

Морзе Н. В., Барна О. В.

Інформатика

«Інформатика»

підручник для 9 класу закладів загальної середньої освіти

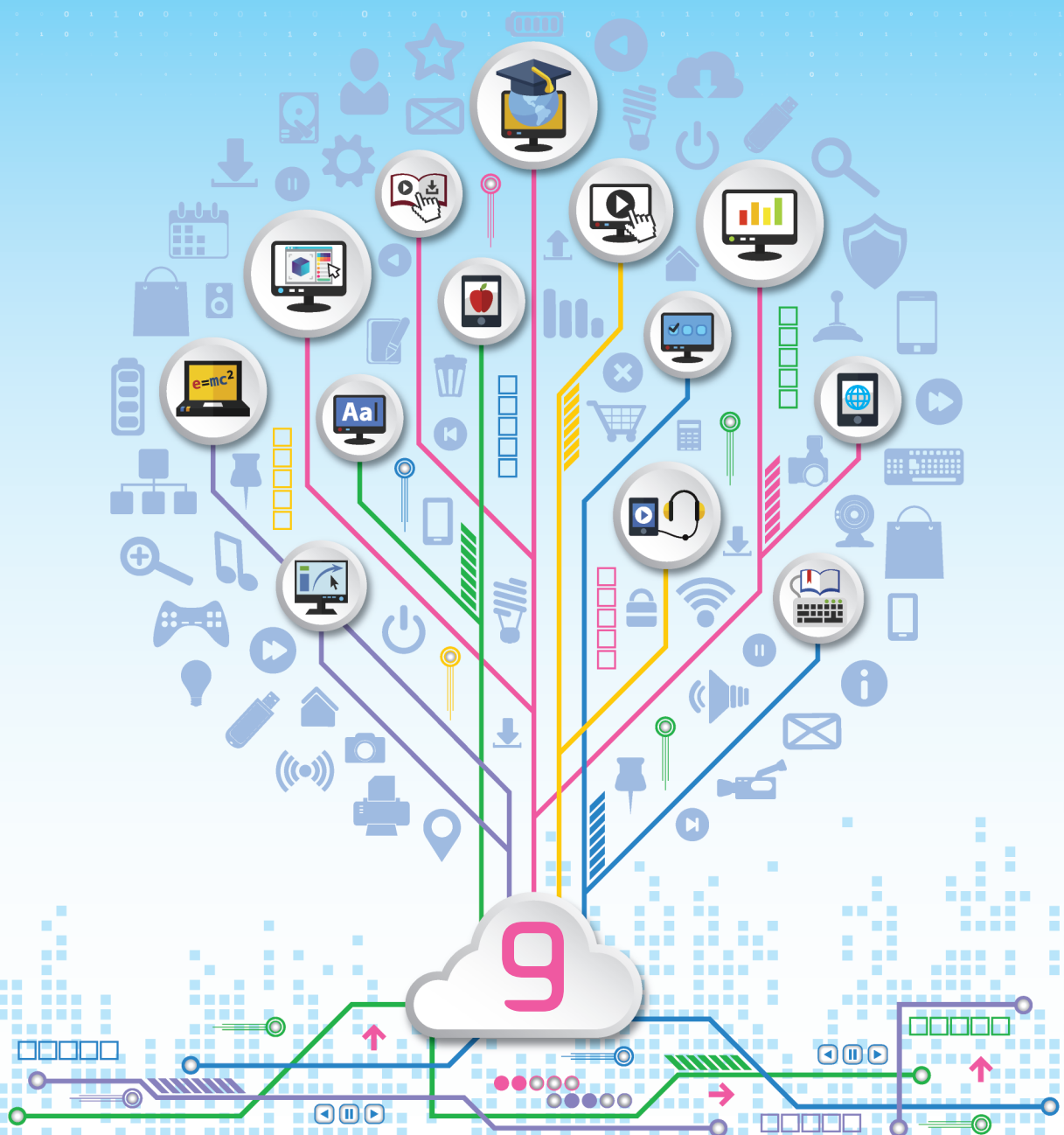
УОВЦ «Оріон»

Оріон

Наталія Морзе,
Ольга Барна



ІНФОРМАТИКА



Н. В. Морзе, О. В. Барна

ІНФОРМАТИКА



Підручник для 9 класу
закладів загальної середньої освіти

Схвалено для використання в освітньому процесі

УДК 004*кл9(075.3)
М79

Схвалено для використання в освітньому процесі
[рішення експертної комісії з інформатики
від 17 листопада 2021 року (протокол № 13) й експертної комісії
з проведення антидискримінаційної експертизи
від 19 листопада 2021 року (протокол № 5),
№ 3.0530-2021 у Каталогі надання грифів навчальній літературі
та навчальним програмам

Н. В. Морзе

М79 Інформатика. Підручник для 9 кл. закладів загальної серед-
ньої освіти / Н. В. Морзе, О. В. Барна. — Київ : УОВЦ «Оріон»,
2022. — с. : іл.

ISBN 000.

УДК 004*кл9(075.3)

ISBN 000

© Н. Морзе, О. Барна, 2022
© УОВЦ «Оріон», 2022

ДОРОГІ ДЕВ'ЯТИКЛАСНИКИ ТА ДЕВ'ЯТИКЛАСНИЦІ!

Цей підручник допоможе вам завершити базовий курс вивчення інформатики й упевнено стартувати в нових відкриттях цифрових технологій, які чекатимуть на вас у майбутньому. Навчальний матеріал згруповано в п'ять розділів. Ви дізнаєтесь про програмне забезпечення комп'ютера та заходи з інформаційної безпеки. Удосконалисте свої навички роботи з табличним процесором. Складатимете програми для опрацювання наборів даних, отримаєте компетентності роботи з великими даними. На завершення — опануєте навички роботи з 3D-графікою. І що найважливіше, отримаєте цифрові та ключові компетентності, які допоможуть вам успішно навчатися, обрати майбутню професію, реалізувати себе в цифровому суспільстві, вирішувати проблеми та бути успішними.

Кожний розділ містить кілька тематичних блоків, які ви вивчатимете протягом кількох уроків. Розділ завершується рубрикою **Оцінюємо себе**. Простий інструмент для самооцінювання допоможе вам визначити свій поступ у навчанні.

Щоб досягти очікуваних результатів навчання та бути компетентними в цифровому світі, радимо:

- ✓ напередодні уроку читати матеріали рубрики **Вивчаємо**, щоб разом з учнями та вчителем або вчителькою обговорити незрозуміле й поділитись вивченим;
- ✓ виконати всі завдання з рубрики **Діємо** й бути готовими надати допомогу тим, хто поки не може оволодіти базовими вміннями без сторонньої допомоги;
- ✓ постійно експериментувати під час виконання завдань рубрики **Досліджуємо**;
- ✓ постійно співпрацювати з іншими, навчати одне одного під час виконання завдань рубрики **Працюємо в парах**;
- ✓ планувати свою навчальну діяльність, мислити логічно, шукати нестандартні розв'язки завдань, робити висновки та самостійно приймати рішення, виконуючи завдання з рубрики **Працюємо самостійно**;
- ✓ обирати не тільки завдання базового рівня, а й ті, що потребують креативного мислення, здатності аналізувати, оцінювати, творити,

— саме вони позначені  та , .

Ви вже дорослі. Тому маєте розуміти, що саме від вас залежить, чого ви можете досягти у своєму житті. Використовуйте цей підручник та різноманітні навчальні ресурси, щоб пізнавати нове, навчатися завжди і скрізь.

Успіхів вам у пізнанні цікавих загадок цього світу!

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ІНФОРМАЦІЙНА БЕЗПЕКА

1

ДЛЯ ЧОГО ПОТРІБНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРА?

ПРИГАДАЙТЕ:

- що називають програмою;
- де відображаються назви програм, встановлених на комп'ютері;
- як запустити програму на виконання.

ВИ ДІЗНАЄТЕСЯ:

- яке програмне забезпечення потрібне для роботи комп'ютера;
- якими є правила використання програмного забезпечення;
- які існують види ліцензій на програмне забезпечення.



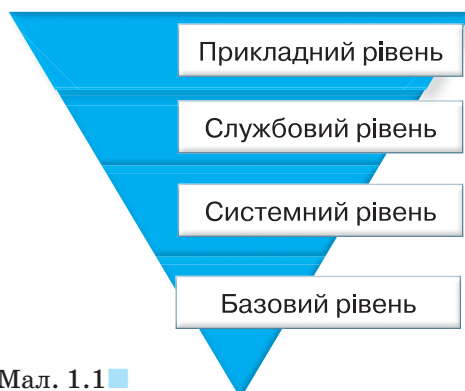
ВИВЧАЄМО

1.1. Яке програмне забезпечення потрібне для роботи комп'ютера?

Ви вже знаєте, що без програм комп'ютер не зможе працювати, а буде лише набором електронних пристроїв.

Для розв'язування задач на комп'ютері потрібно, щоб кожна програма була налагоджена відповідно до його характеристик. Тому стосовно роботи на комп'ютері часто використовують термін **програмне забезпечення** (*software*), під яким розуміють сукупність програм і правил, а також документації щодо управління роботою комп'ютера для опрацювання даних.

Розрізняють програми різного рівня, кожний з яких має відповідне призначення. Схематично структуру програмного забезпечення наведено на малюнку 1.1.



Мал. 1.1

Програми **базового рівня** зберігаються у спеціальних мікросхемах постійного запам'ятовуючого пристрою (ПЗП) та утворюють базову систему введення-виведення — BIOS. Програми й дані записуються в ПЗП на етапі виробництва мікросхем і не можуть бути змінені в процесі експлуатації. Програми цього рівня забезпечують взаємодію з базовими апаратними засобами.

Програми **системного рівня** забезпечують взаємодію інших програм комп'ютера з програмами базового рівня та безпосередньо з апаратним забезпеченням. При під'єднанні до комп'ютера нового обладнання на системному рівні має бути встановлена програма, що забезпечує для решти програм взаємозв'язок із цим пристроєм. Конкретні програми, призначені для взаємодії з конкретними пристроями, називають **драйверами**. Інша група програм системного рівня відповідає за підтримку взаємодії з користувачем. Завдяки таким програмам є можливість вводити дані до комп'ютера, управляти його роботою та отримувати результат опрацювання потрібних даних у зручному для користувача вигляді. Це засоби забезпечення користувацького інтерфейсу, від них залежить зручність і продуктивність роботи з комп'ютером.

Програми **службового рівня** взаємодіють як із програмами базового рівня, так і з програмами системного рівня. Призначення службових програм (утиліт) полягає в автоматизації перевірки та налаштуванні комп'ютера, а також для покращення функцій системних програм, підвищення ефективності роботи комп'ютера та розширення можливостей його використання. До таких програм належать:

- програми для роботи з архівами даних (наприклад, *7-zip*, *WinRar*);
- програми оптимізації розміщення даних на диску (наприклад, *defrag*);
- антивірусні програми (наприклад, *Intel Security-McAfee*, *Symantec Norton™ Security*, *Zillya! Internet Security*);
- програми тестування комп'ютера (наприклад, *ScanDisk*) тощо.

ЦІКАВО

Програми службового рівня називають **утилітами**. У розробці та експлуатації службових програм існує два альтернативні напрямки: *інтеграція з операційною системою та автономне функціонування*.

Програмне забезпечення **прикладного рівня** являє собою комплекс прикладних програм, за допомогою яких користувачі можуть виконати конкретні завдання (від виробничих до творчих, розважальних і навчальних). Між прикладним і системним програмним забезпеченням існує тісний взаємозв'язок.

Прикладні програми призначені для комп'ютерної підтримки виконання прикладних завдань. Розрізняють прикладні програми загально-го та професійного (спеціального) призначення (мал. 1.2).



До прикладного програмного забезпечення **загального призначення** належать: текстові й графічні редактори та процесори, програми створення мультимедійних презентацій, табличні процесори, системи управління базами даних, засоби підтримки комунікацій тощо. До цієї групи програм належать також програми для комп'ютерної підтримки вивчення різних навчальних предметів, іноземних мов, віртуальні фізичні та хімічні лабораторії, програми для електронного перекладу з іноземних мов, швидкого набору тексту на клавіатурі тощо. Досить зручно користуватись електронними довідниками та енциклопедіями. Головною їх перевагою перед паперовими аналогами є компактність і зручність у пошуку відомостей та даних.

Прикладне програмне забезпечення **професійного (спеціального) призначення** потрібне для вузькоспеціалізованого використання. У таких програмах враховується специфіка конкретних задач, і складені вони на основі спеціальних методів подання та опрацювання даних, властивих конкретній галузі практичної діяльності людей. До програм професійного призначення належать програми для проведення математичних обчислень (наприклад, *Mathlab*, *MathCad*), системи автоматизованого проектування (наприклад, *AutoCad*), програми для проведення бухгалтерських операцій (наприклад, *M.E.Doc*), редактори тривимірної графіки та анімації (наприклад, *Blender*) тощо.

Для розробки як прикладного, так і системного програмного забезпечення комп'ютера існують спеціальні програми — **інструментальне програмне забезпечення**. Такими інструментальними засобами є системи програмування, призначені для сприймання та опрацювання програм, записаних однією чи декількома мовами програмування (наприклад, C, C#, C++, Python, Visual Basic). Крім того, є системи, що підтримують кілька мов програмування, наприклад *Microsoft Visual Studio.NET*.

