для елементів з кінця списку використовують і від'ємні цілі індекси. Так, для списку **a** елемент **a[-1]** – це перший елемент із кінця списку (останній елемент), елемент **a[-2]** – це другий елемент із кінця списку (передостанній елемент) і т. д. Наприклад, якщо **a = [-2, 4, 5, 7, 0, 15]**, то **a[-1] = 15**, **a[-2] = 0**, **a[-3] = 7**.

Кілька списків можна **об'єднувати в один**, використавши операцію **+**. Наприклад, якщо **a = [2, 4, 6, 8]** і **b = [10, 12]**, то їх об'єднанням буде такий список: **a + b = [2, 4, 6, 8, 10, 12]**.

Елементи списку можна **дублювати (розмножувати)**, застосувавши операцію `*`. Наприклад, якщо `a = [2, 4, 6]`, то результатом операції `a*2` буде такий список: `a*2 = [2, 4, 6, 2, 4, 6]`, а результатом операції `a*3` буде такий список: `a*3 = [2, 4, 6, 2, 4, 6, 2, 4, 6]`.

Із будь-якого списку можна виокремити його частину. Ця операція називається **добування зрізу зі списку**. Загальний вигляд команди з використанням цієї операції такий: `b = a[i:j:step]`, де `b` – змінна, значенням якої є список, який складається із частини елементів списку `a`; `i` – індекс початку зрізу, `j` – його закінчення, причому елемент із номером `j` у зріз уже не входить; `step` (англ. *крок*) – крок, з яким добуваються елементи, – може бути як додатним, так і від’ємним цілим числом.

Якщо елементи добуваються поспіль, тобто `step = 1`, то параметр `step` можна опустити. Якщо відсутній індекс початку зрізу, то зріз береться від 0-го до (j-1)-го елемента; за відсутності індексу закінчення зрізу, зріз береться від i-го елемента до кінця списку.

Наприклад, якщо `a = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]` (табл. 3.4), то:

Таблиця 3.4

Номери (індекси) елементів списку та значення елементів списку

Номери (індекси) елементів списку	0	1	2	3	4	5	6	7	8	9
Значення елементів списку	2	4	6	8	10	12	14	16	18	20

- `b = a[2:6] = [6, 8, 10, 12]` (`i = 2, j = 6, step = 1`);
- `b = a[1:8:2] = [4, 8, 12, 16]` (`i = 1, j = 8, step = 2`);
- `b = a[4:] = [10, 12, 14, 16, 18, 20]` (`i = 4, j = and, step = 1`);
- `b = a[:3] = [2, 4, 6]` (`i = 0, j = 3, step = 1`);
- `b = a[9:5:-1] = [20, 18, 16, 14]` (`i = 9, j = 5, step = -1`);
- `b = a[::2] = [2, 6, 10, 14, 18]` (`i = 0, j = and, step = 2`);
- `b = a[::-1] = [20, 18, 16, 14, 12, 10, 8, 6, 4, 2]` (`i = and, j = 0, step = -1`);
- `b = a[-4:-1] = [14, 16, 18]` (`i = -4, j = -1, step = 1`).

Задача 1. Дано список із чисел. Чи є цей список паліндромом?

Паліндром (від грец. *πάλιν* – назад, знов і грец. *δρόμος* – біг) – це число, слово, послідовність слів або чисел, які збігаються з перевернутими.

Приклади чисел-паліндромів: 101, 2222, 123321. Приклади слів-паліндромів: око, радар, ротатор, піп, Пилип. Приклади послідовностей слів-паліндромів: «я несу гусеня», «аби ріці риба» (тут пропуски між словами не враховуються). Приклад послідовностей чисел-паліндромів: 3 5 3, 2 4 6 6 4 2.

Розв’яжемо цю задачу двома способами.

Спосіб 1. Перевернемо список, і якщо сам список збігається з перевернутим, то це паліндром.

```
a = list(map(float, input('Уведіть значення елементів списку: ').split()))
b = a[::-1]
if a == b: print('Паліндром')
else: print('Не паліндром')
```

Спосіб 2. Розглянутий спосіб хоча й найпростіший, але не раціональний. Адже якщо вже перший та останній елементи списку не збігаються, то зрозуміло, що такий список не є паліндромом і далі перевіряти нічого не потрібно.

Тому перевірятимемо елементи списку, рівновіддалені від його початку та кінця. І якщо хоча б одна з таких пар елементів не збігається, то такий список не є паліндромом. А якщо всі такі пари елементів збігаються, то такий рядок є паліндромом.

```
a = list(map(float, input('Уведіть значення елементів списку: ').split()))
f = True # Припускаємо, що список є паліндромом
for i in range(len(a)//2): # Переглядаємо значення елементів списку, які рівновіддалені від початку та кінця і до половини довжини списку
    if a[i] != a[-i-1]: # Якщо рівновіддалені елементи не збігаються,
        f = False # то цей список не є паліндромом
        break # І далі нічого перевіряти не потрібно
if f: print('Паліндром')
else: print('Не паліндром')
```

У багатьох задачах потрібно з'ясувати, чи є у списку певне значення, чи немає. Для цього можна застосувати операцію `in`. Ви вже застосовували цю операцію в циклах з лічильником. Наприклад, для списку `a = [2, 4, 6]`: `2 in a = True`, `3 in a = False`.

Наведемо приклади двох таких задач.

Задача 2. У списку містяться річні оцінки учнів / учениць класу з інформатики. Чи є серед цих оцінок оцінка 3?

```
a = list(map(float, input('Уведіть оцінки учнів класу: ').split()))
if 3 in a: print('Оцінка 3 серед оцінок учнів / учениць класу є')
else: print('Оцінки 3 серед оцінок учнів / учениць класу немає')
```

Задача 3. У списку містяться річні оцінки учнів / учениць класу з інформатики. Чи є серед учнів / учениць ті, хто має оцінки початкового рівня?

Оцінки початкового рівня – це 1, або 2, або 3.

```
a = list(map(float, input('Уведіть оцінки учнів / учениць класу: ').split()))
if 1 in a or 2 in a or 3 in a: print('Учні / учениці з оцінками початкового рівня є')
else: print('Учні / учениць з оцінками початкового рівня немає')
```

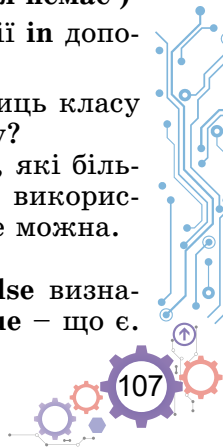
Але не для всіх аналогічних задач використання операції `in` допоможе їх розв'язати.

Задача 4. У списку містяться річні оцінки учнів / учениць класу з інформатики. Чи є серед них оцінки, які більші за введену?

На відміну від **Задачі 3**, ми не знаємо, скільки є оцінок, які більше за ту, яку введуть. Тому ідею розв'язування **Задачі 3** з використанням диз'юнкції (операції `or`) для **Задачі 4** використати не можна.

Скористаємося іншою ідеєю.

Використаємо логічну змінну `f`, і нехай її значення `False` визначатиме, що шуканих оцінок у списку немає, а значення `True` – що є.



Перед початком перегляду значень елементів списку присвоїмо змінній *f* значення **False**, адже шукані оцінки у списку поки що не знайшлися. Перевірятимемо в циклі послідовно значення елементів списку, чи є це значення більше за введене число. І якщо значення хоча б одного з елементів списку буде більше за введене число, то змінимо значення змінної *f* на **True** і перервемо виконання циклу командою **break**. Виведення кінцевого результату реалізуємо залежно від значення змінної *f*.

```
a = list(map(float, input('Уведіть оцінки учнів / учениць класу: ').split()))
x = int(input('Уведіть оцінку, вище якої потрібно шукати: '))
f = False # Шуканих оцінок поки що не знайдено
for i in range(len(a)):
    if a[i] > x:
        f = True # Шукані оцінки є
        break # Перериваємо виконання циклу, бо хоча б одна
              # шукана оцінка знайшлася
if f: print('Оцінки, які більші за введену, є')
else: print('Оцінок, які більші за введену, немає')
```

У мові програмування **Python** цикл **for** має частину **else**, яка виконується, якщо переривання циклу не відбулося. Тому інший спосіб розв'язування **Задачі 4** може бути такий:

```
a = list(map(float, input('Уведіть оцінки учнів / учениць класу: ').split()))
x = int(input('Уведіть оцінку, вище якої потрібно шукати: '))
for i in range(len(a)):
    if a[i] > x:
        print('Оцінки, які більші за введену, є') # Шукані оцінки є
        break # Перериваємо виконання циклу, бо хоча б одна шу-
              # кана оцінка знайшлася
else: print('Оцінок, які більші за введену, немає')
```

ВИКОРИСТАННЯ ФУНКЦІЙ ДЛЯ ОПРАЦЮВАННЯ СПИСКІВ



Пригадайте

- Що таке *функція* у проєкті мовою програмування **Python**?
- Що таке *аргумент функції*? Скільки аргументів може мати функція? Як передаються у функцію значення її аргументів?
- Що таке *результат функції*? Як передаються (повертаються) значення результатів функції в команду її виклику?
- Яке призначення функцій **sum**, **min**, **max** у табличному процесорі **Excel**?

Для опрацювання списків можна використати **функції**. Можна використати стандартні (вбудовані) функції, а можна створювати власні функції (функції користувача). Деякі зі стандартних функцій ви використовували в попередньому пункті. Наприклад, функцію **len** для визначення довжини (кількості елементів) списку.

Наведемо в таблиці 3.5 ще деякі стандартні функції для опрацювання списків, їх призначення та приклади їх використання для списку **a = [2, 4, 6, 8, 4]**.

Таблиця 3.5

Деякі стандартні функції для опрацювання списків

Функції	Призначення	Приклад використання	Результат використання
sum	Знаходить суму значень усіх елементів списку	x = sum(a)	x = 24
min	Знаходить найменше значення серед значень елементів списку	x = min(a)	x = 2
max	Знаходить найбільше значення серед значень елементів списку	x = max(a)	x = 8

Розглянемо, як відбувається обчислення результатів функцій, наведених у таблиці 3.5.

1. Функція **sum**. Аргументом цієї функції буде список, результатом – сума значень його елементів.

Визначення суми значень усіх елементів списку здійснюється з використанням циклу з лічильником. Кількість елементів списку визначимо, використовуючи функцію **len**. Використаємо змінну **s** для накопичування шуканої суми. Перед циклом їй необхідно надати значення 0. У циклі послідовно переглядатимемо елементи списку і додаватимемо їхні значення до змінної **s**.

```
def sum(a):
    s = 0
    for i in range(len(a)):
        s = s + a[i]
    return s
```

або так, переглядаючи елементи списку не за індексами, а безпосередньо за значеннями:

```
def sum(a):
    s = 0
    for x in a:
        s = s + x
    return s
```

Оскільки в **Python** функція **sum** є вбудованою, то у проєкті, у якому вона використовується, її розписувати не потрібно:

```
a = [2, 12, -5, 7, 4]
suma = sum(a)
print(suma)
```

2. Функція **min**. Аргументом цієї функції буде список, результатом – найменше серед значень його елементів.

Спочатку вважатимемо, що значення початкового елемента списку є найменшим. Після чого послідовно переглядатимемо значення всіх інших елементів списку, і якщо трапиться значення, яке менше від того, яке ми на даний момент вважаємо найменшим, то вважатимемо тепер уже це значення найменшим.

```
def min(a):
    m = a[0] # Вважаємо, що найменше значення – значення по-
              чаткового елемента
    for i in range(1, len(a)):
        if a[i] < m:
            m = a[i] # Замінюємо найменше значення
    return m
```

Оскільки в **Python** функція **min** теж є вбудованою, то у проєкті, у якому вона використовується, її розписувати не потрібно:

```
a = [2, 12, -5, 7, 4]
minimum = min(a)
print(minimum)
```

3. Функція **max**.

Обчислення значення цієї функції відбувається аналогічно до обчислення значення функції **min**.

Поміркуйте, що потрібно змінити в командах функції, щоб змінений варіант знаходив найбільше серед значень елементів списку.

Розглянемо задачу, у розв'язуванні якої використовуються функції знаходження найбільшого та найменшого значень у списку.

Задача 5. У списку зберігаються середні температури за кожен із днів місяця. Знайти різницю між найбільшою і найменшою середньою температурою цього місяця.

Звичайно, можна було б, використовуючи вбудовані функції **min** і **max**, знайти найменшу і найбільшу середню температуру і потім знайти їхню різницю. Але за цього проєкту доведеться двічі переглядати список температур. А чи не можна створити проєкт, у якому буде знайдено найбільше і найменше серед значень елементів списку, але список переглядатиметься лише один раз? Від цього час виконання проєкту суттєво зменшиться. І що більшою буде кількість елементів списку (середні температури протягом кількох місяців, протягом року тощо), то економія часу виконання проєкту буде більшою.

Спочатку вважатимемо, що значення початкового елемента списку є і найбільшим, і найменшим. Після чого послідовно переглядатимемо всі інші елементи списку. Якщо значення чергового елемента списку більше за те значення, яке ми на той момент вважаємо найбільшим, то замінюємо найбільше значення серед уже переглянутих. Якщо ні, то, можливо, це значення менше від того, яке ми поки що вважаємо найменшим. І якщо це так, то замінюємо найменше значення серед уже переглянутих.

Оформимо цей фрагмент проєкту як користувацьку функцію з іменем **minimax**.

```
def minimax(a):
    maximum = a[0]
    minimum = a[0]
    for i in range(1, len(a)):
        if a[i] > maximum: # Порівняння значення чергового еле-
                          # мента з найбільшим на цей момент
            maximum = a[i] # Замінюємо значення найбільшого
                          # елемента
```

```

elif a[i] < minimum: # Порівняння значення чергового еле-
                     # мента з найменшим на цей момент
    minimum = a[i] # Замінюємо значення найменшого еле-
                     # мента
rize = maximum - minimum # Обчислення різниці між найбіль-
                           # шим і найменшим значеннями
return rize # Повернення різниці в команду виклику функції

```

ВИКОРИСТАННЯ МЕТОДІВ ДЛЯ ОПРАЦЮВАННЯ СПИСКІВ



Пригадайте

- Які методи вам відомі для опрацювання списків?
- Для чого призначені ці методи?

Для опрацювання списків можна використовувати **методи**. Один із них (метод **append**) ви вже використовували в попередньому пункті для додавання значення в кінець списку. Наведемо в таблиці 3.6 деякі інші методи, їх призначення та приклади використання для списку **a = [2, 4, 6, 8, 4]**.

Таблиця 3.6

Деякі методи для опрацювання списків

Метод	Призначення	Приклад використання	Результат використання
clear (англ. очистити)	Видаляє всі елементи зі списку	a.clear()	a = []
insert (англ. уставити)	Уставляє значення у список як елемент із зазначеним індексом; у дужках спочатку вказується індекс, а потім – значення	a.insert(1,3)	a = [2, 3, 4, 6, 8, 4]
remove (англ. видалити)	Видаляє перший елемент зі списку, що має вказане значення	a.remove(4)	a = [2, 6, 8, 4]
pop (англ. хлоп'як)	Видаляє елемент із зазначеним індексом зі списку і може його зберегти; якщо в дужках індекс не зазначено, то видаляється останній елемент	x = a.pop(2) x = a.pop() a.pop(1)	x = 6 a = [2, 4, 8, 4] x = 4 a = [2, 4, 6, 8] a = [2, 6, 8, 4]
count (англ. кількість)	Визначає кількість входжень вказаного значення у список	x = a.count(4) x = a.count(5)	x = 2 x = 0
index (англ. індекс)	Визначає індекс першого входження вказаного значення у список. Якщо вказаного значення у списку немає, то видає помилку	x = a.index(4) x = a.index(5)	x = 1 ValueError: 5 is not in list



Звертаємо вашу увагу: оскільки за відсутності у списку значення, вказаного в методі **index**, виникає помилка, то використовувати цей метод доцільно тільки тоді, коли відомо, що вказане значення у списку точно є. Наприклад, можна використати такий фрагмент проєкту:

```
if t in a:
    x = a.index(t)
    print(x)
else: print ('Цього значення у списку немає')
```

Розглянемо кілька задач, які розв'язуються з використанням функцій і методів опрацювання списків.

Задача 6. У списку **a** зберігаються семестрові оцінки учня / учениці. Визначити найвищу оцінку цього учня / учениці та зі скількох предметів він / вона має цю найвищу оцінку.

Визначимо найвищу оцінку, використавши функцію **max**. Визначимо, скільки таких є найвищих оцінок, використавши метод **count**.

Оформимо розв'язування цієї задачі як функцію.

До розгляду цієї задачі функції, які ми створювали та використовували, мали тільки один результат. У мові програмування **Python** функції можуть мати кілька результатів. Зокрема, у цій задачі функція повинна мати два результати. Вона має бути такою:

```
def maxmarks (a):
    m = max(a)
    n = a.count(m)
    return m, n
```

Команда виклику цієї функції в основній частині проєкту буде такою: **x, y = maxmarks (a)**. Після її виконання значення змінної **x** дорівнюватиме значенню змінної **m** – найвища оцінка, а значення змінної **y** дорівнюватиме значенню змінної **n** – кількість найвищих оцінок.

Задача 7. У списку **a** зберігаються семестрові оцінки учня / учениці. Визначити кількість предметів, оцінка за які нижча від середньої.

Для визначення середньої оцінки використаємо функцію **sum**, результат якої (сума всіх оцінок) розділимо на кількість предметів. Для визначення кількості предметів, оцінка за які нижча від середньої, уведемо змінну **k**, якій присвоїмо початкове значення **0**. Далі в циклі переглядатимемо оцінки (значення елементів списку), і якщо деяка оцінка виявиться меншою від знайденої середньої оцінки, збільшимо кількість таких оцінок (значення змінної **k**) на 1.

Оформимо розв'язування цієї задачі як функцію:

```
def num(a):
    mid = sum(a) / len(a)
    k = 0
    for x in a:
        if x < mid: k = k + 1 (або скорочено: k += 1)
    return k
```

Команда виклику цієї функції в основній частині проєкту буде такою: **t = num (a)**. Після її виконання значення змінної **t** дорівнюватиме кількості оцінок, які менші від середнього значення оцінок зі списку **a**.



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Задача. Створити список із температур о 8-й годині ранку кожного дня першої декади (перших десяти днів) місяця. Визначити середню із цих температур. Сформувати другий список із номерів днів, коли температура була від'ємною. Вивести значення елементів другого списку.

1. Відкрийте вікно середовища створення проєкту.
2. Уведіть такий текст проєкту:

```

a = []
for i in range(10):
    x = float(input('Уведіть температуру: '))
    a.append(x)
mid = sum(a) / 10
print(mid)
b = []
for i in range(10):
    if a[i] < 0: b.append(i + 1)
if len(b) == 0: print("Від'ємних температур немає")
else: print(*b)

```

3. Збережіть проєкт у вашій папці у файлі з іменем **вправа 3.2.py**.
4. Запустіть проєкт на виконання.
5. Уведіть значення температур: 3 4 -7 0 -3 2.5 -2 -3 5 1.5.
6. Проаналізуйте отримані результати.
7. Запустіть проєкт на виконання другий раз.
8. Уведіть інші значення температур: 6 9 8.2 5 0 4 5 1 29 2 6.
9. Проаналізуйте отримані результати.
10. Запустіть проєкт на виконання.
11. Уведіть свій набір температур.
12. Проаналізуйте отримані результати.
13. Закрийте всі відкриті вікна.



Найважливіше в цьому пункті

Для елементів списку використовують як цілі невід'ємні, так і цілі від'ємні індекси. Так, для списку **a** елемент **a[-1]** – це перший елемент із кінця списку (останній елемент), елемент **a[-2]** – це другий елемент із кінця списку (передостанній елемент) і т. д.

Кілька списків можна **об'єднувати в один**, використавши операцію **+**. Наприклад, якщо **a = [2, 4, 6, 8]** і **b = [10, 12]**, то їх об'єднанням буде такий список:

a + b = [2, 4, 6, 8, 10, 12].

Елементи списку можна **дублювати (розмножувати)**, використавши операцію *****. Наприклад, якщо **a = [2, 4, 6]**, то результатом операції **a*2** буде такий список:

a*2 = [2, 4, 6, 2, 4, 6].

а результатом операції **a*3** буде такий список:

a*3 = [2, 4, 6, 2, 4, 6, 2, 4, 6].

Із будь-якого списку можна виокремити його частину в інший список. Ця операція називається **добуванням зрізу зі списку**. Загальний вигляд команди з використанням цієї операції такий: **b = a[i:j:step]**, де **b** – змінна, значенням якої є список, який складається із частини елементів списку **a**; **i** – індекс початку зрізу, **j** – його закінчення, причому елемент із номером **j** у зріз уже не входить; **step** (англ. крок) – крок, з яким добуваються елементи, – може бути як додатним, так і від'ємним цілим числом.



Якщо елементи добуваються поспіль, тобто **step = 1**, то параметр **step** можна опустити. Якщо відсутній індекс початку зрізу, то зріз береться від **0**-го до **(j-1)**-го елемента; якщо відсутній індекс закінчення зрізу, зріз береться від **i**-го елемента до кінця списку.

Для визначення, чи є у списку певне значення, чи немає, можна використати операцію **in**. Наприклад, для списку **a = [2, 4, 6]**: **2 in a = True**, **3 in a = False**.

Функція **sum** знаходить суму значень усіх елементів списку, функція **min** знаходить найменше значення серед значень елементів списку, функція **max** знаходить найбільше значення серед значень елементів списку.

Метод **clear** видаляє всі елементи зі списку; метод **insert** уставляє значення у список як елемент із зазначеним індексом; метод **remove** видаляє перший елемент зі списку, що має вказане значення; метод **pop** видаляє елемент із зазначеним індексом зі списку і може його зберегти як значення деякої змінної; метод **count** визначає кількість входжень вказаного значення у список; метод **index** визначає індекс першого входження вказаного значення у список; а якщо вказаного значення у списку немає, то видає помилку.



Дайте відповіді на запитання

1. Які ви знаєте операції, що можна виконувати над списком? Для чого можна використовувати кожен з них? Наведіть приклади.
2. Які ви знаєте функції, що можна використовувати для опрацювання списку? Для чого можна використовувати кожен з них? Наведіть приклади.
3. Які ви знаєте методи, що можна використовувати для опрацювання списку? Для чого можна використовувати кожен з них? Наведіть приклади.



Обговоріть і зробіть висновки

1. Для розв'язування яких задач можна використати наведені фрагменти проєктів?

- а)

```
s = 0
for i in range(10):
    if a[i] > 0: s = s + a[i]
```
- б)

```
k = 0
for i in range(10):
    if a[i] < 0: k = k + 1
```
- в)

```
k = 0
for i in range(10):
    if a[i] == x: k = k + 1
```

2. З якою метою в наведеному фрагменті проєкту використано змінну **f**?

```
f = False
for i in range(10):
    if a[i] == x:
        f = True
        break
```

Який тип цієї змінної? Чи можна було б із цією самою метою використати змінну іншого типу? Відповідь поясніть.

3. Як потрібно змінити наведений фрагмент, щоб у ньому визначалося найменше значення елементів списку?

```
maximum = a[0]
for i in range(1,10):
    if a[i] > maximum: maximum = a[i]
```

4. Якщо у списку кілька елементів дорівнюють найбільшому значенню серед значень усіх елементів цього списку, то фрагмент:

```
maximum = a[0]
nommax = 0
for i in range(1,10):
    if a[i] > maximum:
        maximum = a[i]
    nommax = i
```

визначає номер якого з них? Чи зміниться щось, якщо в розгалуженні змінити знак > на знак >=?

5. Для розв'язування якої задачі можна використати наведені фрагменти? Порівняйте їхню ефективність. Відповідь поясніть.

а)

```
maximum = a[0]
nmax = 0
minimum = a[0]
nmin = 0
for i in range(1, 10):
    if a[i] > maximum:
        maximum = a[i]
        nmax = i
    elif a[i] < minimum:
        minimum = a[i]
        nmin = i
a[nmax] = minimum
a[nmin] = maximum
```

б)

```
maximum = a[0]
nmax = 0
minimum = a[0]
nmin = 0
for i in range(1, 10):
    if a[i] > maximum:
        maximum = a[i]
        nmax = i
    if a[i] < minimum:
        minimum = a[i]
        nmin = i
a[nmax] = minimum
a[nmin] = maximum
```

6. Для розв'язування якої задачі можна використати наведений фрагмент?

```
s = 0
k = 0
for x in a:
    if x > 0:
        s = s + x
        k = k + 1
sa = s / k
```

У яких випадках під час виконання проєкту із цим фрагментом може виникнути помилка? Відповідь поясніть.



Виконайте завдання

1. Нехай $a = [2, 3, 6, 11, 10, 15, 4, 16, 28, 40]$. Запишіть у зошит результати виконання команд:

- $b = a[2:6]$;
- $b = a[1:8:2]$;
- $b = a[4:]$;
- $b = a[:3]$;
- $b = a[9:5:-1]$;



- е) $b = a[::2];$
- ж) $b = a[:: -1];$
- з) $b = a[-4:-1].$

2. Створіть проєкт, у якому в списку зберігаються значення середніх денних температур за кожен день тижня і визначається середнє арифметичне від'ємних із них. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.2.**

3. Створіть віконний проєкт, у якому в списку зберігаються кількості відвідувачів музею в кожен із 10 днів місяця і визначається, скільки днів кількість відвідувачів перевищувала задану кількість осіб. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.3.**

4. Створіть віконний проєкт, у якому в списку зберігаються семестрові оцінки учня з 12 предметів і визначається, чи є серед них оцінки, вищі за задану. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.4.**

5. Випускник / випускниця 9 класу може одержати *Свідоцтво з відзнакою*, якщо всі його / її річні оцінки не менші ніж 10. Створіть проєкт для визначення, чи отримає цей / ця учень / учениця *Свідоцтво з відзнакою*. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.5.**

6. Створіть проєкт, у якому для довільного списку з різних дійсних чисел визначається найменший елемент і його значення обмінюється місцями з початковим елементом списку. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.6.**

7. Створіть проєкт, у якому для довільного списку із різних дійсних чисел визначається найбільший елемент і його значення обмінюється місцями з останнім елементом списку. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.7.**

8. Створіть проєкт, у якому для довільного списку з різних дійсних чисел визначаються найменший і найбільший елементи. Найбільший елемент поміняйте місцями з елементом з індексом 2, а найменший – з передостаннім. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.8.**

9. Створіть проєкт, у якому в списку зберігаються семестрові оцінки учня / учениці з 15 предметів і визначається, яких оцінок у нього / неї найбільше: початкового рівня (1–3), середнього рівня (4–6), достатнього рівня (7–9) чи високого рівня (10–12). Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.9.**

10. Створіть проєкт, у якому в списку зберігаються кількості деталей, виготовлених робітником за кожен день протягом першої половини місяця, визначається найбільша кількість деталей, виготовлених упродовж одного дня, і протягом скількох днів він виготовляв цю найбільшу кількість деталей. Створіть два розв'язки задачі. Перше – у якому список переглядатиметься двічі, і друге – у якому список переглядатиметься один раз. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.10.**

11. Створіть проєкт, у якому в списку зберігаються кількості опадів за кожен день протягом місяця, визначається середня кількість опадів протягом цього місяця і в які дати кількість опадів більша за середню, а в які – менша від середньої. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.2.11.**



Практична робота № 4.

«Опрацювання списків»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Створіть проєкт для розв'язування задачі: *Уведіть 8 оцінок учня / учениці, отриманих ним / нею протягом вивчення теми. Визначте середню оцінку учня / учениці за тему. Визначте, скільки оцінок, вищих за 7.*
2. Збережіть проєкт у вашій папці у файлі з іменем **практична 4-1.ру.**
3. Створіть проєкт для розв'язування задачі: *Уведіть час, який показав кожен з учасників змагань вашого закладу освіти з бігу на дистанцію 60 метрів. Визначте найбільший і найменший час. Визначте, скільки спортсменів показали найкращий час. Чи оновлено на цих змаганнях рекорд вашого закладу освіти?*
4. Збережіть проєкт у вашій папці у файлі з іменем **практична 4-2.ру.**



Готуємось до вивчення нового матеріалу

1. Для чого впорядковують дані у списках, таблицях тощо?
2. Які є види сортування даних?
3. Як ви сортували дані в таблицях табличного процесора?



3.3. СОРТУВАННЯ ДАНИХ У СПИСКАХ. ПОШУК У СПИСКАХ ДАНИХ, ЯКІ ВІДПОВІДАЮТЬ ПЕВНИМ УМОВАМ

У цьому пункті ви дізнаєтеся про:

- ▶ сортування даних у списках;
- ▶ пошук у списках даних, які відповідають певним умовам.

СОРТУВАННЯ ДАНИХ У СПИСКАХ



Пригадайте

- Який набір даних вважається відсортованим? ● Як сортуються дані в таблицях Excel? ● Чим відсортований набір даних кращий за невідсортований?

Список **a** називається відсортованим за зростанням (зростаючим), якщо значення кожного його наступного елемента більше за значення попереднього, тобто для всіх i виконується нерівність $a[i+1] > a[i]$.

Список **a** називається відсортованим за спаданням (спадним), якщо значення кожного його наступного елемента менше від значення попереднього, тобто для всіх i виконується нерівність $a[i+1] < a[i]$.

Список **a** називається відсортованим за неспаданням (неспадним), якщо значення кожного його наступного елемента не менше (більше або дорівнює) за значення попереднього, тобто для всіх i виконується нерівність $a[i+1] \geq a[i]$.

Список **a** називається відсортованим за незростанням (незростаючим), якщо значення кожного його наступного елемента не більше (менше або дорівнює) за значення попереднього, тобто для всіх i виконується нерівність $a[i+1] \leq a[i]$.

Як вам уже відомо, пошук потрібних даних у відсортованих наборах даних відбувається значно швидше, ніж у невідсортованих.

Наприклад, у класному журналі список прізвищ учнів / учениць відсортовано в алфавітному порядку для зручності пошуку потрібного учня / учениці.

Для сортування даних у списку можна використати:

- стандартний метод **sort**, який сортує список за зростанням (неспаданням) **sort()** або за спаданням (незростанням) **sort(reverse = True)**; у результаті виконання цього методу список, до якого він застосований, буде відсортований; початковий список не зберігається; приклади відповідних команд:

a.sort() і **a.sort(reverse = True)**.

- стандартну функцію **sorted(список)**, яка утворює новий список, відсортувавши список, який зазначено як аргумент цієї функції; якщо аргументом цієї функції є тільки список, то він буде відсортований за зростанням; якщо вказати ще й другий аргумент **sorted(список, reverse = True)**, то список буде відсортований за спаданням; водночас початковий список зберігається і не змінюється; приклади відповідних команд:

b = sorted(a) і **b = sorted(a, reverse = True)**.

Метод **sort** для сортування списку виконується доволі швидко, навіть для списків великої довжини. Але він досить складний для розуміння учнями / ученицями.

Тому пояснимо деякі ідеї сортування даних у списках, розглянувши два інших методи сортування, які працюють дещо повільніше, але простіші для розуміння. Їх також можна буде використати у проектах, у яких використання лише метода **sort** не дає потрібного результату.

СОРТУВАННЯ МЕТОДОМ ВИБОРУ



Пригадайте

- Як знайти найменше і найбільше значення у списку?
- Як визначити індекс першого входження певного значення у список?

Розглянемо **алгоритм сортування даних у списку методом вибору**. Будемо сортувати дані у списку за зростанням.

Пояснимо ідею цього алгоритму на прикладі.

Нехай маємо такий список із 6 чисел. Він невідсортований.

23	7	4	16	-2	10
----	---	---	----	----	----

Виберемо найменший елемент списку (це – число -2)

23	7	4	16	-2	10
----	---	---	----	----	----

і поміняємо його місцями з початковим елементом списку.

Отримаємо список, у якому початковий елемент є найменшим серед усіх елементів списку. Тобто початковий елемент списку вже зайняв своє місце в майбутньому відсортованому списку.

-2	7	4	16	23	10
----	---	---	----	----	----

Виберемо тепер найменший елемент невідсортованої частини списку (це – число 4)

-2	7	4	16	23	10
----	---	---	----	----	----

і поміняємо його місцями з першим елементом невідсортованої частини списку (це – число 7).

Отримаємо список, у якому перші два елементи відсортовані за зростанням. Тобто вони вже зайняли свої місця в майбутньому відсортованому списку:

-2	4	7	16	23	10
----	---	---	----	----	----

Знову виберемо найменший елемент невідсортованої частини списку (це – число 7)

-2	4	7	16	23	10
----	---	---	----	----	----

і поміняємо його місцями з першим елементом невідсортованої частини списку (це в цьому випадку – те саме число 7).

У цьому разі вони збіглися. І ми отримаємо список, у якому перші три елементи відсортовано за зростанням. Тобто вони вже зайняли своє місце в майбутньому відсортованому списку:

-2	4	7	16	23	10
----	---	---	----	----	----

Ще раз виберемо найменший елемент невідсортованої частини списку (це – число 10) і поміняємо його місцями з першим елементом невідсортованої частини списку (це – число 16). Отримаємо такий список:

-2	4	7	10	23	16
----	---	---	----	----	----

у якому перші чотири елементи зайняли свої місця в майбутньому відсортованому списку.

Ще один раз виконаємо ці операції – і отримаємо відсортований за зростанням список:

-2	4	7	10	16	23
----	---	---	----	----	----

Звертаємо вашу увагу, що на останньому кроці виконання алгоритму свої місця зайняли одразу два елементи списку.

Отже, ми мали список із 6 чисел. І ми 5 разів виконували такі дії:

1. **Вибирали найменший елемент серед елементів поки що невідсортованої частини списку.**
2. **Обмінювали цей вибраний елемент із першим елементом поки що невідсортованої частини списку.**

Створимо проєкт для сортування за зростанням методом вибору елементів списку із 6 дійсних чисел.

```
a = list(map(float, input('Уведіть 6 чисел через пропуск > ').split()))
# Уведення значень елементів списку із 6 дійсних чисел
for i in range(5): # 5 разів повторюємо
    mini = a[i] # Перший елемент невідсортованої поки що частини списку вважаємо найменшим
```

```

nmini = i # Номер першого елемента невідсортованої поки що
           частини списку вважаємо номером найменшого
for j in range(i+1, 6): # Переглядаємо елементи списку, почи-
                        наючи з наступного i до останнього, –
                        можливо, серед них є менший, ніж
                        той, який ми на цей момент вважаємо
                        найменшим у невідсортованій частині
    if a[j]<mini: # Якщо трапляється елемент, менший ніж
                той, який ми вважаємо найменшим у невід-
                сортованій частині списку, він стає наймен-
                шим і його номер стає номером найменшого

        mini = a[j]
        nmini = j
a[nmini] = a[i] # На місце знайденого найменшого елемента
                невідсортованої поки що частини списку ста-
                вимо перший елемент невідсортованої части-
                ни списку
a[i] = mini # На місце першого елемента невідсортованої поки
            що частини списку ставимо знайдений найменший
print(a) # Виведення відсортованого списку

```

І другий варіант, трохи коротший:

```

a = list(map(float, input('Уведіть 6 чисел через пропуск > ').split()))
n = len(a)
first = 0
for i in range(n-1):
    m = mini(a[first:]) # Визначаємо найменше значення в частині
                        списку від first і до кінця
    ind = a.index(m) # Визначаємо його індекс
    a[ind], a[first] = a[first], a[ind] # Міняємо місцями елементи
                                    з індексами ind і first
    first += 1 # Збільшуємо first на 1
print(a)

```



Обговоріть і зробіть висновки

1. Як змінити наведені проекти, щоб вони сортували списки за спаданням?
2. Чи правильно сортуватимуть дані у списку вищенаведені проекти, якщо список міститиме рівні між собою числа? Відповідь поясніть.
3. Перегляньте в інтернеті візуалізацію методу вибору, наприклад за адресами: <https://cutt.ly/HtSyWS40> або <https://www.toptal.com/developers/sorting-algorithms>.

СОРТУВАННЯ МЕТОДОМ ОБМІНУ



Поміркуйте

- Чи правильно для обміну значеннями двох змінних **x** та **y** використати такі дві команди: **x = y** і **y = x**? Відповідь поясніть. Наведіть приклад.
- Які ви знаєте способи поміняти місцями значення двох змінних **x** та **y**?

Розглянемо алгоритм сортування списку методом обміну. Будемо, як і в попередньому алгоритмі, упорядковувати список за зростанням.

Для пояснення ідеї цього алгоритму розглянемо для прикладу також невідсортований список із 6 чисел.

13	6	5	18	11	10
----	---	---	----	----	----

Братимемо по черзі два сусідніх елементи списку (перший і другий, другий і третій, третій і четвертий і т. д.) і, якщо лівий з них більший за правий, мінятимемо їх місцями.

Беремо перший і другий елементи списку: 13 і 6, $13 > 6$, тому міняємо їх місцями й отримаємо такий список:

6	13	5	18	11	10
---	----	---	----	----	----

Беремо другий і третій елементи списку: 13 і 5, $13 > 5$, тому міняємо їх місцями й отримаємо такий список:

6	5	13	18	11	10
---	---	----	----	----	----

Беремо третій і четвертий елементи списку: 13 і 18, 13 не більше 18 , тому не міняємо їх місцями, і список залишається той самий.

Беремо четвертий і п'ятий елементи списку: 18 і 11, $18 > 11$, тому міняємо їх місцями й отримаємо такий список:

6	5	13	11	18	10
---	---	----	----	----	----

Беремо п'ятий і шостий елементи списку: 18 і 10, $18 > 10$, тому міняємо їх місцями й отримаємо такий список:

6	5	13	11	10	18
---	---	----	----	----	----

У результаті першого проходу по списку найбільший його елемент стає останнім і займає своє місце в майбутньому відсортованому списку. Тому в подальшому сортуванні попередній до нього елемент порівнювати з ним уже не потрібно.

Далі потрібно повторити цей процес, знову починаючи з початкового елемента списку.

Тоді після другого проходу по списку вже два останніх елементи займуть свої місця в майбутньому відсортованому списку:

5	6	11	10	13	18
---	---	----	----	----	----

І так далі.

Усього потрібно зробити 5 проходів по списку. На останньому проході свої місця займуть одразу два елементи. Після чого отримаємо відсортований за зростанням список.

Створимо проєкт для сортування за зростанням списку із 6 дійсних чисел методом обміну.

```
a = list(map(float, input('Уведіть 6 чисел через пропуск > ').split()))
# Уведення значень елементів списку із 6 дійсних чисел
for i in range(5): # 5 разів повторюємо прохід по списку
    for j in range(5-i): # Порівнюємо кожен пару сусідніх елементів від першого елемента списку до останнього в невідсортованій частині списку
```

```

if a[j] > a[j+1]: # Якщо лівий із двох сусідніх елементів
                  # більший за правий з них
    x = a[j] # Міняємо два сусідніх елементи списку місця-
             # ми, використовуючи допоміжну змінну x
    a[j] = a[j+1]
    a[j+1] = x

```

```
print(a) # виведення впорядкованого списку
```

Метод обміну інколи називають **методом «бульбашки»**. За аналогією з тим, як бульбашки в газованій воді спливають на поверхню, елементи списку «спливають» у його кінці: спочатку найбільший, потім найбільший серед тих, що залишилися, і т. д.



Обговоріть і зробіть висновки

1. Як змінити наведений проєкт, щоб він сортував списки за спаданням?
2. Чи правильно сортуватимуть дані у списку вищенаведені проєкти, якщо список міститиме рівні між собою числа? Відповідь поясніть.
3. На яку одну команду можна замінити три останні команди циклу в наведеному проєкті?
4. В інтернеті можна подивитися візуалізацію методу обміну, наприклад за адресами: <https://cutt.ly/3tSyE2sA> або <https://www.toptal.com/developers/sorting-algorithms>.



Для тих, хто хоче знати більше

Існують й інші алгоритми сортування даних у списках. Їх візуалізацію можна подивитися в інтернеті, наприклад за адресою: <https://imgur.com/gallery/rainbow-sorting-zurNLYf>.



Цікаві факти з історії

Потреба у створенні швидких та ефективних алгоритмів сортування даних вперше виникла на початку XX ст. у США і була зумовлена значними обсягами даних, які потрібно було сортувати – результатами перепису населення.

З появою комп'ютерів на початку 40-х років XX ст. обсяг даних для опрацювання постійно збільшувався. І для зменшення часу роботи програм опрацювання цих даних розробляли ефективніші алгоритми сортування.

Такі алгоритми розробили відомі математики і програмісти: Джон фон Нейман (1903–1957), Чарльз Хоар (нар. 1934), Дональд Шелл (1924–2015), Дональд Кнут (нар. 1938), Едсгер Дейкстра (1930–2002).

Українські науковці також багато років займаються вдосконаленням та застосуванням алгоритмів сортування для оптимізації обчислень. Проблема сортування даних займалися дослідники Інституту кібернетики імені В. М. Глушкова НАН України для організації обчислень на комп'ютерах із розподіленою пам'яттю. Ці дослідження стосувалися як фундаментальних досліджень символічної обробки даних, так і їхнього практичного застосування (у паралельних обчисленнях, розпізнаванні образів, нейронних мережах, хмарних обчисленнях та іншому).

Суттєвий внесок у дослідження цих питань зробили українські вчені Жалдак М. І. (1937–2021) і Триус Ю. В. (нар. 1957).



Жалдак М. І.



Триус Ю. В.

ПОШУК ДАНИХ У СПИСКАХ



Пригадайте

- Для чого в опрацюванні списків використовується операція **in**?
- Для чого в опрацюванні списків використовуються методи **count**, **index**?



Поміркуйте

- Від чого залежить швидкість знаходження потрібних даних?
- Наскільки важливо знайти потрібні дані якнайшвидше? Наведіть приклади.

Задачі пошуку даних у списках дають відповідь на запитання: *Чи існує елемент з указаним значенням у наборі даних? Скільки таких елементів? На яких місцях вони розташовані?*

Зокрема, у списках учнів / учениць школи часто потрібно з'ясувати: *Чи є учень / учениця з певним прізвищем у школі? Скільки учнів / учениць з таким прізвищем? У яких класах є учні / учениці з таким прізвищем?*

Ви вже знаєте, що наявність чи відсутність певних даних у списку можна визначити, використовуючи операцію **in**.

Розглянемо дві задачі на пошук даних у списках, використавши операцію **in**.

Задача 1. Визначити, чи є введене число у списку. Якщо так, то скільки разів воно трапляється у списку і з якими індексами.

```
a = list(map(float, input('Уведіть числа зі списку через пропуск >').split()))
```

```
x = float(input('Уведіть шукане число: '))
```

```
if x in a:
```

```
    print('Шукане число в списку є')
```

```
    n = a.count(x)
```

```
    print('Воно трапляється у списку ', n, ' разів')
```

```
    b = []
```

```
    for i in range(len(a)):
```

```
        if a[i] == x: b.append(i)
```

```
    print(b)
```

```
else: print('Шуканого числа у списку немає')
```

Задача 2. Є список із чисел, упорядкований за неспаданням, і є окреме число. Уставити це окреме число у список, не порушуючи його впорядкованість.

Оскільки список упорядковано за неспаданням, то кожне наступне число в ньому, починаючи з другого, не менше (більше або дорівнює) від попереднього.

Тому можна послідовно порівнювати це окреме число із числами у списку. І як тільки воно стане меншим від числа зі списку, уставляти його у список. Якщо ж список закінчиться, то це означатиме, що це число не менше від усіх чисел списку і його потрібно додати в кінець списку.

Наводимо проєкт для розв'язування наведеної задачі:

```
a = [2, 4, 4, 6, 8, 10]
```

```
x = int(input())
```



```

f = False # Місце у списку для числа поки що не знайшлося
for i in range(len(a)):
    if x < a[i]:
        a.insert(i, x) # Уставляємо число у список
        f = True # Місце у списку для числа знайшлося
        break
if f == False: a.append(x) # Якщо місце у списку для числа так
                           і не знайшлося, додаємо його в кінець
                           списку

print(a)

```



Поміркуйте

- Як змінити вищенаведений проєкт так, щоб у ньому не використовувалася змінна `f`?



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Задача 1. Створити проєкт для введення значень елементів списку і сортування його за незростанням методом вибору.

1. Відкрийте вікно середовища створення проєкту.
2. Уведіть змінений відповідно до умови задачі фрагмент проєкту для впорядкування списку методом вибору, наведений у тексті пункту. Коментарі вводити не потрібно.
3. Збережіть проєкт у вашій папці у файлі з іменем **вправа 3.3.1.py**.
4. Виконайте проєкт кілька разів з різними вхідними даними. Проаналізуйте результат його виконання.
5. Закрийте вікно середовища створення проєктів.

Задача 2. Скласти проєкт для введення значень елементів одновимірної масиви з 10 дійсних чисел, упорядкування цього масиву за незростанням методом обміну і виведення впорядкованого.

1. Відкрийте вікно середовища створення проєктів.
2. Уведіть змінений відповідно до умови задачі фрагмент проєкту для впорядкування списку методом обміну, наведений у тексті пункту. Коментарі вводити не потрібно.
3. Збережіть проєкт у вашій папці у файлі з іменем **вправа 3.3.2.py**.
4. Виконайте проєкт кілька разів з різними вхідними даними. Проаналізуйте результат його виконання.
5. Закрийте вікно середовища створення проєктів.



Найважливіше в цьому пункті

Список `a` називається відсортованим за зростанням (зростаючим), якщо значення кожного його наступного елемента більше за значення попереднього, тобто для всіх i виконується нерівність $a[i+1] > a[i]$.

Список `a` називається відсортованим за спаданням (спадним), якщо значення кожного його наступного елемента менше від значення попереднього, тобто для всіх i виконується нерівність $a[i+1] < a[i]$.

Список `a` називається відсортованим за неспаданням (неспадним), якщо значення кожного його наступного елемента не менше (більше або дорівнює) від значення попереднього, тобто для всіх i виконується нерівність $a[i+1] \geq a[i]$.

Список а називається відсортованим за незростанням (незростаючим), якщо значення кожного його наступного елемента не більше (менше або дорівнює) ніж значення попереднього, тобто для всіх і виконується нерівність $a[i+1] \leq a[i]$.

Для сортування даних у списку можна використати:

- стандартний метод **sort**, який сортує список за зростанням (неспаданням) **sort()** або за спаданням (незростанням) **sort(reverse = True)**; у результаті виконання цього методу список, до якого він застосований, буде відсортовано; початковий список не зберігається; приклади відповідних команд: **a.sort()** і **a.sort(reverse = True)**.
- стандартну функцію **sorted(список)**, яка утворює новий список, відсортувавши список, який зазначено як аргумент цієї функції; якщо аргументом цієї функції є тільки список, то він буде відсортований за зростанням; якщо вказати ще й другий аргумент **sorted(список, reverse = True)**, то список буде відсортовано за спаданням; водночас початковий список зберігається і не змінюється; приклади відповідних команд: **b = sorted(a)** і **b = sorted(a, reverse = True)**.

Алгоритм сортування методом вибору:

1. Вибрати найменший елемент серед елементів поки що невідсортованої частини списку.
2. Обміняти цей вибраний елемент з першим елементом поки що невідсортованої частини списку.

Алгоритм сортування методом обміну:

Братимемо по черзі два сусідніх елементи списку (перший і другий, другий і третій, третій і четвертий і т. д.) і, якщо лівий з них більший за правий, мінятимемо їх місцями.



Дайте відповіді на запитання

1. Які ви знаєте методи сортування списку? Наведіть приклади.
2. З якою метою сортують списки?
3. Яку ви знаєте функцію для сортування списку? Наведіть приклади їх використання.
4. У чому полягає суть алгоритму сортування списку методом вибору?
5. У чому полягає суть алгоритму сортування списку методом обміну?
6. Як визначити, чи є потрібне значення у списку?
7. Як визначити, скільки разів трапляється потрібне значення у списку?



Обговоріть і зробіть висновки

1. Чим відрізняються результати метода **sort** і функції **sorted** упорядкування списку?
2. У яких випадках зручніше використовувати метод **sort**, а в яких – функцію **sorted**?



Виконайте завдання

1. У списку зберігаються оцінки з інформатики групи учнів / учениць. Відсортуйте цей список за спаданням і визначте, скільки учнів / учениць мають найнижчу в цій групі оцінку. Збережіть проект у вашій папці у файлі з іменем **завдання 3.3.1**.
2. У даному відсортованому списку визначте, яке число трапляється в ньому найбільшу кількість разів. Збережіть проект у вашій папці у файлі з іменем **завдання 3.3.2**.



3. Є відсортований список. Визначте, скільки в ньому різних значень. Наприклад, у списку [1, 2, 4, 1, 2] різних значень 3. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.3.3**.

4. У спортивних змаганнях з бігу на 100 м взяли участь 20 учасників. Розташуйте прізвища учасників змагання за зростанням їхнього часу бігу. Використайте два списки – для зберігання прізвищ і для зберігання результатів. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.3.4**.

5. Змініть проєкт упорядкування списку методом обміну, щоб його виконання переривалося, якщо після чергового проходження списку жодного обміну не відбулося. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.3.5**.

6. Змініть проєкт упорядкування списку методом вибору, щоб обмін найменшого значення і першого значення з поки що не впорядкованої частини списку відбувався лише тоді, коли це доцільно робити. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.3.6**.



7. Створіть власну візуалізацію одного з методів упорядкування списку (анімовану презентацію, відеофільм або інше).



Готуємось до вивчення нового матеріалу

1. Пригадайте, де у проєктах ви використовували тексти.
2. Як перевести текстове подання в число і число в його текстове подання? Де ви це використовували?



3.4. РЯДКОВІ ВЕЛИЧИНИ

У цьому пункті йтиметься про:

- ▶ рядкові величини;
- ▶ операції над рядковими величинами;
- ▶ функції та методи для опрацювання рядкових величин.

РЯДКИ. РЯДКОВІ ВЕЛИЧИНИ



Пригадайте

- Що буде виведено у вікно виконання проєкту командою `print('Середня маса дорівнює ', x, ' кг')`?
- Де ще у проєктах мовою програмування **Python**, зокрема у віконних проєктах, ви використовували тексти?
- Якщо `a` – список, то що таке `a[0]`, `a[3]`, `a[-1]`, `a[-2]`?

У проєктах використовуються та опрацьовуються не тільки числа, а й тексти. Ви вже використовували тексти в командах введення та виведення, у віконних проєктах. Це і паролі, і прізвища, і назви товарів тощо.

Тексти у проєктах безпосередньо пов'язані з поняттями **рядок** і **рядкова величина**.

Рядок – це послідовність символів (літер, цифр, розділових знаків та іншого), узята в одинарні чи подвійні лапки. Наприклад, 'м', "км", 'байт', "Я – учениця 9 класу".

Рядок, який не містить символів, називають **порожнім рядком** і позначають `""`.

У мові програмування **Python** у рядках розрізняються малі та великі літери. Тобто рядки 'кг' і 'Кг' різні.

Величина, значенням якої є рядок, називається **величиною рядкового типу**, або **рядковою величиною**. Наприклад, `x = 'кілобайт'`, `s = 'Ми вивчаємо мову програмування Python'`.

Кожен рядок можна розглядати як список, значеннями елементів якого є символи. Наприклад,

якщо `x = 'кілобайт'`, то `x[0] = 'к'`, `x[1] = 'і'`, `x[2] = 'л'` і т. д.

Для рядків, як і для списків, можна використовувати й від'ємні цілі індекси. Так для рядка з іменем `x` символ `x[-1]` – це перший символ з кінця рядка (останній символ), символ `x[-2]` – це другий символ з кінця рядка (передостанній символ) і т. д. Так для `x = 'кілобайт'` `x[-1] = 'т'`, `x[-2] = 'й'` і т. д.

Звертаємо вашу увагу.

- Якщо символами рядка мають бути лапки, одинарні чи подвійні, або апостроф, які можуть системою сприйматися як початок або кінець рядка, то їх потрібно вставляти в рядок разом із символом `'\'`. Наприклад, `x = 'Мое ім'я Дем'ян'`.
- У мові програмування **Python** символи рядка не можна змінити командою присвоювання. Наприклад, для рядка `x` команда `x[0] = 'і'` неприпустима.

ОПЕРАЦІЇ НАД РЯДКАМИ І РЯДКОВИМИ ВЕЛИЧИНАМИ



Пригадайте

- Як об'єднати два списки в один? • Як продублювати список? • Як отримати зріз зі списку? • Що таке *таблиця кодів символів*? Які таблиці кодів символів ви знаєте?

Кілька рядків можна **об'єднувати в один**, використавши операцію `+`. Наприклад, якщо `a = 'Іванна '` і `b = 'Васильченко'`, то їх об'єднанням буде такий рядок: `a + b = 'Іванна Васильченко'`.

Елементи рядка можна **дублювати**, використавши операцію `*`. Наприклад, якщо `a = '9-A'`, то рядок `a*2` буде `'9-A9-A'`, а рядок `a*3` буде `'9-A9-A9-A'`.

Із будь-якого рядка можна виокремити його частину. Ця операція називається **добування зрізу з рядка**. Загальний вигляд команди з використанням цієї операції такий: `b = a[i:j:step]`, де `b` – змінна рядкового типу, значенням якої є рядок, який є частиною рядка `a`; `i` – індекс початку зрізу, `j` – його закінчення, причому символ з номером `j` у зріз уже не входить; `step` – крок, з яким добуваються символи.

Якщо символи добуваються підряд, тобто `step = 1`, то параметр `step` можна не вказувати. Якщо відсутній індекс початку зрізу, то зріз береться від 0-го до (j-1)-го символу; якщо відсутній індекс закінчення зрізу, зріз береться від i-го символу до кінця рядка.

Наприклад, якщо `a = 'інформатика'`, то:

- `b = a[2:6] = 'форм';`
- `b = a[1:8:2] = 'номт';`
- `b = a[4:] = 'рматика';`
- `b = a[:3] = 'інф';`
- `b = a[9:5:-1] = 'кита';`
- `b = a[::-2] = 'іфраиа';`
- `b = a[::-1] = 'акитамрофні';`
- `b = a[-4:-1] = 'тик'.`



Рядки можна порівнювати на збіги операцією `==`. Результат цієї операції дорівнює **True**, якщо рядки повністю збігаються, і **False**, якщо вони не збігаються.

Рядки можна порівнювати операціями `>` та `<`. Порівнюють коди початкових символів рядків у таблиці кодів символів **Unicode**. Якщо код одного із цих символів більший, то й увесь відповідний рядок вважається більшим. Якщо коди початкових символів рівні, то аналогічно порівнюють коди наступних символів. Наприклад, `'марка' < 'маска'`, тому що коди перших двох символів рядків рівні, а код символу `'р'` менше коду символу `'с'`. Якщо під час порівняння двох рядків коди всіх відповідних символів рівні й один з них закінчився, то він вважається менше від другого. Наприклад, `'маска' < 'маскарад'`.

Якщо потрібно з'ясувати, чи є в рядку певний символ або послідовність символів, чи немає, то для цього можна застосувати операцію `in`. Ви вже застосовували цю операцію в циклах з лічильником, а також в операціях над списками. Наприклад, для рядкової величини `a = 'Інформатика'`:

```
'ф' in a = True, 'форм' in a = True, 'п' in a = False, 'та' in a = False.
```

Символи рядка можна переглядати, використовуючи цикл `for`. Наприклад, для визначення, скільки разів до рядка `a` входить символ `'х'`, можна використати такий фрагмент проєкту:

```
n = 0
for k in a:
    if k == 'х': n = n + 1
```

Розглянемо задачу.

Задача 1. Уведіть окремо своє прізвище та ім'я. Об'єднайте їх в один рядок, розділивши їх пропуском. Чи розпочинається введене прізвище з літери `'Б'`? Чи є в прізвищі або імені сполучення літер `'нд'`?

Проект для розв'язування цієї задачі такий:

```
a = input('Уведіть прізвище: ')
c = input('Уведіть ім'я: ')
ip = a + ' ' + c
if 'Б' == a[0]: print('Прізвище розпочинається з літери Б')
else: print('Прізвище не розпочинається з літери Б')
if 'нд' in ip: print('Таке сполучення літер є')
else: print('Такого сполучення літер немає')
```

ВИКОРИСТАННЯ ФУНКЦІЙ І МЕТОДІВ ДЛЯ ОПРАЦЮВАННЯ РЯДКОВИХ ВЕЛИЧИН



Пригадайте

- Що таке *функція* у проєкті мовою програмування **Python**?
- Які функції ви використовували для опрацювання списків? Скільки аргументів може мати функція? Як передаються у функцію значення її аргументів?
- Що таке *результат функції*? Скільки результатів може мати функція? Як передаються (повертаються) значення результатів функції в команду її виклику?
- Які методи вам відомі для опрацювання списків? Для чого вони призначені?

Для опрацювання рядків можна використати **функції**: стандартні (вбудовані) або створити власні (функції користувача). Деякі зі стан-

дартних функцій ви вже використовували. Так, створюючи віконні проекти, ви використовували функції `int` і `float` для перетворення текстового рядкового подання числа безпосередньо в число, `str` для перетворення числа в його рядкове подання.

Наприклад, `int('268') = 268`, `float('-34.86') = -34.86`, `str(450) = '450'`, `str(23.5) = '23.5'`.

Для рядків можна також використовувати функцію `len` для визначення довжини (кількості символів) рядка. Наприклад, якщо `a = 'Інформатика'`, то `len(a) = 11`.

Для опрацювання рядків можна також використовувати методи. У таблиці 3.7 наведено деякі з них, їх призначення та приклади їх використання для рядка `a = 'величина'`.

У результаті виконання цих методів зміни відбуваються не в самому рядку, а завжди створюється новий рядок.

Таблиця 3.7

Деякі методи для опрацювання рядків

Метод	Призначення	Приклад використання	Результат використання
count	Обчислює кількість входження зазначеного підрядка в даний рядок. Можна в дужках додатково через кому вказати початковий та кінцевий індекси, і тоді пошук буде здійснюватися тільки в межах зазначеного підрядка	<code>b = a.count('и')</code> <code>b = a.count('и', 0, 3)</code>	<code>b = 2</code> <code>b = 1</code>
replace	Замінює одну частину рядка на іншу. У дужках потрібно зазначити, який фрагмент рядка замінити, на який фрагмент замінити і скільки разів замінити. Якщо не вказати кількість разів, то заміняться всі входження зазначеного фрагмента	<code>b = a.replace('ина', 'ний')</code>	<code>b = 'величний'</code>
find	Визначає індекс, починаючи з якого зазначений підрядок перший раз входить до даного рядка. Якщо зазначений підрядок до даного рядка не входить, то результат виконання метода дорівнює -1. Додатково в дужках можна через кому зазначити початковий та кінцевий індекси в рядку, між якими потрібно здійснювати пошук	<code>b = a.find('лич')</code> <code>b = a.find('map')</code> <code>b = a.find('и', 4, 6)</code>	<code>b = 2</code> <code>b = -1</code> <code>b = 5</code>



Метод	Призначення	Приклад використання	Результат використання
startswith	Визначає, чи розпочинається даний рядок із зазначеного підрядка. Якщо в дужках зазначити початковий та кінцевий індекси, то пошук здійснюватиметься в межах зазначеного підрядка даного рядка	b = a.startswith('вел')	b = True
		b = a.startswith('лич', 3, 6)	b = True
strip	Видаляє зазначені символи на початку та в кінці рядка. Якщо в дужках символи не зазначені, то видаляє пропуски. Якщо потрібно видалити тільки на початку рядка, то слід скористатися методом lstrip . Якщо потрібно видалити тільки в кінці рядка, то слід скористатися методом rstrip	b = a.strip('ве')	b = 'личина'
split	Розбиває рядок на підрядки, використовуючи зазначений символ як розділювач. Символ-розділювач зникає. У результаті утворюється список підрядків даного рядка. Якщо в дужках розділювач не зазначений, то розбиття за замовчуванням відбувається за пропуском	b = a.split('и')	b = ['вел', 'ч', 'на']
join	Об'єднує рядки, які є значеннями елементів списку, в один рядок, уставляючи між ними вказаний символ-роздільник	a = ['ln', 'forma', 'tyka'] b = ".join(a) b = '+'.join(a)	b = 'lnforma+tyka'
upper	Переводить усі символи рядка у верхній регістр	b = a.upper()	b = 'ВЕЛИЧИНА'
lower	Переводить усі символи рядка в нижній регістр	b = a.lower()	b = 'величина'

Розглянемо приклади розв'язування задач на опрацювання рядків.

Задача 2. Дано рядок, у якому слова розділені одним пропуском. Визначити кількість слів у цьому рядку.

Очевидно, що кількість слів у рядку на одне більше, ніж кількість пропусків. Тому визначимо кількість пропусків у цьому рядку, використовуючи метод **count**, і збільшимо цю кількість на 1.

```
s = input('Введіть рядок: ')
n = s.count(' ') + 1
print(n)
```

Задача 3. Дано рядок, у якому слова розділені пропусками, одним або більше. Визначити кількість слів у цьому рядку.