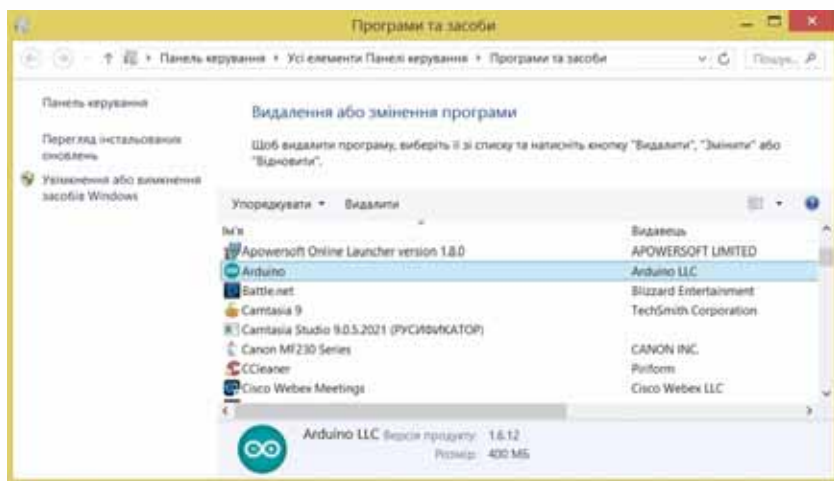


ДІЄМО

Вправа 1. Перелік програм.

Завдання. Створіть інформаційну модель програм, що встановлені на персональному комп'ютері, за яким ви працюєте у школі. Дійте за таким планом.

1. Оберіть форму подання результату виконання завдання. Це може бути карта знань, організаційна діаграма чи таблиця.
2. Оберіть середовище для створення моделі. Наприклад, редактор карт знань, текстовий процесор, електронна таблиця.
3. Створіть модель, яка відповідає структурі програмного забезпечення.
4. Доповніть модель (гілки, блоки чи рядки/стовпці таблиці) прикладами програм. Для цього в контекстному меню *Головного меню* оберіть *Програми та засоби* (мал.1.3). Розгляньте назви програм зі списку. Запишіть їх у відповідну позицію моделі. За потреби знайдіть в інтернеті призначення знайдених програм.



Мал. 1.3

5. Збережіть файл із виконаним завданням та надішліть його на ресурс, який вкаже вчитель/вчителька.

1.2. Якими є правила використання програмного забезпечення?

Комп'ютерні програми створюють програмісти. Розробка нового програмного забезпечення (ПЗ) — трудомісткий і тривалий процес, що потребує глибоких знань і певних навичок, насамперед у галузі математики й інформатики. Будь-які програми мають ціну та власника, тобто комусь належать.

Більшість програмних продуктів є комерційними, тобто передбачають плату за їх використання. Наприклад, на платній основі розповсюджується таке програмне забезпечення, як операційна система *Windows*, пакет прикладних офісних програм *Microsoft Office*, антивірусне програмне забезпечення, програми для комп'ютерної підтримки навчання, комп'ютерні ігри.

Будь-яке програмне забезпечення розповсюджується на основі ліцензійних угод, а не лише продається та купується. Програми, що розповсюджуються безкоштовно, також передбачають ознайомлення з ліцензійною угодою та підтвердження користувачем дотримання правил використання програмного забезпечення, визначених такою угодою.



Ліцензія на програмне забезпечення — правовий документ, що визначає правила використання та поширення програмного забезпечення.

Програмне забезпечення є об'єктом інтелектуальної власності, усі права на нього належать розробнику. Це право захищається Законом України «Про авторське право і суміжні права». За цим законом під час продажу програмного забезпечення розробник не передає кінцевому користувачеві свої права на певну програму, а лише дозволяє використовувати (ліцензує) цю програму. На упаковці програмного продукту, як правило, зазначається стисла ліцензійна угода, яка визначає основні права та обов'язки виробника і власника одержаного програмного продукту.

На жаль, не всі програмні продукти, що продаються, є легальними. З юридичної точки зору до піратського програмного забезпечення належать усі комп'ютерні програми, які розповсюджуються, встановлюються на комп'ютери й використовуються з порушенням умов їх ліцензійної угоди. Наприклад, самостійно створені копії ліцензійної програми перестають бути легальними, а подібні дії є порушенням авторських прав і спричиняють правову відповідальність.

Ліцензійне програмне забезпечення здебільшого може записуватись на окремий компакт-диск і постачатись разом із супроводжувальною документацією в кольоровій картонній коробці або в спеціальній упаковці (мал. 1.4). Доступ до ліцензійного програмного забезпечення можна

отримати також на сайті виробника, вказавши персональний пароль ліцензії.



■ Мал. 1.4 ■

1.3. Які існують види ліцензій на програмне забезпечення?

Розрізняють різні види ліцензій на програми. Основні з них:

- власницькі (пропрієтарні);
- вільні;
- відкриті.

Вони істотно різняться щодо прав кінцевого користувача на використання програми.

Власницька, або пропрієтарна, ліцензія (від англ. *proprietary* — власницький) передбачає, що розробник ПЗ дає дозвіл користувачу використовувати одну або декілька копій програми, але при цьому залишається правовласником усіх цих копій. Таким чином, практично всі права на ПЗ залишено за розробником, а користувач здобуває лише дуже обмежений набір окреслених прав. Для пропрієтарних ліцензій типовим є накладання великої кількості умов, що забороняють певні варіанти використання ПЗ, навіть тих, які без цієї заборони були б дозволені Законом про авторське право і суміжні права. Прикладом пропрієтарної ліцензії може бути ліцензія на операційну систему *Microsoft Windows*, яка включає великий список заборонених варіантів її використання.

Найзначнішим наслідком застосування пропрієтарної ліцензії є те, що кінцевий користувач зобов'язаний прийняти її, бо за законом власником ПЗ є не користувач, а розробник програми. У разі відмови прийняти ліцензію користувач взагалі не може працювати з такою програмою.

Вільні та відкриті ліцензії не залишають права на конкретну копію програми її розробнику, а передають найважливіші з них кінцевому користувачу, який і стає власником. У результаті користувач отримує важливі права, які Закон про авторське право зазвичай дає лише власнику копії. Проте всі авторські права на ПЗ, як і раніше, залишаються в розробника.

Ліцензія Freeware (безкоштовне ПЗ) не потребує виплат правовласнику, не має обмежень щодо функціональності й часу роботи. Проте таке ПЗ можна поширювати без тексту програми, і можуть бути обмеження

щодо комерційного використання чи модифікації програми. Ліцензія *Free software* (вільне ПЗ) надає максимальну кількість прав користувачам: користування, поширення, модифікація. Для цього типу ПЗ створюються спеціальні ліцензії для врегулювання прав й обов'язків авторів і користувачів.

Відмінність між безкоштовним і вільним ПЗ полягає в тому, що згідно з ліцензією *Freeware* користувачі не мають права поширювати програми, дарувати, модифікувати тощо, а згідно з *Free software* — це дозволяється. Часто, хоча не завжди, вони відрізняються ще й тим, що вільне ПЗ надають разом з текстом програми.



ДІЕМО

Вправа 2. Ліцензії та авторське право.

Завдання. Визначте, які типи ліцензій описано у статті Вікіпедії та як така типологія пов'язана із Законом про авторське право і суміжні права.

Дійте за таким планом.

1. Відкрийте пошукову систему та ведіть пошукову фразу для знаходження відомостей.
2. Серед переліку посилань оберіть те, яке вказує як джерело Вікіпедію.
3. Перегляньте статтю про ліцензію на програмне забезпечення. Знайдіть потрібні відомості. Зробіть висновок.



ДОСЛІДЖУЕМО

Вправа 3. Ліцензія на операційну систему.

Завдання. Дослідіть умови використання встановленої операційної системи. Якщо ви працюєте з операційною системою *Windows*, то у контекстному меню *Головного меню* оберіть вказівку *Система*. У вікні, що відкриється, оберіть розділ **Активация Windows**. Прочитайте умови ліцензії на програмний продукт. Зробіть висновок.



ПРАЦЮЕМО В ПАРАХ

1. Поясніть одне одному за поданим на малюнку 1.3 списком програм, для виконання яких завдань можна використати комп'ютер.
2. Схарактеризуйте ліцензії на програмне забезпечення за ознакою «відкритість». Скористайтесь схемою та описом умовних позначень (мал. 1.5).



ліцензія Creative Commons

зазначення авторства

розповсюдження

допускається комерційне використання

без зазначення авторства

модифікація/зміна

комерційне використання заборонене

усі права зареєстровані

без зміни ліцензії

комерційне використання, крім Японії

громадське надбання

без модифікації

комерційне використання, крім країн ЄС

Мал. 1.5



ПРАЦЮЄМО САМОСТІЙНО

3. Доповніть класифікаційну схему програмного забезпечення комп'ютера, збережену в папці *Забезпечення комп'ютера* у файлі *Програмне забезпечення*, прикладами програм із різними ліцензіями. За потреби скористайтесь відомостями з інтернету.



4. Попрацюйте в мініпроекті «Порадник програм». Оберіть професію, для якої складіть список мінімального набору програмного забезпечення, яке, на вашу думку, необхідне для виконання можливих завдань. За потреби знайдіть в інтернеті відомості про особливості професійної діяльності обраного спеціаліста із переліку на постері Професії-2030, який зберігається в папці *Забезпечення комп'ютера* (мал. 1.6).

ЯКІ ПРОФЕСІЇ БУДУТЬ ПОПУЛЯРНИМИ В УКРАЇНІ ЧЕРЕЗ 10 РОКІВ



**РОЗРОБИК
ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ**



**РОЗРОБНИЦЯ
ІНТЕРНЕТ-КОНТЕНТУ**



**ПРОФЕСІОНАЛ ПО
РОБОТІ З КЛІЄНТАМИ**



ФІТНЕС-ТРЕНЕРКА



ОНЛАЙН-ВЧИТЕЛЬ



**СПЕЦІАЛІСТКА
ІЗ ЦИФРОВОГО
МАРКЕТИНГУ**



КОУЧ (НАСТАВНИК)



АНАЛІТИК BIG DATA



ІНЖЕНЕР-ЕКОЛОГ



**МОДЕРАТОРКА
СПІЛЬНОТИ**



ІНЖЕНЕР ЛІКАРНІ



**ФАХІВЧИНЯ
З КІБЕРБЕЗПЕКИ**

Мал. 1.6

ДЛЯ ЧОГО ВИКОРИСТОВУЄТЬСЯ ОПЕРАЦІЙНА СИСТЕМА?



ПРИГАДАЙТЕ:

- що таке інтерфейс операційної системи та які об'єкти його складають;
- як у комп'ютері організовано роботу з файлами.



ВИВЧАЄМО

2.1. Для чого потрібна операційна система, які її функції?

Основою системного програмного забезпечення є операційна система (наприклад, *Windows*, *Linux*, *Android*).

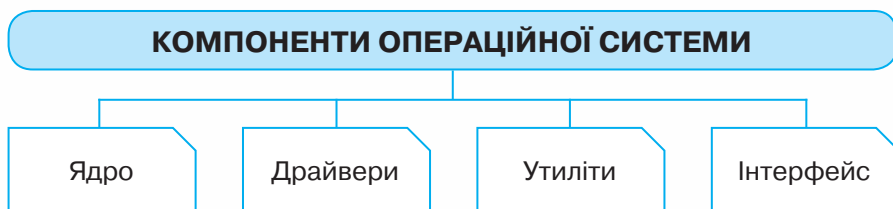


Операційна система (ОС) — це програмний комплекс, що забезпечує:

- управління ресурсами — злагоджену роботу всіх апаратних засобів комп'ютера;
- управління процесами — виконання всіх програм та їхню взаємодію з пристроями комп'ютера й даними;
- взаємодію (обмін відомостями й даними) між користувачем і комп'ютером.

Операційна система починає працювати одразу після ввімкнення комп'ютера. Певну її частину — **BIOS** — розміщено на мікросхемі постійної пам'яті. Ця частина містить сукупність програм, які після ввімкнення комп'ютера автоматично тестують усі його пристрої та в разі їх справної роботи завантажують в оперативну пам'ять частину операційної системи — програму-завантажувач. Далі вже вона завантажує в оперативну пам'ять комп'ютера потрібні для подальшої роботи модулі операційної системи. Після завершення завантаження ОС управління переходить до командного процесора — частини ОС, що забезпечує виконання команд користувача. Поки комп'ютер працюватиме, деяка частина операційної системи завжди залишатиметься в ОЗП. Цю частину ОС називають **резидентною**. За потреби до оперативної пам'яті буде завантажено інші частини ОС.

До складу сучасних операційних систем входять такі основні компоненти (мал. 2.1):



■ Мал. 2.1 ■

- **ядро** — центральна частина ОС, що забезпечує прикладним програмам координований доступ до ресурсів комп'ютера (часу, що витрачається процесором для опрацювання окремих завдань, оперативної пам'яті, зовнішніх пристроїв введення та виведення даних), перекладаючи їх команди з мови прикладних програм мовою двійкових кодів для подальшого опрацювання комп'ютером;
- **драйвери** — програми для перекладу вказівок комп'ютера мовою певного пристрою (принтера, сканера, звукової або відеокарти тощо) та навпаки;
- **утиліти** — допоміжні програми, призначені для обслуговування дисків, перевірки комп'ютера, налаштування параметрів роботи;
- **інтерфейс** — правила взаємодії операційної системи й користувача, які визначають зручність роботи.

До основних функцій операційної системи відносяться:

- створення середовища виконання і взаємодії прикладних програм;
- розподіл апаратних ресурсів комп'ютера між прикладними програмами;
- надання прикладним програмам засобів для ефективного використання пристроїв та виконання типових операцій з введення-виведення даних;
- зберігання даних пристроями пам'яті;
- надання інтерфейсу, за допомогою якого користувачі управлятимуть виконанням прикладних програм та вмістом пристроїв пам'яті;
- забезпечення взаємодії комп'ютерів у мережах.

Останнім часом операційні системи на комп'ютери встановлюють виробники чи компанії, що займаються комплектацією, продажем та обслуговуванням комп'ютерів. Однак користувачі за потреби можуть самостійно встановити чи замінити операційну систему на своєму комп'ютері. Для цього необхідно придбати відповідний пакет програм, перевірити наявність ліцензії та встановити за певними правилами операційну систему на комп'ютер.

2.2. Як класифікують операційні системи?

Операційні системи (ОС) можна класифікувати за такими ознаками (мал. 2.2):

- **цільове призначення:**
 - для великих універсальних високопродуктивних ЕОМ (мейнфреймів);
 - для ПК;
 - для мобільних пристроїв;
 - для вбудованих систем;
- **кількість користувачів, які одночасно працюють із системою:** локальні — однокористувацькі та мережеві — багатокористувацькі;
- **кількість задач, які може розв'язувати користувач за їх допомогою одночасно:** однозадачні й багатозадачні;
- **інтерфейс користувача — основний спосіб взаємодії користувача з ОС:** за допомогою введених із клавіатури вказівок, вибору з меню або вказування на графічні зображення — WIMP (*Window, Image, Menu, Pointer*), SILK-інтерфейсом (*Speech, Image, Language, Knowledge*) — голосом; інтерфейс з рядком вказівок;
- **ресурси, мінімально необхідні для її роботи:** мінімальна ємність оперативної та дискової пам'яті, тип процесора;
- **відкритість:** можливість користувача, що знає мови програмування, вносити потрібні зміни в її окремі модулі;
- **кількість розрядів, що опрацьовують дані у процесорі одночасно:**
 - 32-розрядні;
 - 64-розрядні.

Класифікація ОС за основними ознаками



Мал. 2.2

Найвідомішими є операційні системи: *MS Windows*, *UNIX*, *GNU/Linux*, *OS/2*, *MacOS*, *iOS*, *Android* (мал.2.3). Однією з перших відомих операційних систем була *MS-DOS*, що в минулому встановлювалась на більшість комп'ютерів, проте із часом замінювалась на ОС з графічним інтерфейсом.



■ Мал. 2.3 ■

Наприклад, операційні системи *Windows* і *Linux* забезпечують роботу кількох користувачів одночасно (мережеві), а *MS-DOS* є однокористувацькою операційною системою; операційні системи *MS-DOS* й *UNIX* передбачають введення користувачем усіх вказівок із клавіатури, а під час роботи з операційною системою *Windows* користувачеві, щоб вказати на виконання операції, достатньо вибирати на екрані комп'ютера графічні об'єкти й меню. Операційна система *Linux*, на відміну від ОС *Windows*, має відкритий код.



ДОСЛІДЖУЄМО

Вправа 1. Обираємо операційну систему для мобільних платформ.

Завдання. За матеріалами статті <https://cutt.ly/3YFjZwz> складіть таблицю порівняння операційних систем *iOS* і *Android*. Сформулюйте критерії порівняння та вкажіть переваги кожної з операційних систем (+) чи недоліки (-). Підведіть загальний підсумок за побудованою таблицею та зробіть висновки.

2.3. Коли програмне забезпечення потребує інсталяції та деінсталяції?

Більшість програм постачають для продажу та поширення у стисненому вигляді. Для нормальної роботи їх потрібно розпакувати, а необ-

хідні дані правильно розташувати на комп'ютері, враховуючи відмінності між комп'ютерами й налаштуваннями користувача. У процесі встановлення виконують різні тести на відповідність заданим вимогам, а комп'ютер необхідним чином налаштовують для зберігання файлів і даних, необхідних для правильної роботи програми.

Процес встановлення (інсталяція) програмного забезпечення на комп'ютер користувача може бути здійснений за допомогою:

- *менеджера пакетів* — особливої програми у складі операційної системи (наприклад, *APT* в *Linux*, *Програми та засоби* в *Microsoft Windows*);
- *засобу встановлення* — спеціальної програми у складі самого програмного забезпечення.

Встановлення програмного забезпечення зазвичай включає в себе розташування всіх необхідних програмі файлів у відповідних місцях файлової системи, а також зміну та створення конфігураційних файлів. Менеджери пакетів також виконують контроль залежностей, перевіряючи, чи є в системі необхідні для роботи даної програми засоби, а в разі успішного встановлення реєструють новий програмний засіб у переліку наявних.

Деякі комп'ютерні програми створені таким чином, що їх встановлюють простим копіюванням файлів у потрібне місце. Про такі програми кажуть, що вони не вимагають інсталяції, їх поширюють копіюванням.

Інсталяція (встановлення) великих програм буває:

- типова (*typical, normal*);
- мінімальна (*minimum*);
- повна (*full*);
- керована користувачем (*custom*).

Вилучення (деінсталяцію) програм потрібно здійснювати за допомогою системних чи спеціальних програмних засобів. Просте вилучення файлів не призводить до вилучення програми з реєстру встановленого програмного забезпечення.



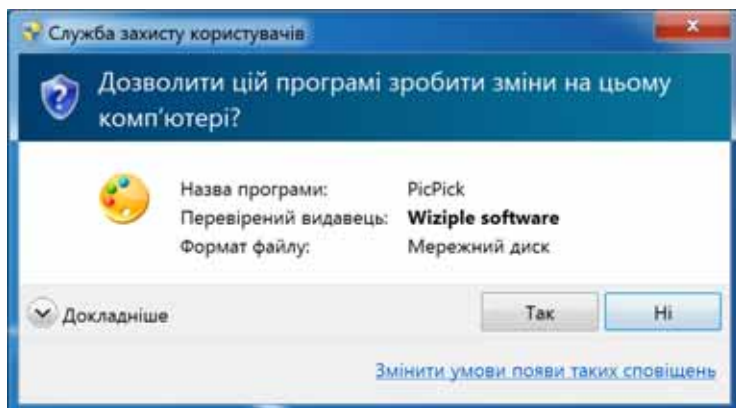
Вправа 2. Інсталяція програми *PicPick*.

Завдання. Інсталюйте графічний редактор *PicPick*.

1. У папці *Забезпечення комп'ютера* знайдіть інсталяційний пакет

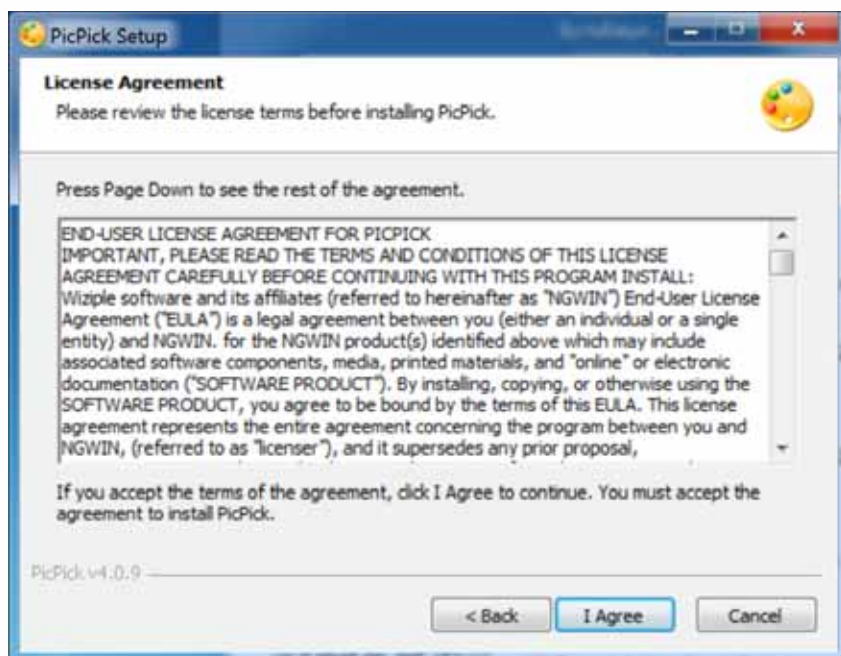
picpick_inst.exe  .

2. У вікні *Служба захисту користувачів* (мал.2.4) підтвердьте намір встановити програму на своєму комп'ютері.



■ Мал. 2.4 ■

3. У вікні встановлення програми прочитайте ліцензійну угоду щодо використання програми. У разі потреби скористайтесь онлайнним перекладачем. Натисніть кнопку *I Agree* (мал. 2.5) — підтвердження того, що ви приймаєте умови угоди та продовжуєте інсталяцію програми.



■ Мал. 2.5 ■

4. Оберіть місце розташування програми: *C:\Program Files(x86)\PicPick* і натисніть кнопку *Install*.
5. Дочекайтесь завершення процесу інсталяції.

2.4. Чи потрібно оновлювати драйвери пристроїв?

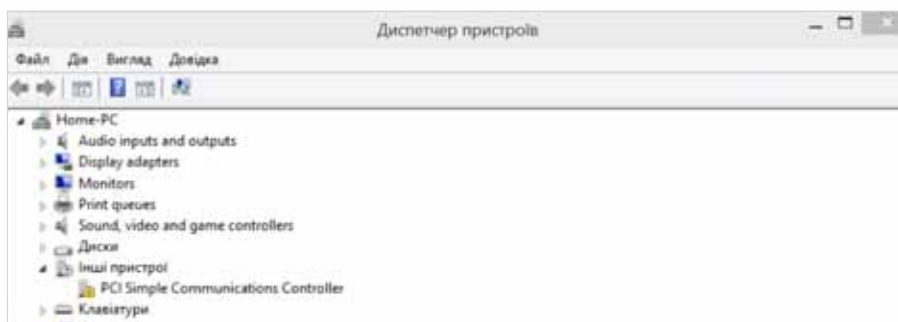
Вам відомо, що до комп'ютера можуть підключатися найрізноманітніші зовнішні пристрої — принтери, флешки, камери, модеми тощо. Незважаючи на величезну різноманітність моделей, у складі одного комп'ютера всі ці пристрої працюють злагоджено, оскільки операційна система «вміє» правильно з ними поводитися. І все завдяки наявності в ОС драйверів.

Драйвер — це невелика, як правило, безкоштовна програма, яка дає змогу операційній системі правильно використовувати будь-який пристрій, що входить до складу комп'ютера. Якщо на комп'ютері відсутній драйвер пристрою, він працюватиме неправильно, або може не працювати зовсім.

Драйвери розповсюджених пристроїв, а також драйвери основних важливих для комп'ютера частин, без яких він взагалі може не працювати, входять до складу операційної системи. Драйвери для всіх інших пристроїв встановлюються на комп'ютері додатково.

Windows влаштована так, що при першому підключенні деякого пристрою до комп'ютера автоматично запускається процес встановлення драйвера для нього. Якщо відповідний драйвер є в системі, встановлення відбувається без участі користувача. В іншому разі користувач отримує повідомлення про відсутність драйвера й необхідність установити його самостійно. Після встановлення на комп'ютері драйвера пристрою його файли копіюються на системний диск. При цьому назва й кількість згаданих файлів, а також їх розташування на диску, залежать від конкретного драйвера. Зазвичай операційна система «розкидає» їх по різних папках, розташованих у каталогах *Windows* і *Program Files*.

Головною ознакою проблем з драйвером якогось пристрою є його неправильна робота. Але щоб повністю переконатися у проблемах у роботі драйвера, потрібно зайти в *Диспетчер пристроїв*. Якщо на комп'ютері є драйвери для всіх пристроїв, диспетчер пристроїв матиме вигляд приблизно такий, як на малюнку 2.6. Якщо якийсь драйвер застарів, або система відслідкувала збої в його роботі, то в *Диспетчері пристроїв* буде позначка .



Мал. 2.6

Виробники комп'ютерних пристроїв періодично випускають для своїх виробів нові драйвери, пропонуючи їх для вільного завантаження на своїх сайтах. Але якщо драйвер пристрою на комп'ютері вже встановлений і проблеми в його роботі відсутні, міняти його на новіший особливо-го сенсу немає.



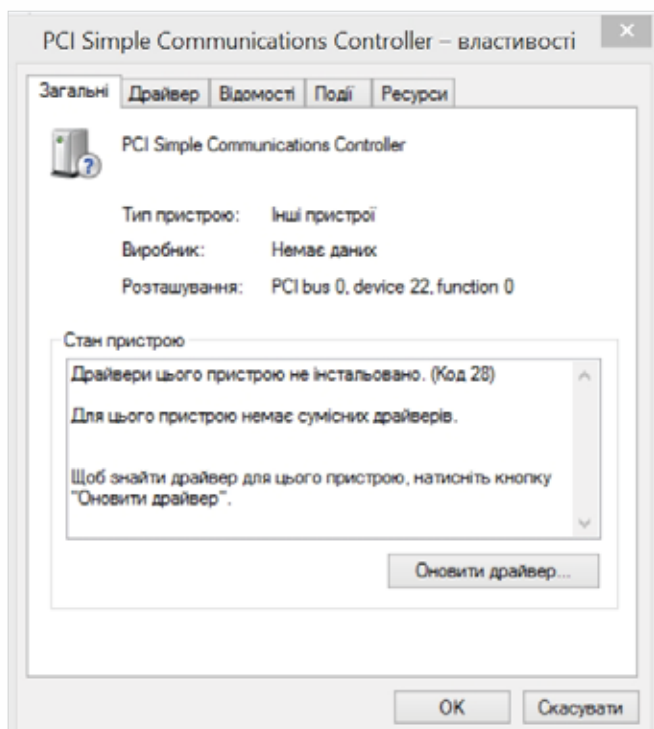
ДІМО

Вправа 3. Драйвери.