Розіб'ємо цей рядок на підрядки за розділовим символом «пропуск». Оскільки підряд можуть розташовуватися більше ніж один пропуск, то у списку, що утворився, елементами можуть бути порожні рядки. Видалимо їх зі списку й підрахуємо кількість елементів в останньому списку. Це й буде кількість слів у даному рядку.

```
s = input('Уведіть рядок: ')
a = s.split(' ')
while '' in a:
    a.remove('')
print(len(a))
```



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Задача 1. Номер телефона абонента мобільного зв'язку розпочинається із символу +, наступні 2 цифри є кодом країни, наступні 3 цифри – код оператора мобільного зв'язку, усі інші цифри – це безпосередньо номер телефону абонента. Є певний номер телефону. Створіть проєкт для визначення, чи є цей телефонний номер з України (+38), чи є оператором цього номера компанія «Lifecell» (063, 073, 093), чи є сума трьох останніх цифр номера парним числом.

Для визначення, чи є даний телефонний номер з України, використаємо метод `startswith`. Для визначення, чи є оператором цього номера компанія «Lifecell», побудуємо із цього номера перші 6 символів і порівняємо цей зріз із кодами України та оператора «Lifecell». Для визначення, чи є сума трьох останніх цифр парним числом, відокремимо три останні цифри, переведемо ці цифри-символи в числа і знайдемо суму цих чисел.

1. Відкрийте вікно середовища створення проєкту.
2. Уведіть нижченаведений текст проєкту:


```
a = input('Уведіть телефонний номер: ')
b = a.startswith('+38')
if b: print('Телефон з України')
else: print('Телефон не з України')
```
3. Збережіть проєкт у вашій папці у файлі з іменем **вправа 3.4.1.py**.
4. Запустіть проєкт на виконання.
5. Уведіть номер «+380564221423».
6. Проаналізуйте отриманий результат.
7. Запустіть проєкт на виконання.
8. Уведіть номер «+410673221525».
9. Проаналізуйте отриманий результат.
10. Додайте до тексту проєкту такий текст:


```
c = a[6:]
if c == '+38063' or c == '+38073' or c == '+38093': print('Lifecell')
else: print('non Lifecell')
```
11. Збережіть оновлений проєкт у тій самій папці під тим самим іменем.



12. Запустіть проєкт на виконання.
13. Уведіть телефон «+380631276633».
14. Проаналізуйте отриманий результат.
15. Запустіть проєкт на виконання.
16. Уведіть телефон «+380651276633».
17. Проаналізуйте отриманий результат.
18. Додайте до тексту проєкту такий текст:


```
s = int(a[-1]) + int(a[-2]) + int(a[-3])
if s % 2 == 0: print('Сума трьох останніх цифр парна')
else: print('Сума трьох останніх цифр непарна')
```
19. Збережіть оновлений проєкт у тій самій папці під тим самим іменем.
20. Запустіть проєкт на виконання.
21. Уведіть телефон «+380631276813».
22. Проаналізуйте отриманий результат.
23. Запустіть проєкт на виконання.
24. Уведіть телефон «+380651276533».
25. Проаналізуйте отриманий результат.
26. Закрийте всі відкриті вікна.



Найважливіше в цьому пункті

Рядок – це послідовність символів (літер, цифр, розділових знаків та ін.), узятя в одинарні чи подвійні лапки. Наприклад, 'м', "км", 'байт', "Я – учениця 9 класу".

Рядок, який не містить символів, називають **порожнім рядком** і позначають "".

У мові програмування **Python** у рядках розрізняються малі та великі літери. Тобто рядки 'кг' і 'Kg' різні.

Величина, значенням якої є рядок, називається **величиною рядкового типу**, або **рядковою величиною**. Наприклад, **x** = 'кілобайт', **c** = 'Ми вивчаємо мову програмування **Python**'.

Кожен рядок можна розглядати як список, значеннями елементів якого є символи. Наприклад, якщо **x** = 'кілобайт', то **x[0]** = 'к', **x[1]** = 'і', **x[2]** = 'л' і т. д.

Для рядків, як і для списків, можна використовувати й від'ємні цілі індекси. Так для рядка з іменем **x** символ **x[-1]** – це перший символ з кінця рядка (останній символ), символ **x[-2]** – це другий символ з кінця рядка (передостанній символ) і т. д. Так для **x** = 'кілобайт' **x[-1]** = 'т', **x[-2]** = 'й' і т. д.

Кілька рядків можна **об'єднувати в один**, використавши операцію **+**. Наприклад, якщо **a** = 'Іванна' і **b** = 'Васильченко', то їх об'єднанням буде такий рядок: **a + b** = 'Іванна Васильченко'.

Елементи рядка можна **дублювати**, використавши операцію *****. Наприклад, якщо **a** = '9-A', то рядок **a*2** буде '9-A9-A', а рядок **a*3** буде '9-A9-A9-A'.

Із будь-якого рядка можна виокремити його частину. Ця операція називається **добуванням зрізу з рядка**. Загальний вигляд команди з використанням цієї операції такий: **b = a[i:j:step]**, де **b** – змінна рядкового типу, значенням якої є рядок, який є частиною рядка **a**; **i** – індекс початку зрізу, **j** – його закінчення, причому символ з номером **j** у зріз уже не входить; **step** – крок, з яким добуваються символи.

Якщо символи добуваються підряд, тобто **step = 1**, то параметр **step** можна не зазначати. Якщо відсутній індекс початку зрізу, то зріз береться від **0**-го до **(j-1)**-го символу; якщо відсутній індекс закінчення зрізу, то зріз береться від **i**-го символу до кінця рядка.

Рядки можна порівнювати на збіги операцією **==**. Результат цієї операції дорівнює **True**, якщо рядки повністю збіглися, і **False**, якщо вони не збіглися.

Рядки можна порівнювати операціями **>** та **<**. Порівнюють коди початкових символів рядків у таблиці кодів символів **Unicode**. Якщо код одного із цих символів більший, то й увесь відповідний рядок вважається більшим. Якщо коди початко-

вих символів рівні, то аналогічно порівнюють коди наступних символів. Наприклад, 'марка' < 'маска', тому що коди перших двох символів рядків рівні, а код символу 'р' менший від коду символу 'с'. Якщо під час порівняння двох рядків коди всіх відповідних символів рівні і один з них закінчився, то він вважається меншим від другого. Наприклад, 'маска' < 'маскарад'.

Якщо потрібно з'ясувати, чи є в рядку певний символ або послідовність символів, чи немає, то для цього можна використати операцію **in**.

Для опрацювання рядків можна використовувати **функції**. Можна використати стандартні (вбудовані) функції, а можна створювати власні функції (функції користувача). Деякі зі стандартних функцій ви вже використовували. Так, створюючи віконні проекти, ви використовували функції **int** і **float** для перетворення текстового рядкового подання числа безпосередньо в число, **str** для перетворення числа в його рядкове подання.

Для рядків можна також використовувати функцію **len** для визначення довжини (кількості символів) рядка. Наприклад, якщо **a** = 'Інформатика', то **len (a) = 11**.

Методи опрацювання рядкових величин:

- **count** – визначає кількість входження зазначеного підрядка в даний рядок;
- **replace** – замінює одну частину рядка на іншу;
- **find** – визначає індекс, починаючи з якого зазначений підрядок перший раз входить до даного рядка;
- **startswith** – визначає, чи розпочинається даний рядок із зазначеного підрядка;
- **strip** – видаляє зазначені символи на початку та в кінці рядка;
- **split** – розбиває рядок на підрядки, використовуючи зазначений символ як роздільник;
- **join** – об'єднує рядки, які є значеннями елементів списку, в один рядок, уставляючи між ними зазначений символ-роздільник.



Дайте відповіді на запитання

1. Що таке *рядок*? Наведіть приклади.
2. Яка величина вважається величиною рядкового типу?
3. Які ви знаєте операції, що можна виконувати над рядками? Наведіть приклади їх застосування.
4. Які ви знаєте функції, що можна виконувати над рядками? Наведіть приклади їх застосування.
5. Які ви знаєте методи, що можна виконувати над рядками? Наведіть приклади їх застосування.



Обговоріть і зробіть висновки

1. Чим опрацювання рядків схоже і чим відрізняється від опрацювання списків?
2. Чим рядок цифр відрізняється від числа?



Виконайте завдання

1. Нехай **a** = 'Ми – учні 9 класу'. Запишіть у зошит результати виконання команд:

- а) **b = a[2:6];**
- б) **b = a[1:8:2];**
- в) **b = a[4:];**
- г) **b = a[:3];**
- д) **b = a[9:5:-1];**
- е) **b = a[::2];**
- ж) **b = a[:-1];**
- з) **b = a[-4:-1].**



2. Створіть проєкт, у якому вводиться рядок і в ньому всі цифри 4 замінюються на слово «чотири». Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.2.**

3. Створіть проєкт, у якому вводиться натуральне чотирицифрове число і в результаті отримується його сума цифр. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.3.**

4. Створіть проєкт, у якому вводиться натуральне число і в результаті отримується перевернуте число. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.4.**

5. Створіть проєкт, у якому вводиться натуральне число, більше за 9, і в результаті отримується запис числа, у якому обмінені місцями перша і остання цифри. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.5.**

6. Створіть проєкт, у якому вводяться прізвище, ім'я та по батькові певної людини і виводяться прізвище та ініціали. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.6.**

7. Дано рядок, у якому слова розділені або одним пропуском, або розділовим знаком (кома, крапка, знак оклику) з одним пропуском після них. Створіть проєкт, у якому визначається кількість слів у цьому рядку. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.7.**

8. Дано рядок, у якому слова розділені або одним пропуском, або розділовим знаком (кома, крапка, знак оклику) з одним пропуском після них. Створіть проєкт, у якому всі ці слова виводяться по одному в кожному рядку без розділових знаків. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.8.**

9. Дано рядок, у якому слова розділені одним або кількома пропусками. Створіть проєкт, у якому видаляються всі зайві пропуски. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.9.**

10. Дано слово. Створіть проєкт, у якому визначається, чи є це слово паліндромом. Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.10.**

11. Дано рядок, у якому слова розділені пропусками. Створіть проєкт для визначення, чи є цей рядок паліндромом, не враховуючи пропусків. Прикладом такого паліндрома є «я несу гусеня». Збережіть проєкт у вашій папці у файлі з іменем **завдання 3.4.11.**



Практична робота № 5.

«Опрацювання рядків»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Задача 1. Дано рядок, у якому слова відокремлені одне від іншого одним пропуском. Створіть проєкт для визначення найбільшої довжини слова в цьому рядку. Збережіть проєкт у вашій папці у файлі з іменем **практична 5-1.**

Задача 2. Дано рядок, у якому слова відокремлені одне від іншого пропусками. Створіть проєкт для визначення найбільшої кількості пропусків, що розташовані поспіль. Збережіть проєкт у вашій папці у файлі з іменем **практична 5-2.**



Готуємось до вивчення нового матеріалу

1. Чим груповий проєкт відрізняється від індивідуального?
2. Як зробити так, щоб результати виконання одного проєкту можна було б у подальшому використати в інших проєктах?



3.5. МОДУЛІ. ВИКОРИСТАННЯ МОДУЛІВ У ПРОЄКТАХ

У цьому пункті йтиметься про:

- ▶ модулі та їх використання у проєктах;
- ▶ модулі в мові програмування **Python**;
- ▶ проєкти мовою програмування **Python**, створені з використанням модулів.



МОДУЛІ ТА ЇХ ВИКОРИСТАННЯ У ПРОЄКТАХ

Поміркуйте

- Як ви організовували роботу групи, виконуючи груповий проєкт? ● Які переваги і недоліки, на вашу думку, є під час групової роботи над проєктом?

Сучасні комп'ютерні проєкти і системи зазвичай великі та складні. Як правило, їх створює не одна людина, а група людей, інколи досить численна. Наприклад, вебсайт або вебпортал фірми чи мережі фірм, застосунок «Дія» та інші.

Для спрощення роботи над такими проєктами, їх створення, налагоджування і тестування роботу розподіляють між різними працівниками чи проєктними групами. Кожен створює свою частину проєкту, налагоджує її, тестує, а потім ці частини об'єднують в один-єдиний проєкт.

Частини, з яких складається комп'ютерний проєкт, називають **модулями**.

Так, наприклад, великий комп'ютерний проєкт може бути розділений на такі модулі:

- модуль введення значень вхідних даних;
- модуль обчислень;
- модуль візуалізації значень кінцевих результатів.

Кожен із цих модулів створюється і тестується окремо, після чого їх об'єднують у єдиний проєкт.

Основні переваги використання модулів під час створення проєктів:

- **Покращена організація коду:** використання модулів структурує програмний код, робить його більш читабельним і зрозумілим.
- **Повторне використання коду:** модулі, які добре спроектовані, можуть бути повторно використані в різних частинах одного проєкту або навіть в інших проєктах, що економить час і зусилля.
- **Спрощення створення, налагодження і тестування:** окремі модулі легше створювати, легше налагоджувати і тестувати, оскільки вони мають обмежену функціональність і менше залежностей усередині кожного модуля.
- **Полегшення командної розробки:** під час розділення проєкту на модулі розробники працюють над різними частинами проєкту одночасно, не заважаючи один одному.
- **Полегшення внесення змін до проєкту у процесі створення:** зміни в одному модулі менше впливають на інші частини проєкту, що полегшує оновлення програмного коду.



- **Зменшення обсягу вихідного коду:** модульне програмування може зменшити обсяг вихідного коду багатьох частин проекту, що покращує їхню читабельність і продуктивність.
- **Ізоляція помилок:** якщо виникає помилка в одному модулі, вона зазвичай не впливає на інші частини проекту.

Кожен модуль проекту зберігається в окремому файлі. Під час об'єднання модулів у єдиний проект усі ці файли не копіюються в єдиний файл, а спеціальним чином підключаються до єдиного проекту.

МОДУЛІ В МОВІ ПРОГРАМУВАННЯ PYTHON



Пригадайте

• Що таке *функція* в мові програмування **Python**? • Для чого у проектах використовуються функції? • Як ви під час створення віконних проектів імпортували команди модуля **tkinter**?

У мові програмування **Python** модулі – це файли, які містять програмний код і зберігаються у файлах з розширенням імені **py**. Модулі можуть містити функції, змінні та інші структури. Для використання коду з модуля його слід імпортувати в інший файл, використовуючи ключове слово **import** і зазначивши ім'я файлу без розширення імені.

Наприклад, команда **import math** імпортує модуль **math**, який містить математичні функції. Після цього у проекті можна використовувати функції з цього модуля, зазначивши ім'я модуля та ім'я функції. Наприклад, можна використати функцію знаходження арифметичного квадратного кореня **sqrt** (англ. *square root* – квадратний корінь): **x = math.sqrt(25)**. Після виконання цієї команди значення змінної **x** дорівнюватиме 5.

Створюючи віконні проекти, ви командою **from tkinter import*** підключали всі функції, константи і методи модуля **tkinter** для роботи з вікном та іншими віконними об'єктами.

Аналогічно підключаються до проекту інші модулі, зокрема створені розробниками проекту.

Розглянемо приклад створення модуля та використання його у проекті.

Задача 1. Створити проект, у якому ввести список із натуральних чисел та обчислити суму цифр кожного із цих чисел. Обчислення суми цифр оформити як функцію в окремому модулі.

Створимо як модуль файл з іменем, наприклад **mymodul1.py**, і включимо до нього функцію **sumnum** для обчислення суми цифр натурального числа. Створимо основний файл проекту з іменем, наприклад **my1.py**, і використаємо в ньому функцію **sumnum** з модуля для обчислення суми цифр кожного натурального числа зі списку.

Ми, звичайно, могли б включити функцію для обчислення суми цифр натурального числа до основного файлу проекту. Але тоді її доведеться включати до файлів усіх проектів, де потрібно буде обчислювати суму цифр натурального числа. Якщо ж ця функція розміщена в окремому модулі, то її можна підключити до будь-якого проекту лише однією командою.

Файл-модуль mymodul1.py	Основний файл проєкту my1.py
<pre>def sumnum(x): # Передаємо у функцію число y = str(x) # Переводимо число в його рядкове подання s = 0 # Початкове значення суми цифр for t in y: # Переглядаємо по черзі ко- жен символ (цифру) рядка s = s + int(t) # Переводимо цифру в її числове подання і додаємо до суми цифр return s # Повертаємо суму цифр у команду виклику цієї функції</pre>	<pre>import mymodul1 # Підключаємо функ- ції вказаного моду- ля a = [126, 54367, 9845, 712] # Уводимо список натуральних чисел b = [] # Порожній список для сум цифр чисел for x in a: # Для чисел зі списку a suma = mymodul1.sumnum(x) # Викликаємо функцію з модуля для обчислення суми цифр b.append(suma) # Додаємо суму цифр чергового числа до списку print('Сума цифр числа: ', *b) # Виво- димо суми цифр числа.</pre>

Якщо модуль містить кілька функцій, то в основну частину проєкту можна підключити не всі з них, а тільки потрібні в цьому проєкті. Для цього потрібно до основної частини проєкту включити команду: **from <ім'я модуля> import <ім'я функції>** або кількох функцій через кому. У такому разі під час виклику цих функцій не потрібно вказувати ім'я модуля.



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Задача 1. Створіть модуль, до якого включіть функцію, яка створює список із перших n чисел Фібоначчі. Використайте цю функцію в основній частині проєкту, у якій обчисліть суму перших n чисел Фібоначчі.

Числами Фібоначчі називають послідовність чисел, перші два з яких дорівнюють 1, а кожне наступне, починаючи з третього, дорівнює сумі двох попередніх: 1 1 2 3 5 8 13 21 34 55 89 ...

Ця послідовність чисел названа на честь італійського вченого Леонардо Пізанського на прізвище Фібоначчі (1170–1250), який у своїх математичних дослідженнях вивчав її властивості.

1. Відкрийте вікно середовища створення проєкту.
2. Уведіть нижченаведений текст модуля:

```
def fb(n): # Передаємо у функцію кількість чисел Фібоначчі
    a = [1,1] # Задаємо початковий список із перших двох чисел
              Фібоначчі
    for i in range(2, n):
        x = a[i-2]+a[i-1] # Обчислюємо наступне число Фібоначчі
                          як суму двох попередніх
    a.append(x) # Додаємо це число до списку чисел Фібоначчі
```



Леонардо
Пізанський



```
return a # Повертаємо в команду виклику цієї функції список
із перших n чисел Фібоначчі
```

3. Збережіть цей модуль у вашій папці у файлі з іменем **mymodul2.py**.
4. Відкрийте нове вікно для створення основної частини проєкту.
5. Уведіть у це вікно нижченаведений текст:

```
import mymodul2 # Підключаємо створений модуль
n = int(input('Уведіть кількість чисел Фібоначчі: ')) # Уводимо
кількість чисел Фібоначчі
b = mymodul2.fb(n) # Викликаємо функцію, яка міститься в мо-
дулі та створює список із перших n чисел
Фібоначчі
if n == 1: # Окремо розглядаємо випадок, коли кількість чисел
Фібоначчі 1
    s = 1 # Сума одного числа Фібоначчі – це саме перше число
Фібоначчі
    print(b[0]) # Виводимо перше число Фібоначчі
else: # Чисел Фібоначчі більше за 1
    print(*b) # Виводимо список перших чисел n Фібоначчі
    s = sum(b) # Сума перших n чисел Фібоначчі зі списку b
print('Сума перших ', n, ' чисел Фібоначчі: ', s) # Виводимо суму
перших n чисел Фібоначчі
```

6. Збережіть цю основну частину проєкту у вашій папці у файлі з іменем **вправа 3.5.1.py**.
7. Запустіть файл **вправа 3.5.1.py** на виконання.
8. Уведіть кількість чисел Фібоначчі – 1.
9. Проаналізуйте результат виконання проєкту.
10. Запустіть ще раз файл **вправа 3.5.1.py** на виконання.
11. Уведіть кількість чисел Фібоначчі – 2.
12. Проаналізуйте результат виконання проєкту.
13. Запустіть ще раз файл **вправа 3.5.1.py** на виконання.
14. Уведіть кількість чисел Фібоначчі – 4.
15. Проаналізуйте результат виконання проєкту.
16. Запустіть ще раз файл **вправа 3.5.1.py** на виконання.
17. Уведіть свою кількість чисел Фібоначчі.
18. Проаналізуйте результат виконання проєкту.
19. Закрийте вікна проєкту.



Найважливіше в цьому пункті

Частини, з яких складається комп'ютерний проєкт, називають **модулями**. Кожен модуль проєкту зберігається в окремому файлі. Під час об'єднання модулів у єдиний проєкт умісти цих файлів не копіюються в єдиний файл, а спеціальним чином підключаються до єдиного проєкту.

Основні переваги використання модулів під час створення проєктів:

- Покращена організація коду.
- Повторне використання коду.
- Спрощення створення, налагодження і тестування.
- Полегшення командної розробки.
- Полегшення внесення змін.
- Зменшення обсягу вихідного коду.
- Ізоляція помилок.

У мові програмування **Python** модулі – це файли, які містять програмний код і зберігаються у файлах з розширенням імені **py**. Модулі можуть містити функції, змінні та інші структури. Для використання коду з модуля його необхідно імпортувати в інший файл, використовуючи ключове слово **import** і зазначивши ім'я файлу без розширення імені.

Під час виклику функції з модуля вказується ім'я модуля і через крапку ім'я функції.



Дайте відповіді на запитання

1. Що таке *модулі* у проєкті? Навіщо їх використовують?
2. Які переваги використання модулів у комп'ютерних проєктах?
3. Як створити і використати модулі у проєкті мовою програмування **Python**?
4. Скільки функцій може містити модуль і чи всі з них потрібно використовувати у проєкті?



Обговоріть і зробіть висновки

1. Як краще: створити один модуль і включити до нього багато різноманітних функцій чи для кожної функції створювати свій окремий модуль? Відповідь обґрунтуйте.
2. Чи не доцільніше було б просто включити функцію з модуля в основний файл проєкту? Відповідь обґрунтуйте.



Виконайте завдання

1. Створіть проєкт, у якому модуль містить функцію, яка відокремлює останню цифру числа. Збережіть модуль у вашій папці у файлі з іменем **modul1.py**. Використайте цей модуль в основній частині проєкту для визначення, чи ділиться введене натуральне число на 5. Збережіть основний файл проєкту у вашій папці у файлі з іменем **завдання 3.5.1.py**.

2. Створіть проєкт, у якому модуль містить функцію, яка відокремлює дві останні цифри числа. Збережіть модуль у вашій папці у файлі з іменем **modul2.py**. Використайте цей модуль в основній частині проєкту для визначення, чи ділиться введене натуральне число на 4. Збережіть основний файл проєкту у вашій папці у файлі з іменем **завдання 3.5.2.py**.

3. Створіть проєкт, у якому модуль містить функцію, яка відокремлює першу літеру слова. Збережіть модуль у вашій папці у файлі з іменем **modul3.py**. Використайте цей модуль в основній частині проєкту для визначення, скільки слів у списку розпочинаються з літери «а». Збережіть основний файл проєкту у вашій папці у файлі з іменем **завдання 3.5.3.py**.

4. Створіть проєкт, у якому модуль містить функцію, яка визначає, чи є певне слово паліндромом. Збережіть модуль у вашій папці у файлі з іменем **modul4.py**. Використайте цей модуль в основній частині проєкту для визначення, скільки слів у рядку є паліндромами. Збережіть основний файл проєкту у вашій папці у файлі з іменем **завдання 3.5.4.py**.

5. Створіть проєкт, у якому модуль містить три функції: чи існує трикутник із заданими довжинами сторін, обчислює периметр трикутника за трьома його сторонами, обчислює площу трикутника за трьома його сторонами. Збережіть модуль у вашій папці у файлі з іменем



modul5.py. Використайте цей модуль в основній частині проекту для визначення периметра і площі трикутника, якщо трикутник з уведеними довжинами сторін існує. Збережіть основний файл проекту у вашій папці у файлі з іменем **завдання 3.5.5.py.**

 **6.** Оформте метод вибору сортування списку як модуль. Збережіть модуль у вашій папці у файлі з іменем **sortselection.py.** Використайте його для створення проекту для розв'язування задачі, умову якої придумайте самі. Збережіть основний файл проекту у вашій папці у файлі з іменем **завдання 3.5.6.py.**

 **7.** Оформте метод обміну сортування списку як модуль. Збережіть модуль у вашій папці у файлі з іменем **sortbubble.py.** Використайте його для створення проекту для розв'язування задачі, умову якої придумайте самі. Збережіть основний файл проекту у вашій папці у файлі з іменем **завдання 3.5.7.py.**



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу «Алгоритми та програми» (початковий, середній, достатній, високий рівень).

- Я знаю, що таке одновимірний масив і для чого його використовують.
- Я знаю, що таке елемент одновимірного масиву, його номер (індекс), як записується його ім'я.
- Я знаю, що таке список, для чого і як його використовують.
- Я вмію використовувати різні способи введення та виведення списку.
- Я знаю операції, які можна виконувати зі списком, і вмію їх використовувати.
- Я знаю функції та методи для опрацювання списків і вмію їх використовувати.
- Я вмію створювати проект, у якому створюються, опрацьовуються і виводяться списки.
- Я вмію створювати проект опрацювання списку, у якому використовуються функції.
- Я знаю, що таке рядок, для чого і як його використовують.
- Я знаю, що таке рядкова величина та її значення.
- Я знаю операції, які можна виконувати зі значенням рядкової величини, та вмію їх використовувати.
- Я знаю функції та методи для опрацювання рядкових величин і вмію їх використовувати.
- Я вмію створювати проект, у якому створюються, опрацьовуються і виводяться значення рядкових величин.
- Я вмію створювати проект опрацювання значень рядкових величин, у якому використовуються функції.
- Я знаю, що таке модуль у проекті, для чого створюються і як використовуються модулі.
- Я вмію створювати проект із використанням модулів.
- Я розумію, як можна організувати групове виконання проекту, розділивши його на окремі модулі.



Готуємось до вивчення нового матеріалу

1. Які види графіки ви знаєте?
2. Чим відрізняються графіки різних видів?
3. У яких випадках доцільно використовувати графіку саме такого виду?



ТРИВИМІРНА ГРАФІКА

У цьому розділі ви здобудете нові, а також поглибите і розширите свої знання й удосконалисте навички з таких тем:

- ▶ поняття й використання тривимірної графіки;
- ▶ класифікація програм тривимірної графіки;
- ▶ основні принципи тривимірного моделювання;
- ▶ створення 3D-моделей з використанням тривимірних примітивів;
- ▶ редагування форми й вигляду тривимірних об'єктів;
- ▶ створення тривимірної анімації;
- ▶ друкування 3D-моделей.

У підручнику цей розділ викладено з використанням програми **Blender**.

Щоб вивчати цей розділ за допомогою онлайн-редактора 3D-графіки **Tinkercad**, перейдіть за посиланням <https://cutt.ly/UtcLG1a9> або QR-кодом.



4.1. ТРИВИМІРНА ГРАФІКА. ПРИНЦИПИ ТРИВИМІРНОГО МОДЕЛЮВАННЯ

У цьому пункті йтиметься про:

- ▶ основні поняття тривимірної графіки;
- ▶ сфери застосування тривимірного моделювання;
- ▶ класифікацію програм тривимірної графіки;
- ▶ основні принципи тривимірного моделювання.

ПОНЯТТЯ Й ВИКОРИСТАННЯ ТРИВИМІРНОЇ ГРАФІКИ



Пригадайте

- Що таке *модель об'єкта*? З якою метою створюють модель об'єкта? Які ви знаєте види моделей?
- Де ви бачили або використовували тривимірні графічні об'єкти?

У попередніх класах ви працювали з растровими та векторними графічними зображеннями. Деякі з цих зображень були моделями об'єктів реального світу, наприклад будинку, літака тощо. Але вони були плоскі – двовимірні, на них людина дивиться з певної точки простору. Побачити на плоскому зображенні інший бік цих об'єктів неможливо.

Щоб можна було розглядати на екрані комп'ютера деталі об'єкта з різних ракурсів, потрібно побудувати його просторову модель, яку називають **тривимірною**, або **3D-моделлю** (англ. *3 Dimensions* – три виміри). Для створення та опрацювання просторових моделей об'єктів використовують засоби **тривимірної графіки**.

Тривимірна графіка, або **3D-графіка**, – це розділ комп'ютерної графіки, у якому вивчають методи та засоби створення графічних моделей просторових об'єктів з використанням комп'ютерних програм.

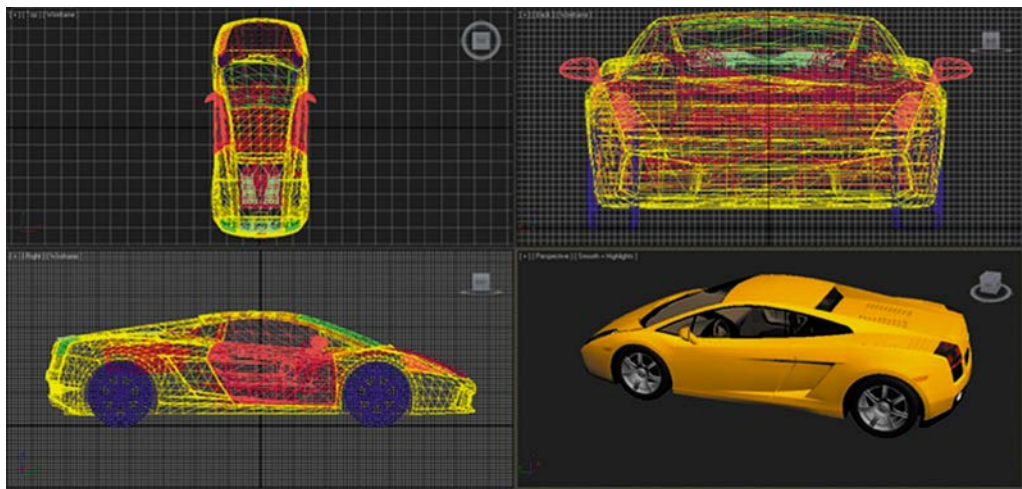
На основі 3D-моделей можуть бути створені плоскі зображення тривимірних об'єктів, які називають **3D-зображеннями**. Процес їх створення називається **рендерингом** (англ. *rendering* – візуалізація, подання). Особливостями 3D-зображень є таке подання об'єкта при погляді крізь об'єкти камери, у якому забезпечено просторову перспективу, тіні, відблиски та інші світлові ефекти, відтворено особливості поверхонь (шорсткість, нерівність, пухнастість тощо), як у реальних об'єктах навколишнього світу.

Також на основі тривимірних моделей можна отримати **3D-анімацію**, у якій моделюється рух об'єктів або їхніх частин для надання наочності та природності відтворюваним процесам.

Тривимірні моделі, створені у програмах 3D-графіки, також можуть бути роздруковані на 3D-принтерах для отримання твердої копії матеріального об'єкта.

Тривимірне моделювання широко використовують у різних сферах людської діяльності. Назвемо деякі з них.

- **Інженерне проєктування.** На виробництві, перш ніж приступати до виготовлення матеріальних виробів, створюють їхні тривимірні моделі. На основі цих 3D-моделей можна здійснювати оцінку зовнішнього вигляду та габаритів об'єктів, провести комп'ютерне дослідження можливостей поєднання декількох об'єктів або міцності конструкцій, а надалі використовувати їх для виготовлення прототипів об'єкта (мал. 4.1).



Мал. 4.1. 3D-модель автомобіля

- **Архітектура.** Засоби 3D-графіки можуть бути використані для створення моделі майбутньої будівлі в ландшафті, на якому вона буде розташована. 3D-зображення майбутніх будівель використовують для узгодження їхнього вигляду та конструкції із замовником, перевірки гармонійності розміщення поруч із навколишніми об'єктами тощо (мал. 4.2).
- **Освіта і наука.** Для демонстрації об'єктів, доступ до яких обмежений або неможливий, створюють тривимірні моделі цих об'єктів. Це можуть бути моделі органів людського тіла, рослинних або



Мал. 4.2. Тривимірне зображення будинку

тваринних клітин, складників комп'ютера (мал. 4.3) або інших пристроїв тощо.

Створені 3D-моделі використовують для демонстрації об'єктів вивчення, їх дослідження та проведення над ними комп'ютерних експериментів.



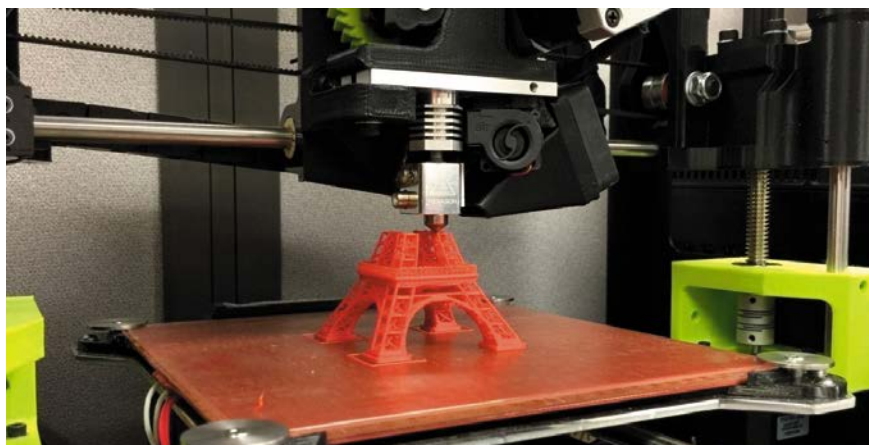
Мал. 4.3. 3D-модель процесора в комп'ютерній програмі для ознайомлення з будовою комп'ютера

- **Кінематографія, мультиплікація, комп'ютерні ігри.** Тривимірні образи персонажів, навіть таких, яких не існує в реальному світі (мал. 4.4), та об'єктів з їх оточення створюють з використанням засобів тривимірної графіки. Комп'ютерну просторову модель легко видозмінювати, а тому може бути швидко створена комп'ютерна анімація, що використовується в цих галузях.



Мал. 4.4. Тривимірні образи мультиплікаційних персонажів

- **3D-друк.** Застосовується в різних галузях діяльності людини для друкування на 3D-принтері матеріального об'єкта на основі попередньо розробленої комп'ютерної 3D-моделі цього об'єкта (мал. 4.5). Використовується для створення прототипів об'єктів промислового виробництва, виготовлення сувенірної продукції, зубного протезування тощо.



Мал. 4.5. Друк на 3D-принтері моделі Ейфелевої вежі

КЛАСИФІКАЦІЯ ПРОГРАМ ТРИВИМІРНОЇ ГРАФІКИ



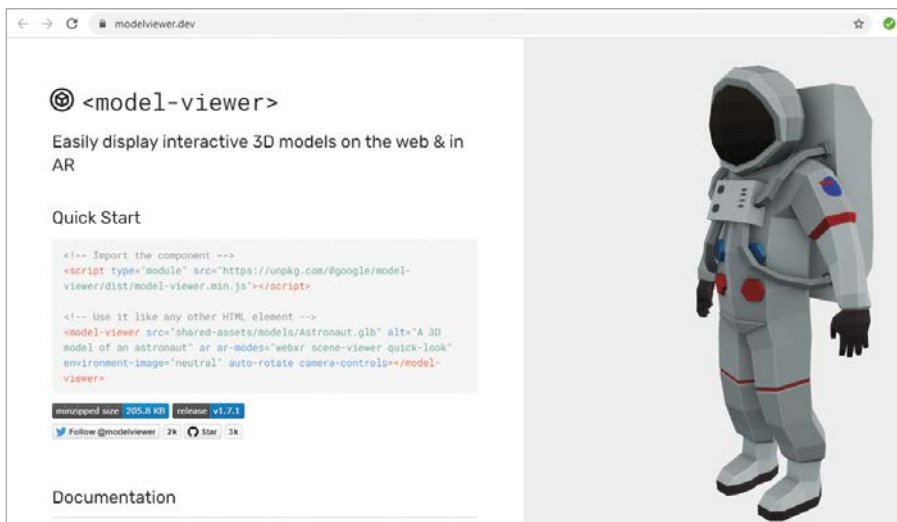
Пригадайте

- Яке призначення програм-переглядачів під час опрацювання мультимедійних даних? ● Яке призначення програм-редакторів під час опрацювання мультимедійних даних?

Програми тривимірної графіки можна класифікувати за їхнім призначенням:

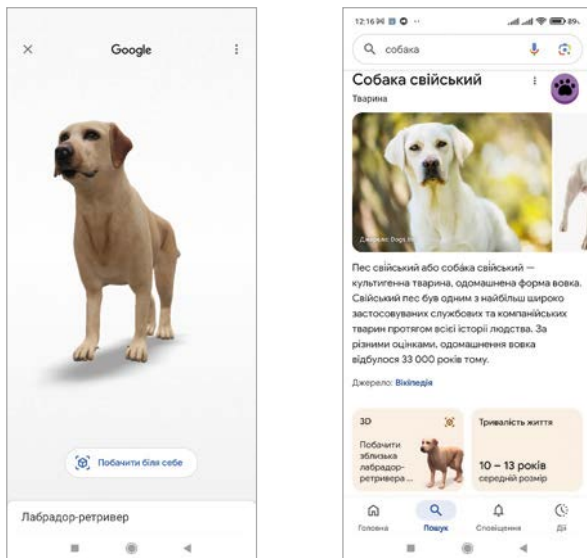
- **для перегляду 3D-моделей.** Особливістю роботи цих програм є можливість повороту тривимірних моделей та їх масштабування. А якщо 3D-модель анімована, то у програмі перегляду може відтворюватися відповідна анімація. Перегляд файлів 3D-моделей можливий, наприклад, у програмах **Word** і **PowerPoint** із пакета

Microsoft Office, у вікні браузера в окремих вебдодатках, наприклад **Model-Viewer** (<https://modelviewer.dev>) (мал. 4.6), **Online 3D Viewer** (<https://3dviewer.net/>), та інших переглядачах.



Мал. 4.6. 3D-модель астронавта у вікні браузера

Великі колекції 3D-моделей можна переглянути і за потреби завантажити для власного використання, наприклад на інтернет-ресурсах **Thingiverse** (<https://www.thingiverse.com>), **Sketchfab** (<https://sketchfab.com>), **Mixamo** (<https://www.mixamo.com>), **TurboSquid** (<https://www.turbosquid.com>), **Open3Dmodel** (<https://open3dmodel.com>) та інших. На смартфонах у додатку **Google** можна переглянути 3D-моделі багатьох об'єктів, опис яких подано у статтях енциклопедії **Вікіпедія** (мал. 4.7).



Мал. 4.7. 3D-модель собаки в застосунку **Google** на смартфоні

- для створення 3D-зображень – 3D-редактори. У 3D-редакторах можна створювати і редагувати графічні просторові моделі об'єктів, зберігати їх у форматі опису моделі та як 3D-зображення. 3D-редакторами є, наприклад, програми **Blender**, **3DStudio MAX**, вебзастосунки **TinkerCAD** (<https://www.tinkercad.com>), **SketchUp** (<https://sketchup.com>) та інші.
- для візуалізації (рендерингу) 3D-об'єктів. Ці програми забезпечують отримання високоякісного реалістичного тривимірного зображення на основі підготовлених в інших програмах описів 3D-об'єктів, виконаних певною мовою програмування. Такими є, наприклад, програми **V-Ray**, **RenderMan**, **Sunflow**.
Інша класифікація програм для роботи з тривимірними об'єктами базується на галузях їх застосування:
- для інженерного проєктування, наприклад **AutoCAD**, **SolidWorks**, **LEGO Digital Designer**;
- для створення архітектурних моделей, наприклад **SketchUp**;
- для проєктування меблів, інтер'єру, наприклад **PRO100**, **Sweet Home 3D**;
- для кінематографії, мультиплікації, наприклад **Cinema 4D**, **Autodesk Maya**, **Houdini**;
- для розробки комп'ютерних ігор, наприклад **Unity**,
та для інших галузей.

Існують також **універсальні** 3D-редактори, які можуть використовуватись у різних галузях діяльності людини, наприклад **3DStudio MAX**, **MODO**, **Blender** та інші.

Зауважимо, що створити 3D-модель можна не лише з використанням 3D-редакторів. Такі завдання можуть виконувати системи штучного інтелекту. Це можуть бути як спеціалізовані ресурси, наприклад **Meshy** (<https://www.meshy.ai>), так і мовні моделі, наприклад **ChatGPT** (<https://chatgpt.com>). Побудова 3D-моделі ресурсами штучного інтелекту може виконуватись як за текстовим описом, так і на основі плоских 2D-зображень.

Існують і технічні засоби – **3D-сканери** (мал. 4.8), призначені для створення 3D-моделей. Їх використовують у галузях архітектури для збереження культурної спадщини людства, у стоматології для точнішого проєктування зубних протезів і в інших галузях.



Мал. 4.8. Побудова 3D-моделі скульптури з використанням 3D-сканера

ПРИНЦИПИ ТРИВИМІРНОГО МОДЕЛЮВАННЯ

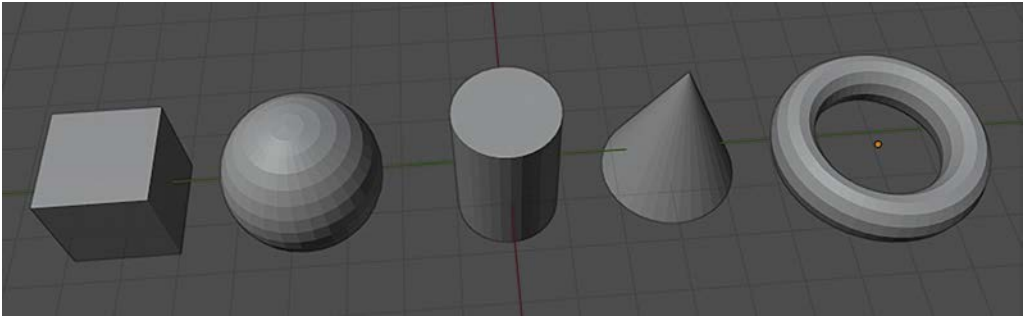


Пригадайте

● Із чого складається векторне графічне зображення? ● Які ви знаєте властивості об'єктів векторного графічного зображення? ● Які об'ємні тіла ви вивчали на уроках математики?

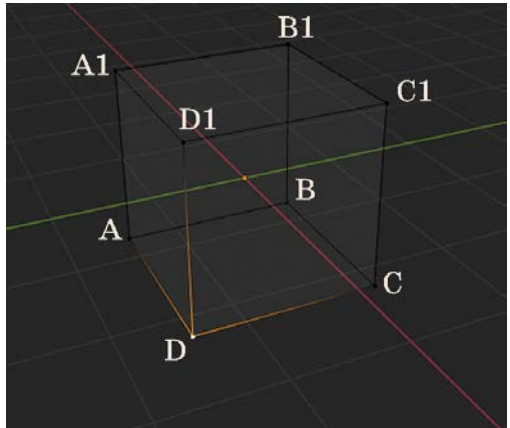
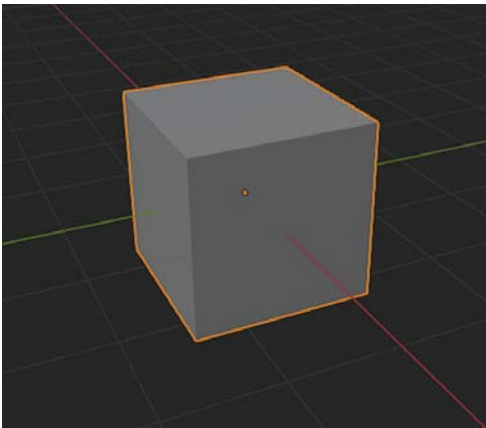
Ви знаєте, що для створення векторного графічного зображення використовують плоскі графічні примітиви, наприклад прямокутники, зірки, овали, фігурні стрілки тощо. Для побудови 3D-моделі використовуються **тривимірні примітиви** – моделі геометричних тіл, з об'єднання та перетину яких утворюється форма поверхні об'єкта моделювання. Такі тривимірні примітиви отримали узагальнену назву **меш** (англ. *mesh* – сітка).

У 3D-редакторах використовують, наприклад, такі тривимірні примітиви (**елементи мешу**): куб, сфера, циліндр, конус, тор (мал. 4.9) та інші.



Мал. 4.9. Приклади тривимірних примітивів: куб, сфера, циліндр, конус, тор

Кожен тривимірний примітив може бути описаний математично. Наприклад, для побудови куба достатньо указати координати трьох його вершин у тривимірному просторі, наприклад вершин А, С, А1 (мал. 4.10). Координати інших вершин куба можуть бути однозначно обчислені на основі властивостей куба.



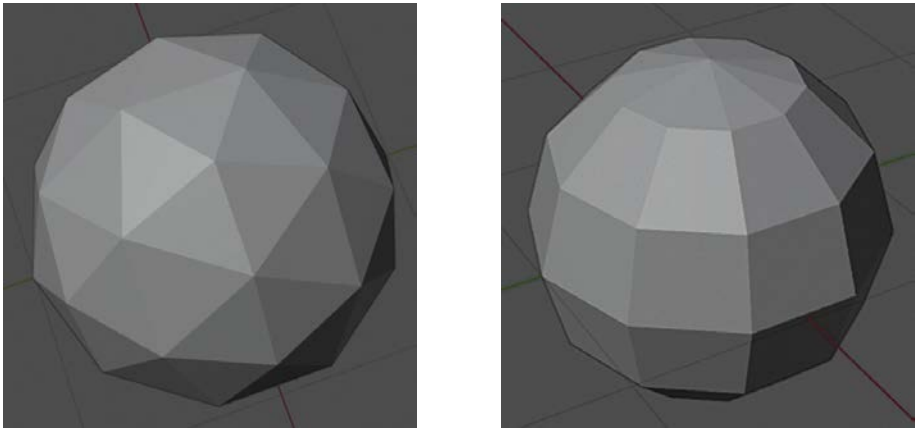
Мал. 4.10. Тривимірний примітив куб



З математичного опису властивостей окремих елементів графічної тривимірної моделі автоматично будується її математична модель, яку використовує 3D-редактор для візуалізації моделі.

Поверхня 3D-моделі будь-якого об'єкта розділяється на невеликі плоскі ділянки – **грані**, що мають форму багатокутника (переважно це трикутники та чотирикутники). Наприклад, куб на малюнку 4.10 має 6 граней: ABCD, A1B1C1D1, AA1B1B, BB1C1C, CC1D1D, AA1D1D. Грань обмежена відрізками – **ребрами**. Наприклад, грань AA1B1B обмежена ребрами AA1, A1B1, B1B, AB. Кожне ребро має дві **вершини**. Наприклад, ребро AA1 має вершини A та A1.

Для побудови моделей круглих тіл (конус, циліндр, сфера, тор) використовують наближене подання їхніх гладких поверхонь поверхнями багатогранників. Наприклад, для побудови 3D-моделі сфери можна використати один із двох елементів мешу – **Сфери Іко** та **Сфери UV** (мал. 4.11).



Мал. 4.11. Вигляд 3D-моделей сфери:
ліворуч – **Сфера Іко**, праворуч – **Сфера UV**

Поверхня **Сфери Іко** складається з однакових рівносторонніх трикутників. Її назва походить від назви правильного двадцятигранника – **ікосаедра** (грец. *είκοσι* – двадцять, грец. *ἔδρα* – площа). Поверхня **Сфери UV** утворюється трикутниками (біля «полюсів») і чотирикутниками, які розташовані вздовж кіл – «паралелей». Літери **U** та **V** у назві сфери є позначеннями осей координат на плоскому зображенні, яке накладається як текстура на об'ємне тіло. Що більше взято трикутних і чотирикутних граней, то більше поверхня побудованого багатогранника візуально подібна до гладкої поверхні.

Над елементами 3D-моделі можна виконувати **операції перетворення: переміщення** всього об'єкта або окремих вершин, ребер, граней, **поворот**, **масштабування** (змінення розмірів) та інші.

Під час 3D-моделювання також обирають матеріали, якими покривають частини поверхні об'єкта, й виконують інші налаштування залежно від призначення і подальшого використання моделі. Наприклад, якщо планується друк 3D-зображення на папері, то вибирають вид, розташування та потужність джерел освітлення, визначають положення камери для вибору точки огляду моделі та інше.

Таким чином, основними властивостями 3D-моделей є кількість вершин, ребер і граней, їх розташування на поверхні моделі та види матеріалів, які покривають поверхню.



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання 1. Перегляньте в онлайн-переглядачі **Online 3D Viewer** кілька 3D-моделей. Дізнайтеся значення їхніх властивостей. Розгляньте 3D-моделі з різних боків.

1. Відкрийте у вікні браузера головну сторінку онлайн-переглядача **Online 3D Viewer** за адресою <https://3dviewer.net/>.
2. Ознайомтеся з переліком форматів, у яких можуть бути збережені файли 3D-моделей.
3. Виберіть кнопку **obj** для перегляду зразка 3D-моделі з файлу формату **OBJ**.
4. Розгляньте, з яких об'єктів складається 3D-модель. Поверніть модель протягуванням будь-якої точки робочого поля. Змініть масштаб перегляду, використовуючи коліщатко прокручування.
5. Створіть у текстовому процесорі документ і запишіть у нього:
 - а) ім'я файлу моделі;
 - б) кількість вершин у моделі (значення властивості **Vertices**);
 - в) кількість граней у моделі (значення властивості **Triangles**);
 - г) розмір моделі по осі OX (значення властивості **Size X**);
 - д) розмір моделі по осі OY (значення властивості **Size Y**);
 - е) висоту моделі (значення властивості **Size Z**).
6. Визначте об'єм моделі (значення властивості **Volume**) та площу її поверхні (значення властивості **Surface**), вибравши відповідні посилання **Calculate** (англ. *calculate* – обчислити). Запишіть отримані значення в текстовий документ.
7. Зробіть скриншот моделі, вибравши кнопку **Create snapshot**  (англ. *create snapshot* – зробити знімок). Збережіть у вашій папці, уставте в текстовий документ.
8. Виберіть логотип  переглядача **Online 3D Viewer** для повернення на його головну сторінку.
9. Виберіть інші формати файлів, наприклад **STL**, **FBX**. Перегляньте 3D-моделі. Запишіть у текстовий документ значення властивостей моделей. Уставте скриншоти.
10. Збережіть документ у вашій папці у файлі з іменем **вправа 4.1.1**.
11. Закрийте всі відкриті вікна.

Завдання 2. Уставте до текстового документа 3D-модель земної кулі. Для цього:

1. Створіть у текстовому процесорі новий документ.
2. Виконайте **Вставлення** ⇒ **Ілюстрації** ⇒ **3D-моделі** ⇒ **Стокові 3D-моделі**.
3. Виберіть у вікні **3D-моделі з інтернету** категорію **Animated for Education** (англ. *animated for education* – анімовані для освіти),

ескіз земної кулі та кнопку **Вставити**. Дочекайтеся вставлення зображення в документ.

4. Увімкніть відтворення анімації тривимірної моделі вибором кнопки . Зупиніть анімацію вибором кнопки . Проаналізуйте, що демонструє анімація.
5. Змініть розмір області перегляду 3D-моделі, використавши маркери на її межах.
6. Зробіть більшим масштаб перегляду 3D-моделі, вибравши на тимчасовій вкладці **3D-модель** у групі **Розмір** кнопку **Панорамування та масштаб** і скориставшись інструментом масштабування  на межі зображення.
7. Вимкніть інструмент масштабування, вибравши повторно кнопку **Панорамування та масштаб**.
8. Змініть точку огляду 3D-моделі так, щоб по центру області перегляду опинилась Україна, використавши інструмент обертання .
9. Збережіть документ у вашій папці у файлі з іменем **вправа 4.1.2**.
10. Закрийте вікно текстового процесора.



Найважливіше в цьому пункті

Тривимірна графіка (3D-графіка) – це розділ комп'ютерної графіки, у якому вивчають методи та засоби створення графічних моделей просторових об'єктів з використанням комп'ютерних програм.

Тривимірна, або 3D-модель об'єкта – це графічна комп'ютерна модель об'єкта, призначена для аналізу його форми, розмірів, пропорцій та інших характеристик з різних ракурсів. Процес створення 3D-моделей називають **тривимірним моделюванням**.

Створені на основі просторових моделей плоскі зображення тривимірних об'єктів називають **3D-зображеннями**. Процес створення плоских зображень на основі 3D-моделей називається **рендерингом**. У **3D-анімації** моделюється рух об'єктів або їхніх частин.

Сферами застосування тривимірного моделювання є інженерне проектування, архітектура, освіта і наука, кінематографія та мультиплікація, комп'ютерні ігри, 3D-друк та інші.

Програми для роботи з тривимірними об'єктами можна класифікувати **за призначенням** та **за галузями їх застосування**.

3D-моделі створюються з **тривимірних примітивів** – моделей геометричних тіл, які отримали узагальнену назву **меш**. Кожен тривимірний примітив (**елемент мешу**) може бути описаний математично.

Поверхня 3D-моделі об'єкта розділяється на невеликі плоскі ділянки – **грані**, що мають форму багатокутника. Грань обмежена відрізками – **ребрами**. Кожне ребро має дві **вершини**. Над елементами 3D-моделі можуть виконуватись **операції перетворення: переміщення** всього об'єкта або окремих вершин, ребер, граней, **поворот, масштабування** (змінення розмірів) та інші.

Основними властивостями 3D-моделей є кількість вершин, ребер і граней, їх розташування на поверхні моделі та види матеріалів, які покривають поверхню.



Дайте відповіді на запитання

1. Що таке *тривимірна графіка*?
2. Які ви знаєте галузі використання тривимірної графіки? Наведіть приклади.

3. Що таке 3D-модель, 3D-зображення, 3D-анімація? У чому їхня сутність та особливості?
4. З яких тривимірних примітивів будують 3D-моделі об'єктів?
5. З яких елементів складається поверхня 3D-моделі об'єкта?
6. Які операції перетворення можна виконувати над елементами 3D-моделі?



Обговоріть і зробіть висновки

1. З якою метою 3D-моделі та 3D-анімація можуть бути використані в навчанні на різних предметах?
2. Які ви знаєте галузі застосування 3D-сканерів? Яким чином ці пристрої спрощують роботу фахівців у цих галузях?
3. Як залежить якість моделювання кривих поверхонь від кількості граней на поверхні 3D-моделі об'єкта? Наведіть приклади.



Виконайте завдання

1. Перегляньте колекцію 3D-моделей на сайті **PixelSquid** (адреса ресурсу: <https://www.pixelsquid.com>). Які категорії об'єктів можна знайти на сайті? Попрактикуйтеся в керуванні переглядом 3D-моделей.
2. Перегляньте колекцію 3D-моделей на сайті **Mixamo** (адреса ресурсу: <https://www.mixamo.com>). Які категорії об'єктів можна знайти на сайті? Попрактикуйтеся в керуванні переглядом і масштабуванні 3D-моделей, виборі анімацій для об'єктів.
3. Створіть у текстовому процесорі документ. Уставте до нього 3D-модель зі стокових моделей категорії **Avatars** (англ. *avatars* – аватари). Виконайте поворот моделі. Збільште масштаб відображення. Збережіть документ у вашій папці у файлі з іменем **завдання 4.1.3**.
4. Створіть презентацію з одного слайда з макетом **Пустий слайд**. Уставте до нього зображення анімованої 3D-моделі зі стокових моделей категорії **Animals** (англ. *animals* – тварини), наприклад собаки.

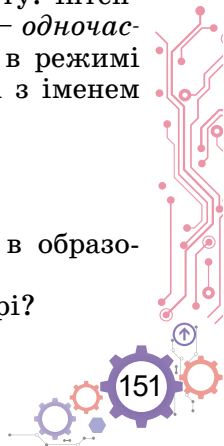
Ознакою анімації є значок  на ескізі моделі. Виконайте поворот моделі, збільште масштаб відображення. Перегляньте презентацію в режимі демонстрації. Переконайтеся у відтворенні анімації в режимі демонстрації. Збережіть презентацію у вашій папці у файлі з іменем **завдання 4.1.4**.

5. Створіть презентацію з одного слайда з макетом **Пустий слайд**. Уставте до нього зображення 3D-моделі зі стокових моделей категорії **Animated Animals**, наприклад бджоли. Додайте ефект анімації **Обертання** з категорії **Тривимірна**. Налаштуйте параметри ефекту: інтенсивність – *безперервна анімація*, напрямок – *уліво*. Початок – *одночасно* з початком перегляду слайда. Перегляньте презентацію в режимі демонстрації. Збережіть презентацію у вашій папці у файлі з іменем **завдання 4.1.5**.



Готуємось до вивчення нового матеріалу

1. Які ви знаєте види графіки? Схарактеризуйте їх.
2. Що таке *перспектива*? Для чого її використовують в образотворчому мистецтві?
3. Які операції можна виконувати у графічному редакторі?



4.2. РЕДАКТОР 3D-ГРАФІКИ BLENDER. УСТАВЛЕННЯ ТА ПЕРЕТВОРЕННЯ ТРИВИМІРНИХ ПРИМІТИВІВ. ПРИНЦИПИ ТРИВИМІРНОЇ НАВІГАЦІЇ

У цьому пункті йтиметься про:

- ▶ особливості роботи з редактором 3D-графіки **Blender**;
- ▶ уставка та перетворення тривимірних примітивів;
- ▶ принципи тривимірної навігації;
- ▶ збереження проєктів, експорт 3D-моделей;
- ▶ рендеринг 3D-моделей, збереження 3D-зображення.

РЕДАКТОР 3D-ГРАФІКИ BLENDER



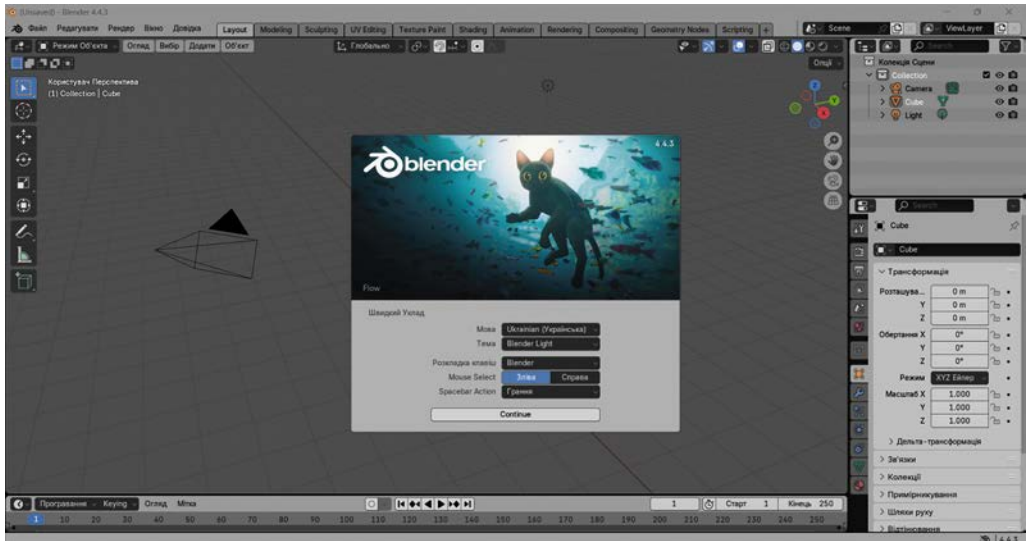
Пригадайте

- Які ви знаєте елементи вікон графічних редакторів? • Які об'єкти можуть бути розміщені у вікнах графічних редакторів? • Як задають координати точок на площині на уроках математики?

Одним із популярних редакторів 3D-графіки є універсальний редактор **Blender**. Його можна установити на локальному комп'ютері, безкоштовно завантаживши файл інсталяції з офіційного сайту **Blender** (<https://www.blender.org>).

Основні операції, які можна виконувати в редакторі **Blender**:

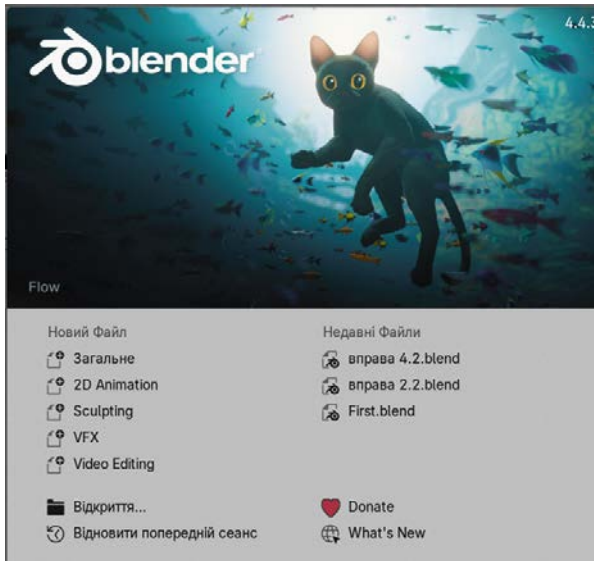
- створення, редагування та перегляд 3D-моделей;
 - використання матеріалів і текстур для оформлення поверхонь моделей;
 - візуалізація (рендеринг) 3D-моделей для отримання тривимірних зображень;
 - створення тривимірної анімації;
 - робота з файлами (відкриття, збереження тощо)
- та інші.