Завдання. Перевірте наявність помилок у драйверах установлених на ПК пристроїв.

Дійте за таким планом.

1. У контекстному меню *Головного меню* операційної системи оберіть *Диспетчер пристроїв*.
2. Перевірте, чи у списку встановлених драйверів є драйвер з позначкою про помилку.
3. У контекстному меню знайденого драйвера прочитайте інформацію та скористайтесь порадами щодо усунення неполадок. Наприклад, як на малюнку 2.7.



■ Мал. 2.7 ■



ДОСЛІДЖУЄМО

Вправа 4. Розв'язник проблем.

Завдання. Проаналізуйте сайт <https://www.driver-soft.com/>. Визначте, які ресурси можна завантажити із сайту та особливості їх завантаження. Чим можуть бути корисні матеріали із сайту для роботи з драйверами?

2.5. У яких випадках говорять про проблеми сумісності програмного забезпечення?

Сьогодні існує багато різних виробників як електронної складової, так програмного забезпечення комп'ютера. Це призвело до проблем їх сумісності при використанні. Якщо проблему сумісності за елементною базою вже майже вирішено, то проблема сумісності програмного забезпечення досі ще є актуальною. Розрізняють види сумісності програмного забезпечення на рівні:

- виконуваних файлів;
- програмних кодів (програма може бути на різних комп'ютерах під управлінням різних операційних систем);
- форматів файлів даних (програми можуть відрізнятися за інтерфейсом, набором функцій, але працювати з однаковими документами);
- мережевої сумісності (здатність програм обмінюватися даними по мережі).

Тестування сумісності (Compatibility Testing) — тестування програмного забезпечення, призначене для з'ясування сумісності програмного забезпечення з певним середовищем — операційною системою, платформою чи обладнанням.

Наприклад, файли з розширенням *doc*, створені в текстовому процесорі *Microsoft Word*, можна відкрити та редагувати у середовищі *LibreOffice Writer*, сайт <https://www.google.com/> має однаковий вигляд у браузері *Chrome* і *Firefox*, на екрані персонального комп'ютера та мобільного пристрою.



ПРАЦЮЄМО В ПАРАХ

1. З переліку програм (драйвер модема (1), прикладні програми забезпечення доступу до інтернету (2), антивірусні програми (3), операційна система (4) та текстовий редактор (5)) складіть «піраміду», що відображає порядок встановлення зазначених програм на комп'ютері для пошуку необхідних даних в інтернеті та створенні на їх основі текстового повідомлення із заданої теми. Обговоріть у парах можливі варіанти виконання цього завдання.

2. Чому в певні комп'ютерні ігри не можна пограти на деяких комп'ютерах? Сформулюйте три можливі причини. Обговоріть у парах.
3. Поясніть одне одному ланцюжок подій при розробці програмного забезпечення: *помилка, дефект (баг), збій* — за прикладом на малюнках. Чи можна це назвати прикладом несумісності програмного забезпечення?



ПРАЦЮЄМО САМОСТІЙНО

4. Знайдіть в інтернеті відомості про популярність сучасних операційних систем. На основі числових даних в табличному процесорі побудуйте діаграми, які ілюструватимуть знайдені відомості. Зробіть висновок про популярність операційної системи, яку ви використовуєте для персонального ПК та мобільного пристрою.
5. У списку інсталюваних на комп'ютері програм визначте три програми, які використовують дуже часто, та дві, які використовують рідко. Встановіть, який обсяг носія даних буде звільнено, якщо деінсталювати одну із програм, яку зовсім не використовують на цьому комп'ютері. Знайдіть в інтернеті відомості про призначення цієї програми та зробіть висновок про доцільність чи недоцільність її деінсталяції.
6. Візьміть участь у мініпроекті «Реклама операційної системи». Уявіть, що ви створюєте рекламу для однієї з фірм-розробників операційних систем. Підготуйте матеріали для такої реклами для потенційних користувачів, врахувавши, що вони краще сприймуть добре структуровану та чітку інформацію. Доберіть спосіб подання: інформаційний плакат, буклет, презентацію. Представте свою роботу в класі.



ПРИГАДАЙТЕ:

- які пристрої складають зовнішню пам'ять комп'ютера;
- як у комп'ютері кодують різні дані;
- як створюють текстовий документ у текстовому процесорі.

ВИ ДІЗНАЄТЕСЯ:

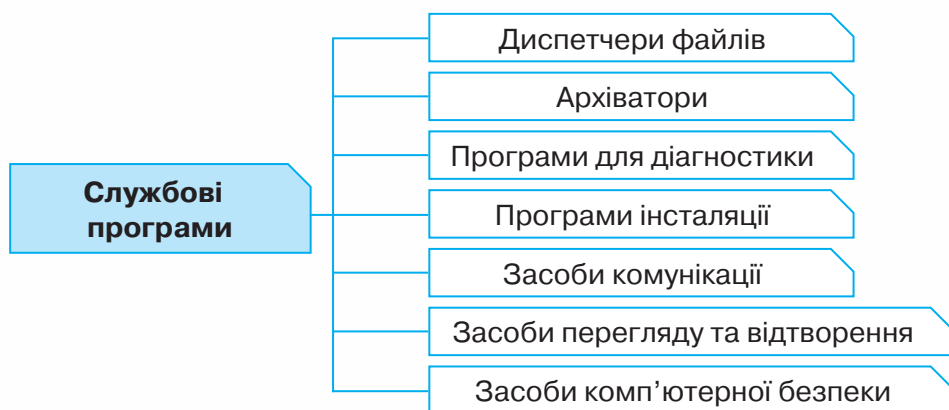
- які програми відносять до службових програмних засобів;
- що таке форматування носіїв даних та за допомогою якого програмного забезпечення воно здійснюється;
- які функції виконують програми-архіватори;
- у чому полягають основні методи стиснення даних.



ВИВЧАЄМО

3.1. Які програми відносять до службових програмних засобів?

До службових програм відносять такі групи програм (мал. 3.1):



■ Мал. 3.1 ■

Диспетчери файлів (файлові менеджери). За їх допомогою виконується більшість операцій з обслуговування файлової структури: копіювання, переміщення, перейменування файлів, створення папок, знищення об'єктів, пошук файлів та навігація у файловій структурі. Базові програмні засоби містяться у складі програм системного рівня і встановлюються разом з операційною системою.

Засоби стиснення даних (архіватори). Призначені для створення архівів, мають підвищену щільність запису даних і за рахунок цього ефективніше використовуються носії даних.

Засоби діагностики. Призначені для автоматизації процесів діагностування програмного та апаратного забезпечення. Їх використовують для виправлення помилок й для оптимізації роботи комп'ютерної системи.

Програми інсталяції (встановлення). Призначені для контролю за додаванням у поточну програмну конфігурацію нового програмного забезпечення. Вони слідкують за станом і зміною навколишнього програмного середовища, відслідковують і протоколюють утворення нових зв'язків. Прості засоби управління встановленням і знищенням програм містяться у складі операційної системи, але можуть використовуватись і додаткові службові програми.

Засоби комунікації. Дають змогу встановлювати з'єднання з віддаленими комп'ютерами, передають повідомлення електронної пошти тощо.

Засоби перегляду та відтворення. Переважно для внесення змін до файлів, їх необхідно відкрити в середовищі програмного засобу, що передбачає створення та редагування файлів відповідного формату. Але якщо вносити зміни не потрібно, існують універсальні засоби для перегляду (у випадку зображень) або відтворення (у випадку звуку чи відео) даних.

Засоби комп'ютерної безпеки. До них належать засоби пасивного й активного захисту даних від пошкодження, несанкціонованого доступу, перегляду та зміни даних. Засоби пасивного захисту — це службові програми, призначені для резервного копіювання. Засоби активного захисту застосовують антивірусне програмне забезпечення. Для захисту даних від несанкціонованого доступу, їх перегляду та зміни використовують спеціальні системи, базовані на криптографії.

3.2. Що таке форматування носіїв даних та за допомогою якого програмного забезпечення воно здійснюється?

До службового програмного забезпечення належать також програми для обслуговування носіїв даних. Найбільш уживаною серед них є програма форматування. Її зазвичай виконують перед установленням усіх програм на жорсткий диск, а також після придбання додаткового носія даних.



Форматування (англ. *formatting*) — процедура створення структур порожньої файлової системи вказаного типу — розподіл доріжок магнітного диска (жорсткого диска) або іншого носія даних (наприклад, флеш-накопичувача чи карти флеш-пам'яті) на фізичні чи логічні записи, що виконується перед першим використанням диска. Форматування при цьому супроводжується втратою даних, що були збережені на носіїві, який форматується.

У процесі форматування також може перевірятися цілісність носія та виправлятися пошкодження.

Форматування носія даних виконується в три етапи:

- низькорівневе форматування — це базова розмітка області зберігання даних, яка виконується на заводі-виробнику як одна з останніх операцій виготовлення пристрою зберігання даних;
- поділ носія на логічні диски (виконується за необхідності);
- високорівневе форматування.

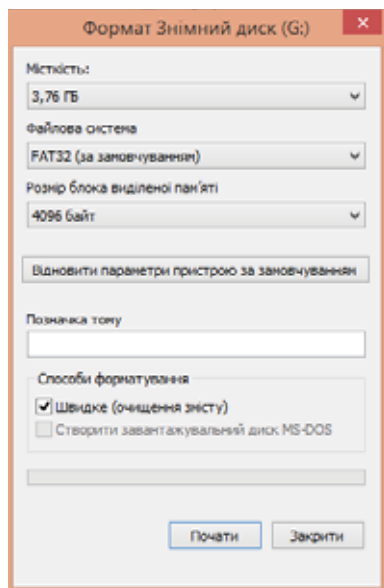
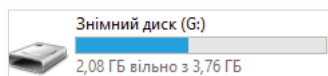
Інші типи пристроїв зберігання даних, зокрема флеш-накопичувачі та карти флеш-пам'яті, можуть надходити в продаж уже відформатованими виробником. Компакт-диски й DVD-диски використовують формат зберігання, відмінний від формату жорстких дисків і знімних запам'ятовуючих пристроїв.



Вправа 1. Форматування флеш-накопичувача.

Завдання. Відформатуйте флеш-накопичувач.

1. Приєднайте флеш-накопичувач до USB-роз'єму. Перевірте, чи відображається назва приєднаного пристрою у списку пристроїв вікна *Мій комп'ютер*.
2. Викличте контекстне меню знімного диску. Оберіть вказівку *Форматувати*.
3. У списку доступних файлових систем оберіть *FAT32*. Задайте швидкий спосіб форматування — очищення змісту (мал. 3.2).



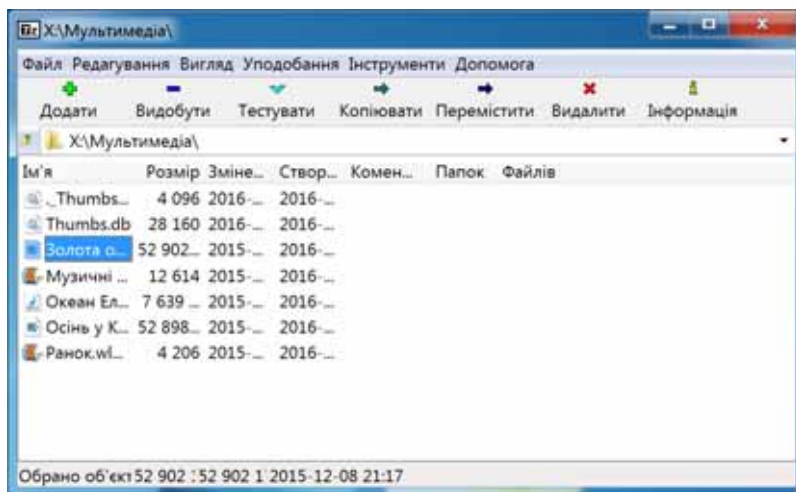
Мал. 3.2

4. Натисніть кнопку *Почати*. Дочекайтесь, поки процес форматування не завершиться. Він буде супроводжуватись індикатором процесу виконання завдання в нижній частині вікна. Завершіть форматування, натиснувши кнопку *Закрити*.
5. Перевірте, чи було в процесі форматування видалено вміст флеш-пам'яті.

3.3. Які функції виконують програми-архіватори?

При передаванні даних через комп'ютерну мережу, а також при збереженні резервних копій файлів суттєвим є їхній обсяг. Тому часто застосовують стискання файлів. Стискати можна не лише один файл, а й папку, що містить кілька файлів чи папок. Результатом стискання є запакований файл, або архів. Створювати такі файли та працювати з ними дають змогу спеціальні програми, які називають архіваторами та програмами резервного копіювання.

Часто вживаними є програми-архіватори *7-Zip* (мал. 3.3), *WinRar*, *WinZip* тощо. Архіви, як правило, мають розширення *7z*, *rar*, *zip* або інші.