Мал. 4.12. Стартове вікно редактора 3D-графіки **Blender** версії 4.4.3

Запустити на виконання редактор **Blender** можна, використавши відповідну команду в меню **Пуск** або ярлик , розташований на **Робочому столі**, на панелі **Швидкий запуск** або в іншому місці. Під час першого запуску редактора у стартовому вікні у списку **Language/ Мова** можна встановити українську як мову інтерфейсу користувача (мал. 4.12).

Після кожного запуску редактора **Blender** на виконання відкривається вікно редактора та поверх нього – заставка з початковим меню (мал. 4.13).



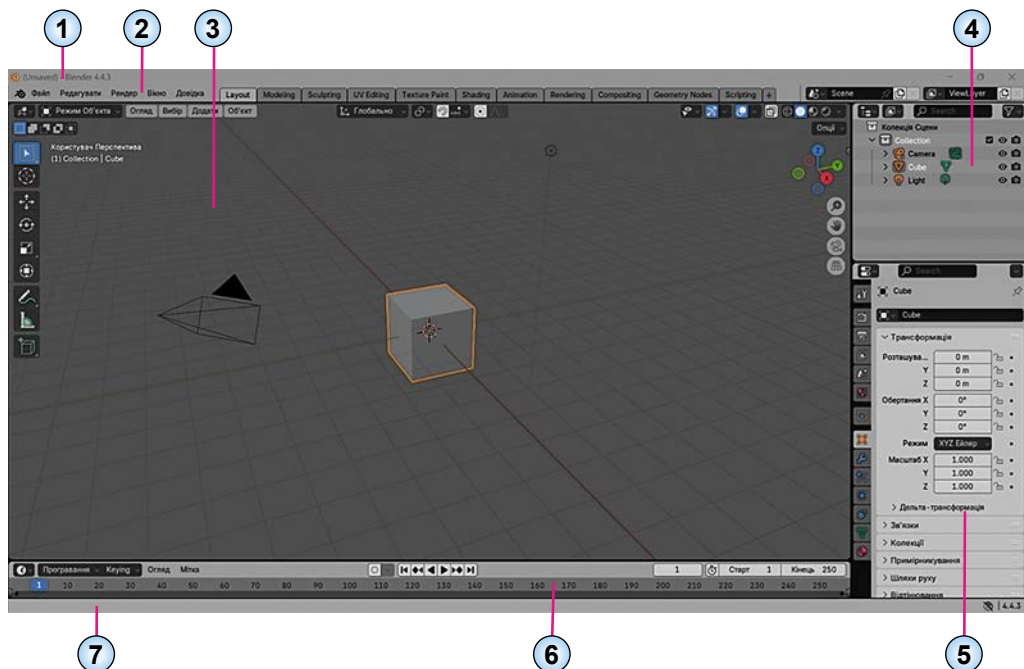
Мал. 4.13. Заставка редактора **Blender** версії 4.4.3 з початковим меню

Для початку створення нової 3D-моделі потрібно вибрати команду **Загальне** у групі **Новий Файл** початкового меню.

Вікно редактора **Blender** (див. с. 154, мал. 4.14) має рядок заголовка (мал. 4.14, 1), головне меню (мал. 4.14, 2), рядок стану (мал. 4.14, 7), а основна частина вікна за замовчуванням поділена на чотири прямокутні області. У кожній області розміщуються інструменти для виконання окремих операцій з об'єктами або переключення режимів роботи редактора. У лівому верхньому куті кожної області міститься кнопка зі списком усіх типів областей, доступних у редакторі. Значок на цій кнопці відповідає типу поточної області.

Основне місце у вікні редактора займає область **Вікно 3D-огляду** (мал. 4.14, 3, значок ). У цій області відбувається побудова 3D-моделей.

У **Вікні 3D-огляду** зображено координатну площину XOY. На ній позначено осі координат: OX – червоного кольору, OY – зеленого кольору. У тривимірному просторі існує третя координатна вісь – OZ, яка проходить через початок координат площини XOY перпендикулярно до цієї площини. Вона не відображається у **Вікні 3D-огляду**



Мал. 4.14. Вікно редактора **Blender**:

1 – рядок заголовка; 2 – головне меню; 3 – область **Вікно 3D-огляду** ;

4 – область **Структуратор** ; 5 – область **Властивості** ;

6 – область **Шкала часу** ; 7 – рядок стану

у звичайному режимі. Осі OZ відповідає лінія синього кольору. Лінії сітки сірого кольору розміщені у площині XOY, і в масштабі за замовчуванням відстань між ними становить 1 метр.

Простір, що відображається у **Вікні 3D-огляду**, називають **сценою**.

Додатні напрямки координатних осей показано на інструменті **гізмо** (англ. *gizmo* – штука) (мал. 4.15), призначеному для управління переглядом сцени.

На сцені після створення нового файлу розміщено три об'єкти (мал. 4.16):

- **камера** (ім'я об'єкта за замовчуванням – **Camera**);
- тривимірний примітив **куб** (ім'я об'єкта – **Cube**);
- **джерело світла** – лампа (ім'я об'єкта – **Light**).

Імена об'єктів відображаються в області **Структуратор** (мал. 4.17,

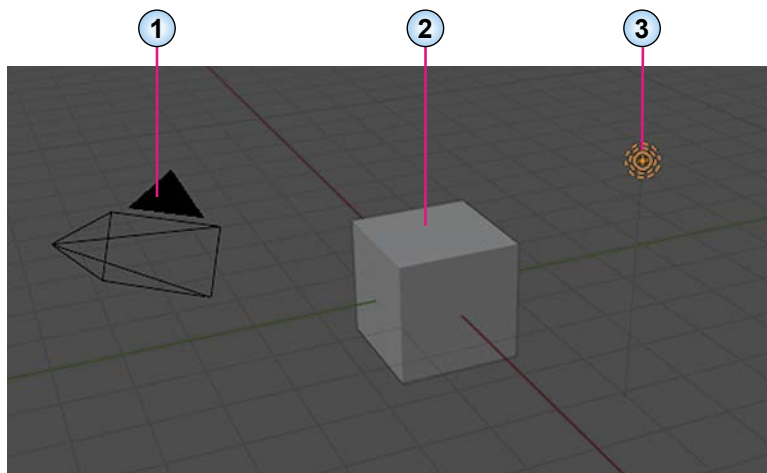
ліворуч, значок ). У цій області можна вибирати об'єкти для ви-

конання з ними операцій. Значення властивостей вибраного об'єкта відображаються й можуть бути змінені в області **Властивості** (мал. 4.17,

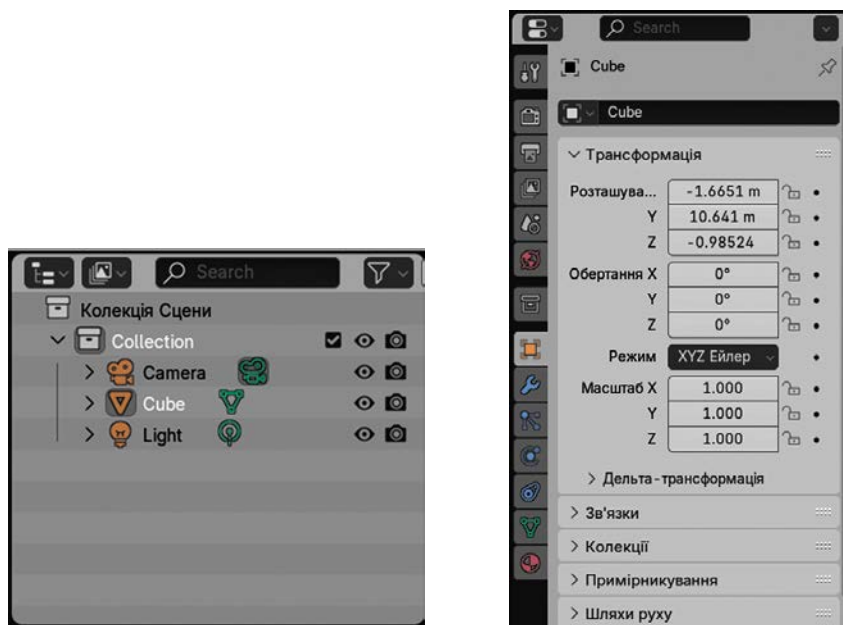
праворуч, значок ).



Мал. 4.15.
Інструмент **гізмо**



Мал. 4.16. Об'єкти сцени: 1 – камера; 2 – куб; 3 – джерело світла



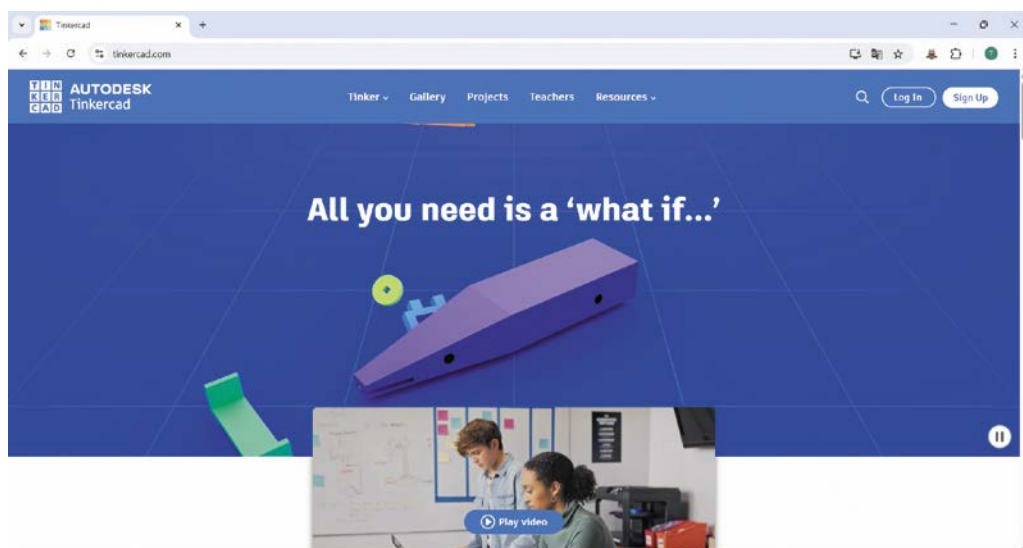
Мал. 4.17. Області **Структуратор** (ліворуч) і **Властивості** (праворуч)

Положення камери та джерел світла, розміщених на сцені, будуть використані у процесі рендерингу 3D-моделі для отримання тривимірного зображення.



Для тих, хто хоче знати більше

Популярним онлайн-редактором 3D-графіки є **Tinkercad** (<https://www.tinkercad.com>) (мал. 4.18) від корпорації **Autodesk** – відомого розробника програмного забезпечення для візуалізації та проектування. Для створення та зберігання власних проєктів на цій платформі потрібна реєстрація. Для користувачів віком до 13 років має бути дозвіл батьків на здійснення реєстрації.



Мал. 4.18. Головна сторінка платформи Autodesk Tinkercad до реєстрації

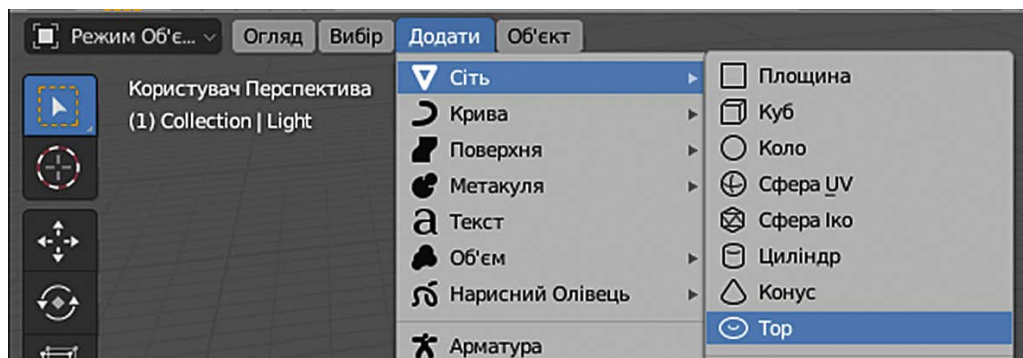
ВСТАВЛЕННЯ ТРИВИМІРНИХ ПРИМІТИВІВ



Пригадайте

- Які ви знаєте тривимірні графічні примітиви?
- Як створити і розмістити новий об'єкт у векторному графічному редакторі?
- Як видалити створений об'єкт у векторному графічному редакторі?

У верхньому лівому куті **Вікна 3D-огляду** розміщена група інструментів для роботи з об'єктами **Огляд** **Вибір** **Додати** **Об'єкт**. Для вставлення на сцену нового тривимірного примітива потрібно вибрати кнопку **Додати**, у її меню – команду **Сіть** і команду вставлення відповідного примітива (мал. 4.19).

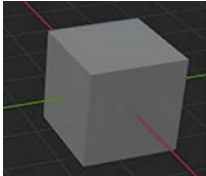
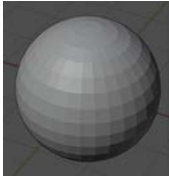
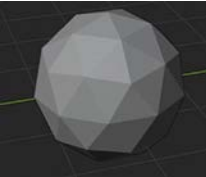
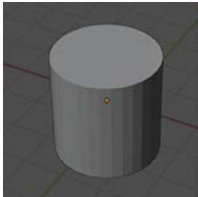
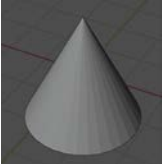
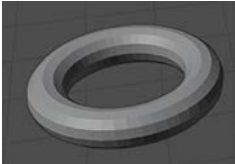


Мал. 4.19. Меню вставлення тривимірного примітива

Приклади деяких тривимірних примітивів наведено в таблиці 4.1.

Таблиця 4.1

Приклади тривимірних примітивів

Команда меню	Приклад примітива	Команда меню	Приклад примітива
 Куб		 Сфера UV	
 Сфера Іко		 Циліндр	
 Конус		 Тор	

Кожен тривимірний примітив має так звану **опорну точку**, яка за замовчуванням розміщена в центрі об'єкта. Після вставлення об'єкта на сцену його опорна точка буде розміщена в точці, позначений

3D-курсором . В області **Структуратор** з'явиться ім'я цього примітива, в області **Властивості** – значення його властивостей за замовчуванням.

Використовуючи меню кнопки **Додати**, можна вставити на сцену також інші об'єкти – криві, поверхні, камери, джерела світла та інші.

Для видалення об'єкта можна вибрати його ім'я в області **Структуратор** і натиснути клавішу **Delete**.

ОПЕРАЦІЇ ПЕРЕТВОРЕННЯ ОБ'ЄКТІВ



Пригадайте

- Які ви знаєте операції перетворення тривимірних об'єктів?
- Як виконати переміщення, обертання та змінення розмірів об'єктів зображення у векторному графічному редакторі?

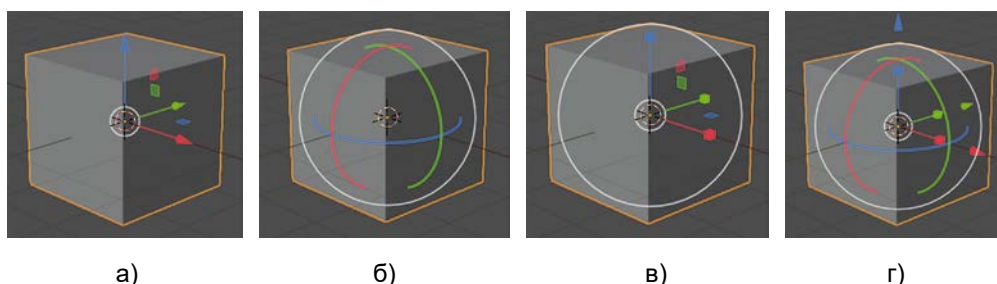
З об'єктами, вставленими на сцену в редакторі **Blender**, можна виконувати операції перетворення: *переміщення, обертання, масштабування* та інші. Для виконання цих та інших операцій призначені елементи керування, розміщені в лівій частині **Вікна 3D-огляду**. Вигляд елементів керування та опис їх призначення наведено в таблиці 4.2.

Елементи керування для виконання операцій над об'єктами

Елемент керування	Призначення
 Вибір Коробкою	Для вибору об'єктів у Вікні 3D-огляду
 Курсор	Для вибору положення 3D-курсора
 Переміщення	Для переміщення вибраного об'єкта
 Оберт	Для обертання вибраного об'єкта
 Масштаб	Для масштабування (змінення розміру) вибраного об'єкта
 Трансформа	Для виконання будь-якого з перетворень – переміщення, обертання або масштабування

Перш ніж виконувати перетворення, потрібно вибрати ім'я об'єкта в області **Структуратор** або скористатись інструментом **Вибір коробкою** .

Після вибору елементів керування, призначених для виконання операцій перетворення, на вибраному об'єкті з'являються **додаткові елементи керування** (мал. 4.20).



Мал. 4.20.Dodatkovі елементи керування для виконання операцій перетворення:
а) переміщення; б) обертання; в) масштабування; г) усіх трьох операцій

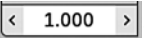
Додатковий елемент керування для переміщення вибраного об'єкта (мал. 4.20, а) має стрілки та прямокутні маркери, колір яких відповідає кольорам координатних осей, та біле коло. Перетягування цих елементів приводить:

- кольорової стрілки – до переміщення об'єкта уздовж відповідної осі;
- прямокутного маркера – до переміщення об'єкта у площині, перпендикулярній до відповідної осі;
- довільної точки в середині білого кола – до вільного переміщення об'єкта.

Подібним чином використовуються і додаткові елементи керування, призначені для виконання інших операцій перетворення.

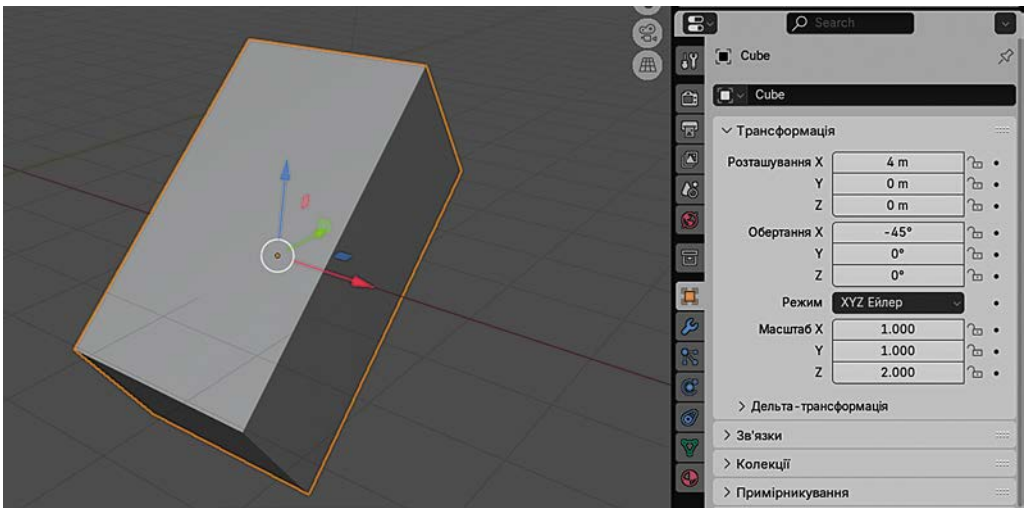
Виконати операцію перетворення вибраного об'єкта також можна, змінивши значення його властивостей в області **Властивості**. На

вкладці **Об'єкт**  у групі **Трансформація** розміщені поля для введення значень властивостей **Розташування**, **Обертання** і **Масштаб** відносно кожної з координатних осей.

Наприклад, щоб збільшити вдвічі висоту вибраного об'єкта, потрібно у групі **Трансформація** у блоці **Масштаб** увести значення **2** в поле **Z**. Значення в поля можна вводити з клавіатури або змінювати кнопками лічильника , що з'являються під час наведення вказівника на поле.

На малюнку 4.21 наведено вигляд об'єкта і значення його властивостей у групі **Трансформація** після виконання перетворень:

- переміщення об'єкта на **4 м** уздовж осі **OX** (**Розташування X – 4 m**);
- обертання на **45 градусів** відносно осі **OX** (**Обертання X – -45°**);
- збільшення *вдвічі* висоти об'єкта (**Масштаб Z – 2.000**).



Мал. 4.21. Вигляд об'єкта після виконання операцій перетворення та відповідні значення властивостей у групі **Трансформація** області **Властивості**

Значення властивості **Розташування** вказують положення опорної точки об'єкта по кожній осі відносно точки перетину координатних осей. Будемо називати їх **координатами опорної точки** і записувати у круглих дужках. Наприклад, якщо опорна точка об'єкта розміщена у площині **XOY** в напрямку осі **OX** за 4 м від точки перетину осей, то її координати (4; 0; 0).

ІНСТРУМЕНТИ ТРИВИМІРНОЇ НАВІГАЦІЇ



Пригадайте

- Як у живописі зображують об'єкти, які розміщені далеко від спостерігача?
- Що означає слово «навігація»? У яких галузях воно використовується?
- Як у підручниках математики зображують об'ємні тіла?

У процесі створення тривимірної моделі потрібно постійно оцінювати її вигляд і взаємне розміщення її частин на сцені. Для цього доводиться змінювати точку огляду сцени та масштаб перегляду.

Дивитися на сцену можна під довільним кутом і бачити об'єкти в перспективі. Це так звана **перспективна проєкція**, за використання якої об'єкти, що розміщені ближче до точки перегляду, здаються більшими, а ті, що далі – меншими. Так виглядають об'єкти навколишнього світу на фотографіях. А в математиці та кресленні об'ємні тіла зображують в **ортогональній проєкції**, у якій розміри об'єктів не залежать від їхньої віддаленості від користувача (мал. 4.22).

Перспектива (лат. *perspicio* – ясно бачу) – система зображення об'ємних тіл на площині, яка враховує їхню просторову структуру й віддаленість окремих частин від спостерігача.

Ортогональність (грец. *ὀρθός* – прямий, *γωνία* – кут) – термін, яким позначають перпендикулярність векторів.



Мал. 4.22. Зображення будинків у перспективній (ліворуч) та ортогональній (праворуч) проєкціях

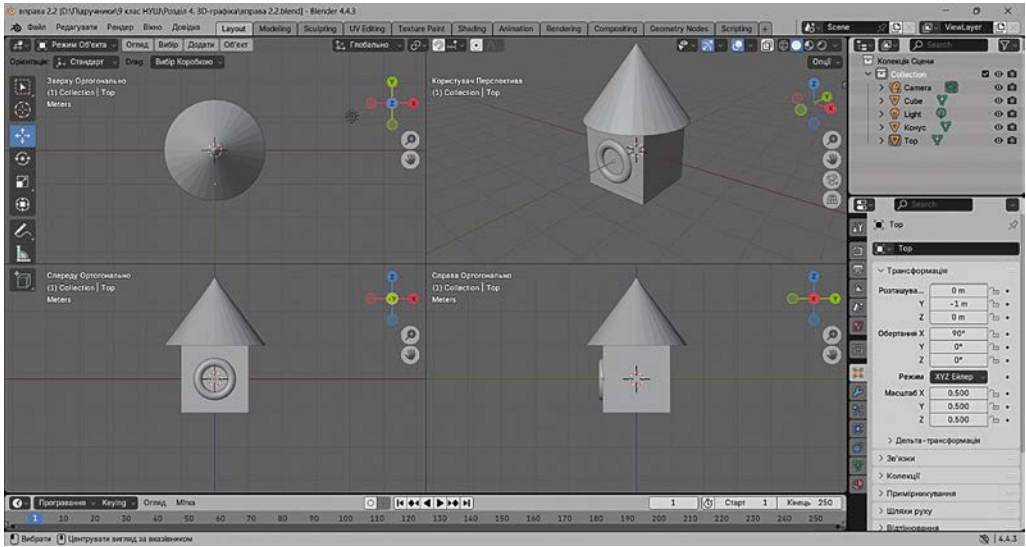
У редакторі **Blender** можна переключати вигляд сцени між перспективною та ортогональною проєкціями.

Операції змінення масштабу, точки перегляду та способу подання виду сцени у **Вікні 3D-огляду** називають операціями **тривимірної навігації**. Для їх виконання призначені різні засоби: інструмент **гізмо**, розташовані поруч із ним інструменти управління переглядом **Вікна 3D-огляду**, команди меню кнопки **Огляд**, «гарячі» клавіші на додатковій цифровій клавіатурі, використання миші після наведення вказівника на **Вікно 3D-огляду**. У таблиці 4.3 наведено деякі зі способів виконання операцій тривимірної навігації.

Способи виконання операцій тривимірної навігації

Операція	Способи виконання
Змінити напрямки координатних осей	Перетягнути кольорові маркери на інструменті гізмо Перемістити вказівник у Вікні 3D-огляду з натиснутим коліщатком прокручування миші
Змінити масштаб перегляду об'єктів у Вікні 3D-огляду	Прокрутити коліщатко миші у Вікні 3D-огляду
Перемістити початок координат у Вікні 3D-огляду	Навести вказівник на інструмент Перемістити перегляд  і перемістити вказівник у Вікні 3D-огляду з натиснутою лівою кнопкою миші Натиснути клавішу Shift і перемістити вказівник у Вікні 3D-огляду з натиснутим коліщатком прокручування
Відобразити виділені об'єкти в центрі сцени	Кнопка Огляд ⇒ Вибране у Кадр
Відобразити у Вікні 3D-огляду всі об'єкти сцени	Кнопка Огляд ⇒ Усе у Кадр
Переключитися між режимами ортогональної та перспективної проєкцій	Вибрати інструмент Перемкнути між перспективною та ортогональною проєкціями поточного огляду  /  Кнопка Огляд ⇒ Перспектива / Ортогонально
Повернути сцену для перегляду вздовж координатних осей: OY – вид спереду / ззаду; OX – вид справа / зліва; OZ – вид зверху / знизу	Вибрати кольоровий маркер з назвою відповідної осі на інструменті гізмо Кнопка Огляд ⇒ Точка зору ⇒ Top / Bottom / Front / Back / Right / Left Натиснути на додатковій цифровій клавіатурі: 1 – вид спереду, 3 – вид справа, 7 – вид зверху, 9 – змінити напрямок перегляду на протилежний
Одночасно відобразити у Вікні 3D-огляду вид сцени зверху, спереду, справа та перспективну проєкцію (мал. 4.23)	Кнопка Огляд ⇒ Area ⇒ Перемкнути Чотирибічний Огляд Натиснути сполучення клавіш Ctrl + Alt + Q. Повторне натискання повертає до попереднього вигляду





Мал. 4.23. Одночасне відображення чотирьох видів сцени

ЗБЕРЕЖЕННЯ ПРОЄКТУ. ЕКСПОРТ МОДЕЛІ. РЕНДЕРИНГ 3D-МОДЕЛІ



Пригадайте

- Чим відрізняється файл проєкту від файлу завершеного мультимедійного продукту, наприклад відеоролика?
- Що означають терміни *експорт* та *імпорт*?
- У чому полягає операція *рендеринг*?

Створену тривимірну модель можна зберегти у файлі проєкту для продовження її редагування в подальшому. Для цього слід:

1. Виконати **Файл** ⇒ **Зберегти (Зберегти як)**.
2. Відкрити у вікні **Перегляд файлу Blender**, що відкриється, папку для збереження проєкту.
3. Увести ім'я файлу в поле **Назва файлу** замість запропонованого за замовчуванням імені *Без назви.blend*.
4. Вибрати кнопку **Зберегти файл Blender**.

Розширення імені файлу проєкту – *blend*. Змінити його під час збереження неможливо. Файли цього формату можна відкрити в редакторі **Blender** командою **Файл** ⇒ **Відкриття** для продовження роботи з проєктом.

Якщо планується відкривати створену 3D-модель для перегляду в інших програмах (наприклад, **Word**, **PowerPoint**, **Online 3D Viewer**), то 3D-модель можна **експортувати** у файли інших форматів, наприклад **STL**, **FBX**, **OBJ**. Для цього призначена команда **Файл** ⇒ **Експортувати**. Модель, збережену в таких форматах, можна також імпортувати у свій проєкт, використавши команду **Файл** ⇒ **Імпортувати**. Імпортована модель буде додаватися до відкритого проєкту.

Для отримання 3D-зображення на основі створеної 3D-моделі потрібно виконати операцію **рендеринг**. Під час рендерингу буде відтворено вигляд моделі через об'єktiv камери з урахуванням положення та потужності джерел світла.

Щоб побачити, як виглядатиме створена модель через об'єктив камери, потрібно виконати *Огляд* $\Rightarrow$ *Точка зору* $\Rightarrow$ *Camera* або вибрати кнопку **Перемкнути огляд камери** . Якщо частини моделі частково виходять за межі отриманого кадру, то можна змінити положення камери. Для цього:

1. Вибрати об'єкт **Camera** в області **Структуратор**.
2. Вибрати вкладку **Властивості Даних Об'єкта**  в області **Властивості**.
3. Змінити значення в полях **Фокусна відстань**, **Зсув по X**, **Зсув по Y** для отримання коректного зображення в кадрі.

Освітлення сцени у режимі перегляду **Суцільний** , що встановлено за замовчуванням, не відповідає тому, що буде отримано у процесі рендерингу. Щоб побачити вигляд об'єктів сцени з урахуванням положення джерел освітлення, потрібно переключитися в режим **Рендерене**, вибравши відповідну кнопку  у верхньому правому

куті **Вікна 3D-огляду**. У випадку невдалого освітлення моделі можна змінити розміщення джерела світла. Для цього потрібно вибрати об'єкт **Light** в області **Структуратор** і використати будь-які способи для його переміщення. Для регулювання потужності лампи на вкладці **Властивості Даних Об'єкта**  в області **Властивості** потрібно змінити значення в полі **Power** (англ. *power* – потужність).

На малюнку 4.24 наведено вигляд моделі в режимі **Рендерене** з оглядом через об'єктив камери та в режимі за замовчуванням **Суцільний**.



Мал. 4.24. Вигляд моделі в режимі **Рендерене** з оглядом через об'єктив камери (ліворуч) та в режимі **Суцільний** (праворуч)

Після корекції освітлення та положення камери для початку рендерингу потрібно виконати **Рендер** $\Rightarrow$ **Рендер Зображення** або натиснути клавішу **F12**. Після завершення розрахунків, потрібних для рендерингу, відкривається вікно **Рендер Blender** з результатами опрацювання зображення. У цьому вікні для збереження отриманого 3D-зображення призначена команда **Зображення** $\Rightarrow$ **Зберегти (Зберегти як)**. За замовчуванням файл 3D-зображення зберігається у форматі PNG.

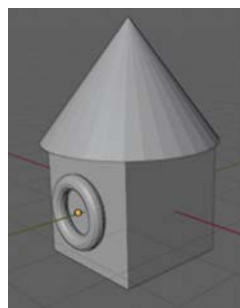


Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання. Створіть у редакторі **Blender** тривимірну модель, розташувавши за зразком куб, конус і тор (мал. 1).

1. Відкрийте програму **Blender**.
2. Виберіть на заставці в початковому меню команду **Загальне**.
3. Ознайомтеся з інтерфейсом програми, знайдіть наявні об'єкти сцени.
4. Уставте на сцену конус, налаштуйте його положення та розміри. Для цього:
 1. Виконайте **Додати** $\Rightarrow$ **Сіть** $\Rightarrow$ **Конус**.
 2. Перемістіть конус уздовж осі OZ так, щоб він торкався верхньої грані куба. Для цього виберіть кнопку **Переміщення**  і перетягніть синю стрілку так, щоб конус зайняв потрібне положення.
3. Збільште в півтора раза радіус основи конуса. Для цього в області **Властивості** на вкладці **Об'єкт** у блоці **Масштаб** установіть значення 1.5 в полях X та Y.
5. Уставте на сцену тор, налаштуйте його положення та розмір. Для цього:
 1. Виконайте **Додати** $\Rightarrow$ **Сіть** $\Rightarrow$ **Тор**.
 2. Поверніть тор на 90° відносно осі OX. Для цього в області **Властивості** на вкладці **Об'єкт** у блоці **Обертання** установіть значення 90 в поле X.
 3. Зменште у 2 рази всі розміри тора, уставивши значення 0.5 в усі поля блока **Масштаб** на вкладці **Об'єкт** в області **Властивості**.
 4. Перемістіть тор уздовж осі OY так, щоб він розмістився перед передньою гранню куба, використовуючи інструмент **Переміщення**.
6. Переконайтеся у правильності розміщення об'єктів, переглянувши їх із різних ракурсів. Для цього виберіть на інструменті **гізмо** маркери: *зелений* (y) – вид спереду / ззаду, *червоний* (x) – вид справа / зліва, *синій* (z) – вид зверху / знизу. Якщо під час перегляду в одному з ракурсів стане помітно, що деякий об'єкт розташовано неправильно, то виберіть його ім'я в області **Структуратор** і змініть положення.



Мал. 1. Зразок до завдання

7. Відновіть на сцені перспективну проєкцію, перемістивши у **Вікні 3D-огляду** вказівник з натиснутим коліщатком прокручування миші.
8. Збережіть проєкт у вашій папці у файлі з іменем **вправа 4.2**. Для цього виконайте **Файл** ⇒ **Зберегти**, відкрийте у вікні **Перегляд файлу Blender** свою папку, уведіть ім'я файлу **вправа 4.2**, виберіть кнопку **Зберегти файл Blender**.
9. Виконайте рендеринг моделі. Для цього:
 1. Розгляньте вигляд моделі через об'єкт камери, вибравши кнопку **Перемкнути огляд камери** .
 2. Змініть положення та фокусну відстань камери так, щоб модель повністю помістилася в кадрі. Для цього:
 1. Виберіть об'єкт **Camera** в області **Структуратор**.
 2. Виберіть вкладку **Властивості Даних Об'єкта**  в області **Властивості**.
 3. Змінюйте значення в полях **Фокусна відстань**, **Зсув по X**, **Зсув по Y** до отримання коректного зображення в кадрі.
 3. Увімкніть режим **Рендерене**, вибравши відповідну кнопку .
 4. Виберіть об'єкт **Light** в області **Структуратор**.
 5. Перемістіть об'єкт **Light** для забезпечення рівномірного освітлення моделі.
 6. Виконайте рендеринг, вибравши **Рендер** ⇒ **Рендер Зображення**.
10. Збережіть зображення у вашій папці у файлі з іменем **вправа 4.2.png**. Для цього у вікні **Рендер Blender** виконайте **Зображення** ⇒ **Зберегти**, уведіть ім'я файлу **вправа 4.2**, виберіть кнопку **Зберегти зображення**.
11. Закрийте всі відкриті вікна.



Найважливіше в цьому пункті

Blender – це універсальний редактор 3D-графіки, який можна встановити на локальному комп'ютері. Основні операції, які можна виконувати в редакторі **Blender**: створення, редагування та перегляд 3D-моделей; використання матеріалів і текстур для оформлення поверхонь моделей; візуалізація (рендеринг) 3D-моделей для отримання тривимірних зображень; створення тривимірної анімації; робота з файлами (відкриття, збереження тощо) та інші.

У **Вікні 3D-огляду** редактора **Blender** зображено координатну площину XOY з осями координат OX та OY. Простір, що відображається у **Вікні 3D-огляду**, називають **сценою**. На сцені відбувається побудова 3D-моделі з тривимірних примітивів. Імена об'єктів, розміщених на сцені, відображаються в області **Структуратор**. Значення властивостей вибраного об'єкта відображаються і можуть бути змінені в області **Властивості**.

Для вставлення на сцену тривимірного примітива потрібно виконати **Додати** ⇒ **Сім'я** і вибрати команду вставлення відповідного примітива.

З об'єктами в редакторі **Blender** можна виконувати операції перетворення: переміщення, обертання, масштабування та інші. Для виконання цих операцій можна використовувати інструменти перетворення або змінювати значення властивостей об'єктів на вкладці **Об'єкт** в області **Властивості**.

Вигляд сцени можна переключати між **перспективною** та **ортогональною** проєкціями. Операції змінення масштабу відображення, точки перегляду та способу по-



дання виду сцени у **Вікні 3D-огляду** називають операціями **тривимірної навігації**. Для їх виконання використовують різні засоби – інструменти управління переглядом **Вікна 3D-огляду**, команди меню кнопки **Огляд**, «гарячі» клавіші, використання миші за наведення вказівника на **Вікно 3D-огляду** та інші.

Створену тривимірну модель можна зберегти у файлі проєкту з розширенням імені *blend* або експортувати у файли інших форматів. Для отримання 3D-зображення на основі 3D-моделі потрібно виконати операцію *рендеринг*. Під час рендерингу буде відтворено вигляд моделі через об'єкти камери з урахуванням положення та потужності джерел світла.



Дайте відповіді на запитання

1. Як додати тривимірний примітив до моделі?
2. Які операції перетворення можуть бути виконані зі вставленими об'єктами? Якими способами їх можна виконати?
3. Які ви знаєте операції тривимірної навігації? Якими способами їх можна виконати?
4. У яких проєкціях можна переглядати вид моделі? Якими способами можна змінити вид проєкції?
5. Як зберегти проєкт, що містить тривимірну модель?
6. У чому сутність операції рендеринг?
7. Як отримати 3D-зображення на основі створеної 3D-моделі?



Обговоріть і зробіть висновки

1. Що спільного та чим відрізняються прийоми роботи у графічних редакторах растрової, векторної та тривимірної графіки? Редактор якого виду графіки має найбільшу подібність до редактора тривимірної графіки?
2. Які переваги та недоліки використання перспективного та ортогонального режимів перегляду 3D-моделі?
3. Чим відрізняються операції збереження та експортування 3D-моделі? У яких ситуаціях кожна з них може бути використана?

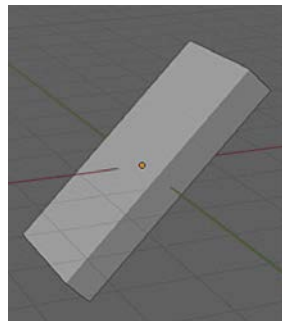


Виконайте завдання

1. Створіть новий файл у програмі **Blender**. Змініть форму куба таким чином, щоб його довжина стала вдвічі більшою за початкову, висота – в 4 рази меншою, а його кут відносно площини XOY став 45° (мал. 4.25). Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.2.1.blend**.

2. Створіть у програмі **Blender** модель ракети, що складається із чотирьох циліндрів та конуса (мал. 4.26). Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.2.2.blend**. Екпортуйте модель у формат **STL** і збережіть у вашій папці у файлі з іменем **завдання 4.2.2.stl**. У переглядачі **Online 3D Viewer** (<https://3dviewer.net/>) відкрийте файл моделі, використавши кнопку

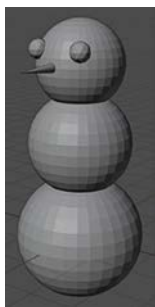
Open from your device  (англ. *open from your device* – відкрити з вашого пристрою). Запишіть у зошит значення властивостей вашої моделі.



Мал. 4.25. Зразок до завдання 1



Мал. 4.26. Зразок до завдання 2



Мал. 4.27. Зразок до завдання 3



Мал. 4.28. Зразок до завдання 4

3. Створіть у програмі **Blender** модель сніговика, що складається із трьох сфер UV, конуса та двох сфер Іко (мал. 4.27). Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.2.3.blend**. Виконайте рендеринг моделі та отримайте 3D-зображення. Збережіть зображення у вашій папці у файлі з іменем **завдання 4.2.3.png**.

4. Створіть у програмі **Blender** модель стільця, що складається із п'яти кубів (мал. 4.28). Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.2.4.blend**. Експортуйте модель у формат **STL** і збережіть у вашій папці у файлі з іменем **завдання 4.2.4.stl**. У текстовому процесорі **Word** уставте модель у текстовий документ (**Вставлення** ⇒ **Ілюстрації** ⇒ **3D-моделі** ⇒ **Цей пристрій**). Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 4.2.4**.



5. Використовуючи довідкову систему, проведіть дослідження щодо використання «гарячих» клавіш для переміщення об'єктів у редакторі **Blender**. Зробіть висновки, які переваги та недоліки їх використання порівняно з іншими інструментами переміщення. Підготуйте текстове повідомлення з поясненням. Збережіть документ у вашій папці у файлі з іменем **завдання 4.2.5**.



6. Проведіть дослідження щодо використання у редакторі **Blender** клавіш додаткової цифрової клавіатури для здійснення тривимірної навігації. Створіть презентацію за результатами дослідження й підготуйте усне повідомлення. Збережіть презентацію у вашій папці у файлі з іменем **завдання 4.2.6**.



Готуємось до вивчення нового матеріалу

1. Як додавати тривимірні примітиви до створюваного проєкту?
2. Як змінити розміри об'єкта та його положення на сцені?
3. Як переглянути створену модель з потрібної точки огляду?



4.3. РЕДАГУВАННЯ ФОРМИ ТА ВИГЛЯДУ ТРИВИМІРНИХ ОБ'ЄКТІВ

У цьому пункті йтиметься про:

- ▶ копіювання, виділення та групування об'єктів моделі;
- ▶ змінення значень властивостей об'єктів у меню останньої операції;
- ▶ змінення форми об'єктів у режимі редагування;
- ▶ застосування кольору і текстури до тривимірних об'єктів.



Пригадайте

● Які ви знаєте способи копіювання і вставлення об'єктів? ● Які ви знаєте способи виділення групи об'єктів? ● З якою метою виконується групування об'єктів у векторних графічних редакторах? Як виконати цю операцію?

Під час створення 3D-моделей може виникати потреба у використанні однакових об'єктів. Наприклад, у моделі автомобіля – чотири однакових колеса, а в моделі будинку – кілька однакових вікон. Щоб не створювати нові об'єкти і налаштовувати значення властивостей кожного з них окремо, можна **скопіювати** попередньо створений об'єкт.

Копіювання виділеного об'єкта і вставлення копії на сцену можна виконати кількома способами:

- вибрати в меню кнопки **Об'єкт** команди **Копіювати Об'єкти** / **Вставити об'єкти**;
- натиснути сполучення клавіш **Ctrl + C** / **Ctrl + V**;
- вибрати команди **Копіювати Об'єкти** / **Вставити об'єкти** в контекстному меню об'єкта.

Ім'я скопійованого об'єкта в області **Структуратор** буде таким самим, як у початкового, але до нього буде додано номер копії, наприклад *Сфера.001*.

Опорна точка копії буде розміщена там само, де і початкового об'єкта, тому одразу після вставлення новий об'єкт потрібно перемістити в інше місце.

У редакторі **Blender** можна одночасно виконувати операції з кількома об'єктами. Для цього потрібно виділити групу об'єктів, використавши один зі способів:

- вибрати кнопку **Вибір коробкою**  та виділити групу об'єктів протягуванням прямокутної області у **Вікні 3D-огляду**;
 - вибирати на сцені об'єкти по одному за натиснутої клавіші **Shift**;
 - вибирати в області **Структуратор** імена об'єктів по одному за натиснутої клавіші **Ctrl**
- та інші.

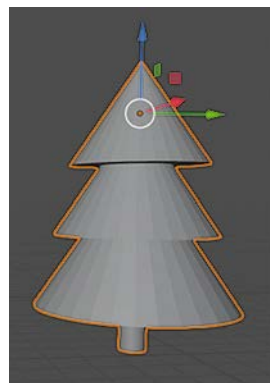
Щоб зняти виділення, достатньо вибрати будь-яку точку у **Вікні 3D-огляду**.

Виділену групу об'єктів можна **згрупувати** (об'єднати) для отримання єдиного об'єкта. Таке групування є засобом запобігання необережним діям користувача, які можуть спотворити вже розроблену частину моделі.

Для групування об'єктів слід:

1. Виділити групу об'єктів, які потрібно об'єднати.
2. Виконати **Об'єкт** ⇒ **Сполучення** або вибрати команду **Сполучення** в контекстному меню виділеної групи.

На малюнку 4.29 зображено об'єкт, який утворився після групування трьох конусів і циліндра.



Мал. 4.29.
Згрупований об'єкт

Утворений об'єкт матиме ім'я та опорну точку останнього з виділених об'єктів. *Зверніть увагу:* розгрупування об'єднаного об'єкта в редакторі **Blender** не передбачено. Відмовитися від небажаного групування можна лише скасуванням операції (**Ctrl + Z**).

ЗМІНЕННЯ ЗНАЧЕНЬ ВЛАСТИВОСТЕЙ ОБ'ЄКТІВ У МЕНЮ ОСТАННЬОЇ ОПЕРАЦІЇ

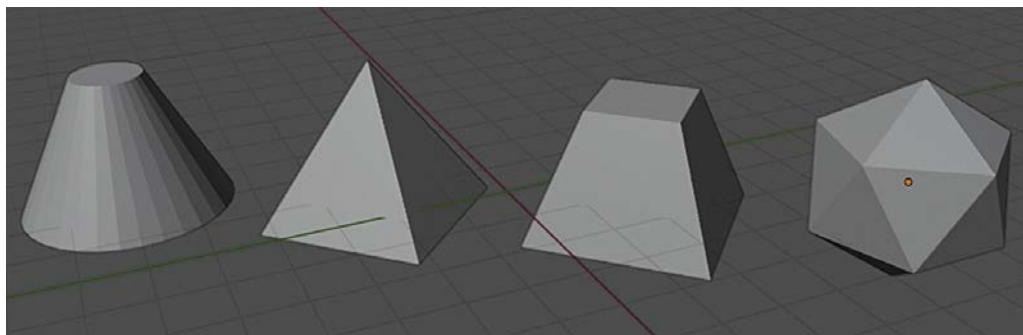


Поміркуйте

- Зважаючи на те, що поверхні тривимірних об'єктів у графічному редакторі складаються з трикутників і багатокутників, як можна перетворити модель конуса на модель піраміди?
- Значення яких властивостей моделі циліндра можна змінити, щоб отримати модель прямокутного паралелепіпеда?

У попередньому пункті ми використовували тривимірні примітиви, форма яких була встановлена за замовчуванням. Але їх недостатньо для створення моделей складних об'єктів. Тому в редакторі **Blender** є засоби для змінення форми існуючих тривимірних примітивів.

Наприклад, змінюючи значення властивостей тривимірних примітивів, можна утворювати тіла, яких немає у стандартному наборі примітивів редактора **Blender**. Так, із примітива конус можна утворити зрізаний конус, піраміду, зрізану піраміду, а зі Сфери Іко – правильний 20-гранник ікосаедр (мал. 4.30) та інше.



Мал. 4.30. Зрізаний конус, піраміда, зрізана піраміда, ікосаедр

Одним зі способів змінення форми тривимірного примітива є використання **меню останньої операції**.

Меню останньої операції з'являється в нижній лівій частині Вікна 3D-огляду одразу після вставлення об'єктів і після виконання операцій перетворення. За замовчуванням меню згорнуто й містить лише назву виконаної операції, наприклад **> Додати циліндр**. Якщо відкрити його вибором кнопки **>**, то можна побачити і змінити значення властивостей об'єкта.

Меню останньої операції відрізняється для різних примітивів та операцій. Вигляд меню останньої операції вставлення Сфери UV та Конуса наведено на малюнку 4.31.



Мал. 4.31. Меню останньої операції вставлення Сфери UV (ліворуч) і Конуса (праворуч)

У меню останньої операції можна встановити значення деяких властивостей залежно від типу вставленого примітива. Наприклад, для Сфери UV можна вказати *кількість кілець*, на які поділена поверхня сфери, *кількість сегментів* у кожному кільці та *радіус* сфери. А для Конуса можна задати кількість точок, що визначають його основу (*Вершини*), висоту конуса (*Глибина*) і радіуси його основ. Якщо радіус верхньої основи (*Радіус 2*) відрізнятиметься від 0, то буде утворено зрізаний конус. А якщо зменшити кількість вершин, то буде утворено піраміду або зрізану піраміду.

Зверніть увагу. Меню останньої операції для вставленого об'єкта можна закрити й відкрити знову натисканням клавіші **F9**. Але після виконання наступної операції перетворення або додавання чергового об'єкта повернутися до попереднього меню та змінити раніше створений об'єкт неможливо.

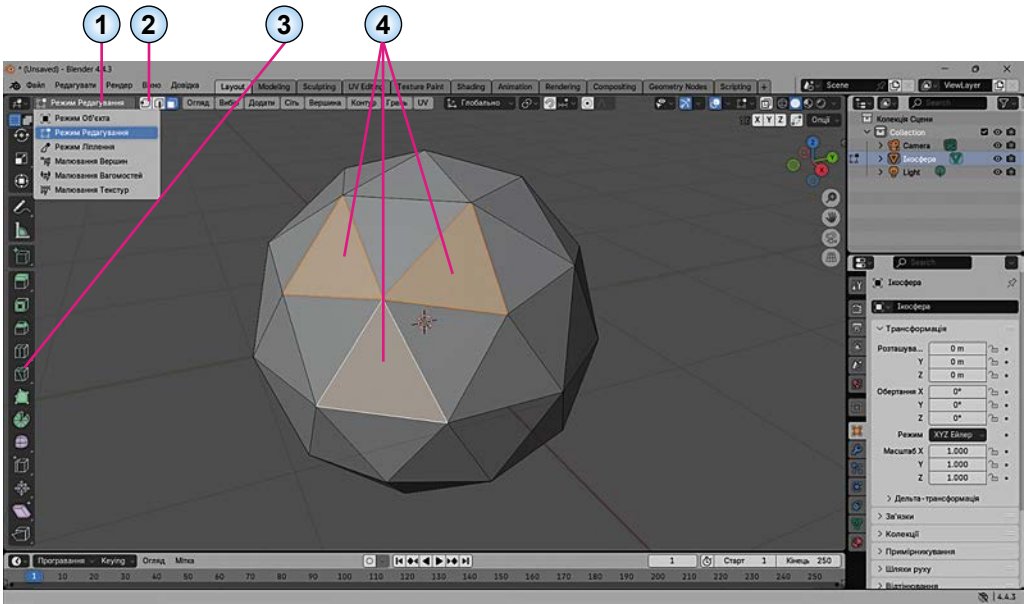
ЗМІНЕННЯ ФОРМИ ОБ'ЄКТІВ У РЕЖИМІ РЕДАГУВАННЯ



Поміркуйте

- Уявіть пляму рідкої фарби на плоскій поверхні. Яка фігура буде утворена, якщо занурити у фарбу кінчик голки та витягнути його за межі плями? Яка фігура буде утворена, якщо занурити у фарбу частину нитки й витягнути її за межі плями? Яка фігура була б утворена, якби вдалося підняти пляму над поверхнею і спостерігати за фарбою, що стікає донизу?

Опрацювання об'єктів у **Вікні 3D-огляду** редактора **Blender** може виконуватись у кількох режимах. За замовчуванням встановлено **Режим Об'єкта**. Усі операції в цьому режимі виконуються з об'єктом як із єдиним цілим. Для опрацювання у вибраного об'єкта окремих елементів його поверхні потрібно встановити для нього **Режим Редагування**, використавши список вибору режимів. На малюнку 4.32 наведено вигляд вікна редактора **Blender** у режимі редагування об'єкта.



Мал. 4.32. Вікно редактора **Blender** у режимі редагування об'єкта:

- 1 – список вибору режимів роботи; 2 – кнопки вибору елементів поверхні об'єкта;
3 – додаткові інструменти; 4 – грані, вибрані на поверхні об'єкта

У режимі редагування змінюється спосіб відображення вибраного об'єкта і стають можливими вибір та перетворення окремих або вершин, або ребер, або граней об'єкта. Для визначення виду елементів використовують кнопки (мал. 4.32, 2) **Вибір вершин** , **Вибір ребер** ,

Вибір граней , розміщені праворуч від списку вибору режиму (мал. 4.32, 1). У лівій частині **Вікна 3D-огляду** з'являються додаткові інструменти (мал. 4.32, 3) для виконання операцій з вершинами, ребрами або гранями об'єкта.

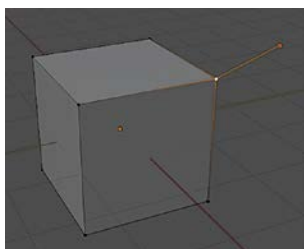
Зверніть увагу: режим редагування буде застосований лише до того об'єкта / об'єктів, які були виділені перед вибором цього режиму. Якщо потрібний об'єкт не був виділений, то потрібно переключитися в режим об'єкта, виділити об'єкт і знову вибрати режим редагування. Швидко переключитися між режимом об'єкта і режимом редагування можна, натиснувши клавішу **Tab** на клавіатурі.

Одним із додаткових інструментів для змінення форми об'єктів є кнопка **Видавити регіон** , яка призначена для виконання операції **екструдуювання** (лат. *extrusio* – виштовхування).

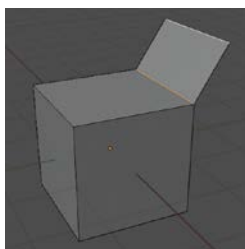
Екструдуювання – це процес змінення форми об'єкта шляхом видавлювання окремих елементів його поверхні (або вершин, або ребер, або граней) з утворенням нових вершин, ребер і граней.

Якщо екструдувати вершину об'єкта, то буде отримано нову вершину, що з'єднана ребром з початковою (мал. 4.33, а). Екструдуювання

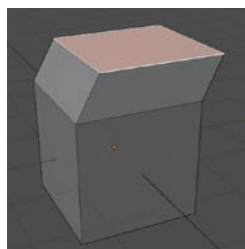
ребра приведе до утворення нової грані, одним з ребер якої буде початкове ребро, а протилежне – рівне за довжиною та паралельне йому (мал. 4.33, б). Під час екструдувannya грані буде отримано одну нову грань, що рівна і паралельна початковій, та кілька граней, що з'єднують початкову грань і паралельну їй (мал. 4.33, в).



а)



б)

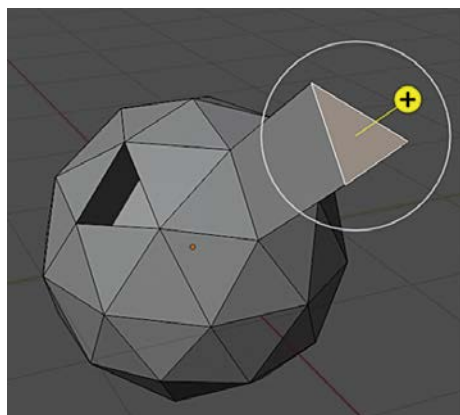


в)

Мал. 4.33. Результат екструдувannya вершини (а), ребра (б) і грані (в) куба

Для виконання операції **екструдувannya** слід:

1. Вибрати в режимі редагування елемент поверхні об'єкта, який потрібно екструдувати, використовуючи інструмент **Вибір коробкою** .
2. Вибрати кнопку **Видаввити регіон** .
3. Перетягнути маркер жовтого кольору  (мал. 4.34) для видавлювання в напрямку жовтої лінії або перемістити область усередині білого кола для вільного видавлювання елементів у будь-якому напрямку.
4. Вибрати інший інструмент або вільну точку **Вікна 3D-огляду**.



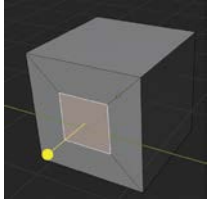
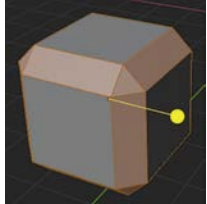
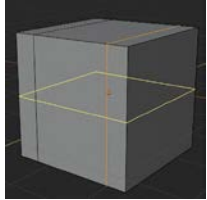
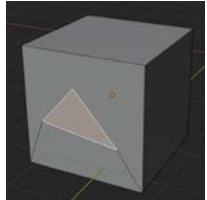
Мал. 4.34. Результат екструдувannya двох граней Сфери Іко

Елементи поверхні можна як виштовхувати назовні, так і заглиблювати всередину об'єкта.

Кілька додаткових інструментів режиму редагування призначені для утворення нових граней на поверхні об'єкта. У таблиці 4.4 наведено опис цих інструментів.

Таблиця 4.4

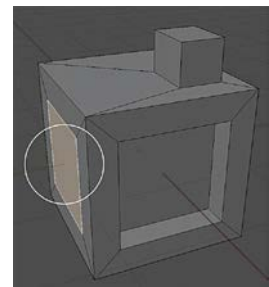
Інструменти для утворення нових граней на поверхні об'єкта

Інструмент	Опис	Приклад
Вставити грані 	Поділяє виділену грань на частини. Утворює одну грань, сторони якої паралельні ребрам виділеної грані, та ще кілька граней, у яких одне з ребер є ребром початкової грані, а протилежне – ребром нової грані	
Фаска 	«Зрізає» ребра та кути об'єкта, утворюючи ефект скосу (фаски)	
Замкнутий розріз 	«Розрізає» об'єкт на частини, поділяючи всі грані, через які проходить розріз	
Ніж 	Поділяє грань на частини, визначені ламаною, вершини якої вказує користувач	

Наприклад, щоб розділити грань об'єкта на частини, використовуючи кнопку **Вставити грані**  , слід:

1. Вибрати в режимі редагування грань об'єкта.
2. Вибрати кнопку **Вставити грані**  .
3. Навести вказівник на лінію кола, що утворилося.
4. Перемістити вказівник у напрямку центра кола із затиснутою лівою кнопкою миші.
5. Відпустити ліву кнопку миші, коли утворені грані набудуть потрібних розмірів.

Отримані в такий спосіб грані можна видаляти натисканням клавіші **Delete**, переміщувати, екструдувати і виконувати з ними інші перетворення (мал. 4.35).



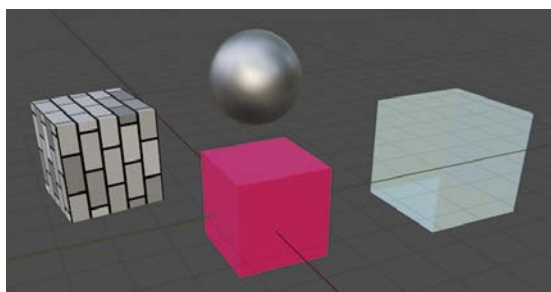
Мал. 4.35.
Утворення нових граней на гранях куба та результат їх екструдування



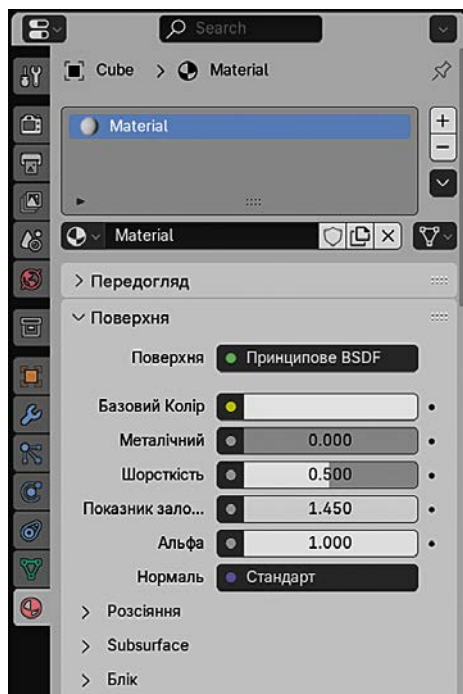
Пригадайте

- Як вибрати або змінити колір об'єктів під час роботи у відомих вам графічних редакторах?
- Які ви знаєте колірні моделі для кодування кольорів? Які їхні базові компоненти?

У навколишньому світі кожен об'єкт створено з певного матеріалу – металу, скла, деревини, каменю, хутра та іншого. Ми розрізняємо ці матеріали за їхнім кольором, блиском, шорсткістю тощо. У тривимірній графіці також передбачено, що поверхня кожного об'єкта вкрита певним матеріалом (мал. 4.36). Властивостями матеріалів є їхні ім'я, колір або малюнок, коефіцієнт прозорості, коефіцієнт відбиття світла та інші.



Мал. 4.36. Різні матеріали на поверхні об'єктів



Мал. 4.37. Вкладка **Матеріал** області **Властивості** об'єкта **Cube**

У редакторі **Blender** для відображення матеріалів на поверхні об'єкта потрібно ввімкнути режим перегляду

Передогляд Матеріалу  або

Рендерене , вибравши відповідну

кнопку в правому верхньому куті **Вікна 3D-огляду**. У режимі **Суцільний** ,

що ввімкнено за замовчуванням, усі об'єкти відображаються у світло-сірому кольорі.

До куба, який після створення нового проєкту міститься у **Вікні 3D-огляду**, застосовано матеріал поверхні з іменем **Material** і значеннями властивостей за замовчуванням. Побачити їх можна на вкладці

Матеріал  області **Властивості**

(мал. 4.37).

Для інших примітивів, які додаються на сцену під час створення моделі, матеріали за замовчуванням не створені. Для створення нового матеріалу потрібно вибрати кнопку **Новий** на вкладці **Матеріал** області **Властивості**. Буде створено новий матеріал з іменем *Material* і кольором та значеннями інших властивостей за замовчуванням. У блоці **Поверхня** з'являться елементи керування для змінення значень властивостей матеріалу.

Після клацання в полі **Базовий колір** відкривається панель (мал. 4.38), на якій можна задати колір і коефіцієнт його прозорості (значення в полі **Alpha**). Колір можна вибрати на кольоровому колі, увести значення його компонентів у кольірних моделях **RGB** або **HSV**, увести 16-ковий код кольору в поле **Hex** або вибрати у вікні редактора потрібний колір інструментом **Піпетка**.