Мал. 3.3

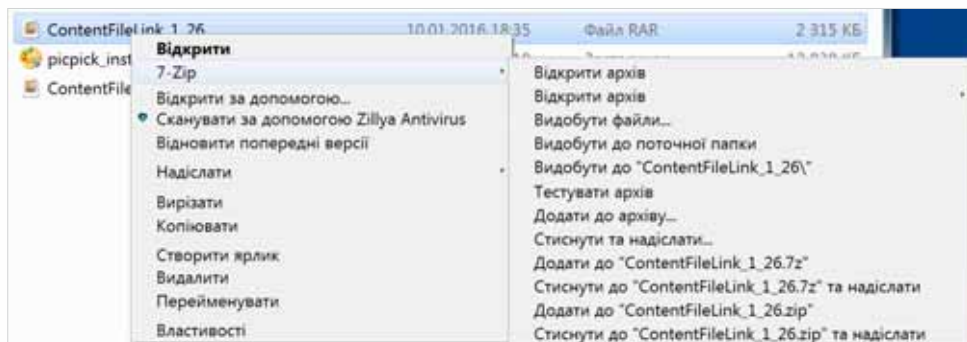
До базових функцій, які виконують більшість сучасних архіваторів, відносять:

- створення нових архівів;
- розпаковування файлів з архівів;
- додавання файлів до архіву;
- створення архівів, що саморозпаковуються;
- створення розподілених архівів на носіях малої ємності;
- тестування цілісності структури архівів;
- повне або часткове відновлення пошкоджених архівів;
- захист архівів від перегляду й несанкціонованої модифікації.

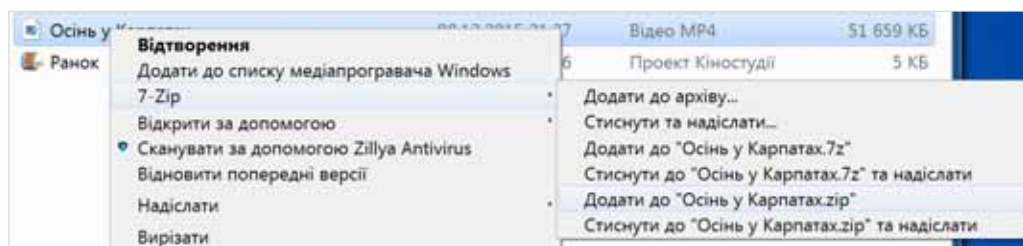
Архівація передбачає упакування та стискання даних. Упакування і стискання (компресія) — не одне й те саме. Упакування — це злиття кількох файлів або папок в єдиний файл, який називають архівом. Стискання ж — скорочення обсягу вихідного файлу або групи файлів.

У різних архіваторах застосовують різні способи стискання, тому обсяг файла архіву порівняно з вихідним файлом може різнитися залежно від програми-архіватора, за допомогою якої його створено.

Швидко створити файл архіву із значеннями його властивостей, що встановлені за замовчуванням, або розпакувати архів можна за допомогою контекстного меню: якщо на комп'ютері встановлено програму-архіватор, то вказівки для виконання найбільш вживаних операцій з архівами виносяться в контекстне меню (мал. 3.4, 3.5).



Мал. 3.4



Мал. 3.5

Для додавання файлів до щойно створеного або відкритого архіву слід вибрати в програмі-архіваторі вказівку *Додати*, а потім знайти та позначити потрібні файли та ще раз скористатися вказівкою *Додати*, тобто підтвердити виконання раніше вибраної вказівки.

Для витягнення з архіву одного або кількох файлів спочатку слід за допомогою програми-архіватора знайти архів, виділити ті файли, які слід розпакувати, та вибрати вказівку розпаковування, вказавши відповідне місце на диску для розміщення файлів, які розкриватимуться. При створенні архіву та занесенні до нього файлів і при його розкриванні залишаються незмінними файли-джерела: при архівуванні — файли, що стискаються; при розкриванні архіву — стиснені файли.

У тих випадках, коли архівація виконується для передавання пакета документів іншим, слід передбачити наявність програмного засобу, необхідного для розпаковування вихідних даних з архіву. У випадку, якщо користувач або користувачка не має необхідної програми-архіватора, на основі звичайного архіву можна створити архів, що саморозпаковується, шляхом приєднання невеликого програмного модуля. Файл архіву отримує розширення *exe*, що свідчить про те, що він є виконуваним файлом. Користувач зможе запустити цей файл як звичайну програму, після чого розпакування архіву відбудеться на його комп'ютері автоматично.

ЦІКАВО

Архів з розширенням *exe* ще називають **SFX архівом** — від англ. *Self eXtracting Archive* — архів, що саморозпаковується.

Крім того, кожна з програм-архіваторів має багато додаткових функцій.



ДІМО

Вправа 2. Архівування папки за допомогою контекстного меню.

Завдання. Заархівуйте папку *Методи стиснення* за допомогою контекстного меню у форматі *7z*.

1. Завантажте папку *Методи стиснення* (<https://cutt.ly/MES4uxR>) і перемістіть її до папки *Комп'ютер та програми* вашої структури папок.
2. У вашій структурі папок клацніть правою кнопкою миші на папці *Методи стиснення* та в контекстному меню оберіть вказівку *7-zip/Додати до «Методи стиснення.7z»*.
3. Переконайтеся, що в папці *Комп'ютер та програми* з'явився відповідний архів.
4. За допомогою вказівки контекстного меню *Властивості* визначте обсяг файлу архіву й обсяг вихідної папки, що підлягала архівуванню.

Вправа 3. Розпаковування з архіву лише окремого файлу за допомогою архіватора 7-Zip.

Завдання. Розпакуйте з архіву *Історія розвитку обчислювальної техніки.7z* (<https://cutt.ly/IES4nZB>) файл *Посилання на ресурси в інтернеті.txt* у власну папку.

1. Завантажте архів *Історія розвитку обчислювальної техніки.7z* у папку *Комп'ютер та програми*.
2. Двічі клацніть на файлі завантаженого архіву. Відкриється вікно програми-архіватора 7-Zip із вмістом зазначеного файлу архіву.
3. У вікні програми-архіватора 7-Zip двічі клацніть на папці *Історія розвитку обчислювальної техніки*, що зберігається в запакованому вигляді.

4. Виділіть файл *Посилання на ресурси в інтернеті.txt* і натисніть кнопку

 **Видобути**

на панелі інструментів вікна 7-Zip.

5. У діалоговому вікні *Видобути:*, користуючись інструментом  для огляду папок, вкажіть папку *Комп'ютер та програми* на вашому комп'ютері, в яку необхідно розпакувати файл, та натисніть кнопку *OK*.
6. Закрийте всі відкриті вікна.

Вправа 4. Створення архіву, що саморозпаковується, за допомогою архіватора 7-Zip.

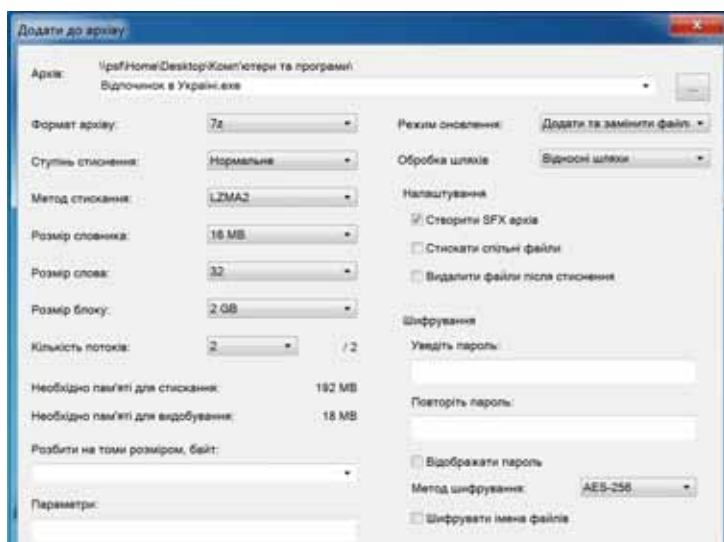
Завдання. Заархівуйте папку *Відпочинок в Україні* таким чином, щоб отримати архів, що саморозпаковується.

1. Завантажте папку *Відпочинок в Україні* (<https://cutt.ly/oES4Vml>). Перемістіть її в папку *Комп'ютер та програми* на вашому комп'ютері.
2. Завантажте програму-архіватор 7-Zip.
3. У списку дисків і папок відкрийте папку *Комп'ютер та програми*.

4. Виділіть папку *Відпочинок в Україні* та натисніть кнопку

 **Додати**

5. У діалоговому вікні *Додати до архіву* оберіть інструмент  і вкажіть папку *Комп'ютер та програми*, у яку буде збережено архів. Ім'я архіву за замовчуванням збігатиметься з іменем папки, яка додається до архіву.
6. В області *Налаштування* увімкніть прапорець *Створити SFX архів* і натисніть кнопку *OK* (мал. 3.6).



Мал. 3.6

7. Відкрийте вікно папки, в яку було створено архів, та визначте, яке ім'я і розширення має створений файл.
8. Закрийте всі відкриті вікна.

3.4. У чому полягають основні методи стиснення даних?

Характерною особливістю більшості форматів даних, з якими традиційно працює користувач, є певна надмірність. Ступінь надмірності залежить від типу даних.

Наприклад, у відеофайлів вона, як правило, в кілька разів менша, ніж у графічних, а в графічних — у кілька разів менша, ніж у текстових. Крім того, ступінь надмірності даних залежить від прийнятої системи кодування.

Кодування текстових даних, поданих українською мовою (з використанням українського алфавіту), дає надмірність у середньому на 20–30 % більше, ніж кодування даних, поданих англійською мовою.

Існує велика кількість форматів стиснення даних, але всі вони працюють за одним принципом — зменшення надмірності даних у файлі за допомогою різних математичних методів. У результаті, залежно від досконалості алгоритму й типу вихідного файлу, його розмір може суттєво зменшитися: типовим значенням для документів є 40–50 % і більше. Наднизькі показники у відео- й аудіофайлів. І це цілком логічно, адже дані, що зберігаються в них, піддавалися компресії і практично не містять надмірності.

В основі роботи програм-архіваторів лежить процедура пошуку та перекодування однакових фрагментів вмісту файлу.

Наприклад, розглянемо одну з можливих таких процедур. Нехай є файл, який містить багато однотипних слів: комп'ютер, комп'ютера, комп'ютерна, комп'ютеризація тощо. Якщо слово «комп'ютер» замінити простою комбінацією символів «чц», то розглянута система слів перетвориться на систему: «чц», «чца», «ччна», «чцизація» тощо. При такій заміні довжина тексту зменшується.

Кожна з програм-архіваторів працює за різними алгоритмами архівування різних типів даних. У реальних програмах-архіваторах процедура пошуку та перекодування даних відбувається значно складніше.



ПРАЦЮЄМО В ПАРАХ

1. Обговоріть у парах, які функції програм-архіваторів є основними, а які — додатковими і чому.
2. Як розкрити раніше створений архів? Сформулюйте узагальнене правило розпаковування архіву. Обговоріть його в парах.
3. На сайті Міжнародного конкурсу з інформатики та комп'ютерної вправності «Бобер» (<http://bober.net.ua/>) перейдіть у сховище

архівів завдань

АРХІВ ЗАВДАНЬ

та завантажте архів із демон-

страційною програмою конкурсу 2020 року. Встановіть, чи можна запуснути програму випробувань безпосередньо з вікна програми 7-zip, не розпаковуючи архіву.



4. Доберіть кожен/кожна один зі способів додавання файлу до архіву: інструмент  вікна програми 7-Zip чи перетягування файлу у вікно архіву, якщо файл міститься на *Робочому столі* або в папці на диску. Продемонструйте одне одному та аргументуйте свій вибір способу.



ПРАЦЮЄМО САМОСТІЙНО

5. Складіть схему-пояснення причин використання операції форматування до флеш-пам'яті. Доберіть засіб для подання результату самостійно.



6. При архівуванні даних використовують різні методи стиснення. Одним з них є алгоритм Хафмана. В основі алгоритму Хафмана лежить ідея кодування бітовими групами. Розглянемо простий приклад, що ілюструє роботу алгоритму Хафмана. Нехай задано текст, у якому літера 'A' входить 10 разів, літера 'B' — 8 разів, 'C' — 6 разів, 'D' — 5 разів, 'E' і 'F' — по 4 рази. Тоді один з можливих варіантів кодування за алгоритмом Хафмана наведено у таблиці:

Символ	Частота входження	Бітовий код	Символ	Частота входження	Бітовий код
A	10	00	D	5	101
B	8	01	E	4	110
C	6	100	F	4	111

Використовуючи алгоритм Хафмана, визначте, яким буде коефіцієнт стиснення даних (без урахування словника) для слів «молоко», «кукуріку».



7. Доповніть презентацію *Інструменти програми-архіватора* (<https://cutt.ly/sES8cTB>) відомостями про призначення інструментів програми 7-Zip.



8. Попрацюйте в групі над виконанням мініпроєкту «Рятівники комп'ютера». Розподіліть обов'язки так, щоб були обрані кожна із службових програм. Наведіть приклад програми, яка належить до обраної групи програм. Створіть листівку з описом програми та презентуйте її у класі. Передбачте, що в листівці має бути наведено аргумент, що дану програму можна використати для «порятунку» комп'ютера, та приклад, коли це може бути потрібно.

ЯКІ ІСНУЮТЬ ЗАГРОЗИ БЕЗПЕЦІ ПІД ЧАС РОБОТИ ЗА КОМП'ЮТЕРОМ?

4

ПРИГАДАЙТЕ:

- правила безпечного користування інтернетом під час пошуку інформаційних матеріалів;
- що таке обліковий запис користувача;
- як критично оцінювати матеріали, знайдені в інтернеті;
- як дотримуватись авторського права.

ВИ ДІЗНАЄТЕСЯ:

- які існують загрози безпеці та пошкодження даних у комп'ютерних системах;
- які розрізняють етичні та правові основи захисту даних;
- які є шляхи захисту даних;
- як програми належать до шкідливих;
- які можуть бути ознаки зараження комп'ютера шкідливими програмами.



ВИВЧАЄМО

4.1. Які існують загрози безпеці та пошкодження даних у комп'ютерних системах?