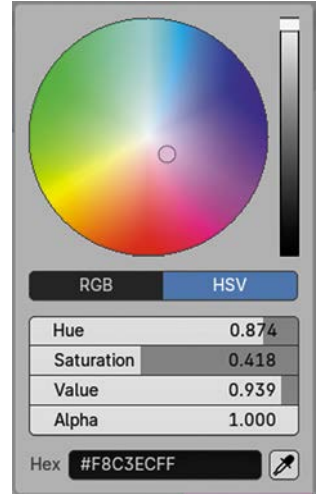
У блоці **Поверхня** можна також установити для матеріалу металевий блиск (значення в полі **Металічний**), шорсткість (значення в полі **Шорсткість**) тощо. У полі зі списком **Поверхня** можна вибрати інші ефекти матеріалу, наприклад ефект скляної (**Скло BSDF**) або глянцевої (**Глянцеві BSDF**) поверхні.

Замість кольору для поверхні об'єкта може бути використана графічна **текстура**. Редактор **Blender** має вбудований набір текстур. Побачити їх список можна, вибравши кнопку



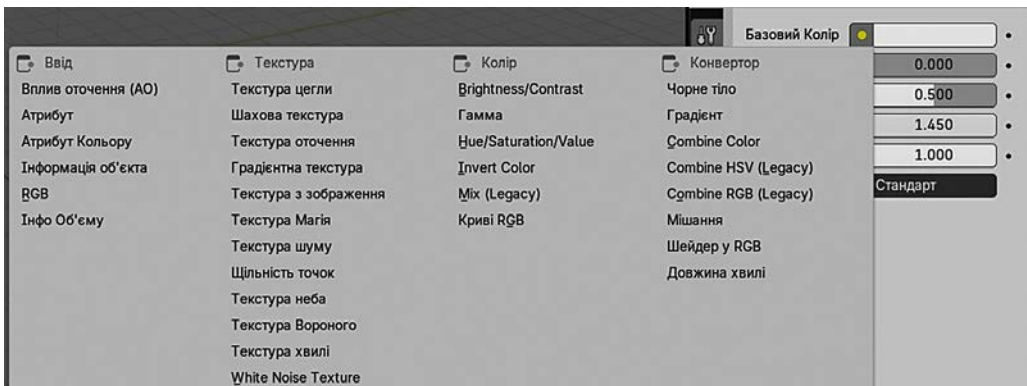
в полі **Базовий**

Колір (мал. 4.39).



Мал. 4.38. Панель вибору кольору

Текстура (лат. *texture* – обробка) – видимий малюнок поверхні об'єкта (текстура деревини, тканини, мармуру або ін.).

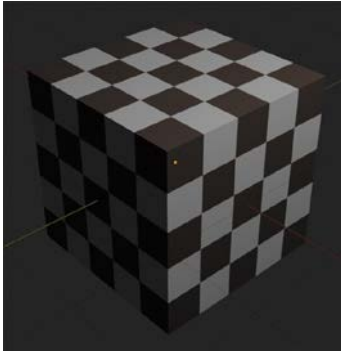


Мал. 4.39. Список вбудованих текстур редактора **Blender**

Як текстуру можна також використати будь-яке растрове графічне зображення. Для цього потрібно вибрати у списку текстуру **Текстура**



з зображення, вибрати кнопку **Відкрити**, що з'явиться у блоці **Поверхня**, вибрати й відкрити зображення. На малюнку 4.40 наведено вигляд куба з текстурою **Шахова текстура** і **Текстура з зображення**.



Мал. 4.40. Вигляд куба з текстурою **Шахова текстура** і **Текстура з зображення**

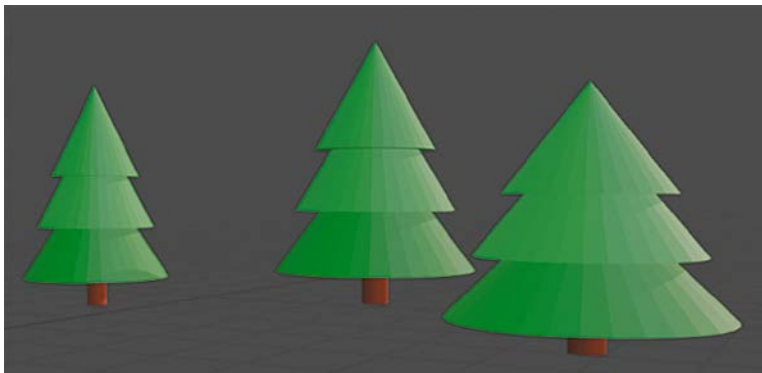


Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання 1. Створіть у редакторі **Blender** модель лісової галявини за зразком (мал. 1).

1. Відкрийте програму **Blender** і створіть новий проєкт.
2. Видаліть зі сцени куб.
3. Уставте на сцену конус.
4. Установіть для конуса на вкладці **Об'єкт** області **Властивості** значення **Масштаб X – 1.5, Y – 1.5**.
5. Увімкніть режим **Передогляд Матеріалу**. Для цього виберіть кнопку цього режиму  у правому верхньому куті **Вікна 3D-огляду**.
6. Створіть матеріал зеленого кольору для конуса. Для цього:
 1. Виберіть в області **Властивості** вкладку **Матеріал** .
 2. Виберіть кнопку **Новий**.



Мал. 1. Зразок до завдання 1

3. Дочекайтеся створення матеріалу та клацніть у блоці **Поверхня** в полі **Базовий колір**.
4. Виберіть на кольоровому колі зелений колір.
7. Скопіюйте конус, натиснувши сполучення клавіш **Ctrl + C**. Уставте копію, натиснувши **Ctrl + V**.
8. Змініть значення властивостей конуса з іменем *Конус.001*, установивши на вкладці **Об'єкт** такі значення:
Розташування Z – 0.9, Масштаб X – 1.2, Y – 1.2.
9. Скопіюйте і вставте третій екземпляр конуса, установіть для нього такі значення властивостей:
Розташування Z – 1.6, Масштаб X – 0.9, Y – 0.9, Z – 0.8.
10. Уставте на сцену об'єкт **Циліндр**.
11. Установіть для циліндра такі значення властивостей:
Розташування Z – -1, Масштаб X – 0.2, Y – 0.2, Z – 0.5.
12. Створіть для циліндра матеріал, установіть коричневий колір.
13. Переконайтеся у правильному розташуванні частин моделі ялинки, переглянувши вигляд сцени з різних ракурсів. За потреби перемістіть неправильно розташовані частини.
14. Згрупуйте об'єкти. Для цього:
 1. Виділіть область сцени, яка буде включати всі створені об'єкти, використовуючи інструмент **Вибір коробкою**.
 2. Виберіть команду **Сполучення** в контекстному меню виділеної групи об'єктів.
15. Створіть ще дві копії отриманої моделі ялинки. Змініть розміри та розташування ялинок за зразком.
16. Збережіть проєкт у вашій папці у файлі з іменем **вправа 4.3.1.blend**.

Завдання 2. Створіть у редакторі **Blender** модель будинку за зразком (мал. 2).

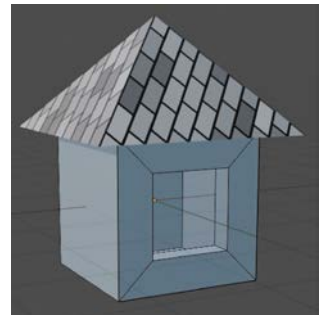
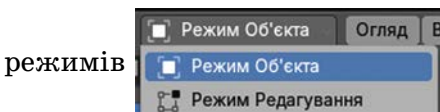
1. Відкрийте програму **Blender** і створіть новий проєкт.
2. Створіть дах будинку. Для цього:
 1. Уставте на сцену конус.

2. Відкрийте меню останньої операції

> Додати Конус

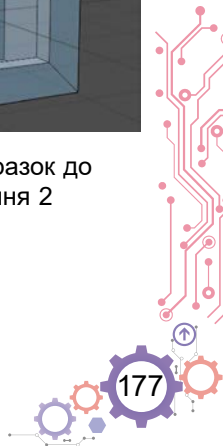
3. Змініть у полях меню значення властивостей: **Вершини – 4, Радіус 1 – 2, Глибина – 1.5, Розташування Z – 1.7, Обертання Z – 45.**

3. Переконайтеся у правильному розташуванні частин моделі.
4. Створіть вікно у стіні будинку. Для цього:
 1. Виберіть куб.
 2. Переключіться в режим редагування, вибравши відповідну команду в списку



Мал. 2. Зразок до завдання 2

3. Виберіть кнопку **Вибір граней**
4. Виберіть передню грань куба.



5. Виберіть кнопку **Вставити грані** .
6. Наведіть вказівник на лінію кола, що утворилося, затисніть ліву кнопку миші та перемістіть вказівник у напрямку центра кола до отримання грані потрібного розміру.
7. Виберіть кнопку **Видавити регіон** .
8. Наведіть вказівник на жовтий маркер, затисніть ліву кнопку миші та перемістіть вказівник у напрямку центра куба, щоб утворити ефект товщини стіни.
9. Видаліть екструдовану грань. Для цього натисніть клавішу **Delete** і виберіть команду **Грані** в меню **Видалення**, що відкрилося.
5. Переключіться в режим об'єкта.
6. Увімкніть режим **Передогляд Матеріалу**.
7. Створіть матеріал для даху будинку.
8. Виберіть **Текстуру цегли** для створеного матеріалу. Для цього:
 1. Виберіть кнопку  в полі **Базовий Колір**.
 2. Виберіть рядок **Текстура цегли**.
9. Створіть матеріал і застосуйте відповідний колір для стін будинку.
10. Збережіть проєкт у вашій папці у файлі з іменем **вправа 4.3.2.blend**.



Найважливіше в цьому пункті

У редакторі **Blender** можна виконувати копіювання і вставлення об'єктів різними способами, аналогічно до інших програм.

Для одночасного виконання операцій з кількома об'єктами можна виділити групу об'єктів, використавши різні способи.

Виділену групу об'єктів можна **згрупувати** для отримання єдиного об'єкта. Таке групування є засобом запобігання необережним діям користувача, які можуть спотворити вже розроблену частину моделі.

У редакторі **Blender** є засоби для змінення форми наявних тривимірних примітивів. Змінюючи значення властивостей тривимірних примітивів у меню останньої операції, можна утворювати тіла, яких немає у стандартному наборі примітивів редактора. Наприклад, із примітива конус можна утворити зрізаний конус, піраміду, зрізану піраміду.

Форму вибраного об'єкта також можна змінювати в режимі редагування. У цьому режимі операції можна виконувати з окремими вершинами, ребрами або гранями об'єкта.

Екструдування – це процес змінення форми об'єкта шляхом видавлювання окремих елементів його поверхні (або вершин, або ребер, або граней) з утворенням нових вершин, ребер і граней.

Кілька додаткових інструментів режиму редагування призначені для утворення нових граней на поверхні об'єкта.

У тривимірній графіці передбачено, що поверхня кожного об'єкта вкрита певним матеріалом. Властивостями матеріалів є їхні ім'я, колір або малюнок, коефіцієнт прозорості, коефіцієнт відбиття світла та інші.

У редакторі **Blender** для відображення матеріалів на поверхні об'єкта потрібно увімкнути режим перегляду **Передогляд Матеріалу** або **Рендерене**. Побачити і створити матеріал для об'єкта, вибрати колір або текстуру можна на вкладці **Матеріал** області **Властивості**.



Дайте відповіді на запитання

1. Якими способами можна копіювати і вставляти об'єкти в редакторі тривимірної графіки?
2. Якими способами можна виділяти групи об'єктів у редакторі тривимірної графіки?
3. Як об'єднати виділену групу об'єктів у один об'єкт?
4. Які способи можна використати для змінення форми тривимірного примітива?
5. У чому полягає операція екструдування?
6. Як змінити колір поверхні об'єктів у редакторі **Blender**?
7. Як застосувати до поверхні об'єктів текстуру з готових зразків і з графічного файлу?



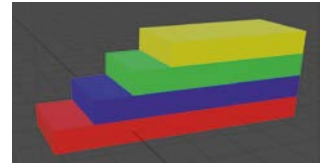
Обговоріть і зробіть висновки

1. Які переваги надає об'єднання групи об'єктів у один об'єкт?
2. Які операції перетворення можна виконувати зі згрупованим об'єктом?
3. Із графічних примітивів яких видів можна отримати модель прямокутного паралелепіпеда? Які значення властивостей потрібно застосувати для перетворення?
4. У чому схожість і відмінність операцій переміщення та екструдування граней об'єкта?

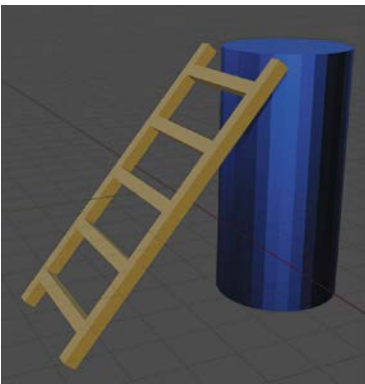


Виконайте завдання

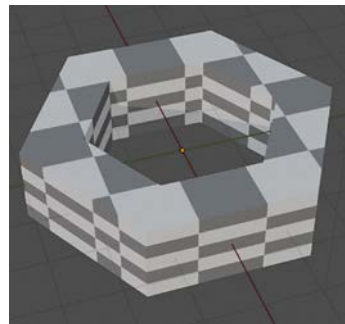
1. Створіть тривимірну модель сходинок за зразком (мал. 4.41). Збережіть проект з іменем **завдання 4.3.1.blend**.
2. Створіть тривимірну модель діжки і драбини за зразком (мал. 4.42). Застосуйте до поверхні діжки ефект **Металічний**. Об'єднайте елементи моделі драбини. Збережіть проект з іменем **завдання 4.3.2.blend**.
3. Створіть тривимірну модель за зразком (мал. 4.43). Використайте для матеріалу поверхні **Шахову текстуру**. Збережіть проект з іменем **завдання 4.3.3.blend**.



Мал. 4.41. Модель сходинок



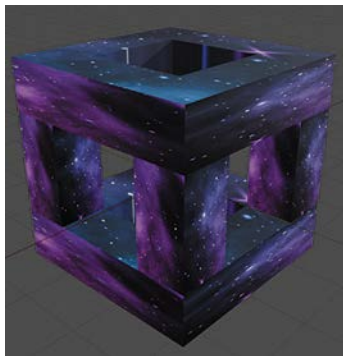
Мал. 4.42. Модель діжки та драбини



Мал. 4.43. Зразок до завдання 3



4. Створіть тривимірну модель за зразком (мал. 4.44). Використайте для матеріалу поверхні **Текстуру з зображення**. Файл із зображенням знайдіть в інтернеті або використайте файл **Розділ 4\Пункт 4.3\завдання 4.3.4.webp**. Збережіть проєкт з іменем **завдання 4.3.4.blend**.



Мал. 4.44. Зразок до завдання 4



5. Дослідіть, використовуючи довідкову систему, особливості різних способів для вирівнювання об'єктів у редакторі 3D-графіки. Зробіть висновки, які переваги дає використання кожного з інструментів. Підготуйте текстове повідомлення з поясненням особливостей використання засобів вирівнювання. Збережіть документ у вашій папці у файлі з іменем **завдання 4.3.5**.



Практична робота № 6. «Створення 3D-моделей»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання. Створіть у редакторі тривимірної графіки модель пінгвіна за зразком (мал. 4.45).

1. Запустіть редактор тривимірної графіки.
2. Додайте примітиви, встановіть для них відповідні розміри, форму та розташування.
3. Застосуйте до об'єктів відповідні кольори.
4. Збережіть проєкт з іменем **практична 6.blend**.



Мал. 4.45. Зразок до практичної роботи № 6



Готуємось до вивчення нового матеріалу

1. Як утворюються анімовані зображення?
2. Які існують види анімації?
3. У файлах якого формату може зберігатися анімація?



4.4. СТВОРЕННЯ 3D-АНІМАЦІЇ

У цьому пункті йтиметься про:

- ▶ етапи підготовки до створення 3D-анімації;
- ▶ створення та редагування ключових кадрів анімації;
- ▶ рендеринг і збереження анімації.

ПІДГОТОВКА ДО СТВОРЕННЯ АНІМАЦІЇ В РЕДАКТОРІ BLENDER



Пригадайте

- Що таке *анімація*?
- Які ви знаєте види анімації? Які ви знаєте види комп'ютерної анімації?
- У чому полягає технологія *морфінг*, яка використовується в комп'ютерній анімації?

Комп'ютерна 3D-анімація широко використовується в рекламі, у кіноіндустрії, комп'ютерних іграх та іншому. Ви вже знаєте, що сутність анімації полягає в тому, що перед очима людини з певною швидкістю змінюються зображення, які мають відмінності у вигляді та положенні об'єктів. У редакторі **Blender** можна анімувати 3D-моделі та зберігати результати у форматі відео.

Ви вже створювали GIF- та WebP-анімацію у графічному редакторі **Krita** й в онлайн-редакторах і знаєте, що анімація в них утворюється з окремих графічних зображень, кожне з яких демонструється протягом певного часу. Для створення анімації в редакторі **Blender** використовується інша технологія – **морфінг**, яка базується на **ключових кадрах**.

У ключових кадрах значення властивостей об'єктів 3D-моделі (розмір, положення, кут обертання, колір поверхні тощо) мають бути точно такими, які встановлює розробник анімації. А редактор **Blender** автоматично визначає вигляд і положення цих об'єктів у проміжних кадрах між ключовими.

Наприклад, у першому кадрі анімації куб повинен мати такі значення властивостей: довжина сторони – 0.2 м , координати опорної точки – $(0; -2; 0)$, колір поверхні – *жовтий*. У кадрі з номером 20 значення властивостей цього куба мають стати: довжина сторони – 1 м , координати опорної точки – $(0; 2; 0)$, колір поверхні – *червоний*. У проміжних кадрах з 2 до 19 значення властивостей куба обчислюються автоматично. Вигляд куба в кадрах з номерами 1, 7, 14 та 20 наведено на малюнку 4.46.



Мал. 4.46. Вигляд куба в кадрах анімації 1, 7, 14 та 20

До початку створення анімації потрібно виконати попередню підготовку:

- розробити сценарій анімації;
- визначити її тривалість;
- запланувати ключові кадри: визначити їхню кількість, номери, положення та вигляд об'єктів у цих кадрах;
- створити всі об'єкти майбутньої анімації;
- налаштувати джерела світла, положення камери й виконати іншу підготовчу роботу.

СТВОРЕННЯ ТА РЕДАГУВАННЯ КЛЮЧОВИХ КАДРІВ

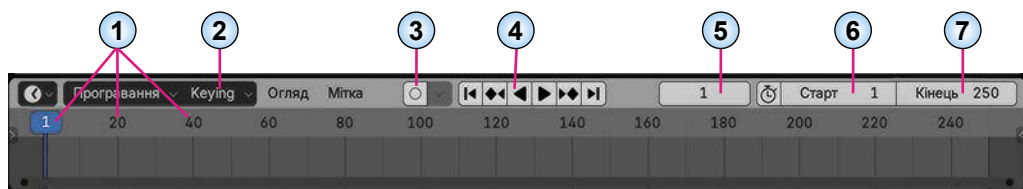


Пригадайте

- Які області відображаються за замовчуванням у вікні редактора **Blender**?
- Як можна змінити значення властивостей тривимірного об'єкта в редакторі **Blender**?

Для створення та перегляду анімації використовується область **Шкала часу** , яка розміщена в нижній частині вікна редактора

Blender (мал. 4.47). Для зручності роботи можна збільшити висоту цієї області, перетягнувши вгору її верхню межу.



Мал. 4.47. Область **Шкала часу**:

- 1 – номери кадрів; 2 – список **Keying**; 3 – перемикач **Автоключування**;
4 – панель керування переглядом анімації; 5 – поле **Поточний кадр**;
6 – поле **Старт**; 7 – поле **Кінець**

У цій області відображаються номери кадрів анімації та можна вибирати кадри, що стануть ключовими. Номери першого та останнього кадрів анімації виведені в полях **Старт** (мал. 4.47, 6) і **Кінець** (мал. 4.47, 7), номер поточного кадру – у полі **Поточний кадр** (мал. 4.47, 5).

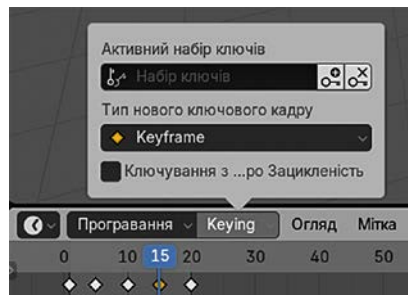
Поточний кадр позначено вертикальним блакитним відрізком, його номер зазначено на блакитному фоні. Зробити поточним інший кадр можна, клацнувши на смужі з номерами кадрів, або увівши потрібний номер у поле **Поточний кадр**, або перетягнувши прямокутник з номером поточного кадру.

За замовчуванням для анімації надається 250 кадрів. Але залежно від запланованої тривалості анімації може бути потрібна інша кількість кадрів. У редакторі **Blender** швидкість відтворення анімації за замовчуванням становить 24 кадри за секунду. Тому, якщо планується створити анімацію тривалістю, наприклад, 5 секунд, то знадобиться $24 \times 5 = 120$ кадрів. А для тривалості у 30 секунд потрібно $24 \times 30 = 720$ кадрів. Для змінення кількості кадрів у спланованій анімації слід увести потрібне значення в поле **Кінець**.

Анімація кожного об'єкта, розміщеного на сцені, створюється окремо. Щоб створити ключовий кадр для вибраного у **Вікні 3D-огляду** об'єкта, слід:

1. Установити в області **Шкала часу** поточним той кадр, який потрібно зробити ключовим.
2. Установити для вибраного об'єкта такі значення властивостей, які сплановані у ключовому кадрі.
3. Відкрити список **Keying** (англ. *keying* – ключування) і вибрати кнопку **Вставити ключовий кадр** у групі **Активний набір ключів**.

Після цього в поточному кадрі шкали часу з'явиться маркер (мал. 4.48), який свідчить про те, що кадр став ключовим в анімації цього об'єкта і в цьому кадрі зафіксовано значення властивос-



Мал. 4.48. Маркери у ключових кадрах і список **Keying**

тей об'єкта. В області **Шкала часу** одночасно відображаються маркери ключових кадрів лише одного з об'єктів анімації.

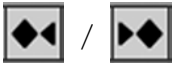
Наведений алгоритм створення ключового кадру потрібно повторити для кожного ключового кадру кожного з об'єктів анімації. Щоб прискорити розробку анімації, можна вибрати перемикач **Автоключування**.



У такому разі кожен кадр, у якому користувач змінюватиме значення властивостей об'єкта, автоматично ставатиме ключовим.

За потреби ключовий кадр можна перемістити, вибравши відповідний маркер в області **Шкали часу** і перетягнувши його в інше положення. Встановлені значення властивостей об'єкта будуть збережені, але зміниться номер кадру, у якому ці значення будуть застосовані.

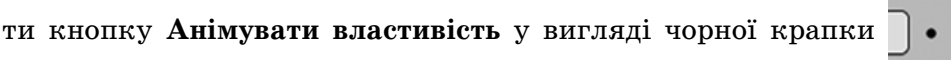
Для змінення значень властивостей об'єкта в наявному ключовому кадрі слід зробити цей кадр поточним і встановити для об'єкта потрібні значення властивостей. Вибір кнопок **Стрибок на Ключкадр**



на панелі керування переглядом анімації приводить до швидкого переходу на попередній / наступний ключовий кадр.

Ключовий кадр можна видалити, вибравши відповідний маркер і натиснувши клавішу **Delete**.

Якщо в анімації планується змінення кольору поверхні об'єкта, то перш ніж установити колір для першого кадру, потрібно указати, що для цієї властивості передбачено анімування. Для цього слід вибрати кнопку **Анімувати властивість** у вигляді чорної крапки



біля правої межі поля **Базовий колір** на вкладці **Матеріали** в області **Властивості**. Після вибору цієї кнопки замість крапки відображатиметься ромб



, а в кожному ключовому кадрі можна буде вказати інший колір поверхні об'єкта.

ПЕРЕГЛЯД, РЕНДЕРИНГ І ЗБЕРЕЖЕННЯ АНІМАЦІЇ



Пригадайте

- У чому сутність операції рендеринг? • Як у редакторі **Blender** змінити налаштування камери та джерел світла? • Як зберегти проєкт у редакторі **Blender**? Як у редакторі **Blender** зберігати 3D-моделі в різних форматах?

Для попереднього перегляду створеної анімації потрібно вибрати кнопку **Відтворення анімації**  на панелі керування переглядом анімації в області **Шкала часу** або натиснути клавішу **Пропуск**. Анімація буде відтворюватися циклічно, тобто після показу останнього кадру відбудеться перехід до першого кадру та показ продовжиться з початку. Для зупинки анімації можна вибрати кнопку  або повторно натиснути клавішу **Пропуск**.

Перед остаточним запуском рендерингу анімації потрібно налаштувати її вигляд:



- через об'єktiv камери – вибрати кнопку **Перемкнути огляд камери**  і налаштувати розташування та фокусну відстань камери;

- з урахуванням джерел світла – увімкнути режим **Рендерене**  і налаштувати розташування й потужність джерел світла.

Коли анімацію створено і попередній перегляд задовольняє користувача, її можна зберегти або як набір графічних файлів із зображенням кожного кадру окремо, або як один файл у форматі відео.

Збереження відбувається під час рендерингу анімації та потребує значного часу і ресурсів комп'ютера.

Перед початком рендерингу анімації потрібно задати папку для збереження файлів і формат вихідного файлу. Для цього слід:

1. Вибрати вкладку **Output (Властивості Виводу)**  в області

Властивості (мал. 4.49).

2. Вибрати у групі **Вивід** кнопку

Прийняти .

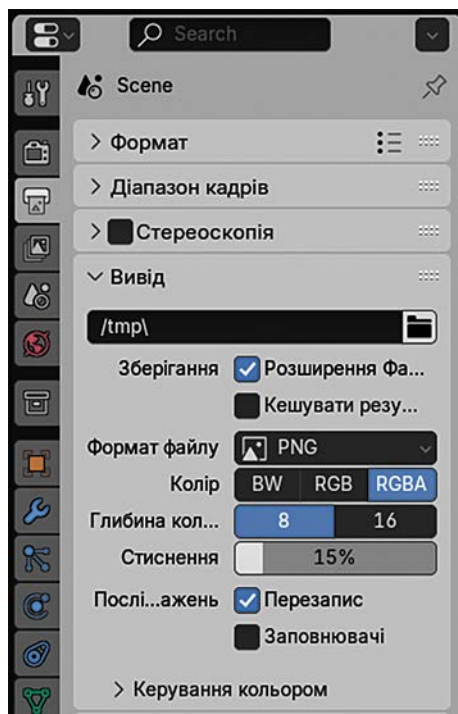
3. Відкрити папку для збереження анімації у вікні **Перегляд файлу Blender**, що відкрилося. За потреби можна створити нову папку, вибравши кнопку **Створити нову теку** .

4. Вибрати кнопку **Прийняти**.

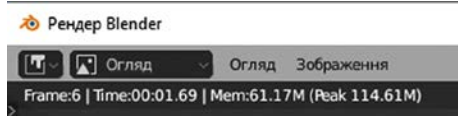
5. Вибрати у списку **Формат файлу** формат графічних файлів або файлу відео для збереження анімації.

За замовчуванням пропонується збереження зображень кожного окремого кадру у файлах формату **PNG**. Для збереження анімації у відеофайлі потрібно вибрати формат **Відео FFmpeg** групи **Фільм**.

Для початку рендерингу потрібно вибрати в головному меню команду **Рендер** ⇒ **Рендер Анімації** або натиснути **Ctrl + F12**. Після цього у вікні **Рендер Blender** буде відображатися процес рендерингу. У верхньому лівому куті можна побачити, який кадр (**Frame**) опрацьовується, скільки витрачено часу та скільки пам'яті використовує процес рендерингу (мал. 4.50).



Мал. 4.49. Вкладка **Output** області **Властивості**



Мал. 4.50. Відомості про хід рендерингу

Після завершення рендерингу у вибраній папці міститиметься відеофайл з анімацією або набір графічних файлів. Ім'я файлу відео складається з номерів початкового та кінцевого кадрів анімації, наприклад **0001-0060.mkv**. Іменами графічних файлів є номери кадрів анімації, наприклад **0016.png**, **0048.png** тощо.

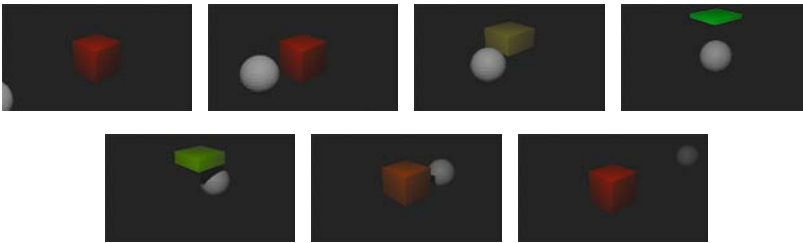


Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання. Створіть анімацію руху сфери (об'єкт **Сфера UV**) та куба (об'єкт **Cube**) тривалістю 3 секунди за таким сценарієм (мал. 1):

1. На початку анімації куб червоного кольору розміщено в центрі кадру під час перегляду через об'єktiv камери, сфера не потрапляє в кадр, розміщена за лівою межею кадру.
2. За першу секунду сфера рухається зліва направо та наближається до нерухомого куба.
3. За другу секунду об'єкт **Cube** піднімається вгору, його розмір по осі Z зменшується в 10 разів, а колір змінюється на зелений. Сфера продовжує рух управо. Після проходження сфери під об'єктом **Cube** він опускається на попередню позицію, збільшується до початкових розмірів і змінює колір на червоний.
4. За третю секунду сфера виходить за межі кадру вправо, куб не змінює положення та кольору.



Мал. 1. Приклади кадрів анімації

Визначимо ключові кадри, у яких можна точно вказати положення, розмір та колір об'єктів. Усього в анімації, що триває 3 секунди зі швидкістю 24 кадри на секунду, $24 \times 3 = 72$ кадри.

Ключові кадри для об'єктів анімації

Час	Кадр	Куб	Сфера UV
0 с	1	Розміщений у центрі кадру. Координати опорної точки (0; 0; 0). Колір червоний. Масштаб по всіх осях 1	Координати опорної точки (0; -7; 0). Не в кадрі
1 с	24	Координати опорної точки (0; 0; 0). Колір червоний. Масштаб по всіх осях 1	Координати опорної точки (0; -3,5; 0).
1,5 с	36	Координати опорної точки (0; 0; 2,5). Колір зелений. Масштаб по осі OZ 0.1	
2 с	48	Координати опорної точки (0; 0; 0). Колір червоний. Масштаб по всіх осях 1	Координати опорної точки (0; 5; 0).



Час	Кадр	Куб	Сфера UV
3 с	72		Координати опорної точки (0; 17; 0)

- Відкрийте програму **Blender** і створіть новий проєкт.
- Додайте на сцену **Сферу UV** та установіть координати її опорної точки (0; -7; 0).
- Установіть на шкалі часу визначену в сценарії кількість кадрів. Для цього уведіть у поле **Кінець** значення **72** і натисніть клавішу **Enter**.
- Перетягніть угору верхню межу області **Шкала часу**, щоб побачити виділені іншим кольором кадри, які увійдуть до анімації.
- Налаштуйте рух куба відповідно до сценарію. Для цього:
 - Виберіть куб у **Вікні 3D-огляду**.
 - Створіть у кадрі 1 перший ключовий кадр для куба, щоб зафіксувати його початкове положення. Для цього у списку **Keying** виберіть кнопку **Вставити ключовий кадр** . *Зверніть увагу* на появу маркера у ключовому кадрі.
 - Зробіть поточним кадр 24, вибравши його номер на шкалі часу або увівши значення **24** в поле **Поточний кадр**.
 - Створіть другий ключовий кадр для куба, щоб зафіксувати його положення перед початком руху вгору. *Зверніть увагу* на появу помаранчевої смуги між маркерами, яка свідчить, що на проміжку між ключовими кадрами значення властивостей об'єкта не змінювалися.
 - Переключіться в режим автоматичного створення ключових кадрів, вибравши перемикач **Автоключування** .
 - Зробіть поточним кадр 36.
 - Установіть для куба в області **Властивості** на вкладці **Властивості об'єкта**:
 - у поле **Z** групи **Розташування** значення **2.5**;
 - у поле **Z** групи **Масштаб** значення **0.1**.*Зверніть увагу* на автоматичну появу маркера ключового кадру в кадрі 36.
 - Створіть четвертий ключовий кадр для куба: кадр 48, значення властивостей **Розташування Z** – 0, **Масштаб Z** – 1.
- Перегляньте анімацію куба. Для цього:
 - Зробіть поточним перший кадр анімації, вибравши кнопку **Стрибок на Кінцеву точку**  на панелі керування переглядом.
 - Виберіть кнопку **Відтворення анімації**  для початку перегляду.
 - Натисніть клавішу **Пропуск** для призупинення перегляду.
- Налаштуйте анімацію кольору куба. Для цього:
 - Увімкніть у **Вікні 3D-огляду** режим перегляду **Рендерене** .
 - Зробіть поточним перший кадр анімації.

3. Виберіть вкладку **Матеріал**  в області **Властивості**.
4. Виберіть кнопку **Анімувати властивість**  біля правої межі поля **Базовий колір**. *Зверніть увагу* на змінення вигляду кнопки.
5. Виберіть червоний як колір поверхні куба.
6. Зробіть поточним другий ключовий кадр, вибравши кнопку **Стрибок на Ключкадр**  на панелі керування переглядом.
7. Виберіть кнопку **Анімувати властивість**  біля правої межі поля **Базовий колір**, щоб зафіксувати червоний колір у кадрі. *Зверніть увагу* на змінення вигляду кнопки.
8. Зробіть поточним третій ключовий кадр і установіть зелений колір куба.
9. Зробіть поточним четвертий ключовий кадр і установіть червоний колір куба.
8. Перегляньте анімацію куба через об'єktiv камери. Для цього:
 1. Виберіть у **Вікні 3D-огляду** кнопку **Перемкнути огляд камери** .
 2. Виберіть в області **Структуратор об'єкт Camera**.
 3. Виберіть в області **Властивості** вкладку **Data** .
 4. Установіть у полі **Фокусна відстань** значення **35 мт**. *Зверніть увагу* на змінення масштабу перегляду в кадрі.
 5. Виберіть в області **Структуратор об'єкт Light**.
 6. Установіть координати опорної точки об'єкта **Light (4; -5; 6)**, щоб були добре освітлені всі грані куба.
 7. Перегляньте анімацію від першого до останнього кадру.
9. Виберіть сферу у **Вікні 3D-огляду**.
10. Налаштуйте ключові кадри анімації сфери відповідно до сценарію.
11. Перегляньте створену анімацію.
12. Збережіть проєкт у вашій папці у файлі з іменем **вправа 4.4.blend**.
13. Виконайте рендеринг і збережіть анімацію. Для цього:
 1. Виберіть вашу папку для збереження анімації, використавши кнопку **Прийняти**  у групі **Вивід** вкладки **Output (Властивості Виводу)**  області **Властивості**.
 2. Виберіть у списку **Формат файлу** команду **Відео FFmpeg** для збереження анімації у форматі відеофайлу.
 3. Виконайте рендеринг, вибравши в головному меню команду **Рендер** ⇒ **Рендер Анімації**. Рендеринг завершиться, коли у верхньому лівому куті вікна **Рендер Blender** буде зазначено **Frame: 72**.
 4. Закрийте вікно **Рендер Blender** після завершення рендерингу.

14. Збережіть проєкт із тим самим іменем і закрийте вікно програми **Blender**.
15. Відкрийте вікно вашої папки, знайдіть файл з іменем **0001-0072.mkv** і перегляньте отримане відео.
16. Закрийте всі відкриті вікна.



Найважливіше в цьому пункті

3D-моделі, створені в редакторі **Blender**, можна анімувати. Для створення анімації в **Blender** використовується технологія **морфінг**, що базується на **ключових кадрах**.

До початку створення анімації в редакторі **Blender** потрібно розробити сценарій анімації, визначити її тривалість, запланувати ключові кадри: визначити їхню кількість, номери, положення та вигляд об'єктів у цих кадрах, створити всі об'єкти, налаштувати джерела світла, положення камери й виконати іншу підготовчу роботу.

Для створення та перегляду анімації використовується область **Шкала часу**.

Загальна кількість кадрів анімації залежить від її запланованої тривалості й від кількості кадрів, які будуть демонструватися за 1 секунду.

Для кожного об'єкта, розміщеного на сцені, анімація створюється окремо, і для кожного потрібно створювати власні ключові кадри. У ключовому кадрі фіксуються значення властивостей об'єкта – його положення, розмір, кут обертання, колір поверхні тощо.

Після завершення створення ключових кадрів можна переглянути анімацію, використовуючи кнопки панелі керування переглядом анімації.

Створену анімацію можна зберегти як набір графічних файлів із зображенням кожного кадру окремо або як один файл у форматі відео. Збереження відбувається під час рендерингу анімації. Перед початком рендерингу потрібно задати папку для збереження файлів і формат вихідного файлу.



Дайте відповіді на запитання

1. Які етапи підготовки до створення анімації?
2. Яка залежність між кількістю кадрів і тривалістю анімації в редакторі **Blender**?
3. Як змінити кількість кадрів анімації в редакторі **Blender**?
4. Як змінити поточний кадр анімації в редакторі **Blender**?
5. Як створити ключовий кадр анімації об'єкта в редакторі **Blender**?
6. Як переглянути анімацію в редакторі **Blender**?
7. Які налаштування потрібно виконувати перед початком збереження анімації на носії даних? Як зробити ці налаштування?
8. У яких форматах можна зберегти анімацію в редакторі **Blender**? Як виконується збереження анімації?



Обговоріть і зробіть висновки

1. Чому, на вашу думку, для кожного об'єкта анімації потрібно створювати окремі ключові кадри, навіть якщо номери ключових кадрів різних об'єктів збігаються?
2. Чи існує аналогія між властивостями ефектів анімації в редакторі презентацій та ключовими кадрами в 3D-анімації? У чому схожість і відмінність?
3. Як залежить вибір ключових кадрів у сценарії анімації від дій об'єктів?

4. У чому схожість та відмінність створення відеоролика та 3D-анімації? Які етапи є спільними, а які унікальними для кожного процесу?

5. Як залежить тривалість рендерингу від складності сцени, кількості кадрів та обраного формату збереження? Як це врахувати під час планування роботи?



Виконайте завдання

1. Створіть анімацію куба тривалістю 2 секунди за таким сценарієм: упродовж першої секунди куб переміщується вздовж осі OY з точки (0; -5; 0) до точки (0; 0; 0), упродовж другої секунди його розміри збільшуються у 2 рази по всіх осях. Збережіть у вашій папці проєкт у файлі **завдання 4.4.1.blend** і файл анімації у форматі **Відео FFmpeg**.

2. Створіть анімацію конуса тривалістю 2 секунди за таким сценарієм: координати опорної точки конуса (0; 0; 0); упродовж першої секунди його розмір збільшується у 2 рази по осі OZ, упродовж другої секунди він обертається на 180° відносно осі OX. Збережіть у вашій папці проєкт у файлі **завдання 4.4.2.blend** і файл анімації у форматі **Відео FFmpeg**.

3. Відкрийте 3D-модель рахівниці з файлу Розділ 4\Пункт 4.4\зразок 4.4.3.blend. Створіть анімацію руху кілець на середній перекладаці рахівниці таким чином, щоб кожне кільце за 0,5 с переміщувалося від одного краю перекладачини до іншого. Збережіть у вашій папці проєкт у файлі **завдання 4.4.3.blend** і файл анімації у форматі **Відео FFmpeg**.



4. Створіть 3D-модель судна та розвідного моста. Створіть анімацію розведення мосту і проходження судна під ним. Вигляд моделі, спосіб розведення, тривалість анімації визначте самостійно. Збережіть проєкт у вашій папці у файлі **завдання 4.4.4.blend** і файл анімації у форматі **Відео FFmpeg**.



5. Створіть анімацію складання сніговика із трьох снігових куль. Тривалість анімації 0,5 с. Збережіть анімацію як набір графічних файлів у форматі PNG та як файл відео. Порівняйте розмір відеофайлу і графічних файлів. Зробіть висновок, чи є залежність між цими розмірами. Таблицю з результатами дослідження та висновки створіть у текстовому файлі. Збережіть файл у вашій папці з іменем **завдання 4.4.5**.



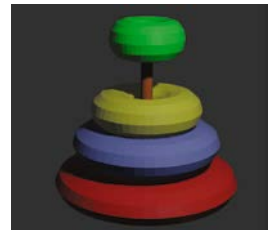
6. Створіть власний сценарій анімації 3D-моделі, що складається з 2–3 об'єктів. Збережіть сценарій у вашій папці в текстовому файлі **завдання 4.4.6**. Створіть об'єкти 3D-моделі та розробіть анімацію на основі вашого сценарію. Збережіть проєкт у вашій папці у файлі **завдання 4.4.6.blend** і файл анімації у форматі **Відео FFmpeg**.



Практична робота № 7. «Створення 3D-анімації»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Запустіть програму **Blender**.
2. Створіть модель іграшки *Пірамідка* за зразком (мал. 4.51).



Мал. 4.51. Зразок до практичної роботи 7

3. Розробіть анімацію тривалістю 3 секунди за таким сценарієм:
 1. На початку анімації в кадрі відображаються стрижень та червоне кільце. Вони залишаються нерухомими упродовж усієї анімації.
 2. Упродовж першої секунди згори опускається і нанизується на стрижень синє кільце.
 3. Упродовж другої секунди згори опускається і нанизується на стрижень жовте кільце.
 4. Упродовж третьої секунди згори опускається і нанизується на стрижень зелене кільце.
4. Збережіть проєкт у вашій папці у файлі з іменем **практична 7.blend**.
5. Виконайте рендеринг анімації та збережіть результат у вашій папці у файлі з іменем **практична 7.mkv**.



Готуємось до вивчення нового матеріалу

1. Які існують способи створення 3D-моделей?
2. У яких галузях діяльності людини використовують тривимірні моделі?
3. Що таке *3D-принтер*? Які об'єкти можуть бути надруковані на 3D-принтері?



4.5. 3D-ДРУК

У цьому пункті йтиметься про:

- ▶ галузі використання 3D-друку;
- ▶ принципи роботи 3D-принтерів;
- ▶ підготовку 3D-моделі до друкування;
- ▶ використання програми-слайсера.

ГАЛУЗІ ВИКОРИСТАННЯ 3D-ДРУКУ. ПРИНЦИП РОБОТИ 3D-ПРИНТЕРІВ



Пригадайте

- Що таке *3D-принтер*?
- Які ви знаєте технології 3D-друку?
- У яких галузях діяльності використовують 3D-друк?

Одним із призначень програм тривимірного моделювання є створення 3D-моделей для виготовлення їхніх твердих копій на 3D-принтерах. Технології виготовлення матеріальних тривимірних об'єктів шляхом накладання послідовних шарів спеціального матеріалу за готовою цифровою 3D-моделлю називають **адитивними технологіями**. Однією з них є **3D-друк**.

3D-друк використовується в різноманітних галузях діяльності людини: у побуті для виготовлення окремих невеликих предметів, у промисловому виробництві для серійного виготовлення продукції різного призначення: сувенірів, декоративних виробів, дрібних деталей та іншого. Існують і специфічні 3D-принтери, які використовують для потреб певної галузі з урахуванням її специфіки, наприклад у будівництві для друкування будинків (мал. 4.52), у медицині – зубних протезів, у військовій справі – деталей для дронів, в індустрії моди – фурнітури для одягу та інших речей.

Ви вже знаєте з 8 класу, що найпоширенішими технологіями, за якими працюють 3D-принтери, є:

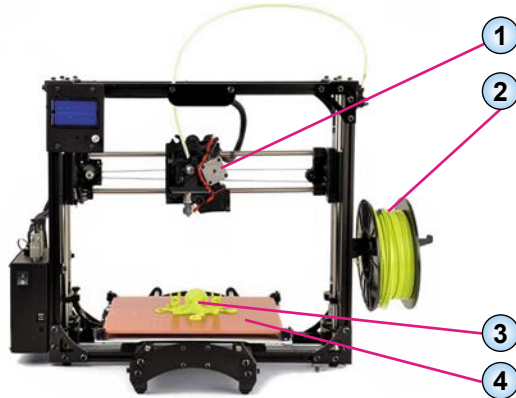
- **SLA** (англ. *Stereolithography apparatus* – стереолітографічна апаратура) – технологія, у якій використовується отвердіння спеціальної смоли під дією лазерного або світлодіодного випромінювання;
- **SLS** (англ. *Selective Laser Sintering* – вибіркове лазерне спікання) – технологія, у якій використовується запікання (плавлення та отвердіння) спеціального порошку під дією лазерного променю;
- **FDM / FFF** (англ. *Fusing Deposition Modeling* – моделювання методом наплавлення, англ. *Fused Filament Fabrication* – виготовлення плавленням ниток) – технологія, у якій використовується отвердіння попередньо розплавлених пластичних мас – **філаменту** (англ. *filament* – нитка).

Залежно від типу принтера та призначення моделі об'єкта використовують різні матеріали для 3D-друкування: полімерні нитки, силікон, бетон, акрил, гіпс, метал, нейлон, стовбурові клітини та інше.

Найбільш розповсюдженими у дрібному виробництві та в закладах освіти є **FDM-принтери** (мал. 4.53).



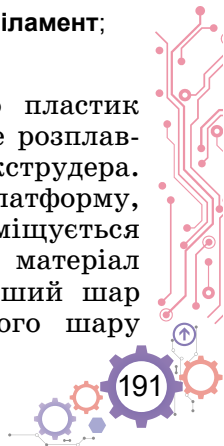
Мал. 4.52. Будівельний 3D-принтер



Мал. 4.53. Складові 3D-принтера типу **FDM**:

- 1 – друкувальна головка – **екструдер**; 2 – пластик для друкування – **філамент**;
3 – об'єкт, що друкується; 4 – робоча платформа

Принцип роботи **FDM-принтерів** полягає в тому, що пластик з котушки подається у друкувальну головку (**екструдер**), де розплавлюється під дією нагрівача та видавлюється через сопло екструдера. Розігріта маса потрапляє у відповідне місце на робочу платформу, де охолоджується і стає знову твердою. Екструдер переміщується паралельно робочій платформі та залишає розплавлений матеріал у певному місці відповідно до 3D-моделі, утворюючи перший шар друкованого виробу. Після завершення друкування одного шару



друкувальна головка піднімається вгору й починається друк наступного шару, у якому матеріал наноситься на попередній шар. І так відбувається наплавлення матеріалу шар за шаром.



Цікаві факти з історії

Ідея пошарового створення тривимірних об'єктів з'явилася ще в XIX ст. Картографи намагалися створювати тривимірні топографічні карти з рельєфом земної поверхні шляхом накладання на папір шарів воску чи картону. Таким чином створювалась об'ємна фігура, хоча й не автоматизованим способом.

Винахідником 3D-друку вважається американський дослідник **Чарльз Халл** (нар. 1939 р.) (мал. 4.54), який 1984 року розробив технологію стереолітографії (**SLA**). 1986 року він заснував компанію **3D Systems**, яка через рік випустила перший комерційний 3D-принтер **SLA-1**.



Мал. 4.54. Чарльз Халл

Ідею технології 3D-друку методом наплавлення **FDM** запропонував 1989 року американський винахідник **Скотт Крамп** (нар. 1957 р.). Заснована ним компанія **Stratasys** 1992 року почала виробництво 3D-принтерів **Stratasys 3D Modeler** за технологією **FDM**.

ПІДГОТОВКА 3D-МОДЕЛІ ДЛЯ ДРУКУВАННЯ



Пригадайте

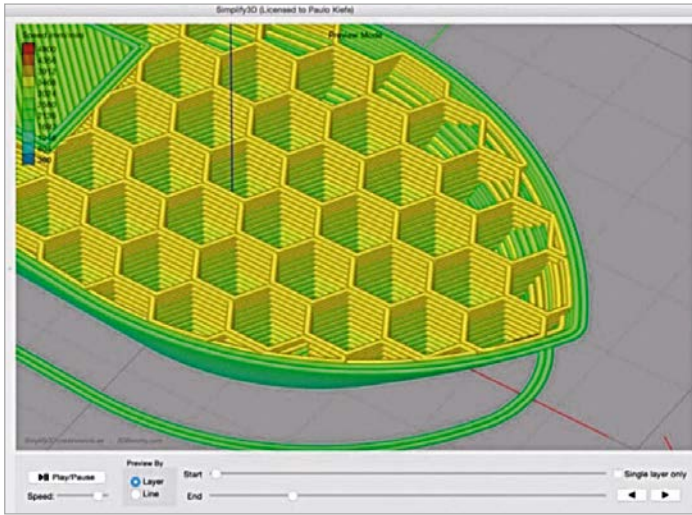
● У яких форматах можна зберігати файли 3D-моделі? У які формати можна експортувати 3D-моделі? ● Який принцип створення тривимірної моделі на 3D-принтері типу **FDM**?

Під час створення 3D-моделі важливо враховувати особливості процесу друкування і не припускати помилок, які можуть зіпсувати результат або призвести до значних витрат матеріалів. Тому важливо врахувати такі рекомендації:

- забезпечити в 3D-моделі точне співвідношення розмірів її окремих елементів;
- забезпечити товщину стінок у моделі не менше від діаметра сопла друкувальної головки;
- робити внутрішні порожнини в середині великих суцільних частин моделі;
- не створювати в моделі дуже дрібних деталей та отворів, мінімальний розмір визначається технологією друкування.

Створену в 3D-редакторі модель потрібно експортувати у файл одного з форматів, який зберігає опис моделі, наприклад: **.STL**, **.GLB**, **.OBJ**, **.FBX**.

3D-принтеру для роботи потрібно мати координати точок, за якими буде переміщуватися його друкувальна головка (екструдер) або промінь лазера. Ці координати визначаються у спеціальній програмі – **слайсері** (англ. *slicer* – нарізка). Слайсер «нарізає» модель шарами (мал. 4.55) і визначає маршрут переміщення друкувальної головки принтера для утворення кожного шару об'єкта.



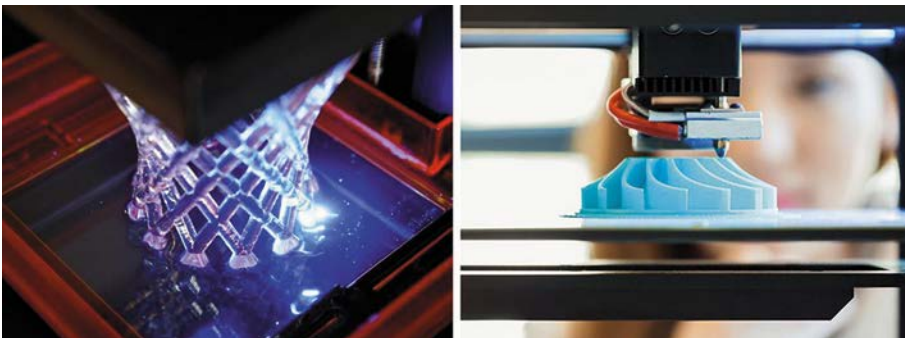
Мал. 4.55. Шари 3D-моделі у програмі-слайсері

У слайсері можна також установити значення інших властивостей друкування: *товщину кожного шару, щільність заповнення суцільних частин моделі матеріалом, швидкість друкування окремих частин об'єкта* та інші. За потреби у слайсері можуть бути автоматично створені підтримувальні елементи для нависаючих частин моделі.

Популярними є програми-слайсери **Ultimaker Cura**, **Simplify3D**, **Astroprint**, **3DPrinterOS** та інші.

Результат опрацювання моделі у слайсері потрібно зберегти у файлі формату **.GCODE**. Цей файл містить інструкції для принтера з координатами переміщення друкувальної головки.

Отриманий файл формату **.GCODE** потрібно завантажити в пам'ять принтера, після чого розпочинається друкування моделі (мал. 4.56). Це забезпечують спеціальні програми, що постачаються із 3D-принтерами.



Мал. 4.56. Друкування моделі:
ліворуч – на принтері типу **SLA**; праворуч – на принтері типу **FDM**

Після завершення друкування потрібно підготувати отриманий виріб до використання: видалити надлишки матеріалу та підтримувальні елементи, очистити поверхню та інше.

ПРОГРАМА-СЛАЙСЕР ULTIMAKER CURA

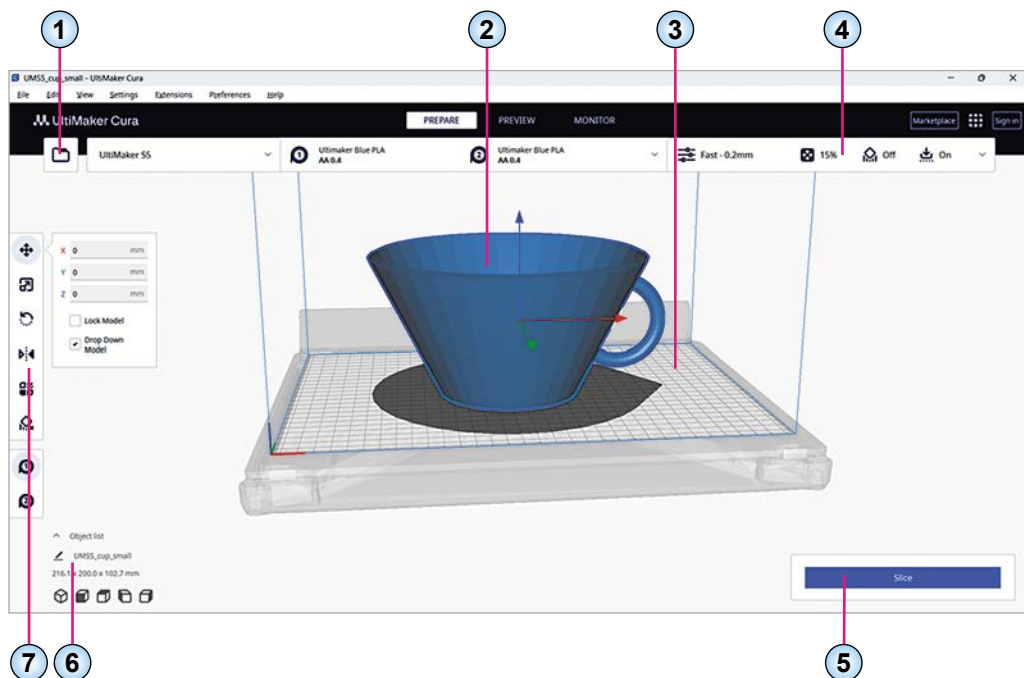


Пригадайте

- Які є етапи підготовки 3D-моделі до друкування? У чому вони полягають?
- Яке призначення програм-слайсерів? • Як виконується друкування об'єкта на 3D-принтері?

Слайсер **Ultimaker Cura** є програмою з відкритим кодом для підготовки готових 3D-моделей для друкування на 3D-принтерах різних виробників. Інсталяційний файл можна безкоштовно завантажити із сайту розробників <https://ultimaker.com/software/ultimaker-cura>. У процесі інсталяції потрібно вказати, на якій моделі 3D-принтера відбуватиметься друкування. Якщо у вас відсутній 3D-принтер, то для моделювання процесу можна вибрати будь-яку модель як мережний принтер.

Після запуску програми **Ultimaker Cura** на виконання у вікні відображається модель робочої платформи 3D-принтера типу **FDM**, на якій можна розмістити 3D-модель. На малюнку 4.57 зображено вікно програми **Ultimaker Cura** після відкриття файлу моделі.



Мал. 4.57. Вікно програми **Ultimaker Cura** після відкриття файлу моделі:

- 1 – кнопка відкриття 3D-моделі; 2 – 3D-модель; 3 – модель робочої платформи 3D-принтера; 4 – панель налаштувань якості друкування; 5 – кнопка **Slice**;
- 6 – відомості про модель; 7 – панель інструментів для перетворення моделі

Для відкриття файлу моделі слід вибрати кнопку  (мал. 4.57, 1), потім файл формату **STL** та кнопку **Відкрити**.

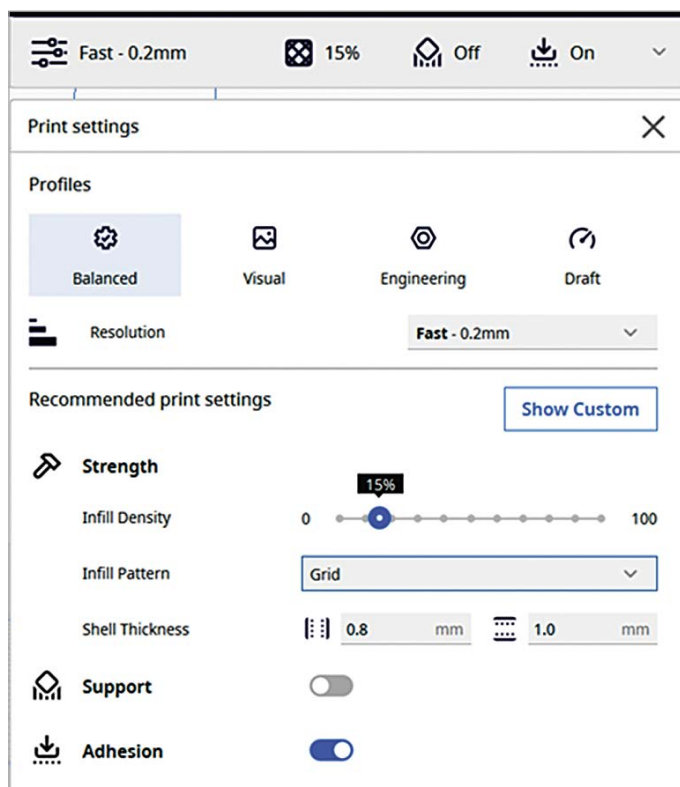
Після відкриття файлу по центру робочої платформи буде розміщено зображення моделі (мал. 4.57, 2). Якщо у вікні програми ви-

брати 3D-модель, то стають доступними інструменти (мал. 4.57, 7), використовуючи які можна перемістити модель , масштабувати для надання їй потрібних розмірів , обернути , віддзеркалити її положення на платформі  для зручності перегляду й виконати інші перетворення. У нижньому лівому куті вікна відображаються відомості про модель (мал. 4.57, 6): ім'я файлу моделі та її розміри в міліметрах.

Для прискорення виготовлення об'єктів можна розміщувати на робочій платформі одночасно кілька моделей.

Після вибору будь-якої точки на панелі налаштувань якості друку (мал. 4.57, 4) відкривається панель з елементами керування (мал. 4.58). Використовуючи їх, можна встановити значення деяких властивостей друкування:

- *якість друкування* – кнопки групи **Profiles** (англ. *profiles* – профілі);
 - *щільність заповнення деталей матеріалом* – шкала **Infill Density** (англ. *infill density* – щільність заповнення);
 - *наявність підтримувальних елементів* – перемикач **Support** (англ. *support* – підтримка)
- та інші.



Мал. 4.58. Елементи керування панелі налаштувань якості друкування

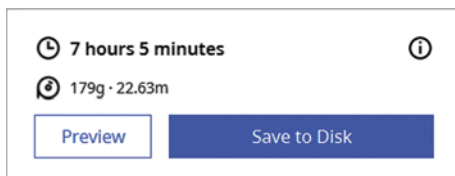
Установлені значення властивостей визначають швидкість та якість друкування, а також кількість витратних матеріалів, потрібних для друкування моделі. Наприклад, профіль **Visual** призначено для найякіснішого друкування, профіль **Draft** (англ. *draft* – чернетка) – для найшвидшого, а профіль **Balanced** (англ. *balanced* – збалансований) – для забезпечення балансу між якістю і швидкістю друкування. Значення щільності заповнення впливає на кількість витратних матеріалів та міцність виробу: що вища щільність, то більше філаменту буде використано під час друкування, водночас міцнішим буде отриманий об'єкт.

Для початку опрацювання моделі (нарізання шарів) потрібно вибрати кнопку **Slice** (мал. 4.57, 5).