Попри корисність використання інтернету для навчання, розвитку та розваг, існують також ризики й загрози безпеці. Під загрозою розуміють показники якості системи й навколишнього середовища, які у відповідних умовах можуть викликати появу небезпечної події.



Загроза — це потенційна можливість певним чином порушити інформаційну безпеку.

Під **інформаційною безпекою** розуміють захищеність даних та інфраструктури, що її підтримує, від будь-яких випадкових або зловмисних дій, результатом яких може стати нанесення шкоди безпосередньо даним, їх власникам або інфраструктурі, що її підтримує.

Існують три різновиди загроз:

1. **Загроза порушення конфіденційності** — полягає в тому, що дані стають відомими тому, хто не має у своєму розпорядженні повноваження доступу до них. Вона має місце щоразу, коли отримано доступ до деяких секретних даних, що зберігаються в комп'ютерній системі чи передаються від однієї системи до іншої. Іноді, через загрозу порушення конфіденційності, використовують термін «витік даних».

2. **Загроза порушення цілісності** — включає в себе будь-яку умисну зміну даних, що зберігаються в комп'ютерній системі чи передаються

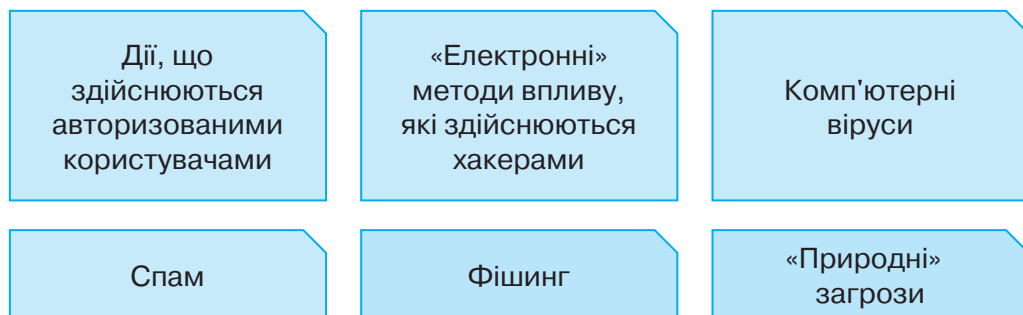
з однієї системи в іншу. Коли зломисники навмисно змінюють дані, тобто порушується їхня цілісність.

Цілісність також може бути порушена, якщо до несанкціонованої зміни даних призводить випадкова помилка програмного або апаратного забезпечення.

Санкціонованими змінами є ті, які зроблено уповноваженими особами з обґрунтованою метою (наприклад, санкціонованою зміною є періодична запланована корекція деякої бази даних).

3. Загроза відмови служб (загроза доступності), що виникає щоразу, коли в результаті навмисних дій, що виконуються іншим користувачем або зломисником, блокується доступ до деякого ресурсу комп'ютерної системи. Реально блокування може бути постійним, якщо запитуваний ресурс ніколи не буде отримано, або воно може викликати тільки затримку запитуваного ресурсу, досить довгу для того, щоб він став непотрібним. У цих випадках говорять, що ресурс вичерпано.

Дії, які можуть завдати шкоди інформаційній безпеці організації, можна розділити на кілька категорій (мал. 4.1):



Мал. 4.1

1. Дії, що здійснюються авторизованими користувачами. У цю категорію потрапляють: цілеспрямована крадіжка або знищення даних на робочій станції або сервері; пошкодження даних користувачами в результаті необережних дій.

2. «Електронні» методи впливу, які здійснюються хакерами.

Хакер — кваліфікований ІТ-фахівець, який розуміється на роботі комп'ютерних систем і здійснює втручання до комп'ютера, щоб без відома власника дізнатися деякі особисті дані, а потім ними

ЦІКАВО

Напад на комп'ютерну систему з наміром зробити комп'ютерні ресурси недоступними користувачам, для яких комп'ютерна система була призначена, називають **DoS-атакою**, або **DDoS-атакою** (англ. *DoS attack*, *DDoS attack*, *(Distributed) Denial-of-service attack* — атака на відмову в обслуговуванні, розподілена атака на відмову в обслуговуванні).

скористатися, або пошкодити дані, що зберігаються в комп'ютері. Їхніми мотивами можуть бути помста, самовираження (дехто робить це задля розваги, інші — щоб показати свою кваліфікацію), винагорода. Останнім часом поняття «хакер» використовують для визначення мережових зломщиків, творців комп'ютерних вірусів й інших кіберзлочинців.

У багатьох країнах злом комп'ютерних систем, розкрадання інформаційних даних, створення і поширення комп'ютерних вірусів і шкідливого програмного забезпечення переслідується законом.

3. Комп'ютерні віруси.

Комп'ютерний вірус (англ. *computer virus*) — комп'ютерна програма, яка має здатність до прихованого самопоширення та спрямовані на знищення, пошкодження, викрадання даних, зниження або унеможливлення подальшої працездатності операційної системи комп'ютера.

Вони являють собою реальну небезпеку, широко використовують комп'ютерні мережі, інтернет і електронну пошту.

4. Спам.

Спам — небажані рекламні електронні листи, повідомлення на форумах, телефонні дзвінки чи текстові повідомлення, що надходять без згоди користувача.

На даний момент поняття спаму включає в себе всі різновиди масової реклами.

5. Фішинг.

Фішинг — один з найпопулярніших і прибуткових (для тих, хто його реалізує) видів атак. Сценарій атак фішингу: зловмисник створює сайт, який у точності копіює дизайн і можливості сайту будь-якого банку, інтернет-магазину або платіжної системи. Далі він замовляє спам-розсилку листів, у яких переконує своїх жертв зайти за посиланням на сайт і заповнити будь-яку форму, з внесенням персональних даних. Як правило, причиною запиту даних зазначають повідомлення про збої в інформаційній системі й загрозу блокування профілю користувача у випадку, якщо не будуть надані дані. Мета — збір конфіденційної інформації — паролі, коди тощо.

Скрипти, що автоматично запускаються при відкритті вебсторінки, можуть виконувати шкідливі дії на вашому комп'ютері, включаючи зміну системного реєстру, крадіжку особистих даних і встановлення шкідливого програмного забезпечення. Використовуючи мережеві технології, зловмисники реалізують атаки на віддалені комп'ютери й сервери компаній. Результатом таких атак може бути виведення ресурсу з ладу, діставання повного доступу до ресурсу, а отже, до даних, що зберігаються на ньому. У зв'язку з появою кредитних карт, електронних грошей і можливістю їх використання через інтернет інтернет-шахрайство стало одним з найбільш поширених злочинів.

6. «Природні» загрози.

На інформаційну безпеку організації можуть впливати різноманітні зовнішні чинники: причиною втрати даних може стати неправильне зберігання, крадіжка комп'ютерів і носіїв, форс-мажорні обставини тощо.



ДІЄМО

Вправа 1. Класифікація джерел загроз.

Завдання. Ознайомтеся з іншою класифікацією джерел загроз інформаційній безпеці (мал. 4.2) та поставте у відповідність загрози, вказані на малюнку 4.1 категоріям запропонованої класифікації.

1. Завантажте файл Класифікація джерел загроз (<https://cutt.ly/IEFUe30>).
2. Ознайомтеся з особливостями запропонованої класифікації.
3. Поставте у відповідність загрози, вказані на малюнку 4.1, категоріям класифікації на малюнку 4.2. Результати подайте у вигляді презентації.
4. Результати роботи збережіть у папці *Безпека* власної структури папок.

Чинники загроз інформаційній безпеці

Людський

Технічний

Стихийний

■ Мал. 4.2 ■

4.2. Які розрізняють етичні та правові основи захисту даних?

Розвиток якісно нового суспільства вимагає створення нової системи міжлюдських взаємин. Інформація, головний ресурс і цінність сучасного суспільства, одночасно є засобом і об'єктом скоєння неетичних, протиправних дій і кримінальних злочинів.

Морально-етичні основи захисту даних передбачають норми поведінки, які традиційно склались або складаються з поширенням комп'ютерів і мереж: соціальна й персональна відповідальність, рівноправність партнерів по комунікації, точне й сумлінне виконання обов'язків тощо. Ці норми здебільшого не затверджені в законодавчому порядку, але їх невиконання часто призводить до падіння авторитету та престижу людини, групи осіб, організації або країни. Морально-етичні норми бувають як неписаними, так й оформленими в деякий статут.

Поряд із загальнолюдськими етичними нормами, існують такі базові права, як:

- *загальнодоступність* — гарантує право на комунікацію і передбачає доступність державних інформаційних ресурсів;
- *таємниця приватного життя* — дотримання конфіденційності довірених даних;

- *недоторканність приватної власності* — основа майнового порядку, дотримання права власності на дані та норм авторського права.

Правові засоби захисту — чинні закони, укази та інші нормативні акти, які регламентують правила користування даними і відповідальність за їх порушення, захищають авторські права програмістів та регулюють інші питання використання ІТ.

У прийнятих в Україні законодавчих нормах зазначено, зокрема, що захисту підлягає:

- відкрита інформація, яка належить до державних інформаційних ресурсів, а також відкрита інформація про діяльність суб'єктів владних повноважень, військових формувань, яка оприлюднюється в інтернеті, інших глобальних інформаційних мережах і системах або передається телекомунікаційними мережами;
- конфіденційна інформація, яка перебуває у володінні розпорядників інформації, визначених частиною першою статті 13 Закону України «Про доступ до публічної інформації»;
- службова інформація;
- інформація, яка становить державну або іншу передбачену законом таємницю;
- інформація, вимогу щодо захисту якої встановлено законом.

Відкрита інформація під час опрацювання в системі має зберігати цілісність, що забезпечується шляхом захисту від несанкціонованих дій, які можуть призвести до її випадкової або умисної модифікації чи знищення.

Усім користувачам має бути забезпечено доступ до ознайомлення з відкритою інформацією. Модифікувати або знищувати відкриту інформацію можуть лише ідентифіковані й автентифіковані користувачі, яким надано відповідні повноваження.

Правовий захист інформації (даних) передбачає:

- наявність прав на інформацію — сертифікація, ліцензування, патентознавство;
- реалізацію прав — захист інтелектуальної власності, захист авторських прав;
- контроль за процедурами реалізації прав — система адміністративного, програмного, фізико-технічного захисту інформації.



ДІЄМО

Вправа 2. Види захисту інформації.

Завдання. Побудуйте хмаринку слів, які описують об'єкти, пов'язані із захистом інформації в нормативних документах, прийнятих в Україні. Скористайтеся статтею Вікіпедії «Список нормативних документів щодо інформаційної безпеки в Україні».

1. У статті Вікіпедії виділіть ключові слова, що описують об'єкти, пов'язані із захистом інформації. Наприклад: інформація, державна таємниця, персональні дані, технічний захист.
2. У браузері відкрийте один із сервісів створення хмаринки слів онлайн. Наприклад, <https://wordart.com/>.
3. Скористайтесь інструкцією по роботі із сервісом <https://cutt.ly/qEFDJHe>.
4. Роздрукуйте отримане зображення та продемонструйте його у класі.

4.3. Які є шляхи захисту даних?

Розрізняють три шляхи захисту даних (мал. 4.3).



■ Мал. 4.3 ■

Захист доступу до комп'ютера. Для запобігання несанкціонованого доступу до даних, що зберігаються на комп'ютері, використовуються облікові записи. Комп'ютер дозволяє доступ до своїх ресурсів тільки тим користувачам, які зареєстровані й ввели правильний пароль. Кожному конкретному користувачеві може бути наданий доступ тільки до певних інформаційних ресурсів. При цьому може проводитися реєстрація всіх спроб несанкціонованого доступу.

Захист даних на дисках. Кожний диск, папка і файл локального комп'ютера, а також комп'ютера, підключеного до локальної мережі, можуть бути захищені від несанкціонованого доступу. Для них можуть бути встановлені певні права доступу (повний, тільки читання, доступ за паролем), причому права можуть бути різними для різних користувачів.

Захист даних в інтернеті. Якщо комп'ютер підключений до інтернету, то будь-який користувач, та-

ЦІКАВО

Нині для захисту від несанкціонованого доступу до важливих даних дедалі частіше використовуються біометричні системи авторизації та ідентифікації користувачів. Використовувані в цих системах характеристики є невід'ємними якостями особистості людини й тому не можуть бути втраченими й підробленими. До біометричних систем захисту даних відносяться системи розпізнавання мови, системи ідентифікації за відбитками пальців, а також системи ідентифікації за райдужною оболонкою ока.

кож підключений до інтернету, може отримати доступ до інформаційних ресурсів цього комп'ютера.

Механізми проникнення з інтернету на локальний комп'ютер і в локальну мережу можуть бути різними:

- вебсторінки, що завантажуються в браузер, можуть містити активні елементи, здатні виконувати деструктивні дії на локальному комп'ютері;
- деякі вебсервери розміщують на локальному комп'ютері текстові файли *cookie*, використовуючи які, можна отримати конфіденційну інформацію про користувача локального комп'ютера;
- електронні листи або дописи в соціальних мережах можуть містити шкідливі посилання;
- за допомогою спеціальних програм можна отримати доступ до дисків і файлів локального комп'ютера тощо.

ЦІКАВО

Для захисту даних під час роботи в інтернеті доцільно використовувати підключення, що захищене шифруванням. Наприклад, за замовчуванням Google шифрує з'єднання з Gmail, а також при виборі інших сервісів Google, наприклад, Google Диск, активується протокол шифрування SSL, який використовується до завершення сеансу роботи.

Щоб визначити, що сайти захищені, слід звернути увагу на їхню URL-адресу – вона починається з *https://*. Це, на відміну від протоколу *http*, протокол зашифрованого підключення, що забезпечує більш ефективний захист даних. У деяких браузерах поруч із назвою протоколу відображається значок замка . Це означає, що з'єднання захищене і більш безпечне.

ЦІКАВО

HTTPS (від англ. *HyperText Transfer Protocol Secure*) — розширення протоколу *http* для підтримки шифрування з метою підвищення безпеки.

4.4. Які програми належать до шкідливих?



Шкідлива програма — комп'ютерна програма або переносний код, призначений для реалізації загроз даним, що зберігаються в інформаційній системі, або для прихованого нецільового використання ресурсів системи, або іншої дії, що перешкоджає нормальному функціонуванню інформаційної системи.

До шкідливого програмного забезпечення належать мережеві хробаки, класичні файлові віруси, троянські програми, хакерські утиліти й інші програми, що завдають явну шкоду комп'ютеру, на якому вони запускаються на виконання, або іншим комп'ютерам у мережі.

Комп'ютерні віруси — це спеціальні програми в машинних кодах або фрагменти програм, здатні без відома та згоди користувача розмножуватися та розповсюджуватися на інші програми шляхом копіювання свого коду у файли, що зберігаються в системі. Вони, як і біологічні віруси, досить малі порівняно з іншими програмами.

Мережеві віруси для свого розповсюдження використовують можливості комп'ютерних мереж. Мережеві віруси — переважно це макровіруси — розповсюджуються через електронну пошту.

Є також шкідливі програми, що розповсюджуються здебільшого через локальну мережу та інтернет.

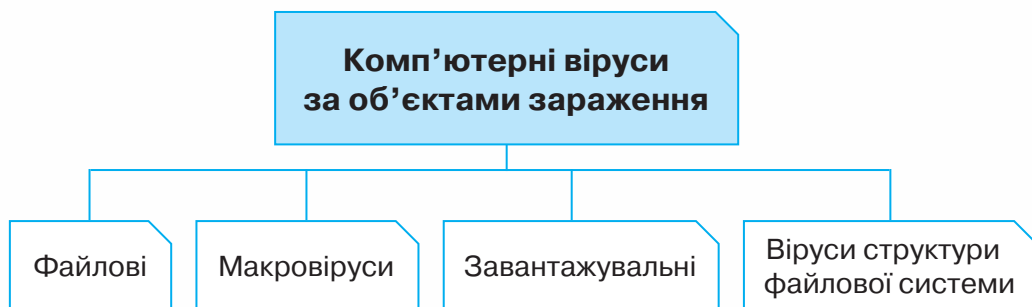
Хробаки — це один з різновидів шкідливих вірусів, що розмножуються та псують дані, збережені на комп'ютері. Часто хробаки розповсюджуються через файли, вкладені в електронні листи, або через деякі вебсторінки, проте можуть також завантажуватися під час спільного користування файлами або програмами миттєвого обміну повідомленнями.

Розповсюджуючись мережею, ці шкідливі програми створюють додаткове навантаження на канали зв'язку, вони можуть потай відправляти дані з вашого комп'ютера. Хробаки розповсюджуються швидше за віруси безпосередньо з одного комп'ютера на інший. Наприклад, хробак електронної пошти може сам відправляти себе на всі адреси електронної пошти в адресній книзі користувача. Інтернет-хробаки шукають підключені до інтернету комп'ютери, які не містять останніх оновлень безпеки.

Програми на кшталт «троянський кінь» розповсюджуються шляхом обману, вони запускаються під виглядом іншої програми для отримання важливих даних. Наприклад, така програма має вигляд діалогового вікна реєстрації в системі. Коли користувач вводить у ньому своє реєстраційне ім'я та пароль, вони запам'ятовуються або передаються власнику програми «троянський кінь». Потім такі дані можуть бути використані для доступу до системи. Відмінність цих програм від вірусів і хробаків полягає в тому, що самостійно відтворюватися вони не можуть.

Є різні підходи до класифікації комп'ютерних вірусів.

Розглянемо класифікацію за об'єктами зараження (мал. 4.4):



Мал. 4.4

- *файлові* — уражають програми (основні й допоміжні, тобто ті, що завантажуються лише під час виконання основних).

Файлові віруси записуються в початкові, об'єктні та завантажувальні модулі. Вони заражають файли, додаючи свій код у початок, середину або кінець файлу. Такі файли залишаються працездатними, але, крім свого коду, виконують код вірусу. Деякі віруси замість початкового змісту файлу записують свій код.

Файлові віруси можуть і не змінювати вміст файлів, які вони заражають.

Деякі віруси собі надають ім'я зараженого файлу, а файлові дають нове ім'я. Існують також віруси, які привертають увагу користувача своїм іменем: *Start.com*, *Install.exe* тощо;

- *макровіруси* — це файлові віруси, які використовують файли документів текстових процесорів та електронних таблиць, зокрема *Microsoft Office*. Для свого розмноження такі віруси використовують можливості макромов (вбудованих макросів) та за їхньою допомогою переміщуються від одного зараженого файлу до інших. Більшу частину таких вірусів написано для текстового процесора *Microsoft Word*;
- *завантажувальні* — уражають завантажувальні сектори дисків.

Завантажувальні віруси записуються в завантажувальний сектор диска або область MBR.

MBR (англ. *Master Boot Record* — головний завантажувальний запис) — таблиця в першому секторі завантажувального диска, що зберігає дані про фізичну та логічну його організацію.

Якщо цей запис зіпсовано, то неможливо завантажити операційну систему. Якщо заражено системний диск, з якого відбувається завантаження системи, тоді код вірусу отримує управління. Під час зараження вірус зчитує потрібні дані із завантажувача та зберігає їх у своєму коді.

Найчастіше завантажувальні віруси є *резидентними*. Це означає, що під час завантаження системи вірус потрапляє в оперативну пам'ять і перебуває там постійно;

- *віруси структури файлової системи* — здатні вносити зміни в службові структури файлової системи таким чином, що вірус включається у файли, призначені для виконання, явно не вкорінюючи в них свій код. Усі записи в папках щодо цих програм модифікуються таким чином, що першим записом програми стає запис, який містить код вірусу. Під час запуску будь-якої такої програми замість неї працює вірус.

За способом зараження середовища мешкання вірусів: резидентні й нерезидентні.

За зовнішнім виглядом:

- *звичайні* — код вірусу видно на диску;
- *невидимі (Stealth-віруси)* — використовують особливі методи маскування, при перегляді код вірусу не видно. *Stealth-віруси* мають

резидентний модуль, який постійно перебуває в оперативній пам'яті комп'ютера;

- *поліморфні* — код вірусу змінний.

За результатами діяльності:

- *безпечні* — лише поширюють себе та інколи призводять до виведення повідомлень, перезавантаження та інших несподіваних, незручних ефектів, але без втрати наявних даних чи шкоди апаратному забезпеченню;
- *небезпечні* — призводять до втрати даних, руйнування як програмного, так і (можливо) апаратного забезпечення інформаційної системи.

4.5. Які можуть бути ознаки зараження комп'ютера шкідливими програмами?

При запуску заражених програм (вірусів) можуть виконуватись різні небажані дії: псування файлів і папок, спотворення результатів обчислень, засмічення або вилучення даних із пам'яті, створення перешкод у роботі комп'ютера тощо. Об'єктами зараження є виконувані файли програм або файли операційної системи, а також документи, що містять програмний код. Коли вірус потрапляє до комп'ютера, його власник може про це навіть не здогадуватися. Часто вірус може себе деякий час не виявляти, і лише після настання певної дати чи події — активізуватися та завдати шкоди комп'ютерній системі. Зараження комп'ютера вірусом відбувається лише тоді, коли на ньому виконується заражений програмний файл або відкривається заражений документ.

Виконання заражених програм автоматично викликає активізацію вірусу, що призводить до зараження нових програм. Процес поширюється дуже швидко.

Ознаками зараження комп'ютера можуть бути такі прояви: на екран виводяться непередбачені повідомлення, зображення або відтворюються непередбачені звукові сигнали; несподівано відкривається й закривається лоток CD/DVD-ROM-пристрою; довільно, без вашої участі, на комп'ютері запускаються які-небудь програми; на екран виводяться попередження про спробу деякої з програм вашого комп'ютера вийти в інтернет, хоча ви ніяк не ініціювали таку її поведінку — з великим ступенем імовірності можна припустити, що комп'ютер уражено вірусом.

Є характерні ознаки зараження вірусом через пошту: друзі або знайомі сповіщають вас про повідомлення від вас, які ви не відправляли; у вашій поштовій скриньці міститься велика кількість повідомлень без зворотної адреси та заголовка.

Про зараження комп'ютера вірусом можна дізнатися і за непрямими ознаками:

- часті зависання і збої в роботі комп'ютера;
- повільна робота комп'ютера при запуску програм;

- неможливість завантаження операційної системи;
- зникнення файлів і папок або спотворення їхнього вмісту;
- часте звернення до жорсткого диска (часто блимає лампочка на системному блоці);
- браузер зависає або поводить ся несподіваним чином (наприклад, вікно програми неможливо закрити).

У 90 % випадків наявність непрямих ознак зараження вірусами викликана збоєм у роботі апаратного або програмного забезпечення.



ПРАЦЮЄМО В ПАРАХ

1. Обговоріть у парах, яка загроза безпеці та пошкодження даних може бути актуальною для комп'ютера у школі, удома. Наведіть свої аргументи.
2. Пограйте у гру «Хто більше?». По черзі називайте спосіб захисту комп'ютера від небезпеки. Виграє той чи та, хто назве найбільше способів.
3. Хто створює віруси та якою є мета їх створення? Обговоріть у парах.
4. Уявіть, що вам запропонували створити свій кодекс законів про захист користувача інтернету від ймовірних небезпек. Обговоріть, з яких розділів ви б склали такий документ, які статті він би мав містити.



ПРАЦЮЄМО САМОСТІЙНО

5. Перевірте себе у вправах:
 - а) якими бувають комп'ютерні віруси (<https://learningapps.org/view3912015>);
 - б) якими можуть бути ознаки зараження комп'ютера вірусами (<https://learningapps.org/view3912102>).
6. Засобами редактора презентацій створіть ілюстрований інтерактивний словник термінів, які виникають при захисті від можливих загроз користувача інтернету. Передбачте, що користувач може обирати термін у списку та переходити на сторінку із тлумаченням терміну. На такий сторінці може бути ілюстрація, що відображає зміст терміну, корисне посилання на ресурс в інтернеті, де можна детальніше ознайомитись із поняттям.
7. Попрацюйте в мініпроекті «ТОП-10 комп'ютерних вірусів в історії». Знайдіть відомості в інтернеті та складіть список з 10 найнебезпечніших комп'ютерних вірусів. Визначте, з якою метою вони були створені. Подайте знайдені відомості в текстовому документі в зручному вигляді.

ЯКІ ЗАСОБИ ДОПОМОЖУТЬ ЗАХИСТИТИ ДАНІ ТА БЕЗПЕЧНО ПРАЦЮВАТИ НА КОМП'ЮТЕРІ?

ПРИГАДАЙТЕ:

- які існують загрози безпеці в комп'ютерних системах;
- які є шляхи захисту даних.

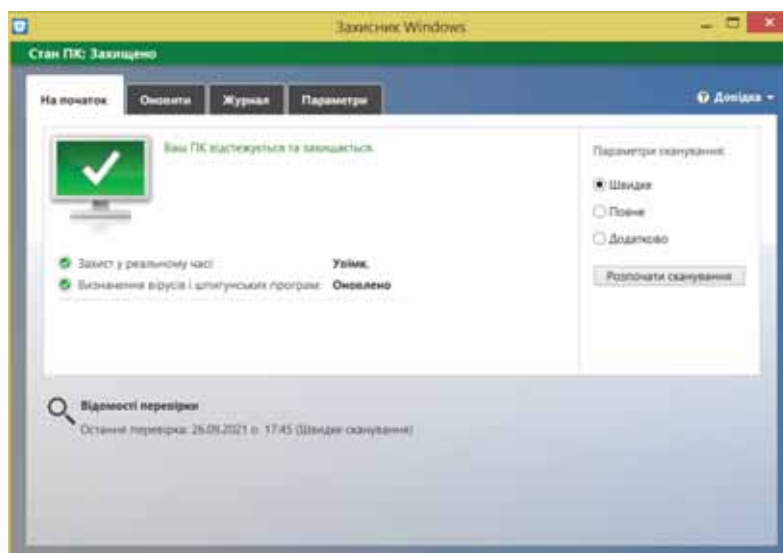


ВИВЧАЄМО

5.1. Якими програмами можна скористатись для підтримки безпечної роботи комп'ютера?