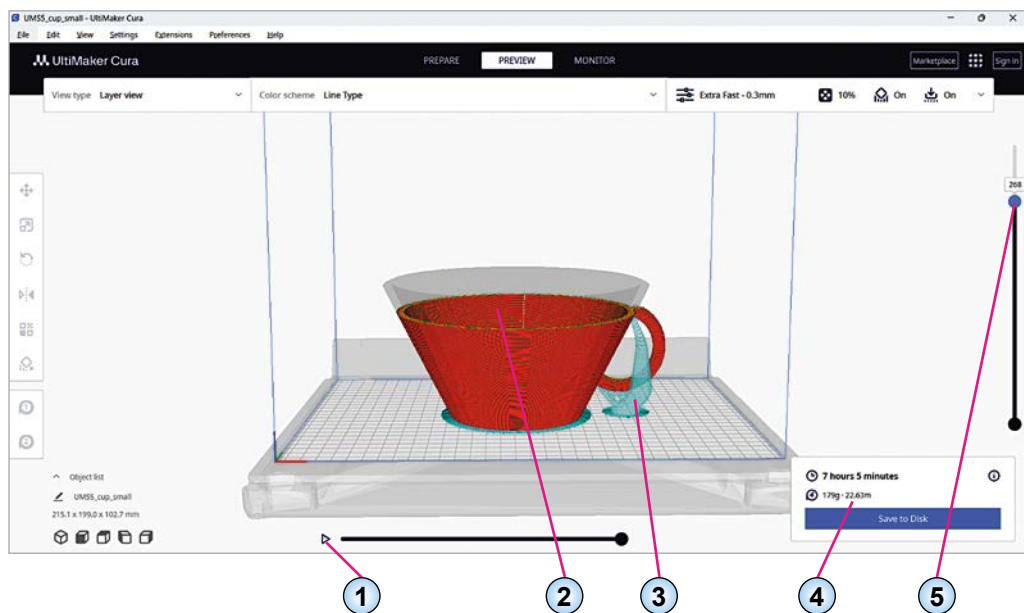
Після завершення процесу нарізання з'являється інформаційна панель (мал. 4.59), на якій можна побачити результати опрацювання моделі:

- тривалість друкування моделі на вибраному 3D-принтері; у прикладі – 7 год 5 хв;
- масу та довжину полімерної нитки, яка потрібна для друкування моделі; у прикладі – 179 грамів, 22,63 метра.

Після вибору на інформаційній панелі кнопки **Preview** відбувається перехід до режиму попереднього перегляду процесу друкування. У цьому режимі можна переглянути шари, отримані під час нарізання, і виконати попередній перегляд процесу друкування (мал. 4.60).



Мал. 4.59. Інформаційна панель з результатами опрацювання моделі



Мал. 4.60. Вікно програми **Ultimaker Cura** в режимі попереднього перегляду:

- 1 – кнопка запуску анімації процесу друкування;
- 2 – шар з вибраним номером;
- 3 – автоматично сконструйовані підтримувальні елементи;
- 4 – інформаційна панель;
- 5 – повзунок для вибору номера шару

На вертикальному повзунку (мал. 4.60, 5) показано загальну кількість шарів моделі та можна вибрати потрібний шар для його перегляду.

Демонстрація процесу друкування вибраного шару моделі розпочинається вибором кнопки  (мал. 4.60, 1).

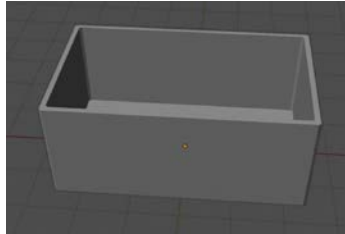
Для збереження файлу з інструкціями для принтера потрібно вибрати кнопку **Save to Disk** на інформаційній панелі (мал. 4.60, 4), указати папку для збереження й вибрати тип файлу **G-code File (*.gcode)**. Отриманий файл є текстовим файлом, уміст якого можна переглянути в будь-якому текстовому редакторі. Саме цей файл потрібен для запуску процесу друкування на 3D-принтері.



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання. Створіть у програмі **Ultimaker Cura** файл-інструкцію для друкування моделі коробки для дрібних речей (мал. 1) на 3D-принтері.



Мал. 1. Зразок до завдання

1. Відкрийте у програмі **Blender** проєкт, наприклад з файлу **Розділ 4\Пункт 4.5\вправа 4.5.blend**.
2. Експортуйте проєкт у файл типу **STL**. Для цього:
 1. Виконайте **Файл** ⇒ **Експортувати**.
 2. Виберіть формат **STL (.stl)**.
 3. Виберіть кнопку **Експорт STL** у вікні **Перегляд файлу Blender**, що відкрилося.
3. Закрийте вікно програми **Blender**.
4. Запустіть на виконання програму **Ultimaker Cura**. Ознайомтеся з інтерфейсом програми.
5. Відкрийте для опрацювання файл **вправа 4.5.stl**. Для цього скористайтесь кнопкою .
6. Виберіть модель на робочій платформі.
7. Виберіть кнопку **Scale**  на панелі інструментів для перетворення моделей.
8. Перегляньте значення властивостей моделі. Змініть розміри моделі, указавши масштаб **1000 %**.
9. Виберіть будь-яку точку на панелі налаштувань якості друку, щоб відкрити відповідну панель.
10. Виберіть профіль друкування **Balanced**.
11. Запустіть процес опрацювання моделі, вибравши кнопку **Slice**.



12. Створіть у текстовому процесорі новий документ. Уставте таблицю за зразком і введіть дані, отримані для профілю друкування **Balanced** у програмі **Ultimaker Cura**.

Значення властивостей	Профіль			
	Balanced	Visual	Engineering	Draft
Infill Density (щільність)				
Тривалість друкування				
Маса філаменту				
Довжина				

13. Вибирайте по чергово інші профілі друкування та кнопку **Slice**. Уведіть у таблицю отримані дані. Зробіть висновок, за вибору якого профілю друкування буде використано найменше філаменту.
14. Збережіть текстовий документ у вашій папці у файлі з іменем **вправа 4.5**.
15. Закрийте панель налаштувань якості друку в програмі **Ultimaker Cura**.
16. Перегляньте демонстрацію процесу друкування моделі. Для цього виберіть на інформаційній панелі кнопку **Preview**, перемістіть повзунок на довільну висоту і запустіть демонстрацію кнопкою . Для зупинки виберіть кнопку .
17. Збережіть файл-інструкцію у вашій папці. Для цього виберіть на інформаційній панелі кнопку **Save to Disk** і в процесі збереження вкажіть тип файлу **G-code File (*.gcode)**.
18. Закрийте вікно програми **Ultimaker Cura**.
19. За наявності 3D-принтера та з дозволу вчителя / вчительки завантажте файл **UMS5_вправа 4.5.gcode** у принтер і виконайте друк моделі.
20. Закрийте всі відкриті вікна.



Найважливіше в цьому пункті

Одним із призначень програм тривимірного моделювання є створення 3D-моделей для виготовлення їхніх твердих копій на 3D-принтерах.

Найпоширенішими технологіями, за якими здійснюється 3D-друк, є **SLA**-технологія – отвердіння спеціальної смоли під дією світла; **SLS**-технологія – запікання спеціального порошку під дією лазерного променя; **FDM**-технологія – отвердіння попередньо розплавлених пластичних мас. Незалежно від типу 3D-принтера під час друкування тривимірний об'єкт утворюється шляхом пошарового нанесення матеріалу.

Підготовка 3D-моделі до друкування відбувається в декілька етапів: експорт готової 3D-моделі в один із форматів – **.STL**, **.OBJ**, **.FBX**, **.3DS**; «нарізання» моделі шарами у програмі-слайсері, збереження результатів «нарізання» у файлі формату **.GCODE**, друкування на 3D-принтері, коригування отриманого виробу.

Слайсер **Ultimaker Cura** є програмою з відкритим кодом для підготовки 3D-моделей для друкування на 3D-принтерах різних виробників. Слайсер «нарізає» модель шарами і визначає координати переміщення друкувальної головки принтера для утворення кожного шару об'єкта. У програмі **Ultimaker Cura** можна виконувати операції перетворення моделі й налаштовувати параметри друкування. Результатом роботи слайсера є файл формату **GCODE** з інструкціями для принтера.



Дайте відповіді на запитання

1. У яких галузях діяльності людини використовують 3D-друк?
2. З яких етапів складається процес підготовки моделі до друкування? Поясніть кожен з них.
3. Якої корекції може потребувати надрукований на 3D-принтері виріб?
4. Яке призначення програми-слайсера? Значення яких властивостей друкування можуть бути встановлені у слайсері?
5. У який формат слід експортувати файл 3D-моделі, створеної у редакторі **Blender**, для подальшого «нарізання» у слайсері?
6. У якому форматі слід зберігати файл з результатами опрацювання моделі у програмі-слайсері для друкування на 3D-принтері? Які дані містяться в цьому файлі?



Обговоріть і зробіть висновки

1. Які переваги та недоліки має виготовлення об'єктів шляхом їх друкування на 3D-принтері порівняно з їх виготовленням іншими способами?
2. У чому полягають подібність і відмінності технологій 3D-друку?
3. У чому особливості створення 3D-моделей, метою яких є отримання тривимірного зображення, отримання 3D-анімації та 3D-друкування?
4. Чому, на вашу думку, у 3D-редакторі **Blender** відсутня команда друку створених 3D-моделей?



Виконайте завдання

1. Створіть і підготуйте для друкування 3D-модель циліндричної підставки для ручок заввишки 120 мм. Збережіть модель у вашій папці у файлі з іменем **завдання 4.5.1.stl**, файл для друку з іменем **завдання 4.5.1.gcode**.

2. Створіть і підготуйте для 3D-друкування модель деталі з конструктора «Лего» зі стороною основи 10 мм і висотою 8 мм за зразком (мал. 4.61). Збережіть модель у вашій папці у файлі з іменем **завдання 4.5.2.stl**, файл для друку з іменем **завдання 4.5.2.gcode**.



Мал. 4.61.

Зразок до завдання 2

3. Створіть з використанням системи штучного інтелекту **Meshy** (<https://www.meshy.ai/>) або іншої 3D-модель будинку. Установіть перед початком створення для властивості **Модель III** значення **Meshy4**. Завантажте створену модель, підготуйте її до друкування. Збережіть файл для друку з іменем **завдання 4.5.3.gcode**.

4. Створіть комп'ютерну презентацію про використання 3D-принтерів у одній з галузей: медицині, кінематографії, будівництві, індустрії моди, військовій справі. Збережіть презентацію у вашій папці у файлі з іменем **завдання 4.5.4.pptx**. Підготуйте виступ із презентацією.



5. Створіть 3D-модель куба зі стороною 4 м, збережіть у вашій папці у файлі з іменем **завдання 4.5.5.1.stl**. Відкрийте 3D-модель порожнього всередині куба з файлу **Розділ 4\Пункт 4.5\завдання 4.5.5.stl**. Опрацюйте обидві моделі у програмі **Ultimaker Cura**, установіть для них однаковий розмір, запишіть час друкування та ви-

трати матеріалу для кожної моделі. Перегляньте процес друкування кожної моделі, проаналізуйте заповнення внутрішньої частини кубів. Проведіть розрахунки в табличному процесорі й визначте, скільки відсотків філаменту буде зекономлено під час друкування порожнього всередині куба порівняно з друком цільного куба таких самих розмірів. Результати обчислень збережіть у вашій папці у файлі з іменем **завдання 4.5.5**.



Готуємось до вивчення нового матеріалу

1. Які ви знаєте ризики для користувача в інтернеті?
2. Що таке *особисті дані* та чому їх потрібно захищати?
3. Як ви розумієте термін *цифровий слід*? Який ваш цифровий слід в інтернеті?



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу «Тривимірна графіка» (початковий, середній, достатній, високий рівень).

- Я знаю, що таке тривимірна графіка.
- Я розумію, що таке 3D-модель об'єкта і для чого вона може бути використана.
- Я знаю, з яких елементів складається тривимірна модель та її основні властивості.
- Я розумію, які засоби можуть бути використані для створення та перегляду тривимірних моделей, і можу пояснити, який з них вибрати в різних ситуаціях.
- Я вмію створювати та редагувати 3D-модель у редакторі 3D-графіки.
- Я вмію здійснювати тривимірну навігацію у програмах для створення, редагування та перегляду тривимірних моделей.
- Я розумію, що таке ключові кадри анімації, та вмію створювати тривимірну анімацію.
- Я розумію, що таке 3D-друк, і знаю галузі його використання.
- Я розумію подібність та відмінності в технологіях 3D-друкування.
- Я знаю, з яких етапів складається підготовка 3D-моделі до друкування, і можу пояснити сутність цих етапів.
- Я вмію готувати 3D-моделі до друкування, використовуючи програму-слайсер.

Повторіть той матеріал, який ви знаєте недостатньо.

У цьому розділі ви здобудете нові, а також поглибите і розширите свої знання й удосконалисте навички з таких тем:

- ▶ інформаційні загрози та інформаційна безпека;
- ▶ цифровий слід у мережах і методи його зменшення, інформаційне «сміття»;
- ▶ захист особистих даних, права людини в інформаційному середовищі;
- ▶ ризики та загрози фізичному, психічному та соціальному здоров'ю у зв'язку з інтенсивним використанням цифрових технологій;
- ▶ шкідливе програмне забезпечення та боротьба з ним;
- ▶ основні дії для захисту персональних комп'ютерів від шкідливого програмного забезпечення.

5.1. ІНФОРМАЦІЙНІ ЗАГРОЗИ. ІНФОРМАЦІЙНА БЕЗПЕКА. ЦИФРОВИЙ СЛІД

У цьому пункті йтиметься про:

- ▶ інформаційну безпеку та інформаційні загрози;
- ▶ цифровий слід у мережах і методи його зменшення;
- ▶ інформаційне «сміття».

ІНФОРМАЦІЙНІ ЗАГРОЗИ ТА ІНФОРМАЦІЙНА БЕЗПЕКА

Пригадайте

● Які ви знаєте загрози, що можуть виникнути під час роботи з комп'ютером? Як можна захистити себе від них? ● Хто такі *хакери*? Що таке *фішинг*? ● Що таке *двофакторна авторизація* під час доступу до акаунту?

Вам уже відома та значна роль, яку відіграють інформаційні технології в житті сучасної людини, в успішному розвитку всього людства. Інформатика як наука вивчає методи і засоби передавання, опрацювання та зберігання повідомлень. Те, що повідомлення можна втратити, знає із власного досвіду більшість користувачів комп'ютерів. Тому проблема захисту даних від втрати внаслідок недбалства, через стихійне лихо чи крадіжку – одна з головних у комп'ютерній індустрії.

Використання автоматизованих інформаційних систем в управлінні підприємствами та установами значно підвищує якість і швидкість опрацювання та передавання різних даних, що позитивно позначається на ефективності управління. Проте цей процес має також зворотний бік – з'являється загроза викрадання конфіденційних даних, їх знищення або використання на користь інших людей, підприємств, установ чи держав.

Наведемо кілька прикладів із реального життя, коли подібні загрози стали жахливою дійсністю і призвели до значних втрат.

2017 року хакери використали вкрадені облікові дані для доступу до інформаційної системи агентства кредитної звітності **Equifax** (США). Унаслідок дій кіберзлочинців було викрадено персональні

дані приблизно 145,5 тисяч клієнтів агентства. Злочинці використовували викрадені дані для отримання кредитів та перерахування коштів на власні рахунки.

Пізніше, 2021 року, у **Фейсбуці (Facebook)** через уразливість у функції пошуку друзів було викрадено й оприлюднено персональні дані понад 530 млн користувачів. Причиною стала недбалість працівників у контролі за системою обміну даними з клієнтами. Наслідками стали системні фішингові атаки на комп'ютери користувачів, інші спроби шахрайства та зламування облікових записів, що призводило до значних репутаційних втрат.

З початку війни росії з Україною росія постійно проводить кібератаки на інформаційні системи критичної інфраструктури України. У грудні 2023 року «Укрзалізниця» зазнала масштабної кібератаки з боку російських злочинців, яка спричинила серйозні збої в роботі онлайн-систем компанії. Зокрема, став недоступним онлайн-продаж квитків, і користувачі змушені були купувати квитки через стаціонарні каси або в потягах. «Укрзалізниця» довелося відкривати додаткові каси з продажу квитків у Києві, Львові та Вінниці. Наслідки кібератаки були ліквідовані протягом доби завдяки діям фахівців з кібербезпеки самої компанії та Служби безпеки України. Зменшити втрати від порушення роботи «Укрзалізниця» вдалося за рахунок наявності дублювальних систем, резервного копіювання даних. Завдяки заходам безпеки хакерам не вдалося внести хаос у рух пасажирських потягів.

12 грудня 2023 року відбулася масштабна атака на засоби мобільного зв'язку «Київстар». За словами представників компанії, було знищено до 70 % віртуальних серверів і комп'ютерів компанії. Близько 24 мільйонів абонентів по всій Україні залишилися без мобільного зв'язку та доступу до інтернету на кілька днів. Це також порушило роботу сирен повітряної тривоги, частини банківських терміналів і банкоматів. Для проникнення в систему, за даними СБУ, було використано скомпрометований обліковий запис одного із співробітників компанії, який не був своєчасно заблокований. До початку деструктивних дій зловмисники більше ніж пів року мали доступ до системи «Київстар» і готували атаку.

Скомпрометований обліковий запис – це обліковий запис, доступ до якого отримали сторонні особи, що може призвести до його блокування, обмеження або несанкціонованого використання.

Наведені приклади говорять про значну шкоду, яку приносить нехтування заходами безпеки окремими громадянами та особливо працівниками, відповідальними за кібербезпеку на підприємствах і в установах.

Більшість держав розглядає безпеку в галузі інформаційно-комунікаційних технологій як одну з основних складових національної безпеки.

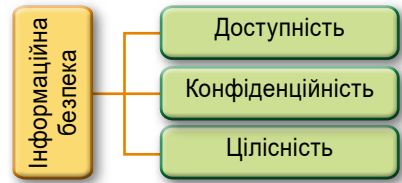
Інформаційна безпека – розділ інформатики, що вивчає методи та засоби захисту інформаційних ресурсів фізичних осіб, підприємств, організацій, державних установ тощо від *втрати, порушення функціонування, пошкодження, спотворення, несанкціонованого копіювання та використання*.

Інформаційна безпека базується на таких принципах (мал. 5.1):

Принцип доступності полягає в забезпеченні доступу до загальнодоступних даних усім користувачам, а до інших даних – лише за умови авторизації. Також передбачено захист від спотворення та блокування зловмисниками. Наприклад, доступ до відкритих сайтів – усім користувачам і в будь-який час, доступ до акаунту – лише після автентифікації (бажано двофакторної).

Принцип цілісності передбачає захист даних від зловмисного або випадкового видалення чи пошкодження.

Принцип конфіденційності полягає в забезпеченні доступу до даних тільки авторизованим суб'єктам – тобто тим особам, що мають підтверджене право здійснювати операції з цими даними.



Мал. 5.1. Принципи інформаційної безпеки

Автентифікація – процес перевірки особи користувача (наприклад, за відбитком пальця).

Авторизація – надання користувачу прав доступу до конкретних ресурсів після успішної автентифікації.

КЛАСИФІКАЦІЯ ЗАГРОЗ ІНФОРМАЦІЙНІЙ БЕЗПЕЦІ



Пригадайте

- Чи знаєте ви випадки шахрайства, пов'язані з розсилкою в соціальних мережах повідомлень із проханням вислати кошти?
- З якою метою зловмисники викрадають дані доступу до облікових записів користувачів у мережах?

Перелік особливо небезпечних деструктивних дій, які здійснюють зловмисники в разі отримання доступу до конфіденційних даних окремих осіб або до інформаційних систем приватних чи державних підприємств та організацій, можна подати у вигляді схеми (мал. 5.2).



Мал. 5.2. Схема деструктивних дій зловмисників



Загрози інформаційній безпеці можна класифікувати за значення-ми різних властивостей. Наприклад, можна класифікувати загрози за джерелом загрози, за принципами інформаційної безпеки, за ступенем впливу (наслідками), за об'єктом атаки.

Класифікація за **джерелом загрози** є однією з найбільш поширених та враховує, хто або що спричиняє загрозу (табл. 5.1).

Інсайдер (англ. *inside* – всередині) – особа, яка має доступ до конфіденційної інформації про діяльність підприємства, організації.

Таблиця 5.1

Класифікація загроз інформаційній безпеці за джерелом загрози

Категорія загроз	Опис	Приклади
Зовнішні	Джерелом деструктивних дій є зловмисники, що не є працівниками певної організації, соціальної групи	Хакерські угруповання, спеціалізовані кіберугруповання іншої (зазвичай ворожої) держави, кіберзлочинці, технологічні конкуренти
Внутрішні	Дії, що здійснюються авторизованими користувачами певних інформаційних систем, співробітниками підприємств та організацій	Зловмисник-інсайдер (незадоволений співробітник) або недбалий інсайдер (помилка, випадкове видалення даних, використання слабкого пароля)
Природні та форс-мажорні	Події, незалежні від людини, але які впливають на фізичну доступність або цілісність систем	Повінь, пожежа, землетрус, відключення електроенергії, результати військової агресії

Класифікація за основними **принципами інформаційної безпеки** базується на порушенні одного або кількох принципів інформаційної безпеки (табл. 5.2).

Таблиця 5.2

Класифікація загроз інформаційній безпеці за принципами інформаційної безпеки

Принцип	Сутність порушення	Приклади загроз, що реалізують порушення
Конфіденційність	Несанкціонований доступ до даних	Крадіжка даних, фішинг, перехоплення мережевого трафіку
Цілісність	Несанкціонована модифікація або знищення даних	Упровадження шкідливого програмного забезпечення в ІС, зміна фінансових транзакцій, пошкодження файлів через збій системи
Доступність	Неможливість авторизованих користувачів отримати доступ до ресурсів	DDoS-атаки, відмова обладнання, блокування системи програмою-вимагачем

Серед інших класифікацій варто відзначити класифікацію за **навмисністю дії** (навмисні, ненавмисні), за **наслідками** (критичні, високого, середнього або низького рівня), за **об'єктами атаки**. Наприклад, за останньою класифікацією розглядають загрози, об'єктами атаки яких є:

- *програмне забезпечення* (операційні системи, службове програмне забезпечення, прикладні програми тощо);
- *апаратне забезпечення* (сервери, робочі станції, засоби збереження даних, мобільні пристрої тощо);
- *пристрої мережі* (мережеве обладнання, роутери, комутатори, системи дротового та бездротового зв'язку);
- *фізичне середовище* (будівлі та приміщення, в яких розміщено цифрові пристрої, системи зв'язку);
- *люди* (окремі користувачі, персонал підприємств та організацій, у тому числі адміністратори ІТ-систем).

DDoS-атака (англ. *Distributed Denial of Service* – розподілена відмова в обслуговуванні) – кібер-атака, метою якої є порушення доступності ІС, вебсайту або сервісу шляхом запитами від великої кількості пристроїв одночасно.

ЦИФРОВИЙ СЛІД



Пригадайте

● Що таке *цифровий слід*? Чи стикалися ви із цим поняттям під час вивчення інформатики? ● Чи знайомі ви з поняттям «кібербезпека»? Що, на вашу думку, воно означає?

Поняття «цифровий слід» не є для вас новим. Ви його вивчали на уроках інформатики в попередніх класах. Розглянемо це поняття детальніше з урахуванням ваших знань з інформатики на сьогодні та з урахуванням його важливості під кутом зору кібербезпеки та обмеження доступу до ваших персональних даних.

Цифровий слід – це сукупність даних, які людина створює та залишає в результаті своєї діяльності в комп'ютерних мережах. Цей слід формується щоразу, коли вона здійснює будь-яку дію, яка реєструється цифровою системою: від відвідування вебсайту до публікації в соціальній мережі.

Компоненти цифрового сліду можна поділити на дві основні категорії: **активний цифровий слід** і **пасивний цифровий слід** (табл. 5.3).

Таблиця 5.3

Характеристики активного та пасивного цифрового сліду

Категорії	Опис	Приклади даних
Активний цифровий слід	Дані, які користувач свідомо та добровільно вносить до сервісів у мережі	Публікації в мережах, у тому числі соціальних, електронні листи, онлайн-форми реєстрації, дані акаунтів, відповіді на анкети (анонімні та персоналізовані), коментарі, електронні документи, фотографії, реакції на повідомлення в соціальних мережах



Категорії	Опис	Приклади даних
Пасивний цифровий слід	Дані, які збираються без прямої згоди користувача, зазвичай у фоновому режимі, під час обміну даними з мережевими сервісами, відвідування сайтів тощо	Тип цифрового пристрою, IP-адреса, тип і версія операційної системи, версія браузера та історія відвідування сторінок сайтів, тематика відвіданих сайтів, час перегляду певної сторінки, файли cookie (містять дані про попереднє відвідування сайту, про налаштування браузера користувача тощо), геолокаційні дані

Є кілька позитивних наслідків утворення цифрового сліду. Розробники сайтів, ураховуючи дії користувачів на сайті, можуть удосконалювати дизайн сторінок, роблячи їх більш зручними та привабливими.

Використовуючи зібрані дані, власники сайтів, особливо комерційного призначення, намагаються підлаштуватися під запити користувача, запропонувати послуги, що пов'язані з тематикою попередніх відвідувань сайтів, пропонують рекламу, що пов'язана з тематикою пошукових запитів користувача. Це може прискорити для користувача пошук потрібних товарів чи послуг.

Історія відвідування сайтів надає користувачу можливість швидко повернутись до раніше знайдених відомостей, а пошукова система, з урахуванням тематики переглянутих сайтів, під час уточнених запитів може надавати більш релевантний результат пошуку.

Цифровий слід також активно використовується для безпекових цілей. Якщо програма, наприклад **Google Chrome**, зафіксує нетиповий вхід у ваш акаунт (з іншого комп'ютера, з іншого міста), то на пошту буде надіслано сповіщення про такий вхід. Якщо це зробив власник цього акаунту, то жодних дій робити не треба, але якщо це зловмисник, то власник може заблокувати акаунт.

Разом з тим цифровий слід є джерелом даних для зловмисників, які на основі повідомлень у соціальних мережах, історії відвідування сайтів, даних геолокації, фото- й відеофайлів та інших даних цифрового сліду можуть визначити персональні дані про користувача: ім'я, прізвище, місце проживання, розпорядок дня, склад сім'ї, періоди відсутності членів сім'ї вдома, інтереси користувача, стан взаємин з іншими людьми тощо. Ці дані в подальшому можуть бути використані для цькування користувача, шантажу, булінгу, фізичних нападів на особу, членів сім'ї, крадіжок через проникнення в житло, коли відсутні мешканці, використання рахунків і банківських карток для викрадення коштів тощо.

Зменшення цифрового сліду – це процес, спрямований на мінімізацію кількості пасивних даних, які ви залишаєте, та контроль над активною складовою цифрового сліду, яку ви публікуєте в мережі.

Для зменшення *пасивного цифрового сліду* варто:

- регулярно очищати комп'ютер від файлів **cookie**;
- налаштувати браузер на автоматичне очищення файлів **cookie** після закриття сторінок сайту;
- під час надання дозволу на використання на сайтах файлів **cookie** обирати мінімальну кількість параметрів, які будуть записуватись у ці файли;

- не надавати сайтам доступ до геолокації, камери, мікрофона без потреби, регулярно перевіряти надані дозволи в налаштуваннях браузера;
- уникати створення акаунтів на сайтах без потреби;
- використовувати для відвідування підозрілих сайтів або відкритих мереж анонімний режим (у Google – **Нове вікно в режимі анонімного перегляду**).

Для зменшення *активного цифрового сліду* варто:

- обмежувати доступ до своїх публікацій у соціальних мережах, наприклад лише «Друзі», а не «Публічно»;
- мінімізувати дані під час реєстрації, за можливості не надавати дані про дату народження, адресу тощо;
- регулярно проводити перевірку старих, невикористовуваних акаунтів і видаляти їх, оскільки вони можуть бути джерелом витоку даних у майбутньому;
- вимикати доступ до даних геолокації в налаштуваннях пристроїв та сервісів, якщо вона не потрібна для їхньої роботи;
- уникати публікацій фото й відео з геомітками та іншими персональними даними.

ІНФОРМАЦІЙНЕ «СМІТТЯ»

Інформаційне «сміття» – це будь-які інформаційні об’єкти, які не мають корисної цінності, заважають ефективній роботі, створюють зайві ризики. Воно накопичується на пристроях зберігання даних, у електронних поштових скриньках, базах даних і робить пошук даних, їх аналіз та прийняття рішень менш ефективними.

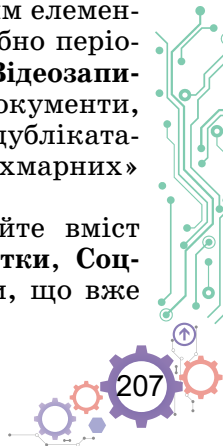
До інформаційного «сміття» належать:

- **файлове сміття** – дублікати документів; численні версії проєктів документів (Доповідь_v1, Доповідь_v2_final, Доповідь_final_final), тимчасові файли (наприклад *.tmp або ~\$document.docx); застарілі резервні копії; інсталяційні пакети, які не використовуються;
- **поштове та комунікаційне сміття** – спам, старі листи, групові чати з великою кількістю повідомлень, які не мають інформаційної цінності;
- **дані в базах і реєстрах** – застарілі записи та окремі дані; акаунти, які тривалий час не використовуються;
- **контент-сміття** – контент, що дублюється на різних сайтах, застарілі сторінки, неправдиві статті (фейки)

тощо.

Своєчасне видалення інформаційного «сміття» є важливим елементом політики інформаційної безпеки та гігієни даних. Потрібно періодично переглядати вміст папок **Завантаження, Документи, Відеозаписи, Зображення**, інших папок, у яких ви зберігаєте свої документи, і видаляти ті з них, які втратили свою актуальність або є дублікатами. Такі самі дії слід виконувати і над вмістом папок у «хмарних» сховищах, наприклад **Google Диск**.

Періодично, наприклад раз на 2–3 місяці, переглядайте вміст вашої електронної пошти (папки **Вхідні, Надіслані, Чернетки, Соцмережі, Оновлення, Пропозиції** та ін.) і видаляйте ті листи, що вже втратили актуальність або є просто спамом.



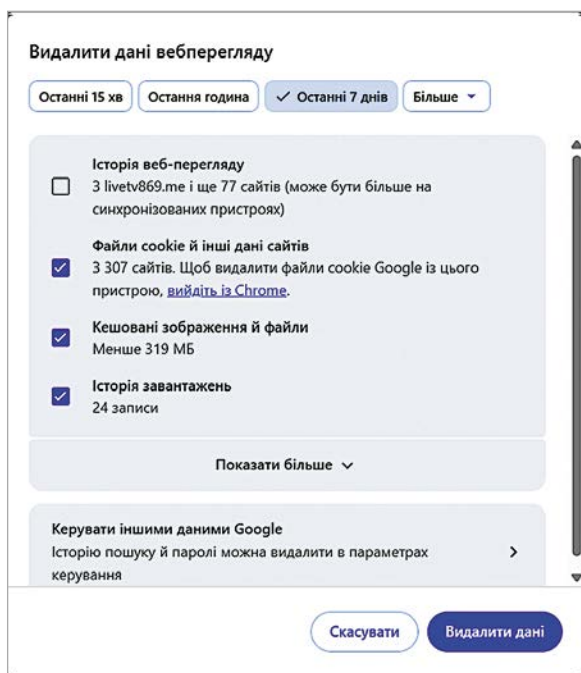


Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання 1. Очищення браузера від файлів cookie та видалення історії вебперегляду.

1. Запустіть на виконання браузер, наприклад **Google Chrome**.
2. Уведіть дані входу до свого акаунту.
3. Виберіть кнопку **Налаштування й керування Google Chrome** 
- у правому верхньому куті браузера.
4. Виберіть у меню, що відкрилося, команду **Видалити дані вебперегляду**.
5. Виберіть у вікні **Видалити дані вебперегляду** кнопку зі списком **Більше** та виберіть вкладку **Останні 7 днів** (мал. 1).
6. Зніміть позначку прапорця **Історія вебперегляду**.



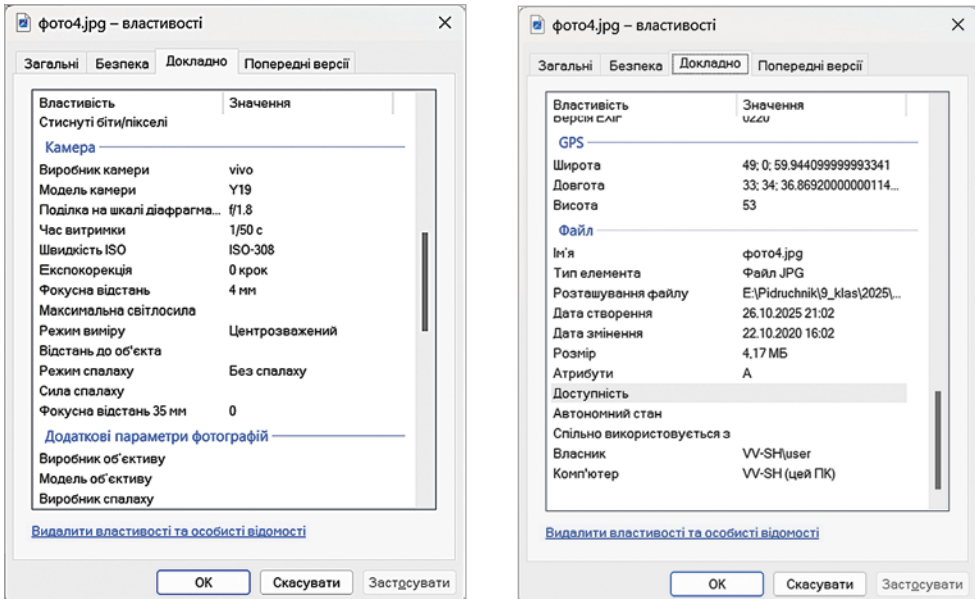
Мал. 1. Вікно **Видалити дані вебперегляду**

7. Виберіть кнопку **Видалити дані**.
8. Повторіть крок 4.
9. Виберіть у вікні **Видалити дані вебперегляду** кнопку зі списком **Більше** та виберіть вкладку **Остання година** (мал. 1).
10. Зніміть позначки прапорців для всіх пунктів, крім **Історія вебперегляду**.
11. Виберіть кнопку **Видалити дані**.

Завдання 2. Видалення персональних даних із графічного файлу.

1. Відкрийте папку з графічними файлами, наприклад **Розділ 5\Фото**.

- Відкрийте контекстне меню одного з графічних файлів, наприклад **фото 4.jpg**.
- Виберіть команду **Властивості**.
- Зробіть поточною вкладку **Докладно** (мал. 2)



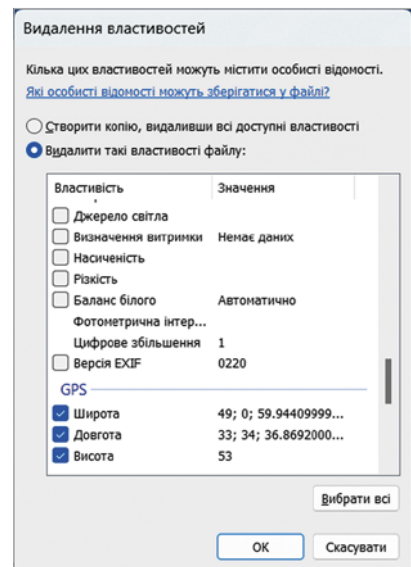
Мал. 2. Вкладка **Докладно** вікна **фото4.jpg-властивості**

- Перегляньте властивості зазначеного файлу. Знайдіть дані про фотокамеру, дані геолокації (GPS).
- Виберіть посилання **Видалити властивості та особисті відомості**.
- Виберіть перемикач **Видалити такі властивості файлу** у вікні **Видалення властивостей**.
- Установіть позначки прапорців, як це подано на малюнку 3.
- Виберіть кнопку **ОК**.
- Повторно перегляньте властивості цього файлу та переконайтеся, що значення властивостей геолокації видалено.



Найважливіше в цьому пункті

Інформаційна безпека – розділ інформатики, що вивчає закономірності забезпечення захисту інформаційних ресурсів фізичних осіб, підприємств, організацій, державних установ тощо від *втрати, порушення функціонування, пошкодження, спотворення, несанкціонованого копіювання та використання*. Інформаційна безпека базується на таких принципах: **доступність, конфіденційність, цілісність**.



Мал. 3. Вікно **Видалення властивостей**

Перелік особливо небезпечних деструктивних дій, які здійснюють зловмисники в разі отримання доступу до конфіденційних даних окремих осіб або до інформаційних систем приватних чи державних підприємств та організацій, включає: знищення та/або спотворення даних; отримання доступу до секретних та/або конфіденційних даних; пошкодження пристроїв інформаційної системи; отримання прав на виконання дій, що закріплені за іншими особами; здійснення фінансових операцій замість власника рахунків; отримання повного доступу до управління інформаційною системою.

Загрози інформаційній безпеці класифікують за значеннями різних властивостей: за **джерелом загрози** (*зовнішні, внутрішні, природні та форс-мажорні*); за основними **принципами інформаційної безпеки** (конфіденційність, цілісність, доступність); за **навмисністю дії** (*навмисні, ненавмисні*); за **наслідками** (*критичні, високого, середнього або низького рівня*); за **об'єктами атаки** (*програмне забезпечення, апаратне забезпечення, пристрої мережі, фізичне середовище, люди*).

Цифровий слід – це сукупність даних, які людина створює та залишає в результаті своєї діяльності в інтернеті та комп'ютерних мережах. Цей слід формується щоразу, коли ви здійснюєте будь-яку дію, яка реєструється цифровою системою: від відвідування вебсайту до публікації в соціальній мережі.

Компоненти цифрового сліду можна поділити на дві основні категорії: **активний цифровий слід** і **пасивний цифровий слід**.

Зменшення цифрового сліду – це процес, спрямований на мінімізацію кількості пасивних даних, які ви залишаєте, і контроль над активною складовою цифрового сліду, яку ви публікуєте в мережі.

Інформаційне «сміття» – це будь-які інформаційні об'єкти, які не мають корисної цінності, заважають ефективній роботі або створюють зайві ризики. Воно накопичується на пристроях збереження даних, у поштових скриньках, базах даних, репозиторіях і робить пошук, аналіз та прийняття рішень менш ефективними.



Дайте відповіді на запитання

1. Що таке *інформаційна безпека*? На яких принципах базується інформаційна безпека?
2. До яких деструктивних дій можуть вдаватися зловмисники, якщо їм вдається подолати систему інформаційної безпеки?
3. На які види поділяються загрози інформаційній безпеці залежно від джерела загрози?
4. Що таке *цифровий слід*? Які є компоненти цифрового сліду? Які дії користувача збільшують цифровий слід?
5. Як зменшити пасивний цифровий слід? Як зменшити активний цифровий слід?
6. Що таке *інформаційне «сміття»*? Як від нього можна позбавитися?



Обговоріть і зробіть висновки

1. Ви отримали повідомлення в соціальній мережі **Telegram** такого змісту: «Я беру участь у конкурсі на кращий виріб із природних матеріалів. Зайди на сайт та проголосуй за мою роботу». Повідомлення надіслала Аня. Ви точно знаєте, що серед ваших друзів у **Telegram** є Аня. Які ваші дії?
2. Чи існує, на вашу думку, цифровий слід установи? Якщо так, то вкажіть, із чого він може складатися, на прикладі вашого навчального закладу.
3. Частина сайтів, які ви часто відвідуєте, виводять повідомлення з проханням надати доступ до відеорекамери на ноутбук. Яким програ-

мам (сервісам), на вашу думку, варто надавати такий доступ? Чи варто надавати доступ усім, хто запитує дозвіл?



Виконайте завдання

1. Проаналізуйте політику конфіденційності сайту **Google Chrome**. Для цього в нижній частині вікна сайту (<https://www.google.com>) виберіть посилання **Конфіденційність і безпека**. Визначте, які дані збираються браузером, та здійсніть перевірку безпеки. Знайдені відомості про дані, що збираються, та результати перевірки безпеки занесіть у текстовий файл і збережіть у вашій папці у файлі **завдання 5.1.1**.

2. Як зазначають фахівці *Державної служби спеціального зв'язку та захисту інформації України*, безліч людей нехтують власною безпекою і встановлюють прості паролі, які хакери без зусиль можуть зламати менше ніж за секунду. Для прикладу: в рейтинг найуживаніших паролів постійно потрапляють такі комбінації, як 123456, qwerty, password, фрази на кшталт i love you або власні імена. Створіть п'ять різних паролів різної складності. Перевірте їхню надійність на сайті <https://zillya.ua/check-password> або подібному. Опис перевірки та її результати внесіть до текстового файлу та збережіть його у вашій папці з іменем **завдання 5.1.2**.

3. Складіть таблицю з п'ятьма прикладами загроз, джерел, наслідків і засобів захисту. Таблицю створіть у текстовому файлі та збережіть його у вашій папці з іменем **завдання 5.1.3**.

Тип загрози	Джерело	Можливі наслідки	Засоби захисту
Фішинг	Шахраї в інтернеті	Викрадення паролів	Уважність, антивірус, двофакторна автентифікація

4. Підготуйте інфографіку або кілька слайдів презентації з порадами на тему «Як захистити себе в інтернеті». Результати роботи збережіть у вашій папці у файлі **завдання 5.1.4**.

5. Перегляньте налаштування приватності свого облікового запису в **Google Chrome** (**Налаштування й керування Google Chrome** : ⇒ ⇒ **Налаштування** ⇒ **Конфіденційність і безпека** ⇒ **Налаштування сайтів** ⇒ **Дозволи**) і з'ясуйте, яким чином сайти отримують дозволи на визначення місцезнаходження, використання вебкамери та мікрофона. З'ясуйте, які сайти мають доступ до визначення місцезнаходження, використання вебкамери та мікрофона. Результати дослідження внесіть до текстового файлу та збережіть його у вашій папці з іменем **завдання 5.1.5**.

6. Виконайте пошук відомостей і створіть у вигляді таблиці довідник із термінів, що стосуються інформаційної безпеки: *Уразливість, Ризик, Кібератака, Фішинг, DDoS-атака, Фаєрвол, VPN (Virtual Private Network), Аудит безпеки*. Створену таблицю збережіть у текстовому файлі у вашій папці з іменем **завдання 5.1.6**.



Готуємось до вивчення нового матеріалу

1. Які є способи захисту особистих даних?
2. Які ризики несе для людини надмірне використання цифрових пристроїв?
3. Що таке *права людини*?



5.2. ЗАХИСТ ПЕРСОНАЛЬНИХ ДАНИХ. ПРАВА ЛЮДИНИ В ІНФОРМАЦІЙНОМУ СЕРЕДОВИЩІ

У цьому пункті йтиметься про:

- ▶ захист персональних даних;
- ▶ ризики і загрози фізичному, психічному та соціальному здоров'ю через користування цифровими технологіями;
- ▶ права людини в інформаційному середовищі.

ПРАВА ЛЮДИНИ В ІНФОРМАЦІЙНОМУ СЕРЕДОВИЩІ



Пригадайте

● Що таке *права людини*? Які права людини ви знаєте? ● Яка відповідальність може настати за порушення прав людини? ● Які випадки порушення прав людини в інформаційному середовищі ви знаєте?

Інформаційне середовище – це простір, де ми спілкуємося, навчаємося, працюємо та отримуємо знання з використанням цифрових технологій. У цьому середовищі класичні права людини набувають нового змісту, а також з'являються нові, суто «цифрові» права. До таких прав людини в інформаційному середовищі належить:

- **право на приватність і захист персональних даних**, що передбачає:
 - недоторканність особистого життя, листування та спілкування (захист повідомлень, дзвінків);
 - контроль над збиранням, зберіганням та використанням персональних даних;
 - можливість видалення персональних даних із будь-якої бази даних на вимогу користувача;
- **право на свободу думки й слова** передбачає:
 - право на розміщення в комп'ютерних мережах повідомлень без додаткової цензури, якщо такі повідомлення не порушують права інших людей;
 - право безперешкодно отримувати інформацію, доступ до якої не обмежується законом;
 - право на рівний доступ до технологій і даних, незалежно від соціального статусу чи фізичних можливостей (інклюзія);
- **право на безпеку в цифровому просторі**, на захист від кіберзлочинів, шахрайства, булінгу, маніпуляцій і дезінформації.

Держава зобов'язана захищати цифрові права громадян, розвивати законодавство у сфері кібербезпеки, цифрової освіти і медіаграмотності. В Україні діє низка законів, які спрямовані на захист цих прав:

- Закон України «Про захист персональних даних»;
- Закон «Про інформацію»;
- Закон «Про основні засади забезпечення кібербезпеки України».

Кожен користувач цифрових технологій має пам'ятати: цифрова свобода передбачає цифрову відповідальність. Кримінальним кодексом України передбачено кримінальну відповідальність за:

- «Порушення таємниці листування, телефонних розмов, телеграфної чи іншої кореспонденції, що передаються засобами зв'язку або через комп'ютер»;
- «Незаконне відтворення, розповсюдження творів науки, літератури і мистецтва, комп'ютерних програм і баз даних, ... їх незаконне тиражування та розповсюдження на ... носіях інформації»;
- «Незаконні дії з документами на переказ, платіжними картками та іншими засобами доступу до банківських рахунків, електронними грошима, обладнанням для їх виготовлення»;
- «Несанкціоноване втручання в роботу електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку»;
- «Створення з метою використання, розповсюдження або збуту шкідливих програмних чи технічних засобів, а також їх розповсюдження або збут»;
- «Несанкціоновані збут або розповсюдження інформації з обмеженим доступом, яка зберігається в електронно-обчислювальних машинах (комп'ютерах), автоматизованих системах, комп'ютерних мережах або на носіях такої інформації»

тощо.

ЗАХИСТ ПЕРСОНАЛЬНИХ ДАНИХ



Пригадайте

- Які дані належать до персональних? ● Які нормативні акти спрямовані на захист персональних даних громадян? ● Як захистити персональні дані від використання без вашого дозволу?

Персональні дані кожної людини в нашій країні захищені низкою нормативних документів. Перш за все це документи, що стосуються прав і свобод людини:

- **Конвенція про захист прав людини та основоположних свобод** (прийнята Радою Європи 04.11.1950 р.) проголошує, що свобода вираження поглядів може обмежуватися законодавством «...для запобігання розголошенню конфіденційної інформації»;
- **Конституція України**: «Стаття 32. ... Не допускається збирання, зберігання, використання та поширення конфіденційної інформації про особу без її згоди, крім випадків, визначених законом...». «Стаття 54. Громадянам гарантується ... захист інтелектуальної власності, їхніх авторських прав ...».

В Україні захист персональних даних унормований Законом України «Про захист персональних даних». Його положення спрямовані на забезпечення основоположних прав і свобод людини під час опрацювання персональних даних.

У законі дано означення **персональних даних** – це «відомості чи сукупність відомостей про фізичну особу, яка ідентифікована або може бути конкретно ідентифікована». Тобто це певні відомості, які надають іншим особам можливість визначити, про яку особу йдеться.



До персональних даних належать:

- дані, що ідентифікують особу: *прізвище, ім'я, по батькові, дата й місце народження; ідентифікаційний код, дані паспорта (серія, номер, ким і коли виданий) або ID-карти, водійського посвідчення; стать, громадянство, сімейний стан, наявність дітей, освіта, місце навчання, посада, місце роботи;*
- контактні дані особи: *адреса проживання або реєстрації, номер телефону, адреса електронної пошти;*
- біометричні дані, що можуть бути використані для отримання доступу до цифрових пристроїв: *відбитки пальців, зображення сітківки ока, зображення обличчя, голос, генетичні (ДНК) дані;*
- медичні дані: *дані про стан здоров'я, медичні діагнози, дані про лікування, відомості про інвалідність, групу крові, щеплення;*
- фінансові дані: *номери банківських рахунків та/або карток, дані про рух коштів і кредити, розміри заробітної плати, дані про сплату податків, дані про власність;*
- відомості про особисті переконання та життя: *релігійні переконання, політичні погляди (належність до політичних партій), сексуальна орієнтація, членство у громадських чи профспілкових організаціях тощо;*
- електронні та мережні ідентифікатори: *IP-адреса, MAC-адреса пристрою, геолокація, дані про цифровий пристрій і браузер користувача, логіни, паролі, нікнейми в онлайн-сервісах, дані облікових записів у соціальних мережах, цифровий підпис.*

MAC-адреса (англ. *Media Access Control* – управління доступом до посередників) – унікальний ідентифікатор пристрою в комп'ютерній мережі.

Наведений перелік складено згідно із Законом України «Про захист персональних даних», а також міжнародними стандартами, зокрема **GDPR** (англ. *General Data Protection Regulation* – *загальний регламент про захист даних*) – нормативно-правовим актом Європейського Союзу щодо захисту персональних даних осіб. Цей перелік не є вичерпним – в окремих нормативних актах ідеться про інші персональні дані, що стосуються професійної діяльності (перелік місць роботи і посад, які обіймала особа, кваліфікація, відгуки керівництва) або результатів проходження різноманітних психологічних тестів та опитувань.

Персональні дані, відповідно до вимог зазначеного Закону України, можуть опрацьовуватися лише за згодою особи. Виняток становлять випадки, пов'язані з безпекою самої особи та інших громадян, державною безпекою тощо. У разі, якщо особа не досягла повноліття, дозвіл на опрацювання її персональних даних надають батьки або особи, що їх замінюють.

Витік (несанкціоноване поширення) персональних даних може призвести до шахрайства, булінгу, крадіжки особистості, дискримінації. Тому і сама особа повинна максимально обмежити доступ до своїх персональних даних, особливо в комп'ютерних мережах.

Найбільш уразливою ланкою в системі захисту даних залишається сама особа – об'єкт захисту даних. Користувачі часто нехтують базовими правилами безпеки, і це надає шахраям можливість скориста-

тися персональними даними в кримінальних цілях. Щоб зменшити ризики, дотримуйтесь основних правил захисту персональних даних у цифровому середовищі:

1. Не розголошуйте свої дані в середовищах з відкритим доступом (соцмережі, форуми, чати).
2. Дотримуйтесь правил роботи з пароллями:
 - використовуйте складні паролі (мінімум 8 символів, великі та малі літери, цифри, спеціальні символи, такі як # \$ % ^ ~ < > _ { } тощо);
 - не передавайте паролі та логіни друзям чи стороннім особам;
 - не використовуйте один і той самий пароль для всіх сервісів (якщо зламують один, зламують усі);
 - увімкніть двофакторну автентифікацію (**2FA – 2-Factor Authentication**), що передбачатиме введення не лише пароля, а й коду з вашого телефону (це найпростіший і найефективніший спосіб захисту);
 - використовуйте програму – **менеджер паролів** (наприклад, **Bitwarden, KeePass**), яка завдяки шифруванню безпечно зберігає всі ваші паролі, і вам потрібно пам'ятати лише один головний пароль: вхід до програми – менеджера паролів;
 - регулярно оновлюйте паролі (раз на 3–6 місяців) для основних сервісів (пошта, браузер, соцмережі).
3. Ніколи не публікуйте в мережі фотознімки документів (ID-картки, свідоцтва про народження, банківських карток, чеків, квитків з персональними даними).
4. Не вводьте персональні дані на сайтах без захищеного з'єднання (адреса повинна починатися з **https://**).
5. Не вводьте паролі або фінансові дані в публічних Wi-Fi-мережах (на вокзалах, в аеропортах, у кафе тощо). Для подібних мереж не передбачено шифрування даних, і дані в них можуть стати доступними злоумисникам.

РИЗИКИ І ЗАГРОЗИ ФІЗИЧНОМУ, ПСИХІЧНОМУ ТА СОЦІАЛЬНОМУ ЗДОРОВ'Ю



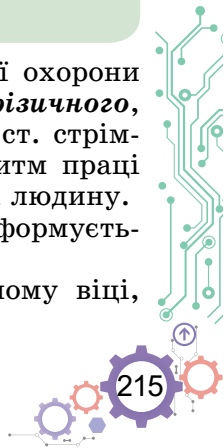
Пригадайте

● Чи знаєте ви ризики фізичному здоров'ю, з якими стикаються користувачі, що понад міру користуються цифровими пристроями та комп'ютерними мережами? Назвіть ці ризики. ● Які санітарно-гігієнічні норми під час роботи з комп'ютерними пристроями слід враховувати?

Здоров'я людини, за означенням Всесвітньої організації охорони здоров'я, – це не лише відсутність хвороб, а **стан повного фізичного, психічного здоров'я та соціального благополуччя**. У XXI ст. стрімкий розвиток технологій, зміни способу життя, високий ритм праці та навчання створюють нові види ризиків, які впливають на людину.

Особливо вразливими є діти й підлітки, чия психіка ще формується, а соціальні зв'язки залежать від цифрового середовища.

Фізичне здоров'я сучасної людини, особливо у шкільному віці, може погіршуватися внаслідок низки ризиків (табл. 5.4).



Ризики фізичному здоров'ю та засоби профілактики

Ризики	Причина	Засоби профілактики
Гіподинамія (малорухливість), може зумовити ожиріння, серцево-судинні захворювання	Надмірний час, проведений за комп'ютером і/або смартфоном	Щоденна рухова активність (спорт, рухливі ігри, прогулянки, ранкова гімнастика)
Погіршення зору (синдром комп'ютерного зору, порушення акомодатії, синдром «сухого ока»)	Тривала робота з екраном, особливо з екранами смартфонів, неправильні налаштування монітора	Дотримання режиму роботи з комп'ютером і смартфоном, правильне налаштування значень властивостей монітора: яскравість, контрастність, кольорова температура
Порушення опорно-рухового апарату (порушення постави, проблеми з хребтом, тунельний синдром зап'ястя)	Неправильно організоване місце з комп'ютером, порушення тривалості роботи з комп'ютером	Правильна організація робочого місця за комп'ютером (мал. 5.3), дотримання норм роботи з комп'ютером
Нераціональне харчування, захворювання системи травлення	Порушення режиму харчування, вживання їжі під час роботи з комп'ютером, надання переваги у їжі фастфудам, солодким газованим напоям	Збалансоване та своєчасне харчування без доступу до гаджетів, зведення до мінімуму споживання фастфудів та солодких газованих напоїв
Недосипання, порушення вироблення гормону сну в організмі	Навчальні перевантаження та постійне перебування онлайн (соціальні мережі, ігри)	Дотримання режиму дня, у якому сну відводиться не менше ніж 8 годин



Мал. 5.3. Організація робочого місця учня / учениці за стаціонарним комп'ютером і ноутбуком (малюнки створені ШІ Leonardo.ai)

Психічне здоров'я є невід'ємною частиною загального здоров'я і впливає на фізичний стан людини, і навпаки – фізичний стан впливає на психічне здоров'я. Тимчасові негативні емоції, як-от смуток або розчарування, є нормальною частиною життя. Проблеми виникають, коли такі почуття домінують і заважають жити повноцінно.

Ознаками нормального стану психічного здоров'я людини є:

- здатність розуміти свої емоції та керувати ними, отримувати задоволення від життя й адекватно проявляти весь спектр почуттів, таких як радість, смуток, гнів (*емоційний добробут*);
- уміння будувати взаємини з іншими людьми, співпрацювати в колективі та робити внесок у життя своєї спільноти (*соціальне благополуччя*);
- здатність успішно долати можливі стресові ситуації, адаптуватися до нових умов життєдіяльності й самостійно приймати рішення (*психологічна стійкість*);
- можливість реалізовувати свої здібності, розвиватися й відчувати себе цінним (*самореалізація*).

Психічне здоров'я людини може погіршуватись унаслідок низки загроз (табл. 5.5).

Таблиця 5.5

Загрози психічному здоров'ю та їх профілактика

Загрози	Причина	Засоби профілактики
<i>Інформаційне перевантаження</i> , що викликає загальну втоми, зниження працездатності	Надмірна кількість повідомлень, які люди на отримують внаслідок неконтрольованого за часом спілкування в соціальних мережах, підписок на різноманітні канали, перегляду сайтів тощо	Унормований час на огляди новин, спілкування в соціальних мережах; відведення часу в розпорядку дня на активні фізичні вправи, читання книжок, прослуховування музики, реалізацію творчих задумів. Вимикання сповіщень від сервісів інтернету, месенджерів, соціальних мереж
<i>Кібербулінг</i> , що викликає тривогу, страх або депресію	Проблеми з комунікацією в учнівському колективі, колективі за місцем проживання чи творчої (спортивної) діяльності. Агресивна поведінка особи чи групи осіб	Звертання за емоційною, а в окремих випадках – і правовою підтримкою до членів родини, педагогів
<i>Інтернет та ігрова залежність</i> , що зумовлює хворобливу потребу постійно бути онлайн та ігнорування реальних потреб	Втрата контролю за часом під час комунікації в соціальних мережах, перегляд різноманітних інтернет-ресурсів, надмірне захоплення відеоіграми	Дотримання розпорядку дня, у якому, крім часу на роботу з комп'ютером, є час на реальне спілкування з друзями / подругами, членами родини; відвідування творчих гуртків, спортивних секцій тощо



Загрози	Причина	Засоби профілактики
Стрес і тривожність, що спричиняють постійну емоційну напруженість, відчуття нестабільності	Має внутрішні (страх невиконання або неякісного виконання завдань, отримання негативних оцінок від педагогів та друзів), а також зовнішні причини (війна, нестабільність у суспільстві, сім'ї)	Розвиток навичок саморегуляції, виконання дихальних вправ, вправ на свіжому повітрі. Спілкування про свої страхи, переживання з батьками, друзями або психологом

Турбота про власне психічне здоров'я допомагає зберегти гарний настрій та енергію для здійснення власних планів і задумів, знаходити час для занять, які роблять життя кращим і підтримують здоров'я – і фізичне, і психічне.

Соціальне здоров'я – це здатність людини взаємодіяти з іншими, відчувати підтримку та приналежність до спільноти. До основних загроз варто віднести:

- *соціальну ізоляцію* – результат надмірного віртуального спілкування, відсутність або обмеження безпосередніх (офлайн) бесід із членами родини, друзями / подругами, однокласниками / однокласницями, педагогами тощо;
- *агресію та нетерпимість* під час спілкування в соціальних мережах і месенджерах;
- *дезінформацію та маніпуляцію*, які впливають на здатність адекватно реагувати на реальні події, підривають довіру до певних осіб та установ;
- *витік персональних даних*, що може призвести до фішингу, фінансових втрат, загроз особистій репутації та безпеці.

Сучасні ризики для фізичного, психічного здоров'я та соціального благополуччя взаємопов'язані й часто мають спільне інформаційне підґрунтя.



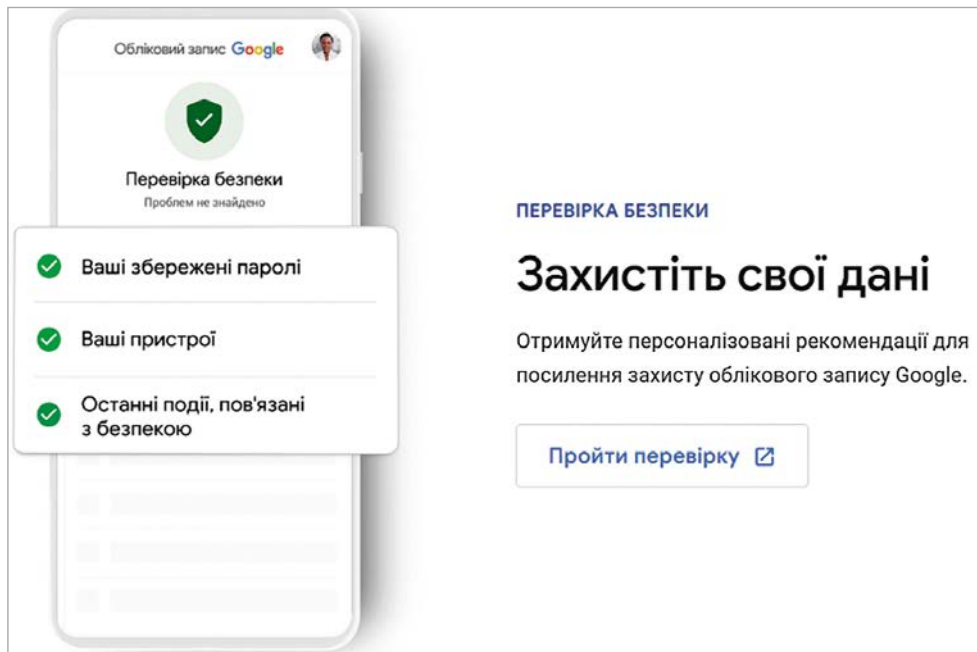
Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання. Перевірка налаштувань безпеки браузера.

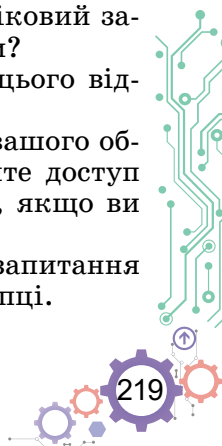
1. Запустіть на виконання браузер **Google Chrome**.
2. Уведіть дані входу до свого акаунту.
3. Відкрийте сторінку **Центру безпеки** <https://safety.google/intl/uk/>.
4. Перегляньте відомості про заходи безпеки, які здійснює **Центр безпеки Google Chrome**.
5. Знайдіть розділ **Пройдіть перевірку безпеки**.
6. Виберіть кнопку **Пройти перевірку** [Пройти перевірку](#)  (мал. 1).
7. Оновіть (уведіть або підтвердьте) дані про резервну електронну пошту. Для цього виберіть кнопку **Так, підтвердити** або **Ні, оновити**.
8. Розкрийте список **2 способи підтвердження** та перевірте правильність даних про вашу резервну електронну пошту і телефон, які використовуються для підтвердження ваших дій в акаунті або від-

новлення доступу до акаунту. За потреби внесіть зміни після вибору кнопки **Редагувати**  .



Мал. 1. Вікно **Центру безпеки Google Chrome**

9. Розкрийте список **Збережені паролі** та виберіть кнопку **Перевірте свої паролі**.
10. Виберіть кнопку **Перевірити паролі** та уведіть дані свого акаунту.
11. Визначте, скільки ваших паролів:
 - зламано;
 - використовуються повторно;
 - є ненадійними.
12. Замініть за потреби проблемні паролі. Для чого відкрийте список відповідної групи паролів і виберіть кнопку **Змінити пароль**  **Змінити пароль** у рядку відповідного сайту.
13. Виберіть кнопку  біля заголовка **Перевірка паролів**.
14. Перегляньте список пристроїв, з яких ви заходили в обліковий запис **Google Chrome**. Чи є в цьому списку підозрілі сеанси?
15. Визначте, чи відбувалися нещодавно підозрілі дії. Для цього відкрийте список **Останні підозрілі дії**.
16. Визначте, чи мають інші сервіси чи програми доступ до вашого облікового запису (список **Сторонні підключення**). Скасуйте доступ програми або сервісу вибором кнопки **Скасувати доступ**, якщо ви не маєте повної впевненості в надійності джерела.
17. Запишіть результати вашої перевірки та відповіді на запитання в текстовий файл **вправа 5.2** і збережіть його у вашій папці.
18. Закрийте всі відкриті вікна.





Найважливіше в цьому пункті

Інформаційне середовище – це простір, де ми спілкуємося, навчаємося, працюємо та отримуємо знання з використанням цифрових технологій. У цьому середовищі є «цифрові» права: *право на приватність та захист персональних даних, право на свободу думки й слова, право на безпеку в цифровому просторі* тощо.

Права людини в інформаційному середовищі – це продовження наших базових прав у цифрову епоху. Кожен користувач цифрових технологій має пам'ятати: цифрова свобода передбачає цифрову відповідальність. Знання своїх прав та вміння їх захищати – це запорука безпечного, свідомого й вільного життя в сучасному інформаційному світі.

Персональні дані кожної людини в нашій країні захищені низкою нормативних документів. Перш за все – це документи, що стосуються прав і свобод людини: **Конвенція про захист прав людини та основоположних свобод, Конституція України**.

До переліку персональних даних належать: *дані, що ідентифікують особу, контактні дані особи, біометричні дані, медичні дані, фінансові дані, відомості про особисті переконання та життя, електронні та мережні ідентифікатори* тощо.

Персональні дані, згідно з вимогами Закону України «Про захист персональних даних», можуть опрацьовуватися лише за згодою особи за винятком окремих випадків, передбачених законодавством.

Щоб зменшити ризики несанкціонованого доступу до персональних даних користувача в комп'ютерних мережах, варто дотримуватися правил захисту персональних даних.

Здоров'я людини – це не лише відсутність хвороб, а й **стан повного фізичного, психічного та соціального благополуччя**.

Фізичне здоров'я сучасної людини, особливо у шкільному віці, може погіршуватися внаслідок низки ризиків: *гіподинамії, погіршення зору, порушення опорно-рухового апарату, захворювання системи травлення, недосипання* тощо.

Психічне здоров'я є невід'ємною частиною загального здоров'я і впливає на фізичний стан людини, і навпаки – фізичний стан впливає на психічне здоров'я. Психічне здоров'я людини може погіршуватися внаслідок низки загроз: *інформаційного перевантаження, кібербулінгу, інтернет- та ігрової залежності, стресу і тривожності* тощо.

Соціальне здоров'я – це здатність людини взаємодіяти з іншими, відчувати підтримку та приналежність до спільноти. До основних загроз варто віднести: *соціальну ізоляцію, агресію та нетерпимість, дезінформацію та маніпуляцію, витік персональних даних*.



Дайте відповіді на запитання

1. Які основні права людини захищаються законами України? Чи має держава право обмежувати доступ до певної інформації? У яких випадках? Наведіть приклади.
2. Що таке *персональні дані*? Назвіть щонайменше три категорії даних, які, на вашу думку, потребують особливої уваги під час захисту.
3. Хто має право збирати та обробляти персональні дані? Які основні права має людина щодо своїх персональних даних?
4. Які вимоги висуваються до надійного пароля? Поясніть, чому використання одного пароля для всіх сервісів є критичною загрозою.
5. Поясніть, чому небезпечно вводити свої облікові дані, використовуючи публічні Wi-Fi-мережі в кафе чи аеропортах.
6. Яких правил має дотримуватися користувач, щоб мінімізувати ризики витоку персональних даних?
7. Як впливають **цифрові пристрої** на зір та поставу учнів / учениць?



Обговоріть і зробіть висновки

1. Що означає згода на опрацювання персональних даних? Які дані не є персональними? Наведіть приклади.
2. Як ви можете захистити дані на своєму комп'ютері від фізичного доступу сторонніх осіб (наприклад, якщо ваш ноутбук украдуть)?
3. Як шкільні електронні журнали, онлайн-тести чи сервіси повинні забезпечувати захист персональних даних учнів / учениць?
4. Опишіть ознаки гіподинамії, спричиненої тривалим сидінням за комп'ютером. Як часто і для чого слід робити перерви під час роботи?
5. Чому, незважаючи на мобільність, використання ноутбука на колінах або лежачи в ліжку може бути більш шкідливим для фізичного здоров'я, ніж робота за стаціонарним ПК?
6. Як дезінформація та фейкові новини впливають на соціальне здоров'я? Які прийоми ви використовуєте для перевірки достовірності інформації?



Виконайте завдання

1. Складіть у текстовому процесорі довідник із термінів, що стосуються захисту персональних даних і прав людини в інформаційному середовищі: *Персональні дані, Інформаційне середовище, MAC-адреса, Витік персональних даних, Менеджер паролів, Психічне здоров'я, Соціальне здоров'я, Приватність, Права людини в інформаційному середовищі, Недоторканність листування, Свобода слова, Кіберзлочин*. Збережіть довідник у вашій папці у файлі з іменем **завдання 5.2.1**.
2. Сформулюйте власне розуміння терміна **Цифрова гігієна** і наведіть один приклад її дотримання, який ви застосовуєте щодня. Уключіть результати виконання завдання до текстового документа і збережіть його у файлі у вашій папці з іменем **завдання 5.2.2**.
3. Відкрийте один із браузерів, наприклад **Google Chrome**. У вікні для введення логіна і пароля виберіть посилання **Створити новий обліковий запис**. Пройдіть усі етапи створення нового облікового запису та зафіксуйте в текстовому документі перелік даних, які слід увести для реєстрації. Які із запитуваних даних не є обов'язковими? Які із запитуваних даних належать до персональних даних? Чи надає програма пояснення, з якою метою запитуються дані, наприклад дата народження? Відповіді на ці запитання також унесіть у той самий текстовий документ і збережіть у вашій папці у файлі з іменем **завдання 5.2.3**.
4. У текстовому процесорі створіть пам'ятку на тему «Як захистити свої персональні дані в інтернеті». Використайте нумерований чи багаторівневий список для переліку дій із захисту і доповніть текст пам'ятки інфографікою. Для створення інфографіки скористайтеся одним із відомих вам сервісів штучного інтелекту. Проведіть редагування та форматування вмісту пам'ятки і збережіть її у вашій папці у файлі з іменем **завдання 5.2.4**.
5. Перевірте своє робоче місце за комп'ютером. Запишіть дані перевірки в текстовий документ:
 - 1) Чи розміщений верхній край монітора на рівні очей?



- 2) Чи утворюють руки в ліктьовому суглобі кут, близький до 90°, якщо долоні розміщуються на клавіатурі?
- 3) Чи опирається спина на спинку стільця (крісла), якщо ви сидите в робочій позі?
- 4) Чи є опора для ніг, якщо вони утворюють у колінних суглобах кут, близький до 90°?

Збережіть дані перевірки та рекомендації щодо покращення свого робочого місця у вашій папці у файлі з іменем **завдання 5.2.5**.



6. Створіть у текстовому документі інструкцію з налаштування нічного підсвічування монітора вашого комп'ютера в період з 20:00 до 07:00 з інтенсивністю 40 %. Додайте скриншоти до пояснення послідовності дій. Шлях до налаштувань нічного підсвічування монітора **Налаштування** ⇒ **Домашня сторінка** ⇒ **Дисплей** ⇒ **Нічник**. В інструкції опишіть, з якою метою вмикається цей режим, і збережіть інструкцію у вашій папці у файлі з іменем **завдання 5.2.6**.



7. Для застосування **Режиму свідомого споживання контенту** зафіксуйте, скільки часу протягом 2–3 днів ви приділяєте безцільному скролінгу соціальних мереж, месенджерів, тобто тій діяльності, яка не зумовлена навчальними та творчими потребами. Визначте три ресурси (канали, блоги, сайти), які сприяють вашому розвитку (навчання, хобі, саморозвиток). Складіть розклад на тиждень, коли ви приділятимете час цим ресурсам замість витрачання його на безцільні «мандри» інтернетом. Збережіть розклад у вашій папці у файлі з іменем **завдання 5.2.7**.

8. Проведіть самооцінювання інтернет-залежності з використанням тесту **Кімберлі Янга**. Для цього відкрийте один із ресурсів інтернету з тестом (<https://cutt.ly/ItA0A1L6> або <https://cutt.ly/jtA0RySo>), ознайомтеся з поясненням до тесту і максимально відверто дайте відповіді на запитання. Ознайомтеся з результатом тесту. Чи відповідає він вашим уявленням про власну інтернет-залежність?



Готуємось до вивчення нового матеріалу

1. Які програми створюються з метою завдання шкоди комп'ютерним даним?
2. Які засоби використовуються для захисту від шкідливих програм?
3. Що таке *комп'ютерний вірус*?



5.3. ШКІДЛИВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА БОРІТЬБА З НИМ

У цьому пункті йтиметься про:

- ▶ шкідливе програмне забезпечення та боротьбу з ним;
- ▶ основні дії для захисту персональних комп'ютерів від шкідливого програмного забезпечення;
- ▶ програми протидії шкідливим програмам, налаштування їхніх основних параметрів.

ШКІДЛИВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ



Пригадайте

● Які ви знаєте загрози, пов'язані з викраденням персональних даних користувача? ● Як зловмисники викрадають кошти в комп'ютерних мережах? ● Що таке вірус? Як він потрапляє в організм людини?

Крім корисних комп'ютерних програм, які користувач використовує для опрацювання даних, існують і **шкідливі комп'ютерні програми**.

Для **шкідливих комп'ютерних програм** характерно:

- *швидке розмноження* шляхом приєднання своїх копій до інших програм, копіювання на інші носії даних, пересилання копій комп'ютерними мережами;
- *автоматичне виконання деструктивних дій*:
 - *знищення даних* шляхом видалення файлів певних типів або форматування дисків;
 - *внесення змін у файли*, зміна структури розміщення файлів на диску;
 - *блокування даних* користувача шляхом їх перекодування з метою подальшого вимагання коштів для відновлення доступу;
 - *змінення або повне видалення даних* з постійної пам'яті комп'ютера;
 - *зниження швидкодії* комп'ютера, наприклад за рахунок заповнення оперативної пам'яті своїми копіями;
 - *постійне (резидентне) розміщення в оперативній пам'яті* від моменту звернення до ураженого об'єкта до моменту вимкнення комп'ютера і ураження дедалі більше нових об'єктів;
 - *примусове перезавантаження* операційної системи;
 - *блокування запуску* певних програм;
 - *збирання і пересилання копії даних* комп'ютерними мережами (шпигунство), наприклад пересилання кодів доступу до секретних даних;
 - *використання ресурсів уражених комп'ютерів* для організації колективних атак на інші комп'ютери в мережах або несанкціонованого майнінгу;
 - *виведення звукових або текстових повідомлень*, спотворення зображення на екрані монітора

Деструктивні дії – це поведінкові акти, що завдають шкоди людині, обладнанню або суспільству загалом.

Майнінг – це процес створення нових одиниць криптовалюти. Для цього потрібний потужний комп'ютер або велика кількість звичайних ПК.

тощо.

За **рівнем небезпечності дій** шкідливі програми поділяють на:

- **умовно безпечні** – проявляються відео- та звуковими ефектами, не змінюють файлову систему, не ушкоджують файли і не виконують шпигунські дії;
- **небезпечні** – призводять до перебоїв у роботі комп'ютерної системи: зменшують розмір доступної оперативної пам'яті, перезавантажують комп'ютер тощо;



- **дуже небезпечні** – знищують дані з постійної та зовнішньої пам'яті, виконують шпигунські дії тощо.

За особливостями розповсюдження і функціонування шкідливі програми поділяють на:

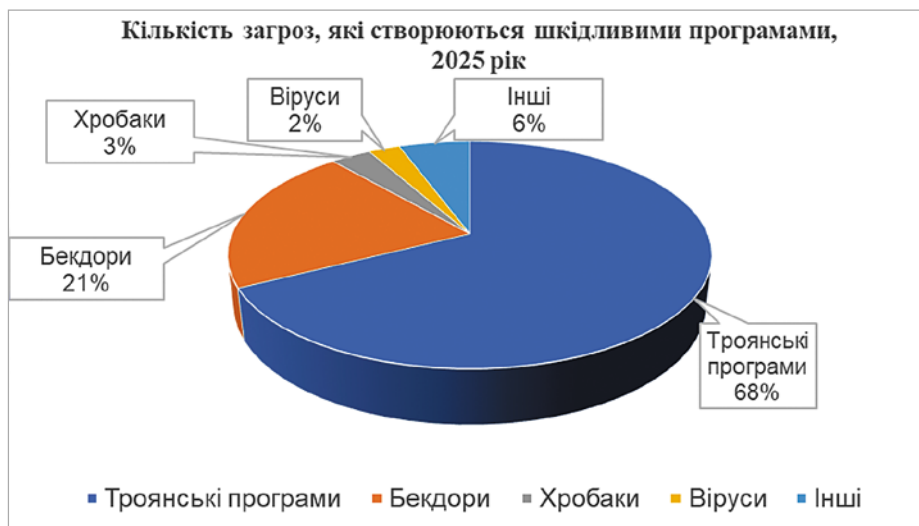
- **комп'ютерні віруси** – програми, здатні саморозмножуватися і виконувати несанкціоновані деструктивні дії на ураженому комп'ютері. Серед них виділяють:
 - **дискові** (завантажувальні) **віруси** – розмножуються копіюванням себе в службові ділянки носіїв даних, яке відбувається під час спроби користувача зчитати дані з ураженого носія;
 - **файлові віруси** – розміщують свої копії у складі файлів різного типу. Зазвичай це файли готових до виконання програм із розширенням імені **exe** або **com**. Однак існують так звані макровіруси, що уражують, наприклад, файли текстових документів, електронних таблиць, баз даних тощо;
- **хробаки** (черв'яки) комп'ютерних мереж – пересилають свої копії комп'ютерними мережами з метою проникнення на віддалені комп'ютери. Більшість черв'яків поширюються, прикріпившись до файлів електронної пошти, електронних документів тощо;
- **троянські програми** – програми, що проникають на комп'ютери користувачів разом з іншими програмами, які користувач «отримує» комп'ютерними мережами або на змінному носії;
- **бекдори** (англ. *back door* – чорний вхід) – програми, що проникають на комп'ютер користувача, використовуючи службові функції окремих ОС для дистанційного доступу до комп'ютерів. Такі функції використовуються розробниками для відновлення доступу до комп'ютерів у разі збоїв у їх роботі;
- **рекламні модулі**, або **Adware** (англ. *Ad* скорочення від *advertisement* – оголошення, реклама, *ware* – товар), – програми, що вбудовуються в браузер користувача для показу реклами під час перегляду вебсторінок.

За цілями діяльності шкідливі програми поділяють на:

- **програми-вимагачі**, або **Ransomware** (англ. *Ransom* – викуп, *ware* від *software* – програмне забезпечення), – програми, що здійснюють шифрування файлів користувача, після чого він не може їх використовувати, та вимагають викуп за розблокування файлів;
- **руткіти** (англ. *root* – кореневий каталог у **Linux**, *kit* – набір інструментів) – програми, що використовуються для приховування наявності шкідливого ПЗ на комп'ютері;
- **завантажувачі**, або **Droppers** (англ. *Dropper* – підвіска, штучна приманка), – програми, що самі не є шкідливими, але використовуються для завантаження та встановлення на комп'ютер іншого ПЗ – шкідливого;
- **криптомайнери**, або **CoinMiners** (англ. *Coin* – монета, *Miners* – шахтарі, рудокопи), – програми, що використовують ресурси комп'ютера без згоди користувача для «добування» криптовалюти;
- **викрадачі даних**, або **Infostealers** (англ. *Info* – від *Information* – інформація, *stealers* – злодії), – програми для крадіжки даних облікових записів, банківських карток, різноманітних паролів тощо.

Шкідливі програми можуть використовувати для проникнення на комп'ютер та здійснення своєї шкідливої діяльності різні механізми, але зазвичай їхні дії передбачають три етапи:

- *проникнення та закріплення в системі* (змінення налаштування ОС для маскування);
- *розповсюдження* (копіювання себе на інші носії, прикріплення до інших файлів і надсилання своїх копій по мережі);
- *безпосереднє виконання шкідливих дій*.



Мал. 5.4. Діаграма кількості атак різними типами шкідливих програм у 2025 році

Приблизний розподіл кількості загроз в ОС **Windows**, які створювали шкідливі програми в 2025 році, за даними сайту **Незалежного інституту ІТ-безпеки** (<https://www.av-test.org/>), наведено на малюнку 5.4.

На сьогодні основними тенденціями розробки шкідливих програм є:

- орієнтація на *фінансову вигоду* (вимагання викупу, крадіжка даних для виконання фінансових операцій, майнінг тощо);
- використання для отримання несанкціонованого доступу до сервісів *штучного інтелекту*;
- значне збільшення кількості *програм-завантажувачів*, які в подальшому «відкривають двері» для інших шкідливих програм з широким діапазоном деструктивних дій.



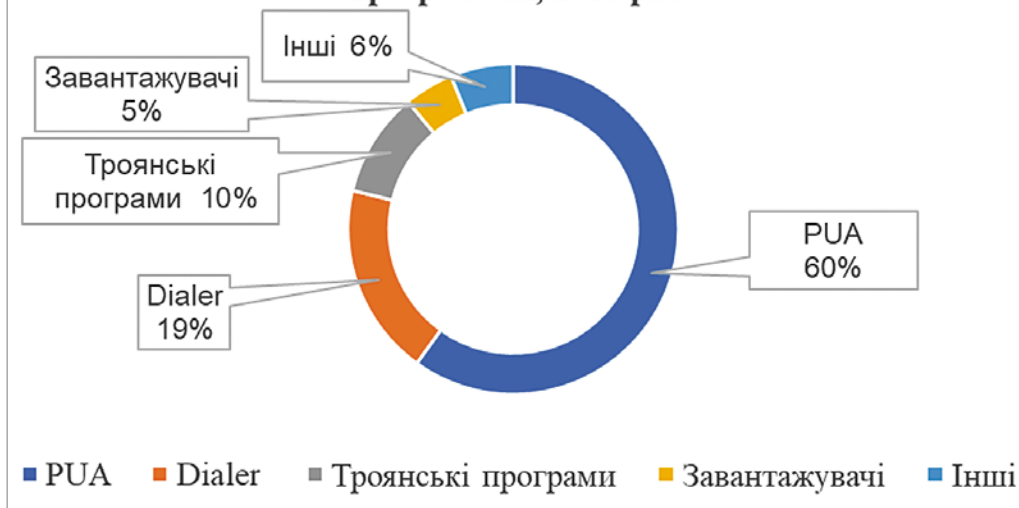
Для тих, хто працює з Android

У пристроях, що працюють з операційною системою **Android**, кількість атак шкідливого програмного забезпечення за останні два роки збільшилась у кілька разів. За даними сайту **Незалежного інституту ІТ-безпеки** (адреса ресурсу: <https://portal.av-atlas.org/malware/statistics>), найчастіше користувач ОС **Android** стикається зі шкідливим програмним забезпеченням типу **PUA**, **Dialer**, **троянські програми** та інші (мал. 5.5).

PUA (англ. *Potentially Unwanted Application* – потенційно небезпечна комп'ютерна програма) – це програмне забезпечення, яке може використовуватися для нав'язливої реклами, змінення налаштувань браузера, установлення небажаних програм, зниження продуктивності пристрою, збирання персональних даних тощо.

Dialer (англ. *Dialer* – номеронабирач) – шкідливе ПЗ, призначене для викрадення коштів з телефонних рахунків. Використовує дзвінки на платні номери зловмисника, приховуючи сам процес дзвінка.

Співвідношення загроз, які створюються шкідливими програмами, 2025 рік



Мал. 5.5. Діаграма співвідношення загроз шкідливих програм для ОС Android у 2025 році



Для тих, хто хоче знати більше

Теоретичні основи створення програм, що можуть самостійно розмножуватись і виконувати дії без втручання користувача, були розроблені у процесі вдосконалення теорії розробки автоматичних пристроїв (теорія автоматів) у 1950–1970-х роках. На початку 1970-х років створювалися експериментальні зразки подібних комп'ютерних програм.

Однією з перших програм такого типу стала програма, що з'явилася в американській військовій комп'ютерній мережі **ARPANet**. Вона отримала назву **Creeper** (англ. *Creeper* – повзуча рослина, рептилія) і могла самостійно поширюватися мережею, створювати свою копію на іншому комп'ютері та виводити на монітор повідомлення «I'M THE CREEPER: CATCH ME IF YOU CAN» (англ. – «Я рептилія: спіймайте мене, якщо зможете»). Водночас ця програма та інші аналогічні програми того часу (наприклад, **Animal**, **Xerox worm**) не завдавали шкоди, а лише демонстрували можливість розробки комп'ютерних програм, здатних до саморозмноження та автоматичного виконання певних дій.

Одними з перших програм, що завдавали шкоди комп'ютерним програмам і даним, були програми **Virus 1, 2, 3** і **Elk Cloner**, розроблені для персональних комп'ютерів **Apple II**. Програма **Elk Cloner** розмножувалася шляхом запису своєї копії в початковій секторі дискет, які були на той час основним носієм даних для ПК. Її шкідливі дії зводилися до перевертання зображення на екрані монітора, мерехтіння тексту, виведення різноманітних повідомлень тощо. Приблизно в цей час за цим видом шкідливих програм закріпилася назва «комп'ютерні віруси».

ПРОГРАМИ ПРОТИДІЇ ШКІДЛИВИМ ПРОГРАМАМ



Пригадайте

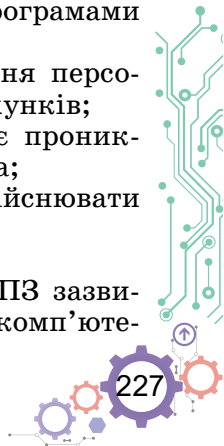
- Які засоби використовують для захисту даних?
- Які засоби захисту даних, як правило, мають сучасні операційні системи?

Для захисту даних і пристроїв комп'ютера від шкідливих програм використовують спеціальне програмне забезпечення. За ним збереглася стара назва – **антивірусні програми**, хоча більшість з них забезпечують захист від різних типів шкідливих програм. Сучасні антивірусні програми – це комплексні програми, що мають набір модулів для захисту від різних загроз. Такими є програми **Bitdefender, Norton 360, Microsoft Defender, Avast Free Antivirus, Eset NOD32 Antivirus, Zillya! Antivirus** та інші.

Наприклад, **Українська антивірусна компанія** (<https://zillya.ua>) пропонує кілька варіантів свого антивірусу **Zillya!** (укр. *зілля* – різноманітна запашна трава, часто з цілющими властивостями): *Для дому та бізнесу, Для бізнесу і держави, Для смартфона та планшета, Антивірус для навчальних закладів* та ін. Залежно від призначення до складу програми (цієї або інших подібних) можуть входити такі модулі:

- *вартовий* (файловий монітор) – постійно (резидентно) міститься в оперативній пам'яті комп'ютера з моменту запуску антивірусної програми і перевіряє всі файли та носії даних, дані про які потрапляють в оперативну пам'ять, блокує дії, що можуть ідентифікуватись як дії шкідливої програми;
- *інспектор* (поведінковий аналізатор) – аналізує стан системних файлів та папок і порівнює їх зі станом, що був на початку роботи антивірусної програми. За певних змін, які характерні для діяльності шкідливих програм, виводить повідомлення про небезпеку ураження;
- *поштовий фільтр* – перевіряє пошту користувача на наявність шкідливих програм, забезпечує блокування надходження пошти з підозрілих поштових адрес або із сайтів, що надсилають спам;
- *USB-захист* – здійснює перевірку підключення змінних носіїв даних, за потреби перевіряє їх на наявність шкідливих програм;
- *евристичний аналізатор* – проводить евристичний (грец. *εῦρηκα* – знайшов, винайшов) аналіз – пошук нових шкідливих програм за їх стандартними діями або фрагментами коду;
- *брандмауер* (мережевий екран) – забезпечує виконання встановлених для програм комп'ютера правил доступу до мережі, блокує за потреби підозрілі з'єднання, відстежує надсилання програмами даних у мережі;
- *антифішинг* – блокує сайти, призначені для викрадання персональних даних користувача з метою доступу до його рахунків;
- *антиспам* (англ. *spam* – нав'язлива реклама) – блокує проникнення рекламних повідомлень на комп'ютер користувача;
- *батьківський контроль* – надає батькам можливість здійснювати контроль за роботою дитини в комп'ютерній мережі та інші.

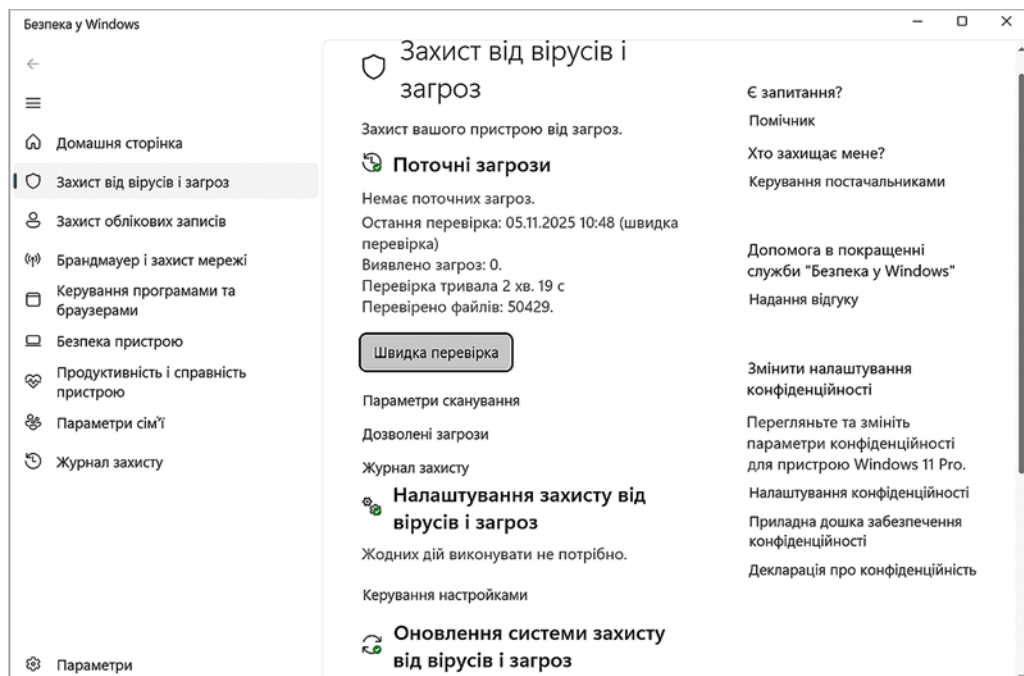
Крім комплексних програм, розробники антивірусного ПЗ зазвичай пропонують окремі програми для швидкого сканування комп'юте-



ра на наявність шкідливих програм і їх знешкодження. Такі програми розповсюджуються безкоштовно із сайтів виробників у вигляді одного файлу. Ці програми називають **сканерами**. Для роботи з ними не потрібна інсталяція. Достатньо скачати файл програми із сайту розробників і запустити на виконання, наприклад, *ZillyaScanner_ua.exe*, **Автономний модуль Microsoft Defender** тощо.

Якщо комплексну антивірусну програму встановлено на вашому комп'ютері, то під час увімкнення ПК вона буде однією з перших автоматично завантажуватися в оперативну пам'ять комп'ютера і виконувати операції з перевірки наявності шкідливих програм і блокування їх дій.

В останніх версіях операційної системи **Windows** програма антивірусного захисту **Microsoft Defender** (англ. *defender* – захисник) входить до складу ОС, і перевірка системи на ураження шкідливими програмами здійснюється автоматично з певною періодичністю. Користувач може увімкнути або призупинити запуск програми, змінити налаштування цієї програми, здійснити додатково швидку або повну перевірку комп'ютера, оновити систему захисту (оновити **антивірусні бази** – сукупності даних про відомі на даний момент часу шкідливі програми і способи боротьби з ними) тощо. Для цього потрібно відкрити вікно **Безпека у Windows** (*Пуск* ⇒ *Налаштування* ⇒ *Конфіденційність і безпека* ⇒ *Безпека у Windows* ⇒ *Захист від вірусів і загроз*) (мал. 5.6).



Мал. 5.6. Вкладка **Захист від вірусів і загроз**

Змінення налаштувань параметрів захисту від вірусів і загроз здійснюється на вкладці **Налаштування захисту від вірусів і загроз**, що відкривається після вибору посилання **Керування настройками**.

Використовуючи елементи керування цієї вкладки, можна вимкнути або увімкнути:

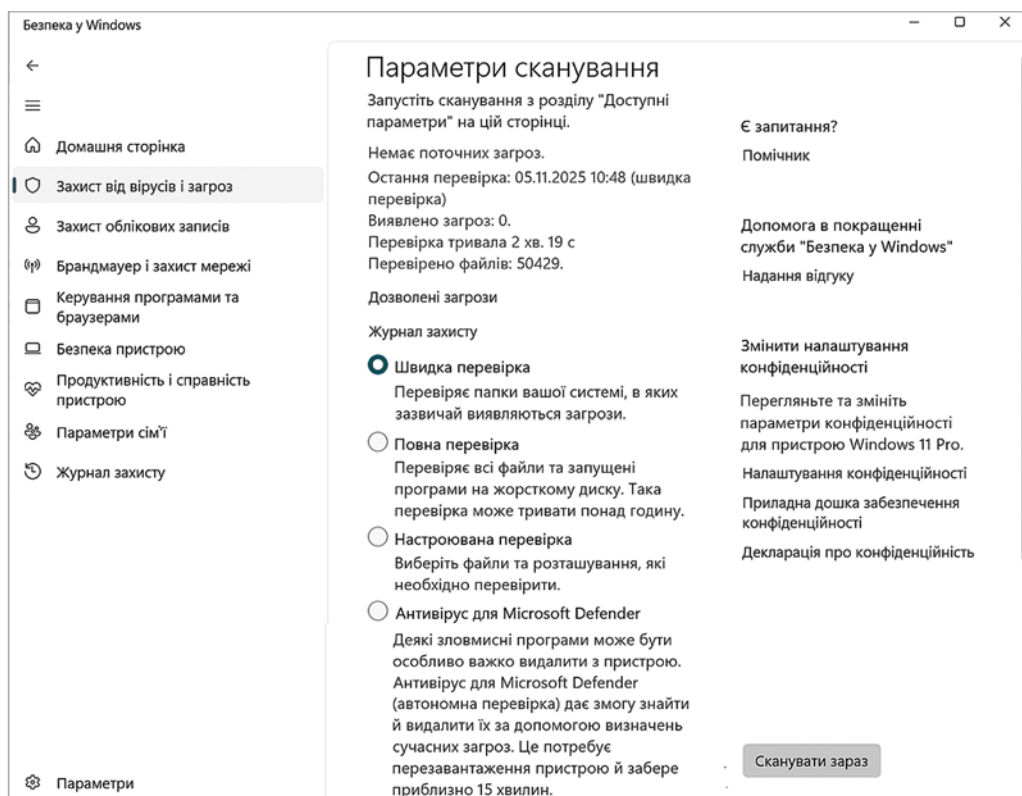
- захист у реальному часі;
- захист у хмарі;
- автоматичне надсилання зразків;
- захист від підробок.

Також можна перейти до змінення налаштувань доступу до певних папок на вашому комп'ютері, збереження критично важливих копій даних у хмарному сервісі **OneDrive**, додати або видалити дозволи на використання певних програм (файлів), змінити характер сповіщень про виявлені загрози.

Після вибору посилання **Параметри сканування** (мал. 5.6) на однійменній вкладці (мал. 5.7) можна:

- розпочати один з видів перевірки комп'ютера на наявність загроз (опис особливостей перевірки подано біля відповідних перемикачів) вибором відповідного перемикача та кнопки **Сканувати зараз**;
- переглянути короткий звіт про результати попередньої перевірки;
- переглянути журнал загроз, у якому відображено, які були загрози і які дії здійснила програма для їх знешкодження;
- переглянути, які дозволи надав користувач на використання програм, позначених антивірусною програмою як потенційно небезпечні

тощо.



Мал. 5.7. Вкладка **Параметри сканування**

ОСНОВНІ ДІЇ ДЛЯ ЗАХИСТУ ПЕРСОНАЛЬНИХ КОМП'ЮТЕРІВ ВІД ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Пригадайте

- Які ви знаєте загрози, що можуть виникнути під час роботи з комп'ютером?
- Які деструктивні дії можуть здійснювати шкідливі програми?

Щоб запобігти ураженню комп'ютера шкідливими програмами і зберегти дані, дотримуйтеся таких вимог:

- використовуйте ліцензійне програмне забезпечення;
- установіть комплексну антивірусну програму або включити і налаштуйте ту, що входить до складу операційної системи;
- регулярно здійснюйте повну перевірку комп'ютера на наявність шкідливих програм;
- постійно оновлюйте антивірусні бази, включіть автоматичне їх оновлення;
- регулярно проводьте резервне копіювання найбільш цінних даних;
- перед використанням перевіряйте всі змінні носії, отримані зі сторонніх джерел, антивірусною програмою;
- не використовуйте піратські («зламані») версії програм, генератори ключів (кейгени) або підозрілі «кряки», оскільки вони є одними з основних каналів розповсюдження шкідливих програм;
- не відкривайте вкладені до листів електронної пошти файли, якщо вони прийшли від невідомого кореспондента;
- не переходьте за підозрілими посиланнями, навіть якщо лист нібито від банку, відомої компанії або відомої вам людини з незвичним проханням. Завжди перевіряйте URL-адресу посилання;
- перевіряйте URL-адреси сайтів у адресному рядку браузера, переконуйтеся, що сайт використовує HTTPS (захищене з'єднання) і що доменне ім'я відповідає очікуваному;
- обмежте коло користувачів вашого комп'ютера, ознайомте їх з правилами антивірусного захисту, вимагайте їх неухильного дотримання.



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання. Перевірте на наявність шкідливих програм файли з папки **Завантаження**.

1. Відкрийте вкладку **Захист від вірусів і загроз** вікна **Безпека у Windows** антивірусної програми **Microsoft Defender** (**Пуск** ⇒ **Налаштування** ⇒ **Конфіденційність і безпека** ⇒ **Безпека у Windows** ⇒ **Захист від вірусів і загроз**).
2. Виберіть посилання **Параметри сканування**.
3. Установіть позначку перемикача **Налаштована перевірка**, а потім виберіть кнопку **Сканувати зараз**.
4. Виберіть для перевірки у вікні **Вибір папки** папку **Завантаження**, а потім кнопку **Вибір папки**.
5. Дочекайтеся завершення сканування та визначте:
 - Скільки знайдено файлів шкідливих програм?

- Скільки файлів шкідливих програм було знешкоджено?
- Скільки часу йшло сканування?

6. Закрийте вікно програми.



Найважливіше в цьому пункті

Крім корисних програм, які допомагають користувачеві опрацьовувати дані, існують і шкідливі програми. Для шкідливих комп'ютерних програм характерним є швидке розмноження та виконання ряду деструктивних дій, пов'язаних із викраденням, знищенням, зміненням та блокуванням доступу до даних користувача, сповільнення роботи комп'ютера, використання ресурсів комп'ютера для організації атак на сервери в мережі та для несанкціонованого майнінгу, крадіжки коштів тощо.

Шкідливі комп'ютерні програми поділяють за особливостями розповсюдження і функціонування на такі види: **комп'ютерні віруси, хробаки комп'ютерних мереж, троянські програми, бекдори, рекламні модулі**. За цілями діяльності шкідливі програми поділяють на: **програми-вимагачі, руткіти, завантажувачі, криптомайнери, викрадачі даних** та інші.

Зазвичай дії шкідливих програм передбачають три етапи: *проникнення та закріплення в системі, розповсюдження, виконання шкідливих дій*.

Для захисту даних і пристроїв комп'ютера від шкідливих програм використовують спеціальне програмне забезпечення – антивірусні програми. Сучасні антивірусні програми – це комплексні програми, що мають набір модулів для захисту від різних загроз: *файловий монітор, поведінковий аналізатор, поштовий фільтр, евристичний аналізатор, брандмауер* та інші.

В останніх версіях операційної системи **Windows** програма антивірусного захисту **Microsoft Defender** входить до складу ОС, і перевірка системи на ураження шкідливими програмами здійснюється автоматично з певною періодичністю.

Щоб запобігти ураженню комп'ютера шкідливими програмами і зберегти дані, користувач повинен дотримуватися правил кібербезпеки.



Дайте відповіді на запитання

1. Що таке *шкідливе програмне забезпечення*?
2. На які деструктивні дії спрямовані шкідливі програми?
3. Які типи шкідливих програм ви знаєте?
4. Як шкідливі програми можуть потрапити на комп'ютер?
5. Для чого потрібне антивірусне програмне забезпечення?



Обговоріть і зробіть висновки

1. Порівняйте принципи дії вірусу, троянської програми та хробака. Які деструктивні дії можуть здійснювати програми цих типів?
2. Як ви вважаєте, що є ефективнішим – профілактика чи лікування зараження? Чому?
3. Як оновлення операційної системи допомагає захиститися від вірусів?
4. Як можна виявити, що програма-шпигун діє на вашому комп'ютері?
5. Які заходи допомагають запобігти поширенню вірусів у комп'ютерній мережі закладу середньої освіти? У вашій домашній комп'ютерній мережі?



Виконайте завдання

1. Укажіть, які деструктивні дії не можуть виконувати шкідливі програми:



- а) знищувати файли;
- б) знищувати мікросхеми оперативної пам'яті;
- в) пересилати дані на інший комп'ютер;
- г) змінювати логічну структуру носія даних;
- д) відтворювати звукові та відеоефекти.

2. Виконайте антивірусну перевірку зовнішнього запам'ятовуючого пристрою, запропонованого вчителем / учителькою, з використанням **Microsoft Defender**. Результати запишіть у зошит.

3. Запишіть у текстовий файл словесний алгоритм перевірки комп'ютера на наявність шкідливих програм. Збережіть цей алгоритм у файлі з іменем **завдання 5.3.3** у вашій папці.

4. Запустіть на виконання антивірусну програму операційної системи **Windows** і виконайте перевірку вашого комп'ютера автономним модулем **Microsoft Defender**. Визначте:

- а) у яких випадках використовується цей модуль;
- б) які операції здійснюються під час перевірки цим модулем і чому відбувається перезавантаження операційної системи;
- в) скільки часу тривала перевірка і які її результати.

Результати роботи запишіть у текстовий документ та збережіть у файлі з іменем **завдання 5.3.4** у вашій папці.



5. Запустіть повну перевірку на наявність загроз на вашому комп'ютері з використанням інстальованої антивірусної програми. Дослідіть:

- а) Яка програма використовується на вашому комп'ютері (смартфоні, планшетному ПК) для захисту від ураження вірусами та шкідливими програмами?
- б) Які дані про результати перевірки надає програма?
- в) Чи були знайдені шкідливі або підозрілі програми (файли)? Якщо так, то які дії пропонує програма для таких програм (файлів)?
- г) Якщо були знайдені шкідливі програми, то до якого типу вони належать?

Результати дослідження запишіть у текстовий документ та збережіть у файлі з іменем **завдання 5.3.5** у вашій папці.

6. Підготуйте презентацію за однією з тем: «Сучасні засоби антивірусного захисту», «Що можуть і чого не можуть комп'ютерні віруси», «Комп'ютерні віруси: міфи і реальність» або «Історія комп'ютерних вірусів».

Створену презентацію збережіть у файлі з іменем **завдання 5.3.6** у вашій папці.

7. Зареєструйтеся та пройдіть безкоштовний онлайн-курс «Основи інформаційної безпеки» за посиланням <https://zillya.ua/prometheus>. Копію сертифіката про проходження курсу надішліть вашому вчителю / учительці.



Готуємось до вивчення нового матеріалу

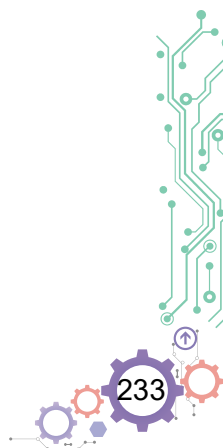
- 1. Як зберігаються дані в інформаційних системах, наприклад в електронних каталогах бібліотек? Як здійснюється пошук потрібних даних у цих каталогах?
- 2. Що таке *посилання в електронній таблиці*? Для яких цілей його використовують?



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу «Інформаційна безпека» (початковий, середній, достатній, високий рівень).

- Я можу пояснити, що таке *інформаційна безпека*.
- Я можу пояснити, що таке *інформаційні загрози*.
- Я розумію, які інформаційні загрози виникають у процесі роботи комп'ютера, під'єданого до комп'ютерної мережі.
- Я можу пояснити, що таке *цифровий слід* і внаслідок чого він виникає та як його зменшити.
- Я вмію реалізувати заходи для зменшення активного та пасивного цифрового сліду.
- Я можу навести приклади утворення цифрового «сміття» та заходів, спрямованих на його зменшення.
- Я періодично виконую дії для зменшення цифрового сліду та видалення цифрового «сміття».
- Я знаю, з якою метою і на якій законодавчій базі здійснюється захист персональних даних.
- Я дотримуюсь правил для забезпечення захисту персональних даних.
- Я розпізнаю ризики та загрози моєму фізичному, психічному і соціальному здоров'ю через користування цифровими пристроями та інтернетом та намагаюсь їх уникати.
- Я знаю, які права людини є в інформаційному середовищі.
- Я дію з дотриманням прав людини в інформаційному середовищі.
- Я знаю правила захисту персональних даних і дотримуюсь їх.
- Я знаю різні класифікації шкідливих комп'ютерних програм та вмію пояснити, які деструктивні дії можуть здійснювати ці програми.
- Я знаю і намагаюсь дотримуватись заходів захисту від шкідливого програмного забезпечення.
- Я вмію використовувати та встановлювати значення властивостей програм для боротьби зі шкідливим програмним забезпеченням.



БАЗИ ДАНИХ



У цьому розділі ви здобудете нові, а також поглибите і розширите свої знання й удосконалисте навички з таких тем:

- ▶ бази даних та їх призначення;
- ▶ комп'ютерні програми для створення та редагування баз даних;
- ▶ особливості додавання, видалення, редагування даних у базах даних;
- ▶ упорядкування, пошук та фільтрування даних у базах даних;
- ▶ створення простих запитів на вибірку даних в автоматизованому режимі.

6.1. БАЗИ ДАНИХ. СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ

У цьому пункті йтиметься про:

- ▶ призначення баз даних;
- ▶ програмне забезпечення для створення та редагування баз даних;
- ▶ основні об'єкти баз даних.

ПОНЯТТЯ ПРО БАЗИ ДАНИХ



Пригадайте

- Що таке *дані*? Які дані опрацьовують і зберігають у комп'ютерних системах?
- Як упорядковуються дані під час зберігання на носіях даних? • Які дані зберігаються в базі даних «Укрзалізниці», пошукових системах інтернету, електронних бібліотеках тощо?

За нашого часу щоденно накопичуються просто вражаючі обсяги даних. У комп'ютерних системах ще десять років тому обсяги даних на персональних комп'ютерах вимірювалися гігабайтами (нагадуємо, що 1 гігабайт = 2^{30} байтів = 1 073 741 824 байтів). Прогнозується, що на кінець 2025 року обсяги даних у світі складуть від 175 до 181 зетабайт (ЗБ) (1 зетабайт = 2^{70} байтів = 1 180 591 620 717 410 000 000 байтів). 2027 року цей обсяг, за прогнозом аналітиків з **International Data Corporation** і **Statista**, зросте до 230 ЗБ.

На сьогодні основними джерелами нових даних є системи, які пов'язані з *інтернетом речей, штучним інтелектом, ігровою індустрією та медіа, хмарними обчисленнями*. Людство вступило в епоху, коли обсяги даних є викликом не тільки для інфраструктури зберігання, а й для розробки нових методів опрацювання даних у реальному часі.

Щодня кожна людина генерує дедалі більше нових даних, коли:

- пересилає повідомлення комп'ютерною чи мобільною мережею;
- ставить «лайк» під фотографією в соціальній мережі;
- прокладає маршрут поїздки з використанням геолокації;
- оплачує товари в касі магазину з карткового рахунку та отримує чек

тощо.

Процес лавиноподібного збільшення обсягу даних у світі із середини 60-х років ХХ ст. отримав назву **інформаційного вибуху**. Водночас збільшення обсягів даних у світі, як правило, відбувається за рахунок невпорядкованих даних у комп'ютерних мережах (до 95 %) та їхнім простим дублюванням. Частина цих даних уже втратила свою актуальність і корисність та утворює **інформаційне «сміття»**, але продовжує зберігатися на носіях даних відокремлених комп'ютерів і в мережі.

Вчені запевняють, що зберігання великих обсягів даних виправдано лише за умови, якщо пошук потрібних даних здійснюється швидко і подаються вони в доступній для розуміння та використання формі. Ці умови забезпечують сучасні технології зберігання даних. Основою цих технологій є комп'ютеризовані (електронні) **бази даних** (БД).

База даних – це впорядкований за певними правилами набір взаємопов'язаних даних.

Основним призначенням бази даних є систематизоване зберігання даних і надання доступу до них користувачеві або комп'ютерній програмі для забезпечення швидкого пошуку потрібних даних. Електронні бази даних активно використовують у банківській сфері (облік клієнтів, їхніх рахунків, проведених виплат і руху коштів), бібліотеках (мал. 6.1), пошукових системах інтернету, системах обліку та здійснення оплати за користування електроенергією, газом, холодною та гарячою водою; під час проведення зовнішнього незалежного оцінювання учнів і в інших галузях діяльності людини.

Державна бібліотека України для юнацтва

Пошук **Відібрані** Персональний кабінет

Автор:

Назва документа:

Анотація:

Тема:

Рік видання: -

Видавництво:

Мова:

Вид документа:

Електронна версія: ☐

Сортування: Автор Назва документа --

За зростанням За зростанням За зростанням

Кількість документів на сторінці 20

Шукати **Очистити**

Пошук теми в класифікаторах та каталогах

Класифікатори та каталоги

- [Новинки у фонд](#)
- [Література іноземними мовами](#)
- [АББД "Молодь України"](#)
- [Виставки](#)
- [Калейдоскоп полій та дат](#)
- [Цифрова колекція](#)
- [Подорожі дослідників та мандрівників](#)
- [Підліткова література](#)
- [Підручники та посібники](#)

Мал. 6.1. Сторінка пошуку відомостей у базах даних на сайті Державної бібліотеки України для юнацтва

База даних є досить складним за своєю структурою об'єктом, тому перед створенням вона заздалегідь проєктується – створюється **модель бази даних**.

Основні типи моделей баз даних подано на малюнку 6.3.

ПОНЯТТЯ ПРО СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ



Пригадайте

- Чи використовували ви бази даних? Які електронні бази даних ви знаєте?
- Чи створювали ви власні бази даних? Чи вносили ви якусь інформацію в існуючі бази даних?
- Які бази даних містить ваш смартфон?

Для роботи з комп'ютерними базами даних використовують спеціальні програми – *системи керування базами даних (СКБД)*.

Системи керування базами даних – це прикладні комп'ютерні програми, призначені для створення, збереження та використання баз даних. До поширених СКБД належать **MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, Interbase** та інші.

Залежно від виду зв'язків між даними, які використовуються в СКБД, їх поділяють на ієрархічні, мережні, реляційні, об'єктно-реляційні тощо.



Для тих, хто хоче знати більше

Інші види класифікації баз даних

Залежно від розміщення баз даних і компонентів, які забезпечують їх опрацювання, СКБД поділяють на *локальні* та *розподілені*. Локальні СКБД забезпечують зберігання та опрацювання даних на локальному комп'ютері, а в розподілених – дані можуть зберігатися й опрацьовуватися на різних комп'ютерах у локальній або глобальній мережі.

Залежно від способу доступу до бази даних СКБД розподіляють на *файл-серверні*, *клієнт-серверні* та *інтегровані (вбудовані)*. У файл-серверних СКБД файли з даними розміщуються на сервері, а на кожному клієнтському комп'ютері встановлено повну версію СКБД. Доступ із клієнтського комп'ютера до даних на файл-сервері здійснюється з використанням мережі.

Найбільш широко в останні роки розвивається клієнт-серверна технологія використання баз даних (мал. 6.2). За цієї технологією на сервері встановлюється серверна версія СКБД, на ньому ж зазвичай розміщена і база даних. На клієнтських комп'ютерах встановлюються тільки невеликі за обсягом і функціональністю клієнтські версії СКБД, у завдання яких входить формування запитів на опрацювання даних та виведення результату опрацювання, надісланого із сервера. Усі операції з даними здійснюються на сервері. За клієнт-серверною технологією організовано роботу залізничних кас продажу квитків, роботу банкоматів тощо.

Інтегровані (вбудовані) СКБД використовуються як складові інших програмних продуктів, наприклад електронних енциклопедій, словників, пошукових систем тощо. Ці системи не потребують окремої інсталяції, можуть мати обмежений набір функцій з керування базою даних, наприклад не забезпечувати оновлення даних. Доступ до даних здійснюється з використанням засобів прикладної програми, у яку інтегрована СКБД.



Мал. 6.2. Схема використання бази даних за клієнт-серверною технологією

Схему основних класифікацій СКБД наведено на малюнку 6.3.



Мал. 6.3. Схема основних класифікацій систем керування базами даних

СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАНИХ LIBREOFFICE BASE



Пригадайте

• Яке прикладне програмне забезпечення ви знаєте? • Для яких цілей використовують прикладне програмне забезпечення загального призначення?

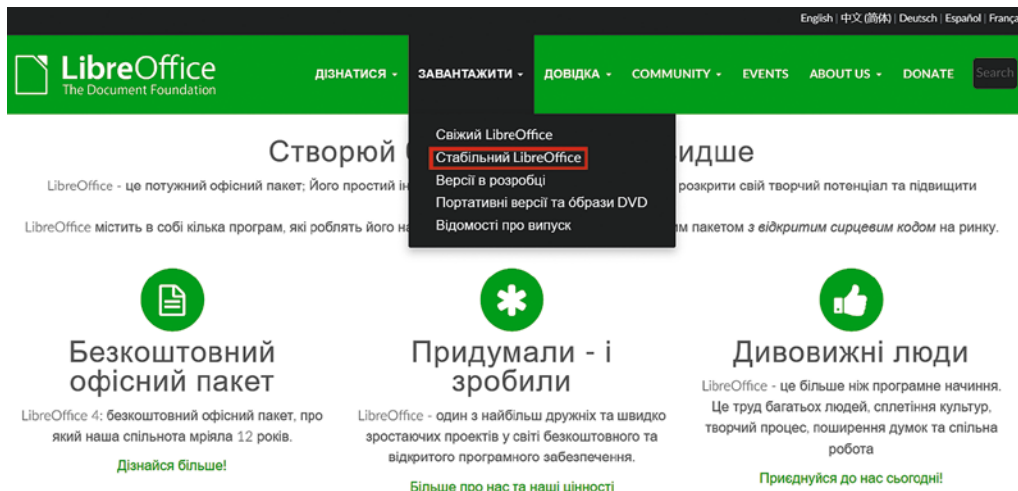
Система керування базами даних **Base** (англ. *Base* – база) із пакета прикладних програм загального призначення **LibreOffice** – це СКБД, що передбачає роботу з порівняно невеликими за обсягом базами даних для особистого користування та для невеликих організацій.

Розробники **LibreOffice** – члени співтовариства **The Document Foundation** (англ. – фундація (фонд) документів), стверджують, «що користувачі повинні мати свободу запускати, копіювати, розповсюджувати, вивчати, змінювати і вдосконалювати програмне забезпечення, яке ми розповсюджуємо... вільне програмне забезпечення – це, перш за все, питання свободи, а не ціни» (адреса ресурсу: <https://www.libreoffice.org/about-us/who-are-we>).

Українськомовну версію із сайту розробників **LibreOffice** можна завантажити за адресою <https://uk.libreoffice.org> (мал. 6.4).

Для інсталяції комплексу програм **LibreOffice** потрібно перейти на сторінку завантаження (посилання **Завантажити / Download**). Буде запропоновано інстальювати, як правило, найновішу або стабільну на поточний момент версію цієї програми. Рекомендуємо завантажувати стабільну версію – **Стабільний LibreOffice** (мал. 6.4).

Для користувачів, що використовують на своїх ПК більш ранні версії операційних систем, варто встановити одну з попередніх версій **LibreOffice**. Наприклад, для операційної системи **Windows 7** можна завантажити інсталятор **Windows x86_64** (Потрібна 7 або новіша) (мал. 6.5).



Мал. 6.4. Сторінка українськомовної версії сайту розробників LibreOffice

Операційні системи

LibreOffice 25.8.1 ідоступний для таких операційних систем/архітектур:

- ☒ Linux Aarch64 (deb)
- ☒ Linux Aarch64 (rpm)
- ☒ Linux x64 (deb)
- ☒ Linux x64 (rpm)
- ☒ macOS (Aarch64/Apple Silicon)
- ☒ macOS x86_64 (Потрібна 10.14 або новіша)
- ☒ Windows (32 bit, deprecated)
- ☒ Windows Aarch64
- ☒ Windows x86_64 (Потрібна 7 або новіша)

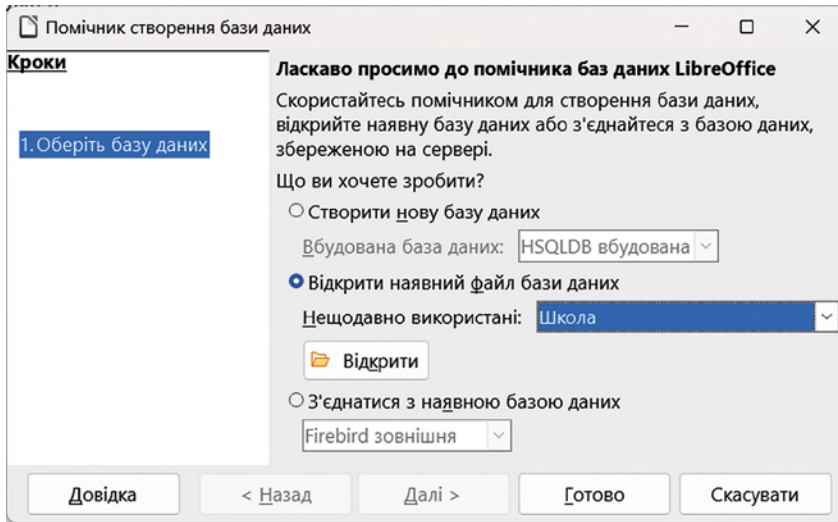
Мал. 6.5. Список можливих варіантів інсталяторів LibreOffice

Після інсталяції **LibreOffice** СКБД **Base** можна запустити на виконання кількома способами. Наприклад, виконати **Пуск** ⇒ **Усі** ⇒ **LibreOffice** ⇒ **LibreOffice Base** або двічі клацнути на значку вже існуючого файлу бази даних. Для збереження баз даних **Base** використовує як основний формат файлів **ODB** (англ. *Open Document Base* – база відкритих документів) і надає користувачу інструменти для імпортування та експортування даних з інших типів файлів баз даних.

У першому випадку після запуску програми **Base** на екрані відкривається початкове вікно програми – **Помічник створення бази даних** (мал. 6.6).

Щоб відкрити вже створену базу даних, слід у цьому вікні:

1. Установити перемикач **Відкрити наявний файл бази даних**.
2. Вибрати потрібну базу даних, використовуючи список **Нещодавно використані**: або кнопку **Відкрити** (вибрати у вікні провідника потрібну папку та файл).
3. Вибрати кнопку **Готово**.

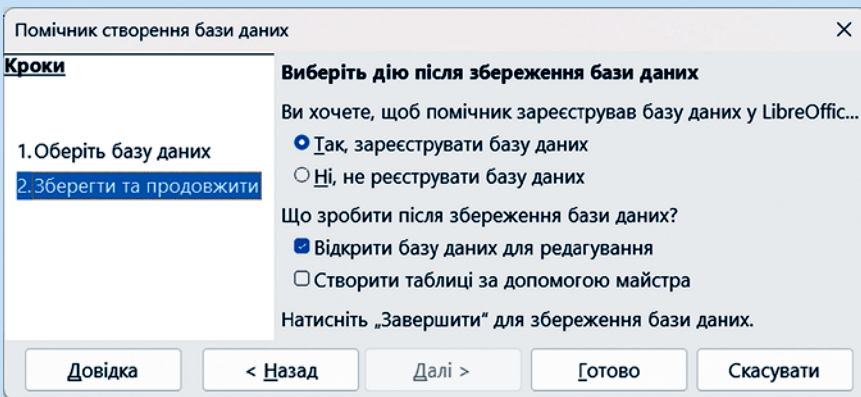


Мал. 6.6. Вікно Помічник створення бази даних

**Для тих, хто хоче знати більше.**

Щоб створити нову локальну базу даних у **LibreOffice Base**, слід після запуску програми:

1. Вибрати у вікні **Помічник створення бази даних** перемикач **Створити нову базу даних**.
2. Вибрати кнопку **Далі**.
3. Установити перемикач і позначку прапорця так, як це подано на малюнку 6.7.

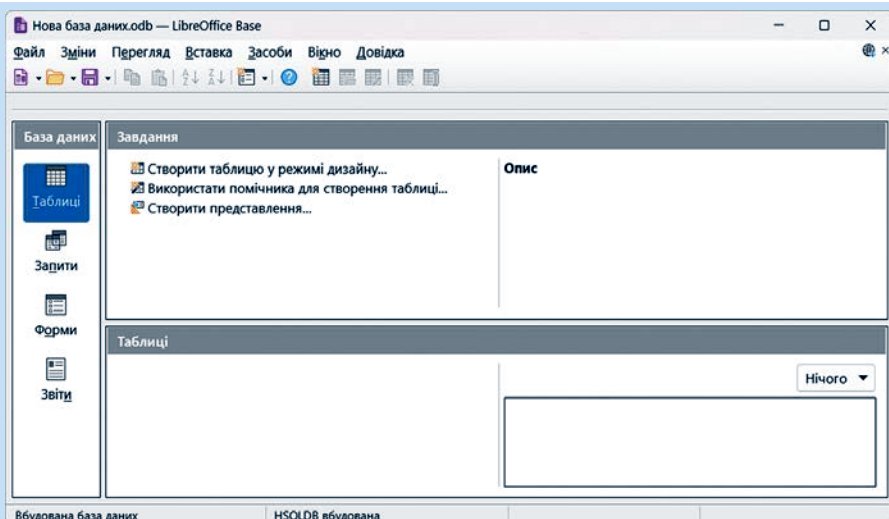


Мал. 6.7. Вікно другого кроку створення нової бази даних

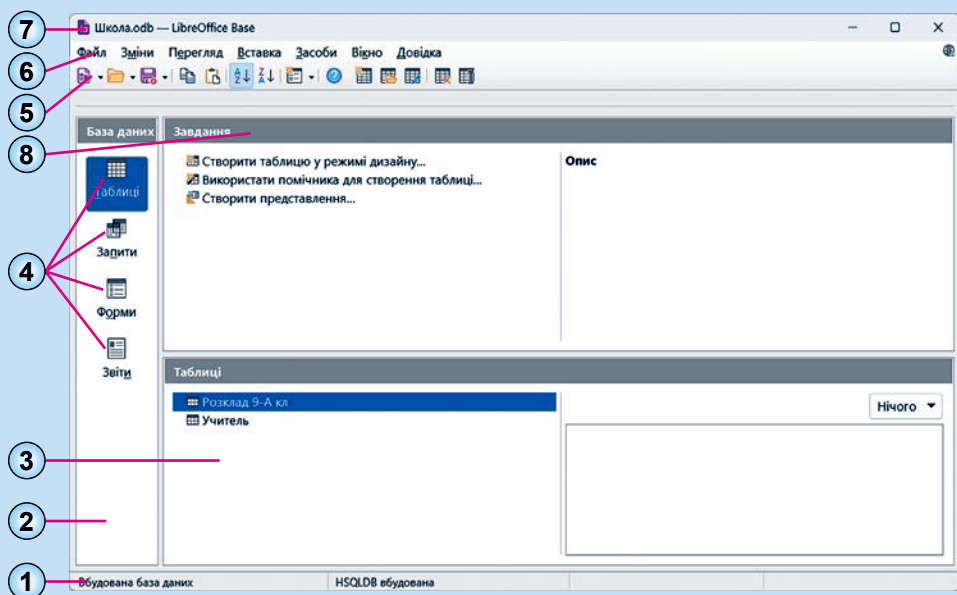
4. Вибрати кнопку **Готово**.
5. Указати папку для збереження та ім'я нової бази даних.

Після цього на екран буде виведено вікно створеної бази даних у СКБД **Base** (мал. 6.8) з вибраною групою об'єктів **Таблиці**. У цій базі даних немає жодного об'єкта і потрібно обов'язково створити перший об'єкт – таблицю з **використанням помічника створення таблиць** або **створити таблицю в режимі дизайну**.

Після відкриття раніше створеної бази даних у **Base** вікно системи керування базами даних буде мати вигляд, поданий на малюнку 6.9.



Мал. 6.8. Вікно нової бази даних **Школа** в LibreOffice Base



Мал. 6.9. Вікно бази даних **Школа** в LibreOffice Base:

- 1 – рядок стану; 2 – панель типів об'єктів бази даних; 3 – панель групи об'єктів **Таблиці**; 4 – кнопки відкриття панелей зі списком об'єктів обраного типу;
5 – панель інструментів; 6 – рядок меню; 7 – заголовок вікна;
8 – панель завдань для поточних об'єктів БД

Основними об'єктами в базах даних **Base** є:

- **таблиці** – для зберігання даних і встановлення зв'язків між ними;
- **форми** – для введення і перегляду даних;
- **запити** – для здійснення вибірки даних, що відповідають певним умовам;
- **звіти** – для подання даних у потрібному користувачеві вигляді, з узагальненням потрібних даних.

Кожна база даних має хоча б одну таблицю, у якій зберігаються дані.

На прикладі бази даних **Школа** (мал. 6.9), яку можна завантажити, наприклад, за адресою **Розділ 6\Школа.odt**, ознайомимося з інтерфейсом **Base** та окремими операціями над об'єктами бази даних.

Після відкриття створеної раніше бази даних поточною є група **Таблиці** (кнопка **Таблиці**  вибрана).

На панелі **Таблиці** (мал. 6.10) розміщено перелік створених у цій базі даних таблиць. У цьому разі є дві таблиці: **Розклад 9-А кл** та **Учитель**.

У цьому самому поданні ми можемо виконати певні операції зі створення нових таблиць, використовуючи посилання на панелі **Завдання** (мал. 6.9, 8).

Для перегляду списків інших об'єктів слід вибрати потрібну кнопку – **Запити**



Мал. 6.10. Панель **Таблиці**

Для відкриття потрібного об'єкта, наприклад таблиці **Розклад 9-А кл**, потрібно двічі клацнути по назві цього об'єкта на панелі **Таблиці**. Для закриття вікна поточного об'єкта слід вибрати кнопку закриття вікна  або виконати послідовність **Файл** ⇒ **Закрити**.

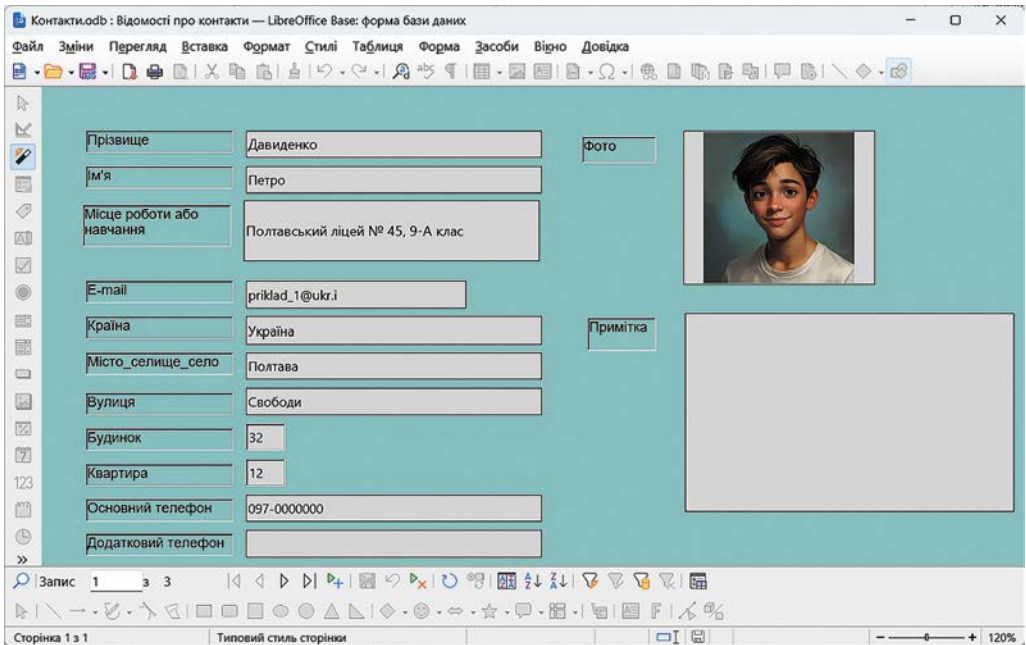


Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання 1. Дослідіть базу даних у СКБД LibreOffice Base.

- Відкрийте базу даних **Контакти** у СКБД LibreOffice Base. Для цього:
 - Запустіть програму **Base**, виконавши **Пуск** ⇒ **Усі** ⇒ **LibreOffice** ⇒ **LibreOffice Base**.
 - Установіть позначку перемикача **Відкрити наявний файл бази даних**.
 - Виберіть кнопку **Відкрити**.
 - Виберіть у вікні **Відкриття файлу** файл бази даних **Контакти.odt** у папці, яку вкаже вчитель / учителька, наприклад **Розділ 6\Пункт 6.1**.
- Збережіть файл бази даних під іменем **Контакти_Прізвище.odt** у вашій папці, для чого виконайте **Файл** ⇒ **Зберегти як**.
- Виберіть на панелі типів об'єктів БД (мал. 6.9, 4) кнопку **Таблиці**. Ознайомтеся з даними, що вже містяться в таблиці: дані про скількох осіб занесено в таблицю, які відомості містяться про кожну особу.
- Виберіть на панелі типів об'єктів БД (мал. 6.9, 4) кнопку **Форми**.
- Відкрийте в окремому вікні форму **Відомості про контакти** (мал. 1). Для цього двічі клацніть на імені форми на панелі **Форми**.
- Перегляньте відомості про осіб, занесених у базу даних, використовуючи кнопки переходу **Наступний запис**  та **Попередній запис**  у нижній частині вікна форми.



Мал. 1. Вікно форми **Відомості про контакти**

7. Виберіть кнопку **Додати запис** .
8. Уведіть у поле **Прізвище** прізвище одного / однієї з ваших одно-
класників / однокласниць.
9. Уведіть відповідні дані в поля **Ім'я**, **Місце роботи_навчання** (уве-
діть назву вашого закладу освіти і клас), **Е-mail**, **Мобільний те-
лефон**, **Місто_селище_село** (уведіть назву вашого населеного
пункту). Для переходу до наступного поля, якщо введення здійс-
нюється з клавіатури, можна використовувати клавішу **Tab**.
10. Збережіть введені дані одним зі способів:
 - виберіть кнопку **Зберегти запис**  на Панелі інструментів;
 - виберіть кнопку **Наступний запис**  у нижній частині вікна
форми.
11. Перейдіть до введення даних про іншого учня / ученицю, для чого
виберіть кнопку **Наступний запис**  (якщо ви скористалися пер-
шим із наведених способів збереження даних).
12. Уведіть дані про іншого учня / ученицю вашого класу.
13. Перегляньте введені дані, використовуючи кнопки **Наступний**
запис , **Попередній запис** , **Перший запис** , **Останній**
запис .
14. Видаліть третій запис, для чого зробіть поточним цей запис і вибе-
ріть кнопку **Видалити запис** .

15. Відредагуйте дані про заклад навчання у другому записі з *Полтавська гімназія № 97, 9 клас* на назву вашого закладу освіти і клас. Для цього зробіть поточним відповідне поле запису 2, видаліть наявний текст і введіть нові дані.
16. Перегляньте введені дані в таблиці бази даних. Для цього:
 1. Закрийте вікно форми **Відомості про контакти**.
 2. Виберіть на панелі типів об'єктів БД (мал. 6.9, 4) кнопку **Таблиці**.
 3. Відкрийте в окремому вікні таблицю **Відомості про контакти**. Для цього двічі клацніть на імені таблиці на панелі **Таблиці**.
17. Переконайтеся в тому, що введені вами дані збереглися в цій таблиці, а видалені – зникли.
18. Уведіть дані про ще одну особу, для чого використайте нижній порожній рядок.
19. Послідовно закрийте вікна таблиці, бази даних і програми **Base**.



Найважливіше в цьому пункті

База даних – це впорядкований за певними правилами набір взаємопов'язаних даних. Основним призначенням бази даних є систематизоване зберігання даних і надання доступу до них користувачеві або комп'ютерній програмі для забезпечення швидкого пошуку потрібних даних.

Системи керування базами даних (СКБД) – це прикладні комп'ютерні програми загального призначення, які використовуються для створення, збереження та використання баз даних. До поширених СКБД належать **MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, Interbase** та інші.

Система керування базами даних **Base** із пакета прикладних програм **LibreOffice** – це СКБД, що передбачає роботу з порівняно невеликими за обсягом базами даних для особистого користування та для невеликих організацій.

Для збереження баз даних у **LibreOffice Base** використовується як основний формат файлів **ODB**, користувач може також використати інструменти для імпортування та експортування даних з інших типів файлів баз даних.

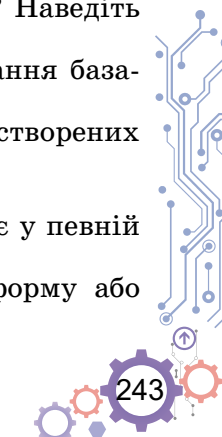
Основними об'єктами в базах даних **Base** є:

- **таблиці** – для зберігання даних і встановлення зв'язків між ними;
- **форми** – для введення і перегляду даних;
- **запити** – для здійснення вибірки даних, що відповідають певним умовам;
- **звіти** – для подання даних у потрібному користувачеві вигляді, з узагальненням потрібних даних.



Дайте відповіді на запитання

1. Для чого потрібно впорядковане зберігання даних?
2. Що таке *база даних*? Де використовують бази даних? Наведіть приклади.
3. Що таке *СКБД*? Для чого призначена система керування базами даних **Base**?
4. Яке призначення основних об'єктів баз даних, створених у СКБД **Base**?
5. Як відкрити вже існуючу базу даних у СКБД **Base**?
6. Як відобразити список таблиць або список форм, які є у певній базі даних у СКБД **Base**?
7. Як відкрити об'єкт бази даних, наприклад певну форму або таблицю?





Обговоріть і зробіть висновки

1. Які бази даних, на вашу думку, будуть корисні у вашій школі, як їх зможуть використовувати учасники / учасниці освітнього процесу?
2. Яку базу даних ви запропонуєте створити для власної бібліотеки? Які відомості має зберігати подібна база даних?
3. Що спільного та відмінного у роботі з даними у базах даних та електронних таблицях?



Виконайте завдання

1. Об'єднайтесь у малі групи і дослідіть особливості використання баз даних у таких галузях:

- | | |
|---|---------------|
| а) освіта; | б) медицина; |
| в) торгівля; | г) транспорт; |
| д) комп'ютерні мережі, у тому числі інтернет. | |

Для дослідження використайте матеріали інтернету. За результатами дослідження підготуйте повідомлення. Форму подання повідомлення доберіть самостійно.

2. Підготуйте повідомлення (презентацію) про роль В. М. Глушкова в розвитку напрямків інформатики, пов'язаних з використанням баз даних.

3. Відкрийте файл бази даних (наприклад, Розділ 6\Пункт 6.1\зразок 6.1.3.odt) та:

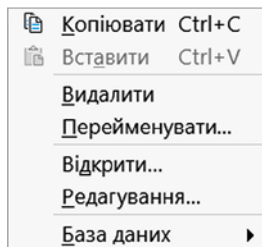
1. Виведіть список таблиць бази даних. Визначте, які таблиці створено в цій базі даних.
2. Відкрийте таблицю **Навчальні комп'ютерні програми**.
3. Визначте, які властивості містить таблиця. Скільки таких властивостей?
4. Визначте, скільки та яких об'єктів усього включено до бази даних.
5. Результати дослідження включіть у текстовий файл, доповнивши його скріншотами об'єктів бази даних і поясненнями до них.
6. Збережіть текстовий файл у вашій папці з іменем **завдання 6.1.3**.

4. Відкрийте файл бази даних (наприклад, Розділ 6\Пункт 6.1\зразок 6.1.4.odt) та:

1. Відкрийте форму **Відомості про монітори**.
2. Уведіть у форму дані про три монітори, відомості про які знайдіть в інтернеті.
3. Видаліть перший запис про монітори.
4. Відкрийте таблицю **Відомості про монітори**.
5. Переконайтеся, що таблиця містить введені вами дані.
6. Уведіть дані про ще один монітор, використовуючи цю таблицю. Який із способів введення даних, у форму чи у таблицю, зручніший?
7. Збережіть змінену базу даних у вашій папці з іменем **завдання 6.1.4**.

5. Відкрийте файл бази даних (наприклад, Розділ 6\Пункт 6.1\зразок 6.1.5.odt). Відкрийте існуючу форму **Принтери** та змініть її дизайн: додайте логотип, змініть колір тла, упорядкуйте розміщення

об'єктів. Для переходу в режим редагування використайте команду контекстного меню **Редагування** (мал. 6.11). Для визначення послідовності дій для виконання завдання скористайтеся довідкою програми або відомостями з інтернету. Збережіть змінену базу даних у вашій папці з іменем **завдання 6.1.5**.



Мал. 6.11. Контекстне меню форми



Готуємось до вивчення нового матеріалу

1. Що таке *система керування базами даних*?
2. Як упорядкувати дані в електронній таблиці?
3. Як з використанням фільтрів в електронних таблицях відбувається пошук даних, що відповідають певним умовам?



6.2. ПОНЯТТЯ ТАБЛИЦІ, ПОЛЯ, ЗАПISУ, КЛЮЧА ТАБЛИЦІ БАЗИ ДАНИХ. СОРТУВАННЯ ТА ФІЛЬТРУВАННЯ ДАНИХ. ЗАПИТИ

У цьому пункті йтиметься про:

- ▶ поняття таблиці, поля, запису, ключа таблиці в базах даних;
- ▶ сортування та фільтрування даних у базах даних;
- ▶ автоматизоване створення запитів на вибірку даних.

ПОНЯТТЯ ТАБЛИЦІ, ПОЛЯ, ЗАПISУ, КЛЮЧА ТАБЛИЦІ В БАЗАХ ДАНИХ



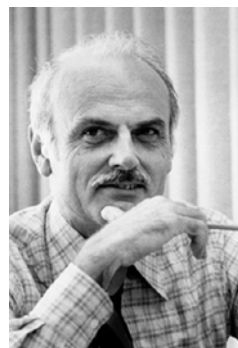
Пригадайте

- Як подаються дані в електронних таблицях, як позначаються в них рядки і стовпці?
- Які ви знаєте основні об'єкти бази даних? Для чого вони призначені?
- Як можна вводити і змінювати дані в таблицях баз даних?

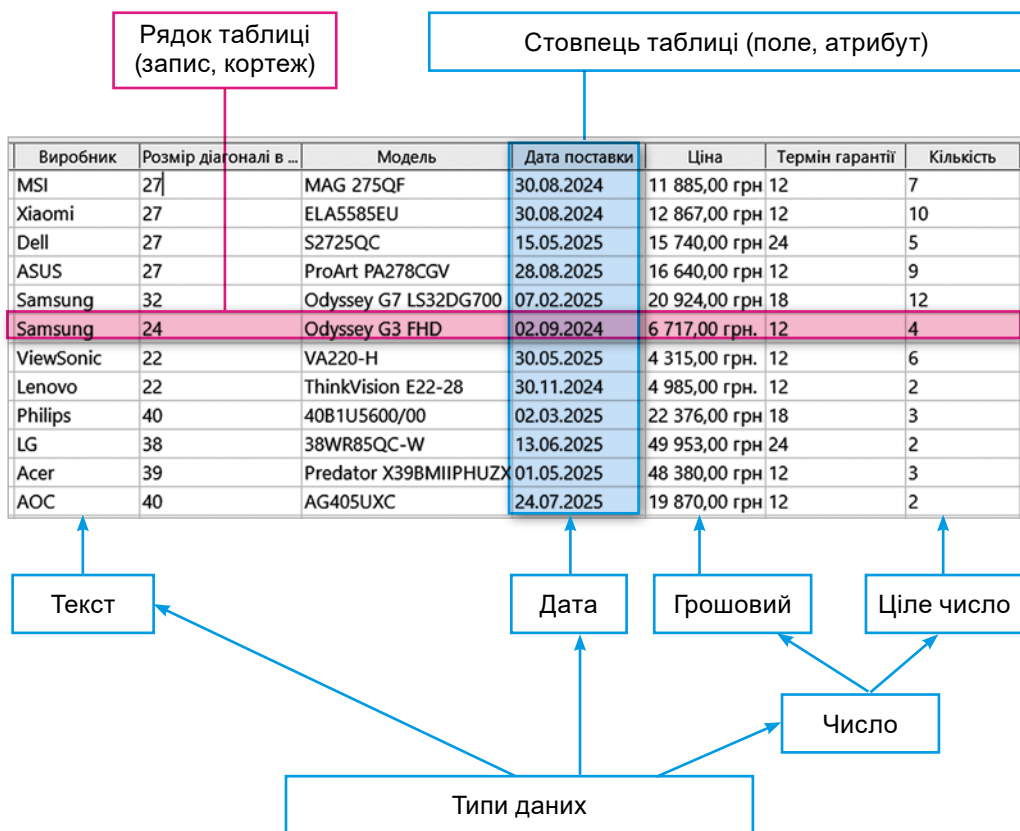
Сучасні бази даних за структурою упорядкування даних належать до реляційних або до об'єктно-реляційних. Структуру реляційної бази даних запропонував 1970 року математик і програміст корпорації IBM **Едгар Франк Кодд** (1923–2003) (мал. 6.12). Для пришвидшення роботи з базами даних, спрощення опису об'єктів і зв'язків між ними 1970 року було запропоновано реляційну модель даних.

Основним об'єктом реляційної бази даних є **таблиця**. У таблицях кожен рядок містить набір значень властивостей одного з об'єктів певної множини. Кожен стовпець таблиці містить набір значень певної властивості об'єктів цієї множини. Об'єкти множини в таблиці мають спільні властивості та відрізняються тільки значеннями властивостей. Ознайомимося з основними поняттями реляційної бази даних на прикладі таблиці **Монітори**, поданої на малюнку 6.13.

Рядок таблиці реляційної бази даних називають **записом**. Запис містить значення властивостей одного об'єкта із множини об'єктів одного виду. Наприклад, у таблиці **Монітори** (мал. 6.13) шостий зверху запис містить такий набір даних: *Samsung; 24; Odyssey*



Мал. 6.12.
Едгар Франк Кодд



Мал. 6.13. Таблиця **Монітори** бази даних

G3 FHD; 02.09.2024; 6 717 грн; 12; 4. Ці дані є значеннями відповідних властивостей одного об'єкта множини **Монітори**: **Виробник**, **Розмір діагоналі**, **Модель**, **Дата поставки**, **Ціна**, **Термін гарантії**, **Кількість**. Інший запис містить набір значень тих самих властивостей, але вже іншого об'єкта – іншого монітора. У попередньому пункті ви вже мали можливість заповнювати й оновлювати записи про значення властивостей окремих елементів множин.

Стовпець таблиці реляційної бази даних називають **полем**. Поле має ім'я, яке пов'язане з назвою властивості. Поле містить значення однієї властивості всіх об'єктів множини. Дані, що містяться в одному полі таблиці, є однотипними. Для кожного поля під час проектування таблиці бази даних встановлюється тип даних. Наприклад, для поля **Виробник** (мал. 6.13) встановлено *текстовий* тип даних, для полів **Ціна**, **Кількість** і **Термін гарантії** – *числовий*, тільки для поля **Ціна** – *грошовий*, а для полів **Кількість** і **Термін гарантії** – *ціле число*.

Для зберігання даних про об'єкти деякої предметної області може бути використано кілька таблиць, які можуть бути пов'язані між собою.

У нижньому рядку таблиці бази даних виводиться номер поточного запису й кількість записів у таблиці

Запис	1	з 4
-------	---	-----

Ключі. У базах даних передбачена обов'язкова вимога, яка передбачає ідентифікацію кожного з об'єктів. Тобто в таблиці бази даних не повинно бути двох різних записів з однаковим набором значень властивостей. Для таблиці бази даних це означає, що значення хоча б в одному з полів не повинні повторюватися. Таке поле називають **ключовим**, або **ключем таблиці**. Ключові поля ще називають **ідентифікаторами**.

Ідентифікація (фр. *identification* – впізнання) – процес установлення особи або об'єкта.

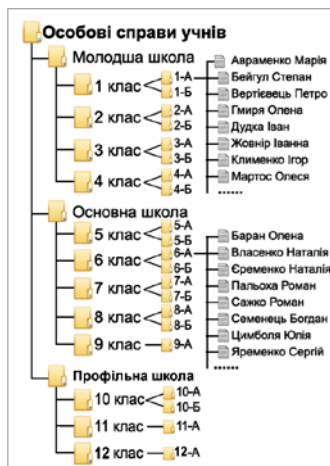
У таблиці бази даних, поданій на малюнку 6.13, значення властивостей об'єктів у кожному з наведених полів можуть повторюватися. У таких випадках потрібно додатково вводити ще одне поле, значення в якому не будуть повторюватись у жодному із записів (рядків) таблиці. Таким полем може бути порядковий номер запису.

Для тих, хто хоче знати більше

Для зберігання даних у комп'ютерах часто використовують ієрархічні структури. Вони передбачають організацію даних про об'єкти у вигляді дерева. У цій структурі бази даних у кожного об'єкта є лише один об'єкт вищого рівня, якому він підпорядкований (батьківський об'єкт), і може бути кілька підпорядкованих об'єктів (нащадків). Виняток становить лише найвищий за ієрархією об'єкт – у нього немає батьківського об'єкта.

Прикладом ієрархічної організації даних є файлова структура (мал. 6.14), що використовується під час розміщення даних на дисках.

Основними недоліками ієрархічної структури даних є складність розміщення великих обсягів даних і тривалий час на пошук потрібних даних у таких структурах.



Мал. 6.14. Приклад ієрархічного розміщення даних про учнів / учениць школи

Цікаві факти з історії

Розробка теорії реляційної бази даних мала величезне значення для розвитку всієї комп'ютерної науки і практики. Для опису табличної структури даних **Е. Ф. Кодд** використав термін **relation** (англ. *relation* – відношення), а цю модель стали називати **реляційною**.

За визначні здобутки в галузі **Е. Ф. Кодда** було визнано 1976 року «Людиною IBM», 1981 року – нагороджено однією із найпрестижніших в інформатиці відзнак – **премією Тюрінга**. 2002 року за версією часопису **Forbes** реляційну модель бази даних було визнано однією з найважливіших інновацій за попередні 85 років.

СОРТУВАННЯ ТА ФІЛЬТРУВАННЯ ДАНИХ

Пригадайте

- Які правила сортування даних в електронних таблицях ви знаєте?
- Як здійснити сортування за даними в кількох стовпцях електронних таблиць?
- Що таке фільтр в електронних таблицях? Як його використати?

Як і в таблицях текстового і табличного процесорів, дані в таблицях баз даних можна сортувати. За замовчуванням дані в таблиці **Base** сортуються в порядку зростання значень за ключовим полем.

Щоб відсортувати записи в таблиці за значеннями в одному з полів, слід виконати таку послідовність дій:

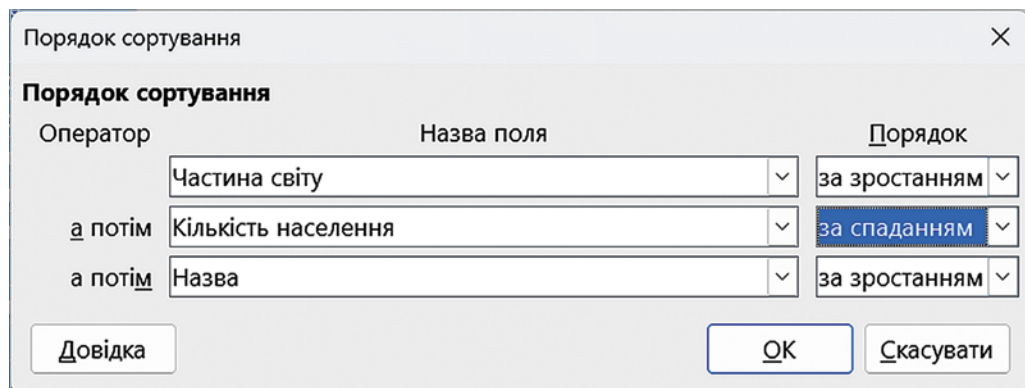
1. Відкрити таблицю бази даних, дані в якій потрібно відсортувати.
2. Установити курсор у межах поля, за даними якого буде виконано сортування записів.
3. Виконати **Дані** ⇒ **Сортування за зростанням** (або **Сортування за спаданням**). Можна також вибрати відповідні кнопки на панелі інструментів: **Сортування за зростанням**  або **Сортування за спаданням** .

Фрагмент таблиці **Країни** бази даних **Країни світу**, відсортований за спаданням за даними поля **Кількість населення**, наведено на малюнку 6.15. Кількість населення у відповідному полі зазначено в тисячах осіб. Зазвичай одиниці вимірювання величин зазначають у підписах полів у формах, запитах і звітах. Щоб відмінити сортування, слід виконати **Дані** ⇒ **Очистити фільтр / сортування** або вибрати кнопку **Очистити фільтр / сортування**  на панелі інструментів **Дані таблиці**. Повернеться сортування записів таблиці за зростанням значень за ключовим полем.

Для сортування за даними кількох полів використовується команда **Дані** ⇒ **Сортувати...** або кнопка **Сортувати**  на панелі інструментів **Дані таблиці**. У вікні **Порядок сортування** (мал. 6.16) у спис-

ID	Назва	Частина світу	Кількість населення	Площа	Дата утворення	Форма правління	Держав
48	Індія	Азія	1439323,78	3287,59	01.01.1947	Парламентська республіка	Індійська республіка
63	Китайська Народна Республіка	Азія	1380004,39	36197	01.10.1949	Однопартійні системи	Юань
126	Сполучені Штати Америки	Америка	331002,65	9833,52	04.07.1776	Президентська республіка	
49	Індонезія	Азія	273523,62	1904,57	17.08.1945	Президентська республіка	Індонезійська Республіка
20	Бразилія	Америка	212559	8511,97	01.01.1822	Президентська республіка	Крузадо
10	Бангладеш	Азія	164689	144	16.12.1972	Парламентська республіка	Така
112	Росія	Європа	145491	17075,4	25.12.1991	Парламентсько-президентська республіка	Рубль
103	Пакистан	Азія	141256	803,94	01.01.1947	Парламентсько-президентська республіка	Пакистанська Республіка
155	Японія	Азія	127096	377,84	22.12.1559	Конституційна монархія	Йена
42	Ефіопія	Африка	114963,59	1127,13	Х ст. до н.е.	Парламентська республіка	Ефіопський імператор
96	Нігерія	Африка	113862	923,77	01.10.1960	Президентська республіка	Найра
43	Єгипет	Африка	102334,4	1001,45	01.01.1922	Парламентсько-президентська республіка	Єгипетський президент
89	Мексика	Америка	98872	1972,55	01.01.1821	Президентська республіка	Мексиканський президент
23	В'єтнам	Азія	97338	331,34	01.09.1945	Однопартійні системи	Донг
38	Демократична Республіка Конго	Африка	89561,4	2345,41	30.06.1960	Президентська республіка	Конголезький президент
51	Іран	Азія	83783,94	1648	01.04.1979	Унітарна теократична президентська республіка	Іранський президент
99	Німеччина	Європа	82017	357,59	01.01.1871	Парламентська республіка	Євро
24	Великобританія	Європа	67886	244,8	24.03.1603	Конституційна монархія	Фунт стерлінгів
137	Туреччина	Азія	66668	780,58	01.01.1923	Парламентська республіка	Турецька ліра
133	Таїланд	Азія	62806	514	24.12.1349	Конституційна монархія	Бат

Мал. 6.15. Фрагмент таблиці **Країни** бази даних **Країни світу**, відсортованої від найбільшого до найменшого за даними поля **Кількість населення**

Мал. 6.16. Вікно **Порядок сортування**

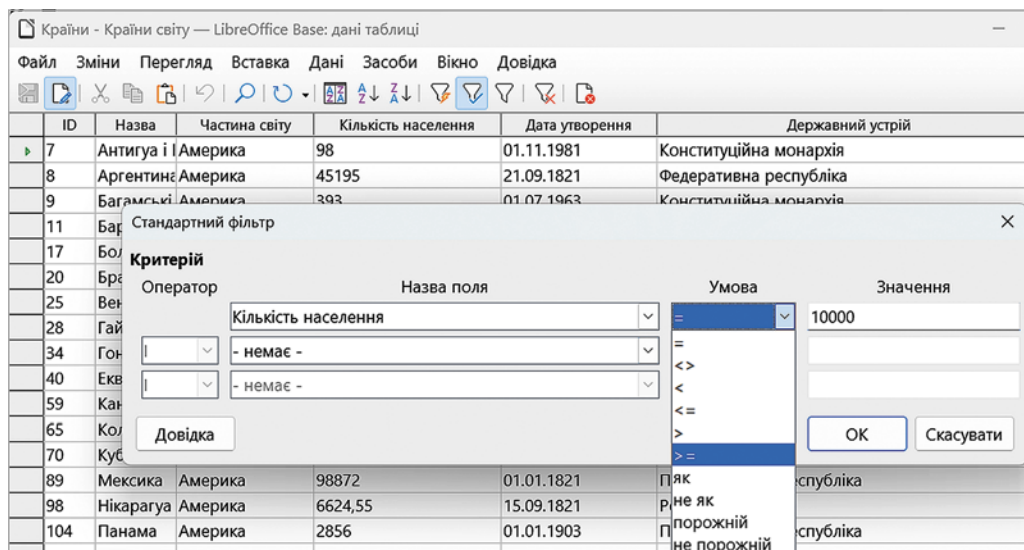
ках **Назва поля** потрібно вибрати назви полів, за якими буде здійснено сортування. Усього можна обрати до трьох полів. Порядок сортування відповідає порядку списку полів від верхнього до нижнього, тобто сортування в наведеному прикладі відбувається спочатку за назвами частин світу (від А до Я); якщо назви частин світу збігаються, то далі записи сортуються за кількістю населення (від найбільшого значення до найменшого). Якщо ж і ці значення збігаються, то відбувається сортування за назвами країн (від А до Я).

Для відбору записів, дані в деякому полі яких відповідають певним умовам, використовують **фільтри** подібно до використання фільтрів у табличному процесорі. Для фільтрування даних у вікні відкритої таблиці використовують елементи керування панелі інструментів **Дані таблиці**, а також команди меню **Дані**.

Щоб знайти записи, дані в одному з полів яких відповідають певному значенню, слід відкрити вікно таблиці, виділити це значення у відповідному полі таблиці й виконати **Дані** ⇒ **Швидкий фільтр...** або вибрати кнопку **Швидкий фільтр**  на панелі інструментів **Дані таблиці**.

Для повної відміни дії фільтра потрібно виконати **Дані** ⇒ **Очистити фільтр / сортування** або вибрати кнопку **Очистити фільтр / сортування**  на панелі інструментів **Дані таблиці**. Щоб можна було тимчасово відмовитися від фільтра, а потім його повторно застосувати, вибирають **Дані** ⇒ **Застосувати фільтр** або кнопку **Застосувати фільтр**  на панелі інструментів **Дані таблиці**. Для повторного застосування фільтра слід ще раз виконати зазначену команду або вибрати кнопку **Застосувати фільтр** .

Якщо потрібно застосувати фільтр із кількома умовами, слід виконати команду **Дані** ⇒ **Типовий фільтр** або вибрати кнопку **Типовий фільтр**  на панелі інструментів **Дані таблиці**. Далі у вікні **Стандартний фільтр** (мал. 6.17) потрібно встановити значення властивостей фільтрування:



Мал. 6.17. Вікно **Стандартний фільтр**

1. Вибрати у списку **Назва поля** назву поля, за значеннями якого буде здійснено фільтрування, наприклад *Кількість населення*.
2. Обрати у списку **Умова** одну з можливих умов фільтрування, наприклад **=**.
3. Увести у поле **Значення** дані, за значенням яких будуть перевірятися дані у відповідних полях таблиці на відповідність умові, наприклад *10000*.
4. Обрати за потреби наступну складову умови (*назва поля, умова, значення*) та вибрати у списку **Оператор** логічний оператор (**І** чи **АБО**), яким будуть з'єднані ці дві частини умови.
5. Повторити дії попереднього кроку, якщо є потреба у використанні ще однієї умови фільтрування.
6. Вибрати кнопку **ОК**.

Для тимчасової відмови від фільтра, а також повторного застосування фільтра слід використовувати команду **Дані** ⇒ **Застосувати фільтр** або кнопку **Застосувати фільтр**

В усіх випадках після застосування фільтрів у нижньому рядку таблиці буде вказано номер поточного запису і загальну кількість запитів, що відповідають умові

Запис	1	з 4
-------	---	-----

.

ЗАПИТИ НА ВИБІРКУ ДАНИХ



Пригадайте

- Як здійснюється пошук потрібних даних у пошукових системах, наприклад **Google**? Які інструменти використовуються для уточнення пошукових запитів?
- Які дії виконує користувач для ознайомлення з розкладом руху потягів в онлайн-системі «Укрзалізниці»?

Запити є одним з основних інструментів опрацювання даних у базах даних. Запити можуть використовувати для пошуку даних, які відповідають певним умовам, а також для виконання операцій над даними і збереження результатів пошуку й опрацювання даних у вигляді таблиці.

По суті, запит – це програма пошуку потрібних даних, створена з використанням мови **SQL** (англ. *Structured Query Language* – мова структурованих запитів).

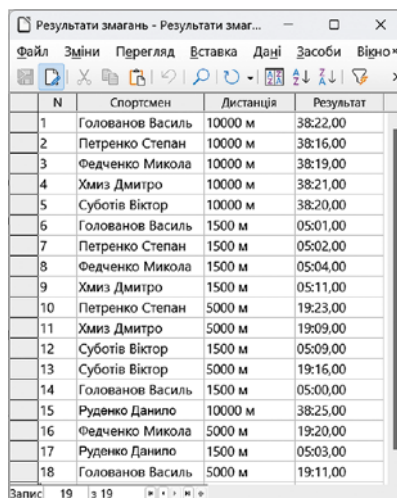
Основними операціями, які може здійснити користувач з використанням запитів, є:

- створення таблиць як результату виконання програми запиту на основі аналізу даних у вже існуючих таблицях бази даних, наприклад створення таблиці призерів з бігу на 10 000 м на основі таблиці, наведеної на малюнку 6.18. Зазвичай під час змінення даних в основній таблиці дані в таблиці, що є результатом виконання запиту, теж змінюються під час чергового виконання запиту;
- знаходження узагальнювальних даних (суми, максимального чи мінімального значення тощо) для заданих полів, наприклад найкращого (мінімального) результату з бігу на 5000 м;
- знаходження значень інших властивостей шляхом виконання операцій над даними з полів однієї або кількох таблиць, наприклад знаходження густоти населення певної країни на основі значень площі та кількості населення;
- оновлення даних тощо.

Серед різних видів запитів виокремлюють як одні з найпоширеніших **запити на вибірку даних** – запити, з використанням яких на основі наявних таблиць створюються таблиці з даними, які відповідають певним умовам (мал. 6.19).

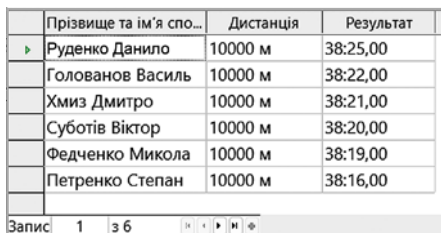
СКБД **Base** для створення запитів, крім режиму написання запитів мовою **SQL**, має засоби створення запитів у автоматизованому режимі з використанням спеціального інструмента – **Помічника для створення запиту**. Для створення нового запиту з використанням помічника слід:

1. Вибрати на панелі груп об'єктів бази даних кнопку **Запити**.
2. Вибрати на панелі **Завдання** посилення **Використати помічника для створення запиту**.



N	Спортсмен	Дистанція	Результат
1	Голованов Василь	10000 м	38:22,00
2	Петренко Степан	10000 м	38:16,00
3	Федченко Микола	10000 м	38:19,00
4	Хмиз Дмитро	10000 м	38:21,00
5	Суботів Віктор	10000 м	38:20,00
6	Голованов Василь	1500 м	05:01,00
7	Петренко Степан	1500 м	05:02,00
8	Федченко Микола	1500 м	05:04,00
9	Хмиз Дмитро	1500 м	05:11,00
10	Петренко Степан	5000 м	19:23,00
11	Хмиз Дмитро	5000 м	19:09,00
12	Суботів Віктор	1500 м	05:09,00
13	Суботів Віктор	5000 м	19:16,00
14	Голованов Василь	1500 м	05:00,00
15	Руденко Данило	10000 м	38:25,00
16	Федченко Микола	5000 м	19:20,00
17	Руденко Данило	1500 м	05:03,00
18	Голованов Василь	5000 м	19:11,00

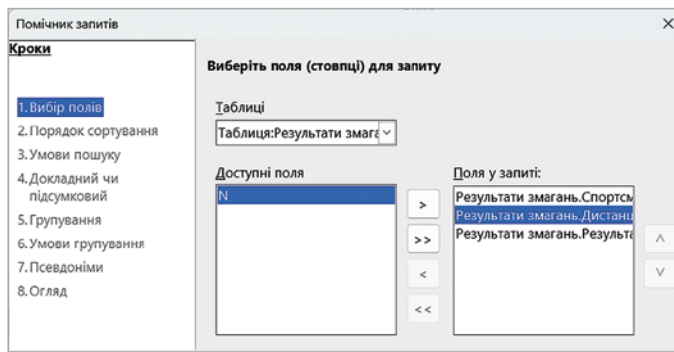
Мал. 6.18. Таблиця
Результати змагань



Прізвище та ім'я спо...	Дистанція	Результат
Руденко Данило	10000 м	38:25,00
Голованов Василь	10000 м	38:22,00
Хмиз Дмитро	10000 м	38:21,00
Суботів Віктор	10000 м	38:20,00
Федченко Микола	10000 м	38:19,00
Петренко Степан	10000 м	38:16,00

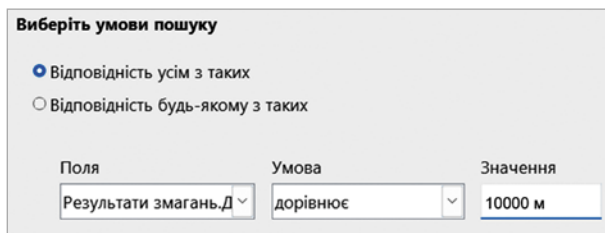
Мал. 6.19. Таблиця запиту Найкращі
результати з бігу на 10 000 м

- Вибрати у вікні **Помічник запитів** (мал. 6.20) у списку **Таблиці** таблицю, яка буде джерелом даних, наприклад **Результати змагань**.



Мал. 6.20. Вікно Помічник запитів

- Перенести з поля **Доступні поля** в поле **Поля у запиті** потрібні для запиту імена полів, використовуючи кнопки **>** та **>>**, наприклад поля **Спортсмен**, **Дистанція**, **Результати**.
- Вибрати кнопку **Далі**.
- Вибрати у списку **Сортувати за** поля, за даними в яких буде здійснено сортування, наприклад поле **Результати**, і встановити з використанням перемикача вид сортування, наприклад **За спаданням**.
- Вибрати кнопку **Далі**.
- Установити умови пошуку, наприклад так, як показано на малюнку 6.21.



Мал. 6.21. Вікно вибору умов запиту

- Вибрати кнопку **Далі**.
- Вибрати тип запиту, наприклад **Докладний запит**.
- Вибрати кнопку **Далі**.
- Увести, за бажанням, інші імена (псевдоніми) для підпису даних у результатах виконання запиту, наприклад ім'я поля **Спортсмен** замінити на **Прізвище та ім'я спортсмена**.
- Вибрати кнопку **Далі**.
- Увести у поле **Назва запиту** назву цього запиту, під яким він буде збережений у базі даних, наприклад **Результати змагань з бігу на 10 000 м**.
- Обрати дію, яку варто виконати після завершення роботи з помічником створення запитів, наприклад **Показати запит**.
- Вибрати кнопку **Завершити**.

У результаті цих дій буде створено запит **Результати змагань з бігу на 10 000 м**, результат виконання якого – таблицю – буде виведено на екран (мал. 6.19).

Якщо потрібно внести зміни в наявний запит, слід його відкрити в поданні конструктора запиту. Для цього потрібно обрати ім'я запиту на панелі **Запити** та виконати **Зміни** ⇒ **Редагування** або обрати цю саму команду в контекстному меню запиту.

У цьому поданні (мал. 6.22) у верхній частині вікна розміщено список імен полів таблиці, на основі якої було створено запит.

Поле	Спортсмен	Дистанція	Результат	
Псевдонім	Прізвище та ім'я	Дистанція	Результат	
Таблиця	Результати змаг	Результати змаг	Результати змаг	
Сортувати			за спаданням	
Видиме	☒	☒	☒	☐
Функція				
Критерій		10000 м		
Або				
Або				
Або				
Або				

Мал. 6.22. Фрагмент вікна запиту **Результати змагань з бігу на 10 000 м** у поданні конструктора

У нижній частині вікна розміщено таблицю конструктора. Вона містить такі рядки:

- **Поле** – відображаються імена обраних полів;
- **Псевдонім** – відображаються підписи полів у результатах виконання запиту;
- **Таблиця** – відображається ім'я таблиці, до складу якої входить відповідне поле;
- **Сортувати** – може встановлюватися порядок сортування, що буде застосований до записів даного поля в таблиці – результаті запиту;
- **Видиме** – містить прапорець, установлена позначка якого свідчить про те, що дані цього поля будуть відображатися в результаті запиту;
- **Функція** – може містити функції для проведення обчислень над даними поля;
- **Критерій** – може містити умову для вибірки даних з відповідного поля;
- **Або** – для запису інших умов, які є частиною складеної умови та які поєднуються з умовами в рядку **Критерій** логічною операцією **АБО** (диз'юнкція).

Якщо в одному і тому самому рядку **Критерій** та **Або** введено умови в кількох стовпцях, то вони об'єднуються в одну умову логічною операцією **I** (кон'юнкція). Наприклад, щоб у запиті відображалися лише записи про результати забігів на дистанцію 10 000 м Голованова Василя, слід у рядку **Критерій** до наявного запиту про дистанцію у стовпці **Спортсмен** записати **'Голованов Василь'** (мал. 6.23).

Поле	Спортсмен	Дистанція	Результат	
Псевдонім	Прізвище та ім'я сп	Дистанція	Результат	
Таблиця	Результати змагань	Результати змаг	Результати змаг	
Сортувати			за спаданням	
Видиме	☒	☒	☒	☐
Функція				
Критерій	'Голованов Василь'	'10000 м'		
Або				

Мал. 6.23. Фрагмент вікна запиту з двома умовами, об'єднаними логічною операцією **I** (кон'юнкція)



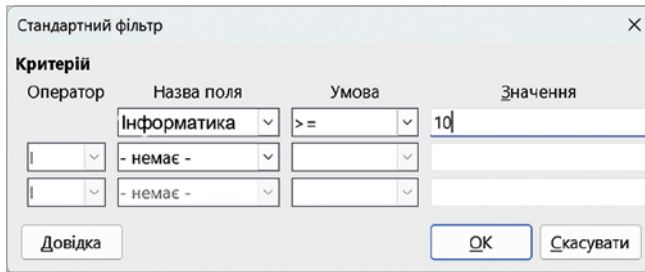
Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання 1. Виконайте сортування та фільтрування даних у таблицях. Для цього:

1. Запустіть програму **Base**.
2. Відкрийте файл бази даних, наприклад **Розділ 6\Пункт 6.2\вправа 6.2.1.odt**.
3. Збережіть файл бази даних у вашій папці під тим самим ім'ям.
4. Визначте, які три учні / учениці мають найвищий середній бал. Для цього:
 1. Відобразіть уміст таблиці **Підсумкові оцінки**. Для цього двічі клацніть на імені таблиці на панелі **Таблиці**.
 2. Здійсніть сортування даних за спаданням по полю **Середній бал**. Для цього:
 1. Зробіть поточним поле **Середній бал**.
 2. Виконайте **Дані** ⇒ **Сортування за спаданням**.
5. Запишіть у зошит прізвища перших трьох учнів / учениць у списку.
6. Відмініть сортування, виконавши **Дані** ⇒ **Очистити фільтр / сортування**.
7. Використовуючи фільтр, визначте, скільки дівчат з іменем **Юлія** є у класі. Для цього:
 1. Знайдіть у полі **Ім'я** значення **Юлія** та встановіть курсор у цю клітинку.
 2. Виконайте **Дані** ⇒ **Швидкий фільтр**.
 3. Визначте кількість записів, що відповідають умовам фільтра.
 4. Відмініть фільтр, вибравши кнопку **Очистити фільтр / сортування**  на панелі інструментів.

8. Знайдіть прізвища учнів / учениць, що мають високий рівень навчальних досягнень з інформатики. Для цього:
 1. Виконайте команду **Дані** ⇒ **Типовий фільтр**.
 2. Установіть значення властивостей фільтрування у вікні **Стандартний фільтр** (мал. 1).
 3. Виберіть кнопку **ОК**.
 4. Визначте кількість записів, що відповідають умовам фільтра.
 5. Відмініть фільтр, вибравши кнопку **Очистити фільтр** / **сортування**  на панелі інструментів.

Мал. 1. Вікно **Стандартний фільтр**

9. Закрийте вікно таблиці. За потреби підтвердьте збереження таблиці.
10. Закрийте вікно СКБД.

Завдання 2. База даних магазину комп'ютерної техніки містить дані про комп'ютери (стаціонарні та ноутбуки), їхні комплектуючі (монітори) та додаткові пристрої (сканери, принтери, SSD) (мал. 2). Потрібно створити й виконати запит на вибірку даних про всі ноутбуки, що надійшли до магазину після 01.11.2025 року. Для цього:

Отримання товару - Отримання товару — LibreOffice Base: дані таблиці						
Файл Зміни Перегляд Вставка Дані Засоби Вікно Довідка						
	N	Номер накладної	Дата отримання	Тип товару	Назва товару	Кількість
1	123	11.10.2025	Комп'ютер	Настільний	4	15 660,00 грн
2	123	11.10.2025	Монітор	LCD 19"	6	3 103,00 грн.
3	123	11.10.2025	Сканер	Планшетний	2	4 641,00 грн.
4	124	21.10.2025	Комп'ютер	Ноутбук	10	22 307,00 грн
5	124	21.10.2025	SSD	Зовнішній USB	3	1 649,00 грн.
6	125	23.11.2025	Комп'ютер	Настільний	15	29 285,00 грн
7	125	23.11.2025	Принтер	Лазерний	12	5 861,00 грн.
8	125	23.11.2025	Принтер	Струменевий	13	3 221,00 грн.
9	126	27.11.2025	Комп'ютер	Ноутбук	5	36 245,00 грн
10	126	27.11.2025	Сканер	Планшетний	14	3 956,00 грн.
11	123	11.10.2024	Монітор	LCD 22"	7	4 421,00 грн.
12	124	21.10.2024	Монітор	LCD 23"	6	6 233,00 грн.
13	124	21.10.2024	Монітор	LCD 27"	4	7 308,00 грн.
14	125	23.11.2025	Комп'ютер	Ноутбук	12	33 456,00 грн
15	126	27.11.2025	Комп'ютер	Настільний	3	28 560,00 грн
+						
Запис: 6 з 15						

Мал. 2. Вікно таблиці **Отримання товару**

1. Відкрийте базу даних **Отримання товару** у СКБД LibreOffice Base із папки, яку вкаже вчитель / учителька, наприклад **Розділ 6\Пункт 6.2.**
2. Збережіть файл бази даних під іменем **Вправа 6.2.2.odt** у вашій папці.
3. Виберіть на панелі груп об'єктів бази даних кнопку **Запити** .
4. Виберіть на панелі **Завдання** посилання **Використати помічника для створення запиту.**
5. Виберіть у вікні **Помічник запитів** у списку **Таблиці** таблицю **Отримання товару**, яка буде джерелом даних.
6. Перемістіть поля **Номер накладної**, **Дата отримання**, **Тип товару**, **Назва товару**, **Кількість**, **Ціна** з поля **Доступні** до поля **Поля у запиті**, використовуючи кнопки  та , і виберіть кнопку **Далі**.
7. Установіть сортування даних за зростанням значень у полі **Дата отримання** і виберіть кнопку **Далі**.
8. Установіть умови вибірки даних (мал. 3).

Поля	Умова	Значення
Отримання товару.Дата	більше, ніж	01.11.2025
Отримання товару.Назва	дорівнює	Ноутбук

Мал. 3. Панель для встановлення умови запиту

9. Виберіть кнопку **Далі**.
10. Установіть перемикач **Докладний запит (показувати всі записи за-питу)** і виберіть кнопку **Далі**.
11. Уведіть за потреби псевдоніми до полів запиту і виберіть кнопку **Далі**.
12. Уведіть у поле **Назва запиту** назву запиту – **Ноутбуки**, виберіть перемикач **Показати запит** і кнопку **Завершити**.
13. Перегляньте і проаналізуйте результати виконання запиту. Визначте кількість знайдених записів.
14. Закрийте вікно створеного запиту. За потреби підтвердьте збереження запиту.
15. Закрийте вікно СКБД.



Найважливіше в цьому пункті

Сучасні бази даних за структурою впорядкування даних належать до реляційних або до об'єктно-реляційних. Основним об'єктом реляційної бази даних є **таблиця**. У таблицях кожен рядок містить набір значень властивостей одного з об'єктів певної множини. Кожен стовець таблиці містить набір значень певної властивості об'єктів цієї множини.

Рядок таблиці реляційної бази даних називають **записом**. Запис містить значення властивостей одного об'єкта із множини об'єктів одного виду.

Стовпець таблиці реляційної бази даних називають **полем**. Поле має ім'я, яке зазвичай пов'язане з назвою властивості. Поле містить множину значень однієї властивості всіх об'єктів множини. Дані, що містяться в кожному полі таблиці, є однотипними.

Поле, у якому значення не повторюються, називається **ключовим**, або **ключем** таблиці.

Дані в таблицях баз даних можна сортувати. За замовчуванням дані в таблиці сортуються в порядку зростання значень за ключовим полем. Для зміни значень властивостей сортування слід установити курсор у будь-яку клітинку потрібного поля й виконати **Дані** ⇒ **Сортування за зростанням** (або **Сортування за спаданням**).

Щоб відмінити сортування, слід виконати **Дані** ⇒ **Очистити фільтр / сортування**.

Для відбору записів, дані в яких відповідають певним умовам, використовують фільтри. Для знаходження записів, які відповідають певному значенню, слід виділити це значення у відповідному полі таблиці й виконати **Дані** ⇒ **Швидкий фільтр**.

Якщо потрібно застосувати фільтр з різними умовами, слід виконати команду **Дані** ⇒ **Типовий фільтр**.

Запити є одним з основних інструментів опрацювання даних у базах даних. Запити можуть використовувати для пошуку даних, які відповідають певним умовам, а також для виконання операцій над даними і збереження результатів пошуку та опрацювання даних у вигляді таблиці. Запит – це програма пошуку потрібних даних, створена з використанням мови **SQL**.

Запити на вибірку даних – це запити, з використанням яких на основі наявних таблиць створюються таблиці з даними, які відповідають певним умовам. Для створення запитів використовується спеціальна мова, у **Base** такою мовою є мова **SQL**.

СКБД **Base** для створення запитів, крім режиму написання запитів мовою **SQL**, має засоби створення запитів в автоматизованому режимі з використанням спеціального інструмента – **Помічника для створення запиту**.

Якщо потрібно внести зміни в наявний запит, слід його відкрити в поданні конструктора запиту. Для цього треба обрати ім'я запиту на панелі **Запити** і виконати **Зміни** ⇒ **Редагування** або обрати цю саму команду в контекстному меню запиту.



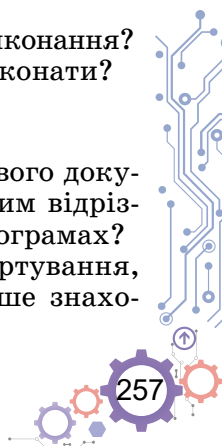
Дайте відповіді на запитання

1. Що є основним об'єктом реляційної бази даних?
2. Яка структура таблиці реляційної бази даних? Які об'єкти вона має?
3. Що таке *поле* в таблиці бази даних? Що таке *запис* у таблиці бази даних?
4. Що таке *ключове поле*? Які вимоги до ключового поля?
5. Які операції можна виконувати з таблицями?
6. Як виконати сортування за даними одного з полів? Як відмінити сортування?
7. Яка послідовність операцій під час здійснення сортування за даними кількох полів?
8. Як визначити кількість записів, що задовольняють умови фільтрування?
9. Для чого призначений запит? Що є результатом його виконання?
10. Як відредагувати запит? Які дії при цьому можна виконати?



Обговоріть і зробіть висновки

1. Порівняйте особливості сортування в таблицях текстового документа, електронної таблиці та бази даних. Що спільного і чим відрізняється виконання цих операцій у вказаних прикладних програмах?
2. Поясніть, для яких випадків ми використовуємо сортування, а для яких – фільтрування даних. У яких випадках швидше знаходяться дані в той чи інший спосіб?



3. У яких випадках використовують запити на вибірку даних? Наведіть приклади їх використання.

4. Що спільного і в чому відмінності використання запитів та фільтрів у базі даних?



Виконайте завдання

1. Відкрийте файл бази даних (наприклад, Розділ 6\Пункт 6.2\завдання 6.2.1). Відкрийте таблицю **Країни** та виконайте сортування даних:

- за даними поля **Кількість населення** за спаданням. Визначте п'ять країн з найбільшою кількістю населення. Визначте, яке місце посідає Україна у світі за кількістю населення;
- за даними полів **Кількість населення** та **Частина світу** для визначення країн, що мають найбільшу кількість населення в кожній частині світу. Визначте, у якій послідовності слід виконувати сортування для швидкого знаходження вказаних значень. Для якого поля (полів) слід виконати сортування за зростанням, а для якого (яких) – за спаданням? Збережіть базу даних у вашій папці у файлі з іменем **завдання 6.2.1.б**;
- за даними полів **Форма правління** та **Частина світу** для визначення кількості країн, що мають державний устрій *Президентська республіка* в кожній частині світу. Визначте, у якій послідовності слід виконувати сортування для швидкого знаходження вказаних значень. Для якого поля (полів) слід виконати сортування за зростанням, а для якого (яких) – за спаданням? Збережіть базу даних у вашій папці у файлі з іменем **завдання 6.2.1.в**.

2. Відкрийте файл бази даних (наприклад, Розділ 6\Пункт 6.2\зразок 6.2.2). Відкрийте таблицю **Географічні та фізичні карти** та виконайте фільтрування даних:

- Відмініть за потреби всі фільтри. Виберіть карти України з масштабом перегляду *M1:2 100 000* та ціною понад *20 грн*. Скільки таких карт знайдено? Збережіть базу даних у вашій папці у файлі з іменем **завдання 6.2.2.а**.
- Відмініть усі фільтри. Виберіть усі карти, крім карт з даними про Україну і ціною від *5 до 40 грн*. Скільки таких карт знайдено? Збережіть базу даних у вашій папці у файлі з іменем **завдання 6.2.2.б**.

3. Відкрийте файл бази даних (наприклад, Розділ 6\Пункт 6.2\завдання 6.2.3.odt). Створіть з використанням **Помічника** для створення запиту запит на вибірку даних **Ціна карт** із використанням полів таблиці **Географічні та історичні карти**. Запит має містити дані з полів **Країна**, **Регіон**, **Вид карти**, **Розмір**, **Ціна**. Установіть такі умови відбору даних: за полем **Вид карти** – *Політична*; за полем **Ціна** – *менше ніж 45 грн*. Запустіть запит на виконання, проаналізуйте результати. Переконайтеся у правильності виконання запиту. Збережіть базу даних у вашій папці у файлі з іменем **завдання 6.2.3**.

4. Відкрийте файл бази даних (наприклад, Розділ 6\Пункт 6.2\завдання 6.2.4.odt). Створіть з використанням **Помічника** для створення запиту запит на вибірку даних **Площа країн** із таблиці **Країни світу**. Запит має містити дані з полів **Назва**, **Частина світу**, **Площа**. Установіть такі умови відбору даних та сортування: за значеннями

в полі **Площа країн** – сортування за спаданням; за значеннями в полі **Частина світу** – Африка. За результатами запиту визначте, які п'ять країн мають серед країн Африки найбільшу площу. Збережіть базу даних у вашій папці у файлі з іменем **завдання 6.2.4**.



Практична робота № 8.

«Створення запитів на вибірку даних»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте файл бази даних, наприклад **Розділ 6\Практична 8\практична 8**.
2. Збережіть базу даних у вашій папці у файлі з іменем **практична робота 8**.
3. Відкрийте таблицю **Країни** та відсортуйте дані за зростанням значень у полі **Форма правління**.
4. Установіть фільтрування даних: для поля **Площа** – менше ніж 1 тис. кв. км, а для поля **Кількість населення** – понад 1 млн осіб.
5. На основі даних таблиці **Країни** створіть запит **Країни Азії**, уключивши в нього поля: **Назва, Частина світу, Площа, Кількість населення**.
6. Установіть для запиту умову відбору за полем **Частина світу** – уключити до запиту тільки країни Азії.
7. Установіть сортування даних у запиті **Країни Азії** по полю **Кількість населення за спаданням**.
8. Збережіть усі зміни у вашій папці у файлі з іменем **практична робота 8**.



Готуємось до вивчення нового матеріалу

1. Що таке *навчальний проєкт*? На яких уроках ви виконували навчальні проєкти?
2. Які цифрові пристрої та програмні засоби ви застосовували для виконання завдань проєкту?
3. Яка послідовність виконання навчального проєкту?



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу **«Бази даних»** (початковий, середній, достатній, високий рівень).

- Я можу пояснити призначення баз даних.
- Я знаю основні об'єкти реляційної бази даних (таблиця, поле, запис, ключ таблиці) та можу пояснити їхнє призначення.
- Я розумію, як використовуються бази даних у різних галузях життя людини.
- Я вмію виконувати операції додавання, видалення, редагування даних у базах даних з використанням СКБД.
- Я можу аргументовано обирати способи впорядкування та відбору потрібних даних, використовуючи операції сортування та фільтрування даних.
- Я можу навести приклади використання операцій сортування та фільтрування даних у навчальній діяльності та у побуті.
- Я вмію створювати запити на вибірку даних з використанням помічника для створення запитів.
- Я вмію застосовувати різні стратегії пошуку та відбору потрібних відомостей у базі даних.

ПРАКТИКУМ З ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

У цьому розділі ви здобудете нові, а також поглибите і розширите свої знання й удосконалисте навички з таких тем:

- ▶ навчальний проєкт та етапи його виконання;
- ▶ імпорт та експорт даних із різних програм;
- ▶ створення інтегрованих інформаційних продуктів;
- ▶ оцінювання результатів виконання проєкту.

НАВЧАЛЬНИЙ ПРОЄКТ ТА ЕТАПИ ЙОГО ВИКОНАННЯ

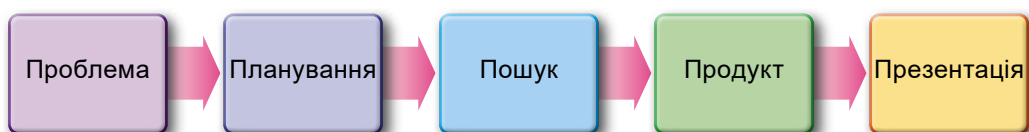


Пригадайте

- У яких навчальних проєктах ви брали участь? Які етапи виконання проєкту були найскладнішими?
- Як цифрові інструменти можуть покращити якість проєктної роботи?

У попередніх класах ви вже ознайомилися з основами проєктної діяльності, зокрема з поняттям навчального проєкту, його структурою та етапами реалізації. Ви брали участь у групових дослідженнях, створювали інформаційні продукти, презентували результати своєї роботи перед аудиторією. Цей досвід сформував у вас навички командної взаємодії, критичного мислення і творчого самовираження.

Нагадаємо, що **навчальний проєкт** – це форма організації навчальної діяльності, яка передбачає самостійне або групове дослідження певної проблеми з метою створення завершеного продукту. Основою проєктної роботи є п'ять ключових кроків – так звані «**п'ять П**» (мал. 7.1):



Мал. 7.1. Ключові компоненти навчального проєкту

Пригадаємо ці етапи:

1. **Визначення проблеми.** На початку слід чітко окреслити, яку життєву / навчальну проблему ми намагаємося розв'язати в цьому проєкті, яка мета діяльності та які для цього ставимо перед собою завдання.
2. **Розробка плану дій.** Наступним кроком є створення докладного плану реалізації проєкту з визначенням термінів виконання. Завдання розподіляються між учасниками, кожному призначаються конкретні ролі та обов'язки.
3. **Пошук і збирання інформації.** Проводимо пошук потрібних даних у різноманітних джерелах: книжках, періодичних виданнях, мережі «Інтернет», а також шляхом опитування та анкетування.

4. **Аналіз та опрацювання матеріалів.** Зібрану інформацію аналізуємо, упорядковуємо, створюємо продукт і формулюємо висновки.
5. **Підготовка та представлення проєкту.** Презентуємо виконану роботу перед цільовою аудиторією (класом, учителем / учительською або батьками тощо).

Упродовж цього навчального року під час виконання навчальних проєктів ви створювали інтегровані цифрові продукти, які поєднують текстові, графічні, числові, мультимедійні дані. Використання кількох програмних засобів під час створення інформаційних продуктів робить їх не тільки функціонально насиченими, а й естетично привабливими.

Для створення таких інтегрованих продуктів, наприклад, ви можете:

- **імпортувати дані** з різних джерел (**Google Форми**, Excel-таблиці, 3D-моделі, бази даних);
- **експортувати результати** у зручні формати (PDF, презентації, інтерактивні звіти, відео чи аудіо);
- **використовувати ШІ-інструменти** для генерації текстів, аналізу даних, створення візуалізацій;
- **спільно редагувати документи** у хмарних середовищах (**Google Workspace, Canva, Microsoft 365**);
- **використовувати цифрові шаблони** для ефективного оформлення результатів тощо.

Наведемо приклад навчального проєкту зі створення інтегрованого інформаційного продукту.

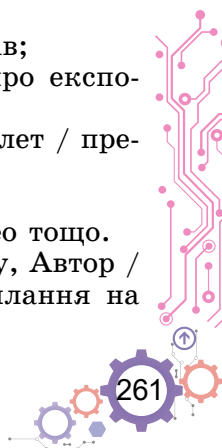
Назва проєкту: «Віртуальний музей винаходів людства».

Мета: Створити інтерактивну цифрову експозицію, яка представляє найвизначніші винаходи людства.

Завдання: Учні / учениці збирають інформацію про різні винаходи, аналізують і структурують її, створюють базу даних про ці винаходи, створюють прості 3D-моделі експонатів, а потім оформлюють інтерактивну презентацію або вебсторінку тощо, інтегруючи туди створені об'єкти.

Етапи реалізації навчального проєкту:

1. Визначення проблеми:
 - Які винаходи людства змінили хід історії?
 - Як подати їх у форматі віртуального музею?
2. Розробка плану дій, об'єднання у групи та розподіл ролей:
 - **дослідники** – шукають інформацію про видатні винаходи людства, які змінили хід історії;
 - **модельовальники** – створюють 3D-моделі цих винаходів;
 - **архіваріуси** – створюють і заповнюють базу даних про експонати музею;
 - **дизайнери** – оформлюють інтерактивний плакат / буклет / презентацію / вебсторінку / електронну книгу.
3. Збирання інформації відповідно до мети проєкту:
 - **Джерела:** енциклопедії, наукові сайти, Вікіпедія, відео тощо.
 - **Відомості для наповнення бази даних:** Назва винаходу, Автор / країна, Рік створення, Призначення винаходу, Посилання на джерело.



4. Створення 3D-моделей:
 - У **Blender** або **Tinkercad** – прості моделі винаходів (наприклад, телескоп, парова машина, мікроскоп).
 - Уставлення створених моделей в інформаційний продукт.
5. Створення бази даних:
 - У **Base** або **Google Таблиці** – таблиця з усіма експонатами.
 - Додати фільтри, форми, посилання на 3D-моделі.
6. Вибір інтерактивного продукту для представлення результатів роботи з використанням інструментів ІІІ для отримання рекомендацій щодо кращого оформлення інтерактивного продукту.
7. Оформлення інтерактивного продукту (плаката / буклета, презентації тощо)
 - Уставити зображення моделей.
 - Додати текстові описи, звукозаписи або відео анімованих моделей.
 - Уставити QR-коди з посиланнями на джерела.
 - Додати інтерактивні елементи для навігації (переходи, посилання, зміст, кнопки) по експозиції.
8. Презентація результатів проєкту:
 - Демонстрація інформаційних продуктів на шкільному заході.
 - Розміщення повідомлення на шкільному сайті або в соцмережах.

ІНТЕГРОВАНЕ ВИКОРИСТАННЯ ЗАСОБІВ ОПРАЦЮВАННЯ ЕЛЕКТРОННИХ ДОКУМЕНТІВ



Пригадайте

- Об'єкти яких типів можна вставити в текстовий документ? Як це можна зробити?

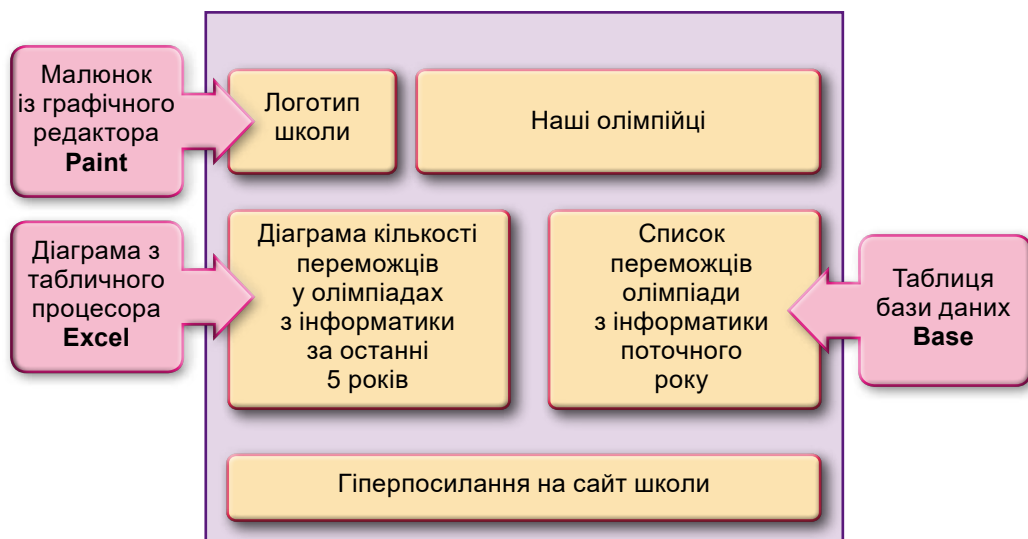
Протягом вивчення курсу інформатики ви ознайомилися з різними прикладними програмами – графічними редакторами **Paint** та **Krita**, текстовим процесором **Word**, табличним процесором **Excel**, програмою підготовки мультимедійних презентацій **PowerPoint**, системою управління базами даних **Base** і навчилися створювати й опрацьовувати з їх використанням різноманітні *електронні документи*: текстові документи, електронні таблиці, комп'ютерні презентації, бази даних, графічні зображення, вебсторінки та інше.

Але досить часто під час створення електронного документа в певній програмі виникає потреба вставити в нього фрагменти з документів, які створено в інших програмах. Наприклад, на слайд презентації (мал. 7.2), підготовлений у **PowerPoint**, потрібно вставити растровий малюнок, створений у графічному редакторі **Paint** (логотип школи), та діаграму, яку розроблено в табличному процесорі **Excel** (кількість переможців олімпіад). А також на слайді потрібно розмістити таблицю бази даних зі списком переможців.

У результаті таких дій користувач отримує документ, що включає фрагменти, які створено з використанням різних програмних засобів. Подібний документ отримав назву **інтегрований**.

Інтегрований (лат. *integrum* – ціле) **документ** – це документ, у який вставлено фрагменти з даними різних типів, що створювалися в інших прикладних програмах. Програма (документ, файл), із якої виконується вставлення фрагмента у складений документ, називається

Слайд Microsoft PowerPoint



Мал. 7.2. Приклад фрагмента інтегрованого документа

джерело, а програма (документ, файл), куди вставляються об'єкти з інших джерел, – **приймач**.

ТЕХНОЛОГІЯ OLE



Пригадайте

- Що таке **Буфер обміну**? Які операції можна виконувати з його використанням?
- Якими способами можна виконати операції **Копіювати**, **Вирізати**, **Вставити**?

Інтегрований документ можливо створити завдяки технології OLE. Технологія OLE (англ. *Object Linking and Embedding* – зв'язування та вбудовування об'єктів) дає змогу вставляти об'єкти в документ, забезпечуючи різні способи зв'язку цього об'єкта з документом-джерелом та програмою-джерелом.

Технологія OLE передбачає два способи вставлення об'єктів:

- **вбудовування об'єкта** – забезпечує вставлення в документ-приймач копії об'єкта з автоматичним встановленням зв'язку з програмою-джерелом, але зв'язок з документом-джерелом втрачається. У такий спосіб жодні зміни оригінального об'єкта не впливають на вигляд його копії у складеному документі, але для опрацювання вставленого об'єкта можна користуватися не тільки засобами програми-приймача, а й засобами програми-джерела.
- **зв'язування об'єкта** – у документ-приймач вставляється не сам об'єкт, а лише посилання на нього у вигляді образу цього об'єкта з автоматичним встановленням зв'язку і з документом-джерелом, і з програмою-джерелом. За такого способу будь-які зміни об'єкта в документі-джерелі приводять до автоматичної зміни образу об'єкта в документі-приймачі. Для опрацювання об'єкта можна використовувати засоби програми-джерела.

Технологію **OLE** в повному обсязі підтримують не всі прикладні програми. Деякі з них можна використовувати тільки як приймач, інші – тільки як джерело, але є й ті, що можуть виступати в обох ролях. Наприклад, прикладні програми комплексу **Microsoft Office** повністю забезпечують роботу за технологією **OLE**, а графічний редактор **Paint** можна використовувати за цією технологією тільки як джерело об'єктів.

Операційна система **Windows** надає користувачу кілька способів вставляння даних з різних програм до інтегрованого документа. З деякими з них ви вже ознайомилися раніше: метод **drag-and-drop** (англ. *перетягнути та кинути*) і використання **Буфера обміну**.

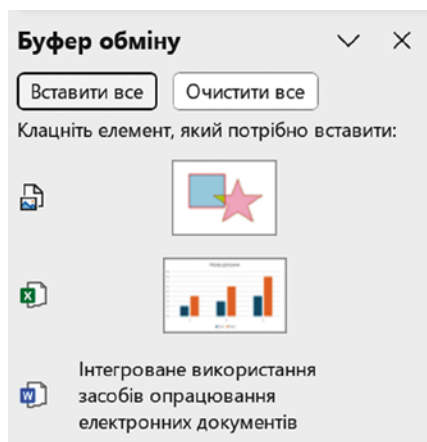
Так, у багатьох програмах, які працюють під керівництвом ОС **Windows**, для вставляння об'єктів передбачена можливість перетягування потрібного фрагмента чи об'єкта з використанням миші з вікна однієї програми до вікна іншої (метод **drag-and-drop**) або з використанням «швидких» клавіш **Ctrl + C** (копіювати) та **Ctrl + V** (вставити). Під час виконання цієї операції об'єкт переміщується з документа-джерела у документ-приймач.

Такий спосіб рекомендується використовувати за невеликої відстані між вікнами програми-джерела і програми-приймача. Для цього вікна цих програм доцільно розташувати поруч у віконному режимі.

Також уставити потрібний об'єкт у документ-приймач можна, використавши **Буфер обміну** операційної системи **Windows**. Для цього слід об'єкт із документа-джерела скопіювати в **Буфер обміну** (команда **Скопіювати**), а потім уставити його з **Буфера обміну** в документ-приймач (команда **Вставити**). Під час роботи з програмами **Microsoft Office** використовується **Буфер обміну Microsoft Office**, який може одночасно зберігати 24 об'єкти (мал. 7.3). Це можна використати для швидкого збирання фрагментів по документах-джерелах та їх уставлення в документ-приймач (команда **Вставити все**). Під час виконання цієї операції вікно **Буфера обміну** в усіх програмах **Microsoft Office**, які задіяні в цій роботі, має бути відкритим (для цього слід натиснути кнопку відкриття діалогового вікна відповідної групи **Буфер обміну**  на вкладці **Основне**).

Використовуючи ці способи, у документ-приймач вставляється копія об'єкта з документа-джерела, водночас значення деяких властивостей уставлених об'єктів можуть змінитися.

За таких способів уставлення жодного зв'язку між програмами, документами та об'єктами не встановлюється. Це призводить до того, що жодні зміни об'єкта-оригіналу не відображаються в документі-приймачі. І в подальшому опрацювання вставленого об'єкта забезпечується засобами програми, у якій здійснюється робота з інтегрованим



Мал. 7.3. Відкритий **Буфер обміну Microsoft Office**

документом. Наприклад, уставивши до текстового документа малюнок із програми **Paint**, застосувавши метод **drag-and-drop**, користувач уже не зможе використовувати для змінення малюнка інструменти графічного редактора, такі як **Пензель**, **Ластик**, **Розпилювач**; не зможе здійснити інвертування кольорів, виділення довільної області тощо. Він може скористатися тільки засобами, які надає текстовий процесор для роботи з графічними зображеннями.

Однак якщо вставлення об'єктів у інтегрований документ у такий спосіб відбувається в межах програм пакета **Microsoft Office**, то вставлені об'єкти в подальшому можна редагувати вбудованими редакторами цих програм.



Для тих, хто хоче знати більше

Інший спосіб уставлення об'єктів у інтегрований документ у програмах **Microsoft Office** – **вбудовування або зв'язування** – здійснюється через **Буфер обміну** командою **Спеціальне вставлення** за таким алгоритмом:

1. Виділити в документі-джерелі об'єкт (фрагмент документа), який уставляється, і виконати **Оснoвне** ⇒ **Буфер обміну** ⇒ **Копіювати**.
2. Розмістити курсор у потрібному місці документа-приймача і виконати **Оснoвне** ⇒ **Буфер обміну** ⇒ **Вставити** ⇒ **Спеціальне вставлення**.
3. Указати в діалоговому вікні **Спеціальне вставлення** спосіб уставлення об'єкта, вибравши потрібний перемикач:
 - для вбудовування – перемикач **Вставити**;
 - для зв'язування – перемикач **Зв'язати**.
4. Вибрати у списку **Як тип об'єкта**, у форматі якого об'єкт-копію буде вставлено в документ-приймач.

Залежно від обраного типу в подальшому користувачу будуть надаватися відповідні засоби опрацювання об'єкта-копії. Набір можливих варіантів для вибору залежить від типу об'єкта-оригіналу. Для вставлення об'єкта з тими самими властивостями, які він має в документі-джерелі, рекомендується вибирати тип **Об'єкт**.

5. Вибрати спосіб відображення об'єкта-копії в документі-приймачі, установивши за потреби позначку прапорця **У вигляді значка**:
 - позначку прапорця не встановлено – об'єкт вставляється в такому самому вигляді, як і в документі-джерелі;
 - позначку прапорця встановлено – у документ замість об'єкта вставляється значок даного типу об'єкта (його можна змінити, вибравши кнопку **Змінити піктограму**), який є посиланням на оригінальний об'єкт. Для відкриття об'єкта-оригіналу потрібно буде двічі клацнути на вставленому значку.
 6. Вибрати кнопку **ОК**.
- Використовуючи зв'язування об'єктів, **слід пам'ятати**:
- програма, яка була використана для створення об'єкта-оригіналу, має бути обов'язково встановлена на тому комп'ютері, де відбувається перегляд та опрацювання інтегрованого документа;
 - файли документа-джерела і документа-приймача потрібно заздалегідь зберегти;
 - під час змінення місця збереження інтегрованого документа або документів-джерел може відбутися **розрив (втрата) зв'язків**, і тоді в інтегрованому документі вставлені об'єкти не відображатимуться.

Таким чином, використання команд **Спеціального вставлення** дає змогу більш гнучко керувати інтеграцією об'єктів у документ. Це відкриває додаткові можливості для створення комплексних матеріалів, де поєднуються тексти, таблиці, графіка та інші елементи.



ІМПОРТ ТА ЕКСПОРТ ДАНИХ ІЗ ФАЙЛІВ



Пригадайте

• Що таке *конвертація файлів*? • У яких форматах можна зберегти текстовий документ; електронну таблицю; мультимедійну презентацію; базу даних; вебсторінку?

Часто обмін даними між різними програмами можна організувати з використанням операцій *імпорту* та *експорту*, якщо такі операції передбачені засобами цих програм.

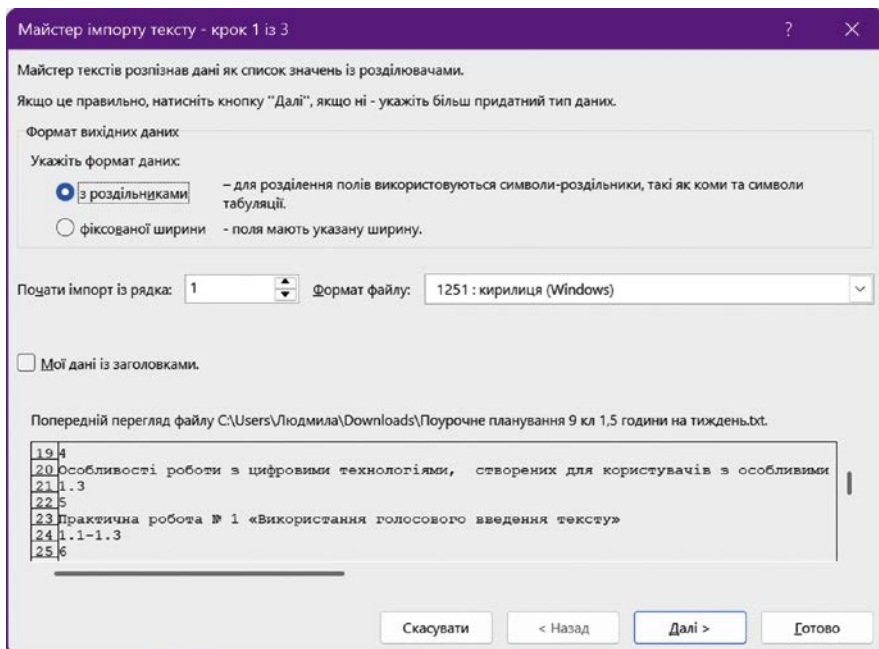
Операція **імпорту** передбачає відкриття програмою даних із файлів, які підготовлені в інших програмах і формат яких не є основним для цієї програми. Після відкриття таких файлів (імпортування даних) програма опрацьовує отримані дані як стандартні для своєї роботи. **Експорт** полягає у збереженні даних у файлі, формат якого не є основним для цієї програми. Насправді під час здійснення операцій імпорту й експорту відбувається *конвертація (перекодування) даних*, тому перетворення даних може відбуватися із частковою втратою даних, елементів форматування тощо.

Розглянемо на прикладах, як здійснюються деякі операції імпорту та експорту в прикладних програмах **Microsoft Office**.

Приклад 1. Імпорт текстових файлів у програми **Microsoft Office**.

Для імпорту текстових файлів у програму **Microsoft Excel** потрібно виконати операцію відкриття файлу, вибравши у вікні **Відкрити** у списку **Тип файлу** формат *Текстові файли*. Імпортуванню в **Excel** підлягають тільки текстові файли формату **TXT**.

Після цього запускається **Майстер імпорту текстів** (мал. 7.4). Виконуючи кроки інструкції, потрібно зазначити, чи використовуються



Мал. 7.4. Майстер імпорту тексту в **Excel**

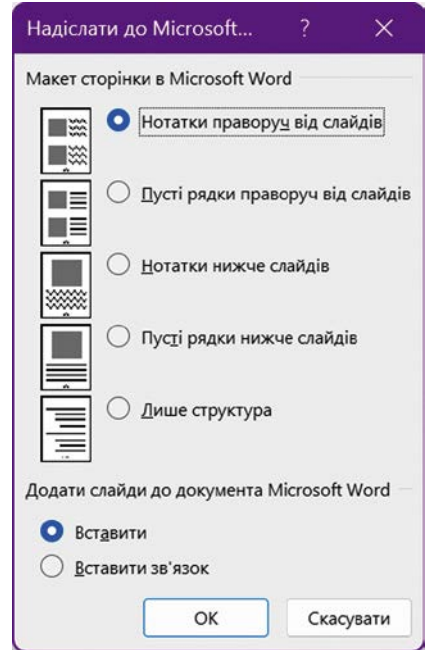
розділювачі в початковому документі, який саме символ-розділювач використано, яке кодування символів використано, який тип і формат окремих стовпців даних потрібно встановити тощо. У результаті текст буде конвертовано і вставлено у вигляді таблиці Excel.

Приклад 2. Експорт даних у програмі PowerPoint.

Створену презентацію можна експортувати (мал. 7.5.) в текстовий документ двома способами (**Файл** ⇒ **Експорт** ⇒ **Створити супровідні матеріали в Microsoft Office Word**):

- у вигляді *нотаток*, коли кожен слайд презентації вставляється в текстовий документ у вигляді графічного ескізу слайда, поруч із яким вставляється додаткове місце для нотаток. Кількість слайдів на аркуші, місце розташування нотаток обираються додатково;
- у вигляді *структури* презентації, коли текстові дані з кожного слайда розміщуються в документі у вигляді послідовного тексту.

Також зручним видом експортування презентацій є збереження презентації у вигляді набору графічних зображень, де кожен слайд представлено окремим файлом заданого формату (**Файл** ⇒ **Зберегти як** і вибрати тип файлу *Малюнок*). Усі зображення слайдів будуть зібрані в одну папку, ім'я якої буде збігатися з іменем файлу презентації.



Мал. 7.5. Експорт презентації в текстовий документ

ОЦІНЮВАННЯ ЯКОСТІ РОЗРОБЛЕНИХ ІНФОРМАЦІЙНИХ ПРОДУКТІВ



Пригадайте

- Які комп'ютерні програми можуть бути використані для оформлення результатів проєкту?
- Які можливості використання систем штучного інтелекту в навчанні?

Коли ви завершуєте роботу над навчальним проєктом, важливо визначити, наскільки він **якісний**. Для цього потрібно оцінити свою роботу і створений продукт за певними критеріями:

Критерій	Що оцінюється
Змістовність	Чи повністю розкрито тему? Чи є інформація достовірною?
Структура	Чи логічно побудовано продукт? Чи легко ним користуватися?
Оформлення та дизайн	Чи привабливо виглядає продукт? Чи є єдиний стиль?

Критерій	Що оцінюється
Інтеграція даних	Чи вдало поєднано різні типи даних (текст, графіка, таблиці, відео, аудіо)?
Використання цифрових інструментів	Чи доцільно обрано відповідні програми? Чи використано їх ефективно?
Командна робота	Чи всі учасники брали участь? Чи була співпраця?
Застосування ШІ	Чи використано штучний інтелект доречно та етично?

Кожен критерій може оцінюватися за шкалою, наприклад: 3 бали – відмінно, 2 – добре, 1 – наступного разу буде краще.

Критерії оцінювання можна використати як для самооцінювання, так і для оцінювання проєктів інших. Це допомагає оцінити роботу справедливо і зрозуміло для всіх.

Оцінювання – це не просто оцінка в журналі. Це можливість навчитися аналізувати свою роботу, бачити прогрес і ставати кращим. Коли ви знаєте, за що і за якими критеріями оцінюється ваш проєкт, ви можете планувати роботу більш свідомо та впевнено.



Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Завдання.

- Відкрийте у графічному редакторі **Paint** малюнок із файлу **Розділ 7\Пункт 7.1\Вправа\Логотип.bmp**.
- Відкрийте в табличному процесорі **Excel** діаграму з файлу **Розділ 7\Пункт 7.1\Вправа\Кількість.xls**.
- Відкрийте в СКБД **Base** таблицю бази даних із файлу **Розділ 7\Пункт 7.1\Вправа\Олімпійці.odt**.
- Відкрийте програму **PowerPoint** і створіть слайд, на який уставте через **Буфер обміну** об'єкти з відкритих документів з розміщенням відповідно до малюнка 7.2: графічне зображення з файлу **Логотип**, діаграму з файлу **Кількість**, список учнів та учениць із файлу **Олімпійці**.
- Збережіть презентацію у вашій папці у файлах з іменем **Вправа 7.1** у форматах **PPTX, PDF, JPG**.
- Експортуйте базу даних **Олімпійці** в електронну таблицю і текстовий документ.
- Збережіть створену електронну таблицю і текстовий документ у вашій папці у файлах з іменами **Олімпійці.xlsx** та **Олімпійці.txt**.
- Закрийте вікна всіх програм.



Найважливіше в цьому пункті

Навчальний проєкт – це форма навчальної діяльності, яка передбачає самостійне або групове дослідження певної проблеми з метою створення завершеного інформаційного продукту. Основними етапами реалізації проєкту є: визначення проблеми, розробка плану дій, збір інформації, аналіз і обробка матеріалів, підготовка результатів і презентація.

Під час роботи над проектом створюють інтегровані цифрові продукти, які поєднують текст, графіку, таблиці, відео, 3D-моделі та інші типи даних. Для цього використовують різні програмні засоби, зокрема **Word, Excel, PowerPoint, Canva, Tinkercad, Google Форми**, а також системи штучного інтелекту, які допомагають генерувати тексти, аналізувати дані й створювати візуалізації.

Інтеграція даних із різних програм здійснюється за допомогою технології **OLE** (метод **drag-and-drop** або **Буфер обміну**). **Інтегрований** документ – це документ, що містить фрагменти, створені в різних програмах. Технологія **OLE** дозволяє вставляти об'єкти з можливістю збереження зв'язку з програмою-джерелом, що дає змогу редагувати їх у середовищі, де вони були створені.

Імпорт та експорт даних – це процеси перенесення інформації між програмами. Імпорт дає змогу відкривати файли з іншого програмного середовища, а експорт – зберігати дані у форматі, придатному для використання в інших програмах або для публікації в інтернеті.

Завершальним етапом роботи над проектом є оцінювання якості створеного інформаційного продукту. Це включає самооцінювання власної роботи й оцінювання за критеріями, які враховують змістовність, логічність структури, оформлення, інтеграцію даних, ефективність використання цифрових інструментів, командну взаємодію та етичне застосування штучного інтелекту. Такий підхід допомагає усвідомити власний прогрес, розвивати критичне мислення і вдосконалювати навички цифрової грамотності.



Дайте відповіді на запитання

1. Що таке *навчальний проєкт*? Назвіть основні етапи реалізації навчального проєкту.
2. Що таке *інтегрований документ*? Наведіть приклади.
3. Які способи вставлення об'єктів у документ-приймач підтримує операційна система **Windows**?
4. У чому полягає суть технології **OLE**? Яке її призначення?
5. Що таке *імпорт* та *експорт* даних у прикладних програмах?



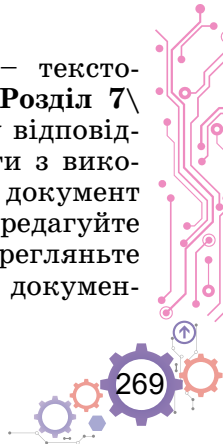
Обговоріть і зробіть висновки

1. Чому навчальний проєкт краще виконувати в команді? Які переваги це дає? Які виникають труднощі?
2. У чому особливості опрацювання інтегрованого документа порівняно зі звичайним?
3. Поясніть, чим вбудовування об'єкта відрізняється від зв'язування об'єкта.
4. Які труднощі можуть виникнути під час імпорту й експорту даних між різними програмами?
5. Які критерії, на вашу думку, можуть бути використані для оцінювання якості інформаційного продукту?



Виконайте завдання

1. Відкрийте вказані вчителем / учителькою файли – текстовий документ та електронну книгу (наприклад, із папки **Розділ 7/ Пункт 7.1** файли **зразок 7.1.docx**, **зразок 7.1.xlsx**). Уставте у відповідні місця текстового документа діаграми з електронної книги з використанням **Буфера обміну MS Office**. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 7.1.1.docx**. Відредагуйте довільним чином діаграму в файлі електронної книги і перегляньте текстовий документ. Чи відбулися якісь зміни в текстовому документі? Поясніть результат.



2. Відкрийте мультимедійну презентацію, запропоновану вчителем / учителькою (наприклад, **Розділ 7\Пункт 7.1\зразок 7.2.pptx**). Збережіть файл у форматах **HTML, JPG, PDF** у вашій папці. Експортуйте презентацію в текстовий документ у вигляді файлу з нотатками **завдання 7.1.2**. Перегляньте створені файли.

3. Відкрийте запропоновані вчителем / учителькою файли – мультимедійну презентацію і малюнок (наприклад, із папки **Розділ 7\Пункт 7.1** файли **зразок 7.3.bmp, зразок 7.3.pptx**). На слайд № 1 уставте малюнок через **Буфер обміну**, на слайд № 2 – убудуйте малюнок, на слайді № 3 – зв'яжіть малюнок. Збережіть презентацію у вашій папці у файлі з іменем **завдання 7.1.3.pptx**. Відредагуйте малюнок у документі-джерелі. Перегляньте презентацію та поясніть отриманий результат.

4. Відкрийте запропонований учителем / учителькою файл бази даних (наприклад, **Розділ 7\Пункт 7.1\зразок 7.4.adb**). Експортуйте таблицю бази даних в електронну таблицю, а звіт – у текстовий файл. Збережіть створені файли у вашій папці з іменами **завдання 7.1.4**.

5. Імпортуйте вказаний текстовий документ (наприклад, **Розділ 7\Пункт 7.1\зразок 7.5.rtf**) у базу даних та електронну таблицю. Збережіть створені файли у вашій папці з іменами **завдання 7.1.5**.

ПІДСУМКОВИЙ ПРОЄКТ

Реалізуйте навчальний проєкт у групі однокласників / однокласниць за однією з наведених тем. Оберіть об'єкти дослідження відповідно до мети, використайте 2–3 різних цифрових засоби для опрацювання даних (наприклад: електронні таблиці, текстові процесори та графічні редактори, 3D-моделювання, презентації, відео, аудіо, сайти, бази даних тощо). Інтегруйте результати у завершений інформаційний продукт – інтерактивний плакат, буклет, презентацію, сайт, відео або електронну книгу. Застосуйте систему штучного інтелекту у зручний для вас спосіб – для генерації текстів, аналізу даних, створення візуалізацій або формулювання висновків тощо. Збережіть результат роботи в папці вашої групи у файлі з іменем **завдання 7.2**.

Орієнтовна тематика навчальних проєктів:

1. Ми в мережі: цифрова репутація під мікроскопом.

Мета: дослідити цифровий слід вашої групи – пости, лайки, коментарі, фото – та оцінити його вплив на майбутнє (навчання, кар'єру, репутацію).

2. Наша маленька батьківщина крізь час і простір.

Мета: створити віртуальну експозицію про рідне місто / село – його минуле, сучасність та уявне майбутнє.

3. Онлайн 24/7: як ми живемо в екрані.

Мета: дослідити, скільки часу учні / учениці проводять онлайн, для чого саме і як це впливає на здоров'я, навчання та спілкування.

4. Сімейний архів: цифрова пам'ять поколінь.

Мета: зібрати фото, документи, спогади родини і створити цифровий альбом, що зберігатиме історію для майбутніх поколінь.

5. Фізика в цифрах: моделюємо реальність.

Мета: змодельовати фізичний процес (наприклад, рух тіла, теплопередачу, електричне коло) за допомогою цифрових засобів.

6. Школа майбутнього: навчання без меж.

Мета: дослідити, як учні / учениці уявляють школу майбутнього – з учителями / учительками чи без, із гаджетами чи на природі, самонавчання чи наставництво.

7. Цифрове мистецтво: створено учнями / ученицями.

Мета: представити власні графічні роботи – растрові, векторні, 3D – у форматі цифрової галереї.

8. Від лічби на пальцях до штучного інтелекту.

Мета: дослідити, як змінювалися засоби обчислення – від давніх рахівниць до сучасних суперкомп'ютерів та ШІ.

9. Професії майбутнього: хто ми через 10 років?

Мета: дослідити нові професії, що з'являються завдяки ІТ, і яким чином це впливає на вибір професії.

10. Технології, що змінюють світ: від генетики до штучного інтелекту.

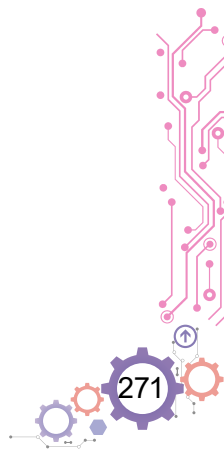
Мета: дослідити проривні технології, які впливають на життя людини та суспільства.



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу «Практикум з використання інформаційних технологій» (початковий, середній, достатній, високий рівень).

- Я розумію, що таке навчальний проєкт, етапи його виконання і можу визначити проблему дослідження, цілі та завдання.
- Я вмію планувати проєктну діяльність, розподіляти ролі в команді та ефективно взаємодіяти з учасниками / учасницями, ураховуючи їхні думки та емоції.
- Я знаю, як збирати, аналізувати і структурувати інформацію з різних джерел для виконання проєкту.
- Я усвідомлюю роль штучного інтелекту в навчальному проєкті й можу визначити напрямки його використання у проєкті.
- Я вмію створювати інформаційні продукти в різних форматах та інтегрувати в них текстові, графічні, числові, мультимедійні дані.
- Я розумію, що таке інтегрований документ, знаю способи вставлення об'єктів з одних програм в інші та можу імпортувати й експортувати дані між ними.
- Я можу якісно презентувати результати проєкту, враховуючи потреби аудиторії та використовуючи сучасні цифрові інструменти.



А

Адитивні технології – технології виготовлення матеріальних тривимірних об'єктів шляхом накладання послідовних шарів спеціального матеріалу за готовою цифровою 3D-моделлю, с. 190.

Активний цифровий слід – дані, які користувач свідомо та добровільно вносить до сервісів у мережі, с. 205.

Антивірусні програми – вид спеціального програмного забезпечення, призначеного для захисту даних і пристроїв комп'ютера від шкідливих програм, с. 227.

Б

Багаторівневий список – ієрархічний перелік елементів, де кожен пункт може мати підписки; використовується для планів, тестів, структурованих описів, с. 50.

База даних – упорядкований за певними правилами набір взаємопов'язаних даних, с. 235.

Бекдори (англ. *back door* – чорний вхід) – шкідливі комп'ютерні програми, що проникають на комп'ютер користувача разом з файлами або використовують службові функції окремих ОС для організації дистанційного доступу до комп'ютерів, с. 224.

Брайлівський дисплей – пристрій, що використовують для перетворення тексту з екрана комп'ютера на тактильний шрифт Брайля, с. 32.

Брайлівська клавіатура – пристрій, що використовують для введення даних із клавіатури з тактильним шрифтом Брайля, с. 32.

Брандмауер (мережевий екран) – комп'ютерна програма, що забезпечує дотримання правил доступу до мережі, блокує за потреби підозрілі з'єднання, відстежує надсилання програмами даних у мережі, с. 227.

Буфер обміну – область пам'яті операційної системи, у якій тимчасово зберігаються дані (текст, зображення, об'єкти) для їх копіювання і вставлення, с. 254.

В

Вбудовування об'єкта – спосіб вставлення копії об'єкта в документ-приймач без зв'язку з документом-джерелом, але з можливістю редагування засобами програми-джерела, с. 265.

Викрадачі даних, або Infostealers (англ. *Info* – від *Information* – інформація, *stealers* – злодії), – шкідливі програми для крадіжки даних облікових записів, банківських карток, різноманітних паролів тощо, с. 224.

Вільного використання, або freeware (англ. *freeware* – вільний товар), **ліцензія** – ліцензія на ПЗ, що передбачає вільне використання програм без виплат винагороди автору, але не передбачає можливості внесення змін у програму, с. 12.

Г

Гізмo – інструмент, призначений для управління переглядом сцени в редакторі 3D-графіки **Blender**, с. 154.

Гіперпосилання – об'єкт у тексті (слово, малюнок), що забезпечує перехід до іншої частини документа або зовнішнього ресурсу, с. 85.

Грані – невеликі плоскі ділянки поверхні 3D-моделі будь-якого об'єкта, що мають форму багатокутника (переважно це трикутники та чотирикутники). Грань обмежена відрізками – **ребрами**. Кожне ребро має дві вершини, с. 148.

Д

Деінсталяція програми – процес видалення компонентів програмного забезпечення з комп'ютера, с. 20.

Деструктивні дії – поведінкові акти, що завдають шкоди самій людині, іншим або суспільству загалом, с. 203.

Динамічні субтитри – програма, яка автоматично перетворює будь-який звук, що відтворюється на комп'ютері, у текст у реальному часі, с. 35.

Дистрибутив (англ. *distribute* – розподіляти) – інсталяційний пакет програми, с. 19.

Драйвери пристроїв (англ. *driver* – водій, керуючий пристроєм) – програми, які забезпечують обмін даними між операційною системою і конкретною моделлю пристрою, с. 18.

Е

Експорт даних – операція збереження даних у форматі, що не є основним для програми, з метою використання в інших середовищах, с. 266.

Екструдуння – процес змінення форми тривимірного об'єкта шляхом видавлювання окремих елементів його поверхні з утворенням нових вершин, ребер і граней, с. 171.

Екранний диктор – програма для читання з екрана, озвучування тексту, елементів інтерфейсу, дій користувача та інших подій на екрані, с. 36.

Елемент одновимірного масиву – кожна змінна, що входить до одновимірного масиву, с. 96.

З

Завантажувачі, або Droppers (англ. *dropper* – підвіска, штучна приманка), – програми, що самі не є шкідливими, але використовуються для завантаження і встановлення на комп'ютер іншого ПЗ – шкідливого, с. 224.

Закладка – позначене місце в документі з унікальним ім'ям, яке використовується для швидкого переходу чи створення гіперпосилань, с. 85.

Запис – рядок таблиці реляційної бази даних. Запис містить значення властивостей одного об'єкта із множини однорідних об'єктів, с. 245.

Запит – інструмент опрацювання даних у базах даних, що використовується для пошуку, узагальнення та оновлення даних, с. 251.

Запити на вибірку даних – запит, з використанням якого на основі наявних таблиць створюється таблиця з даними, які відповідають певним умовам, с. 251.

Зв'язування об'єкта – спосіб усталення посилання на об'єкт у документ-приймач зі збереженням зв'язку з документом-джерелом; зміни в

оригіналі автоматично відображаються у приймачі, с. 263.

Здоров'я людини – стан повного фізичного, психічного здоров'я та соціального благополуччя, с. 215.

Зменшення цифрового сліду – процес, спрямований на мінімізацію кількості даних, які користувач залишає в комп'ютерних мережах, шляхом контролю над активною і частково пасивною складовою цифрового сліду, с. 206.

Зміст документа – автоматично сформований перелік заголовків структурних частин із номерами сторінок; кожен елемент є гіперпосиланням, с. 83.

Зростаючий список – список, у якому значення кожного наступного елемента більше за значення попереднього, с. 117.

І

Ідентифікатори – у базах даних імена полів таблиць та елементів керування, с. 247.

Ідентифікація – ототожнення, порівнювання, уподібнення, розпізнавання об'єктів, с. 247.

Ім'я елемента одновимірного масиву складається з імені цього масиву і порядкового номера цього елемента в одновимірному масиві, який узятो у квадратні дужки, с. 96.

Імпорт даних – відкриття файлів, створених іншими програмами, у поточному середовищі для подальшого опрацювання, с. 266.

Інсталяція програмного забезпечення (англ. *installation* – установлення, розміщення, монтаж) – спеціальна операція встановлення програмного забезпечення, у тому числі операційної системи, на комп'ютер, с. 19.

Інклюзія – залучення осіб (дітей і дорослих) з особливими освітніми потребами до навчання в загальноосвітніх закладах разом з іншими учнями / ученицями із забезпеченням необхідної підтримки, с. 29.

Інтегрований документ – документ, що містить об'єкти різних типів (текст, таблиці, діаграми, малюнки), усталені з інших програм, с. 262.

Інформаційна безпека – розділ інформатики, що вивчає закономірності забезпечення захисту інформаційних ресурсів фізичних осіб, підприємств, організацій, державних установ тощо від втрати, порушення функціонування, пошкодження, спотворення, несанкціонованого копіювання та використання, с. 202.

Інформаційна система (ІС) – це організована сукупність технічних, програмних, інформаційних, організаційних і людських ресурсів, призначена для здійснення інформаційних процесів, с. 5.

Інформаційне середовище – це простір, де ми спілкуємося, навчаємося, працюємо та отримуємо знання з використанням цифрових технологій, с. 212.

Інформаційне «сміття» – будь-які інформаційні об'єкти, які не мають цінності, заважають ефективній роботі, створюють зайві ризики. Інформаційне «сміття» накопичується на пристроях зберігання даних, у електронних поштових скриньках, базах даних і робить пошук даних, їх аналіз та прийняття рішень менш ефективними, с. 207.

Інформаційний вибух – процес лавиноподібного, із середини 60-х років ХХ ст., збільшення обсягу даних у світі, с. 235.

К

Ключове поле – у базах даних це поле, у якому значення не повторюються і яке використовується для ідентифікації записів у таблиці, с. 247.

Ключовий кадр – кадр комп'ютерної анімації, у якому значення властивостей об'єктів 3D-моделі (розмір, положення, кут обертання, колір поверхні тощо) мають бути точно такими, які встановлює розробник анімації, с. 181.

Колонтитул – службова частина сторінки (верхня, нижня, бічна), де розміщують номери сторінок, назву документа, автора, дату тощо, с. 64.

Комерційна (пропрієтарна) ліцензія – ліцензія на ПЗ із оплатою ко-

ристувачем вартості використання даної програми на одному чи кількох зазначених у ліцензії комп'ютерах, с. 11.

Конфіденційний – той, який не підлягає розголошенню, с. 203.

«Коробкові» версії ліцензії – ліцензії на ПЗ, що продаються зазвичай для одного екземпляра програми; можуть містити носій, на якому записана програма, та інструкцію щодо її використання разом із ключем для встановлення, с. 11.

Корпоративна ліцензія – ліцензія на кілька копій програми для використання, наприклад, на комп'ютерах закладів освіти, с. 12.

Криптомайнери, або **CoinMiners** (англ. *Coin* – монета, *Miners* – шахтарі, рудокопи), – програми, що використовують ресурси комп'ютера без згоди користувача для «добування» криптовалюти, с. 224.

Л

Ліцензія – дозвіл власника авторських прав (компанії-виробника або розробника програми) на використання програмного забезпечення. Ліцензія фіксує взаємовідносини між користувачем і правовласником програми, с. 11.

М

Меню останньої операції – спосіб змінення форми тривимірного примітива; з'являється у **Вікні 3D-огляду** одразу після вставлення об'єктів або виконання операцій перетворення, с. 169.

Меш – узагальнена назва графічних тривимірних примітивів. У 3D-редакторах використовуються такі **елементи мешу**: куб, сфера, циліндр, конус, тор та інші, с. 147.

Модулі – частини, з яких складається комп'ютерний проект, с. 135.

Н

Незростаючий список – список, у якому значення кожного наступного елемента не більше (менше або дорівнює) за значення попереднього, с. 117.

Неспадний список – список, у якому значення кожного наступного еле-

мента не менше (більше або дорівнює), ніж значення попереднього, с. 117.

О

Об'єкти текстового документа – елементи, що складають документ: символи, абзаци, сторінки, малюнки, таблиці, діаграми; кожен має певні властивості, с. 43.

Одновимірний масив – упорядкований за номерами набір змінних одного типу, с. 96.

Операції перетворення 3D-моделі – **переміщення** всього об'єкта або окремих вершин, ребер, граней, **поворот**, **масштабування** (змінення розмірів) та інші, с. 148.

Ортогональна проєкція – спосіб перегляду 3D-моделі у вікні редактора тривимірної графіки, за якого розміри об'єктів не залежать від їх віддаленості від точки перегляду сцени, с. 160.

П

Пасивний цифровий слід – дані, які збираються без прямої згоди користувача, зазвичай у фоновому режимі, під час обміну даними з мережевими сервісами, відвідування сайтів тощо, с. 206.

Персональні дані – відомості чи сукупність відомостей про фізичну особу, яка ідентифікована або може бути конкретно ідентифікована, с. 213–214.

Перспективна проєкція – спосіб перегляду 3D-моделі у вікні редактора тривимірної графіки, за якого об'єкти, які розміщені ближче до точки перегляду сцени, здаються більшими, а ті, що далі, – меншими, с. 160.

Поле – стовпець таблиці реляційної бази даних, що має ім'я, яке пов'язане з назвою властивості. Поле містить множину значень однієї властивості всіх об'єктів множини, с. 246.

Прикладне програмне забезпечення – програмне забезпечення, призначене для реалізації конкретних завдань опрацювання даних, які користувач здійснює в ході своєї професійної (навчальної) та побутової діяльності, с. 9.

Принцип доступності – один із принципів інформаційної безпеки, який полягає в забезпеченні доступу до загальнодоступних даних усім користувачам, а до інших даних – лише за умови авторизації, с. 203.

Принцип конфіденційності – один із принципів інформаційної безпеки, який полягає в забезпеченні доступу до даних лише авторизованим суб'єктам – тобто тим особам, що мають підтверджене право здійснювати операції з цими даними, с. 203.

Принцип цілісності – один із принципів інформаційної безпеки, який передбачає захист даних від зловмисного або випадкового видалення чи пошкодження, с. 203.

Проблема (у проєкті) – навчальне чи життєве питання, яке потребує вирішення й визначає мету і завдання проєкту, с. 260.

Пробна, або **trial** (англ. *trial* – випробування), ліцензія – ліцензія на комерційне ПЗ, яка надається для пробного використання програми протягом певного часу (зазвичай приблизно на місяць) або на певну кількість запусків програми, с. 12.

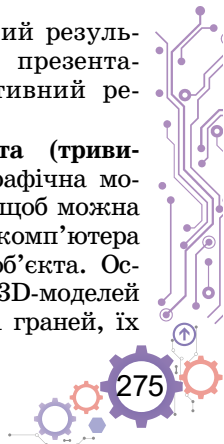
Програма розпізнавання тексту – OCR (англ. *Optical Character Recognition* – оптичне розпізнавання символів) – здійснює перетворення тексту, поданого графічним способом, на текст, який можна редагувати, с. 12.

Програми для читання з екрана (англ. *Screen Readers*) – програми, які перетворюють текст на екрані на звукові повідомлення або шрифт Брайля, с. 12.

Програмне забезпечення (ПЗ) – сукупність програм для комп'ютерів, с. 9.

Продукт проєкту – кінцевий результат діяльності (доповідь, презентація, база даних, інтерактивний ресурс), с. 260.

Просторова модель об'єкта (тривимірна, або 3D-модель) – графічна модель, призначена для того, щоб можна було розглядати на екрані комп'ютера в різних ракурсах деталі об'єкта. Основними властивостями 3D-моделей є кількість вершин, ребер і граней, їх



розташування на поверхні моделі та види матеріалів, які покривають поверхню, с. 141.

Р

Режими перегляду документа – способи відображення тексту на екрані: макет друку, режим читання, вебмакет, структура, чернетка, с. 44.

Рекламні модулі, або Adware (англ. *Ad* скорочення від *advertisement* – оголошення, реклама, *ware* – товар), – шкідливі програми, що вбудовуються у браузер користувача для показу реклами під час перегляду вебсторінок, с. 224.

Рендеринг – процес вимальовування вигляду моделі при погляді на неї крізь об'єктив камери з урахуванням положення та потужності джерел світла, с. 142.

Рецензування (відстеження змін) – режим роботи у **Word**, що дає змогу бачити внесені правки різними користувачами, приймати чи відхилити їх, с. 88.

Розділ документа – частина тексту з власними параметрами сторінки (орієнтація, поля, колонтитули), відокремлюється розривом розділу, с. 62.

Руткіти (англ. *root* – кореневий каталог у **Linux**, *kit* – набір інструментів) – шкідливі комп'ютерні програми, що використовуються для приховування наявності шкідливого ПЗ на комп'ютері, с. 224.

Рядок – це послідовність символів (літер, цифр, розділових знаків та іншого), узятя в одинарні чи подвійні лапки, с. 126.

С

Система опрацювання текстів – прикладне програмне забезпечення для створення, редагування, форматування та збереження текстових документів, с. 41.

Системи керування базами даних (СКБД) – прикладні комп'ютерні програми, призначені для створення, збереження та використання баз даних, с. 236.

Системи програмування – програми, призначені для створення нових про-

грам із використанням мов програмування, с. 10.

Системне програмне забезпечення – програмне забезпечення, призначене для керування роботою складових комп'ютера, забезпечення обміну даними між пристроями комп'ютера, діагностування та усунення недоліків у роботі комп'ютера, автоматизації процесу опрацювання даних, організації обміну даними між користувачем і комп'ютером, с. 9.

Системний диск – диск комп'ютера, на який встановлено операційну систему, с. 20.

Системи керування поглядом (англ. *Eye-Tracking Devices*) – програмні та апаратні засоби для відслідковування спрямованості зору людини та синхронізації з ним руху курсора на екрані, подання команд опусканням вій чи заплуцуванням очей, с. 33.

Слайсер – комп'ютерна програма, призначена для опрацювання 3D-моделей, що полягає в розділенні моделі на окремі шари з метою визначення координат переміщення друкувальної головки 3D-принтера під час друкування моделі, с. 192.

Соціальне здоров'я – здатність людини взаємодіяти з іншими, відчувати підтримку та приналежність до спільноти, с. 218.

Спадний список – список, у якому значення кожного наступного елемента менше, ніж значення попереднього, с. 117.

Спеціальні символи – символи, що не вводяться з клавіатури (математичні знаки, умовні позначки, піктограми), уставляються через меню **Вставлення** ⇒ **Символ**, с. 54.

Списки – особливий вид форматування абзаців текстового документа, які використовують для переліку об'єктів, опису порядку дій тощо, с. 50.

Стиль – набір значень властивостей для об'єкта (шрифт, розмір, колір, інтервали), що має власне ім'я та застосовується одним вибором, с. 71.

Стовпці – спосіб розміщення тексту в кілька вертикальних блоків, як у газетах чи журналах, с. 63.

Структура документа – ієрархічна схема складових частин документа (розділів, пунктів та підпунктів), с. 76.

Т

Таблиця – основа структури даних реляційної бази даних та основний об'єкт СКБД LibreOffice Base, с. 245.

Табуляція – засіб, який дає змогу розміщувати об'єкти в рядку текстового документа в чітко визначених місцях (позиціях табуляції), с. 66.

Тема документа – узгоджений набір стилів для всіх об'єктів документа (текст, таблиці, малюнки), що забезпечує єдиний дизайн, с. 73.

Технологія drag-and-drop – спосіб перенесення об'єктів між програмами шляхом перетягування мишею, с. 264.

Тривимірна графіка, або **3D-графіка**, – розділ комп'ютерної графіки, у якому вивчають методи і засоби створення графічних моделей просторових об'єктів з використанням комп'ютерних програм, с. 141.

Тривимірна навігація – операції змінення масштабу відображення, точки перегляду та способу подання виду сцени в редакторах 3D-графіки, с. 160.

Тривимірні примітиви – моделі геометричних тіл, з об'єднання та перетину яких утворюється форма поверхні об'єкта 3D-моделювання, с. 147.

Троянські програми – вид шкідливих програм, які проникають на комп'ютери користувачів разом з іншими програмами, які користувач «отримує» комп'ютерними мережами або на змінному носії, с. 224.

У

Універсальний дизайн програмних продуктів – передбачає створення програм, які можуть бути використані людьми з різними потребами без додаткової адаптації, с. 34.

Утиліти – програми, призначені для діагностування апаратної та програмної складових комп'ютера, розширення можливостей операційних систем, с. 23.

Ф

Файл-джерело – документ, із якого копіюється об'єкт для вставлення в інший документ, с. 263.

Файлова система – визначає структуру збереження даних на зовнішніх носіях та сукупність програм, які забезпечують роботу із цією структурою, с. 18.

Файл-приймач – документ, у який вставляється об'єкт із зовнішнього джерела, с. 263.

Формула (рівняння) – математичний чи науковий вираз, що вставляється в документ за допомогою конструктора формул (Word Equation Editor), с. 54.

Х

Хакер – особа, метою якої є втручання в комп'ютерні системи інших користувачів зі злочинними намірами або для власного самоствердження, с. 201.

Хробаки (черв'яки) комп'ютерних мереж – вид шкідливих програм, які пересилають свої копії комп'ютерними мережами задля проникнення на віддалені комп'ютери. Більшість черв'яків поширюються, прикріпившись до файлів електронної пошти, електронних документів тощо, с. 224.

Ц

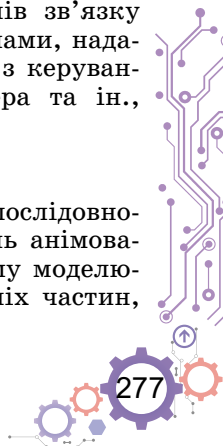
Цифровий слід – сукупність даних, які людина створює і залишає в результаті своєї діяльності в комп'ютерних мережах, с. 206.

Я

Ядро операційної системи – набір програм, з використанням яких виконуються команди, розподіляються ресурси пам'яті та каналів зв'язку між пристроями і програмами, надаються розширені функції з керування пристроями комп'ютера та ін., с. 18.

З

3D-анімація – створене з послідовності тривимірних зображень анімоване 3D-зображення, у якому моделюється рух об'єктів або їхніх частин, с. 142.



3D-друк – процес виготовлення на 3D-принтерах твердих копій 3D-моделей, с. 144.

3D-зображення – створене на основі просторової моделі плоске зображення тривимірного об'єкта, с. 142.

3D-сканери – технічні пристрої, призначені для створення 3D-моделей, с. 146.

В

BIOS – базова система введення / виведення – незалежний від конкретної версії операційної системи набір базових команд, які використовуються для забезпечення обміну даними між пристроями, с. 18.

D

DDoS-атака (англ. *Distributed Denial of Service* – розподілена відмова в обслуговуванні) – кібератака, метою якої є порушення доступності ІС, вебсайту або сервісу шляхом перевантаження його запитами від великої кількості пристроїв одночасно, с. 205.

I

install.exe – файл, що зазвичай містить алгоритм (програму) встановлення програмного забезпечення на комп'ютер в операційній системі **Windows**, с. 20.

L

LibreOffice Base – це СКБД для роботи з порівняно невеликими за об-

сягом базами даних для особистого користування та для невеликих організацій, с. 238.

М

MAC-адреса (англ. *Media Access Control* – управління доступом до посередників) – унікальний ідентифікатор пристрою в комп'ютерній мережі, с. 214.

О

OEM (англ. *Original Equipment Manufacturer* – оригінальний виробник обладнання) – ліцензія, що надається разом з новим обладнанням, наприклад з новим ноутбуком, с. 11.
OLE (Object Linking and Embedding) – технологія зв'язування та вбудовування об'єктів у документи, яка забезпечує інтеграцію даних з різних програм, с. 263.

S

setup.exe (англ. *set-up* – постава, система, план) – файл, який зазвичай містить алгоритм (програму) встановлення програмного забезпечення на комп'ютер в операційній системі **Windows**, с. 20.

W

Windows Defender (англ. *захисник Windows*) – програма захисту комп'ютерів з операційною системою **Windows** від шкідливих програм, с. 227.

ЗМІСТ

Дорогі дев'ятикласники та дев'ятикласниці!	
Шановні вчителі та вчительки!	3

Розділ 1. Програмне забезпечення

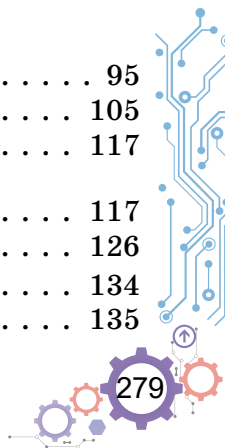
1.1. Інформаційні системи, інформаційні процеси в них. Програмне забезпечення персонального комп'ютера	5
1.2. Операційні системи. Сумісність програмного забезпечення. Драйвери	17
1.3. Особливості роботи із цифровими технологіями, створеними для користувачів з особливими освітніми потребами, зокрема для осіб з інвалідністю	28
<i>Практична робота № 1. «Використання голосового введення тексту»</i>	40

Розділ 2. Опрацювання текстових даних

2.1. Системи опрацювання текстів. Робота з текстовими документами.	41
2.2. Створення в текстовому документі багаторівневих списків, вставлення спеціальних символів і формул	50
2.3. Створення в текстовому документі стовпців, розділів, колонтитулів.	62
<i>Практична робота № 2. «Створення текстового документа, що містить об'єкти різних типів»</i>	70
2.4. Використання стилів у текстових документах. Структура текстового документа	71
2.5. Зміст документа. Гіперпосилання в документі. Спільна робота над документом.	83
<i>Практична робота № 3. «Стилі. Структура документа. Автоматизоване створення змісту. Гіперпосилання. Рецензування документів»</i>	93

Розділ 3. Алгоритми та програми

3.1. Одновимірні масиви. Списки в Python , їх уведення та виведення.	95
3.2. Опрацювання списків	105
<i>Практична робота № 4. «Опрацювання списків»</i>	117
3.3. Сортювання даних у списках. Пошук у списках даних, які відповідають певним умовам	117
3.4. Рядкові величини	126
<i>Практична робота № 5. «Опрацювання рядків»</i>	134
3.5. Модулі. Використання модулів у проєктах.	135



Розділ 4. Тривимірна графіка

4.1. Тривимірна графіка. Принципи тривимірного моделювання	141
4.2. Редактор 3D-графіки Blender . Уставлення та перетворення тривимірних примітивів. Принципи тривимірної навігації	152
4.3. Редагування форми та вигляду тривимірних об'єктів	167
<i>Практична робота № 6. «Створення 3D-моделей»</i>	<i>180</i>
4.4. Створення 3D-анімації	180
<i>Практична робота № 7. «Створення 3D-анімації»</i>	<i>189</i>
4.5. 3D-друк	190

Розділ 5. Інформаційна безпека

5.1. Інформаційні загрози. Інформаційна безпека. Цифровий слід	201
5.2. Захист персональних даних. Права людини в інформаційному середовищі	212
5.3. Шкідливе програмне забезпечення та боротьба з ним	222

Розділ 6. Бази даних

6.1. Бази даних. Системи керування базами даних	234
6.2. Поняття таблиці, поля, запису, ключа таблиці бази даних. Сортування та фільтрування даних. Запити	245
<i>Практична робота № 8. «Створення запитів на вибірку даних»</i>	<i>259</i>

Розділ 7. Практикум з використання інформаційних технологій

Навчальний проєкт та етапи його виконання	260
Інтегроване використання засобів опрацювання електронних документів	262
Технологія OLE	263
Імпорт і експорт даних із файлів	266
Оцінювання якості розроблених інформаційних продуктів	267
Підсумковий проєкт	270
Словник	272

Навчальне видання

**РИВКІНД Йосиф Якович,
ЛИСЕНКО Тетяна Іванівна,
ЧЕРНІКОВА Людмила Антонівна,
ШАКОТЬКО Віктор Васильович**

ІНФОРМАТИКА

**Підручник для 9 класу
закладів загальної середньої освіти**

*Рекомендовано
Міністерством освіти і науки України*

**Видано за рахунок державних коштів.
Продаж заборонено**

**Підручник відповідає Державним санітарним нормам і правилам
«Гігієнічні вимоги до друкованої продукції для дітей»**

У підручнику використано ілюстративний матеріал з відкритих джерел інтернету, зокрема сайтів *vecteezy.com*, *depositphotos.com*. Усі матеріали в підручнику використано з навчальною метою відповідно до законодавства України про авторське право і суміжні права.

**Редактор *Наталія Дашко*
Обкладинка *Олени Мамаєвої*
Макет, художнє оформлення,
комп'ютерна обробка ілюстрацій *Василя Марущинця*
Комп'ютерна верстка *Юрія Лебедєва*
Коректори *Ірина Хмельовська, Олена Симонова***

**Формат 70×100/16. Ум. друк. арк. 22,75. Обл.-вид. арк. 19,50.
Тираж 238 222 пр. Вид. № 0113. Зам. № 26-04-0705.**

**ТОВ «Гене́за»,
вул. Генерала Алмазова, 18/7 (літ. В), офіс 404, м. Київ, 01133, Україна.
Свідоцтво суб'єкта видавничої справи серія ДК № 7692 від 24.10.2022.**

**Віддруковано у ТОВ «ПЕТ»,
вул. Максиміліанівська, 17, м. Харків, 61024, Україна.
Свідоцтво суб'єкта видавничої справи серія ДК № 6847 від 19.07.2019.**

ЯК ЗМЕНШИТИ ЗАГРОЗИ ПІД

ЧАС РОБОТИ З КОМП'ЮТЕРОМ?