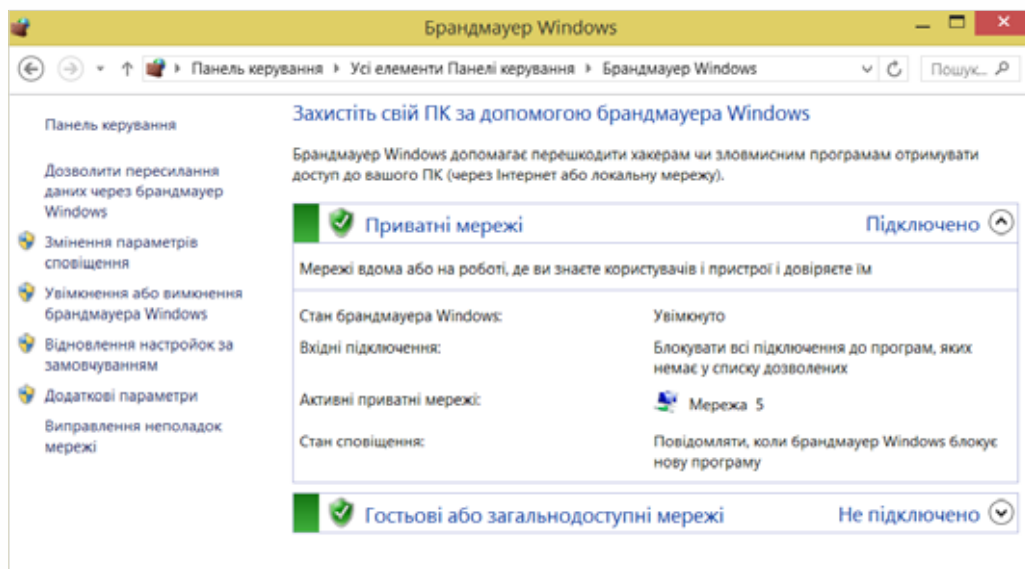
В операційній системі *Windows 10* передбачено ряд засобів для підтримки безпечної роботи комп'ютера.

Вбудована у *Windows 10* програма *Захисник Windows* забезпечує виявлення, попередження інсталяції та вилучення зловмисного програмного забезпечення в реальному часі за допомогою захисту в хмарі. Він призначений для використання вдома, на підприємствах малого бізнесу, а також для використання корпоративними клієнтами (мал. 5.1).



Мал. 5.1

Загальний захист мережевого під'єднання здійснюють за допомогою брандмауерів (міжмережевих екранів) — окремих пристроїв або спеціальних програм, які для захисту створюють бар'єр між комп'ютером і мережею (мал. 5.2).



Мал. 5.2

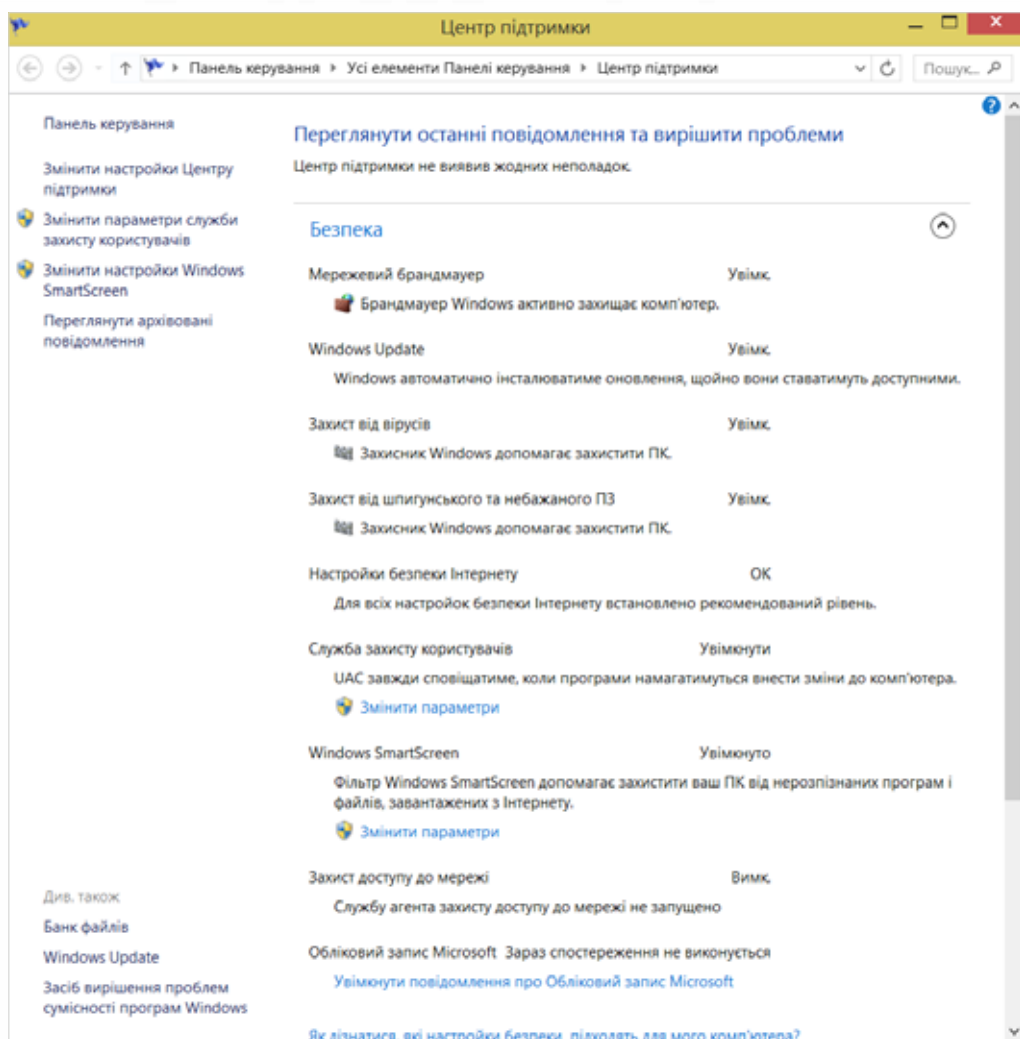
За допомогою програм-брандмауерів відслідковують усі під'єднання і за необхідності дозволяється чи блокується доступ до комп'ютера. Брандмауер може блокувати доступ до комп'ютера вірусів та хробаків, однак він не в змозі знайти їх і знищити.

ЦІКАВО

Термін **брандмауер** походить від нім. *brand* — пожежа, *mauer* — стіна, його англійський еквівалент — *firewall*, асоціюється з вогнестійкою капітальною стіною, що перешкоджає поширенню пожежі. Термін виник приблизно в 1995 р.

Служба захисту користувачів запитує дозвіл на інсталяцію на ПК програмного забезпечення або відкривання певних типів програм, які можуть зашкодити комп'ютеру чи зробити його вразливим до інших загроз системі безпеки.

Перегляд і внесення змін у налаштуваннях безпеки здійснюють у вікні *Центру підтримки* (мал. 5.3). Важливо переконавшись, що брандмауер увімкнено, програму захисту від зловмисного програмного забезпечення оновлено, а на ПК налаштовано автоматичну інсталяцію оновлень.



■ Мал. 5.3 ■

5.2. Для чого використовують антивірусні програми?

Для боротьби з комп'ютерними вірусами можна придбати спеціальні програми, які називають антивірусними.

Антивірусні програми призначені для виявлення і лікування програм і файлів, що заражені комп'ютерним вірусом, а також для запобігання зараження файлу вірусом. Вони виконуються на комп'ютері й перевіряють на наявність вірусів усі файли, які завантажуються з інтернету, отримуються електронною поштою, записуються на жорсткий диск або запускаються на виконання з компакт-диска чи флеш-накопичувача.

Антивірусне програмне забезпечення складається з комп'ютерних програм, які намагаються знайти, запобігти розмноженню і видалити комп'ютерні віруси й інші шкідливі програми.

Антивірусне програмне забезпечення зазвичай використовує два різних методи для виконання своїх задач:

- перегляд (сканування) файлів для пошуку відомих вірусів, що відповідають визначенню в переліку вірусів;
- знаходження підозрілої поведінки будь-якої з програм, що схожа на поведінку зараженої програми.

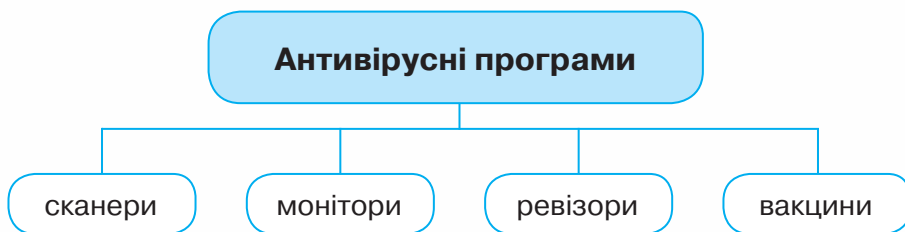
Незалежно від того, яку з антивірусних програм встановлено, важливо постійно її оновлювати. Зазвичай за певну річну оплату можна завантажувати оновлення такої антивірусної програми із сайту виробника.

Щоб захистити свій комп'ютер, бажано постійно використовувати антивірусні програми та брандмауери, періодично перевіряти свій комп'ютер для виявлення шкідливих або інших небажаних програм та одразу вилучати їх.

ЦІКАВО

Найбільш нищівний з усіх хробаків на ім'я **I LOVE YOU** завдав збитків на 7 млрд доларів.

Класифікувати антивірусні продукти можна відразу за кількома ознаками, такими, як: використовувані технології антивірусного захисту, функціонал продуктів, цільові платформи. Залежно від принципу дії антивірусні програми поділяють на сканери, монітори, ревізори та вакцини (мал. 5.4).



■ Мал. 5.4 ■

Сканери (наприклад, *Eset Smart Security (NOD32)*, *MS Anti Virus*) — визначають наявність комп'ютерного вірусу за базою даних, у якій зберігаються відповідні відомості про кожний вірус. Їх ефективність визначається актуальністю вірусної бази та наявністю спеціальної підпрограми, яка може проводити аналіз на наявність вірусу за спеціальним алгоритмом. Потребують постійного оновлення своїх баз даних.

Ревізори (наприклад, програма *Adinf*) — контролюють уразливі для вірусної атаки компоненти комп'ютера: запам'ятовують стан файлової системи, аналізують стан файлів та службові відомості про диск і порівнюють їх з вихідним станом, який було збережено в одному з файлів даних ревізора.

Програми монітори, або фільтри, — розташовуються в оперативній пам'яті й реагують на звернення до операційної системи, які роблять

віруси для свого розмноження або виконання руйнівних дій, пропонуючи користувачеві відповідні запити на дозвіл чи заборону операції. Повідомляють користувача про всі дії на його комп'ютері. Якщо троянська програма або вірус захочуть проникнути в ПК або викрасти пароль і відправити його зловмисникові, фільтр миттєво спрацює і запитає: «Дозволити чи заборонити виконання операції?». Багато сучасних антивірусів мають вбудований брандмауер, що дозволяє контролювати мережевий трафік і стежити за змінами в системі.

Вакцини, або імунізатори (наприклад, *Anti Trojan Elite*, *Trojan Remover*), — модифікують програми й диски так, що це не відображається на роботі програм, але вірус, від якого виконується вакцинація, вважає такі програми та диски вже зараженими.

Лікарі або фаги — відривають від інфікованих програм тіло вірусу й відтворюють початковий стан програм.

Програми, що використовуються на практиці, як правило, належать до антивірусних комплексів, що поєднують функції сканера, ревізора й монітора.

КОРИСНЕ ПОСИЛАННЯ

Найкращі антивірусні програми для Windows — стаття від <https://uk.wizcase.com/> (<https://cutt.ly/KEGN37J>)

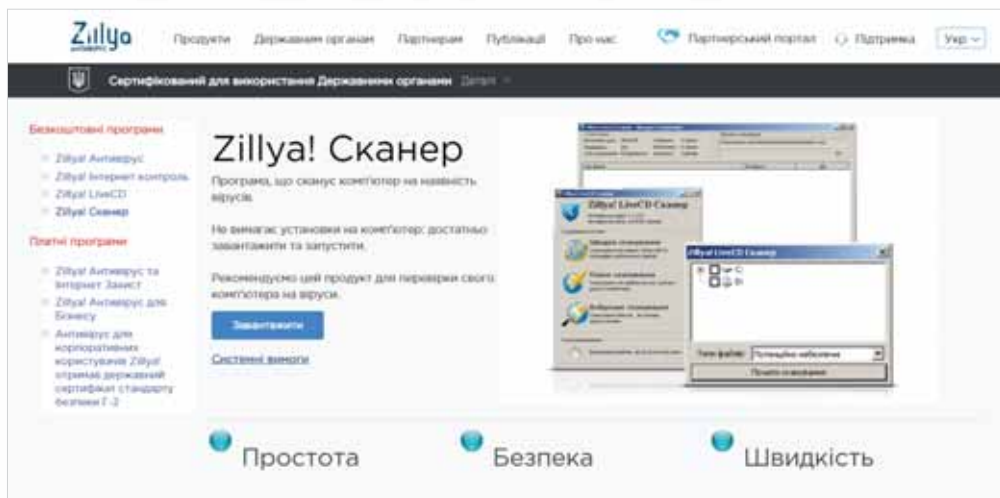
Сьогодні здебільшого використовують антивірусні програми: *Захисник Windows Microsoft Security Essentials*, *avast!* (avast.ua), *AVG* (www.avg.com), *Norton by Symantec* (us.norton.com), *ESET NOD32* (eset.ua), *McAfee* (www.mcafee.com), *Malwarebytes* (www.malwarebytes.com), *Bitdefender* (www.bitdefender.com).

КОРИСНЕ ПОСИЛАННЯ

Звіт про частку ринку антивірусного програмного забезпечення Windows: <https://www.metadefender.com/stats/anti-malware-market-share-report#!/>

Піктограма антивірусної програми, що встановлена на комп'ютері, відображається в правій частині панелі завдань. Двічі клацнувши на ній, можна відкрити вікно антивірусної програми, наприклад, український антивірус *Zillya!*, в якому можна обрати різні режими роботи: оновлення бази антивірусів, налаштування програми, сканування всього комп'ютера або обраних компонентів.

Антивірусні програми передбачають різні режими сканування, наприклад, швидке сканування, повне сканування, вибіркове сканування (мал. 5.5). Повне сканування всього комп'ютера, як правило, відбувається протягом тривалого часу.



Мал. 5.5



Вправа 1. Перевірка файлів за допомогою онлайнного антивіруса.

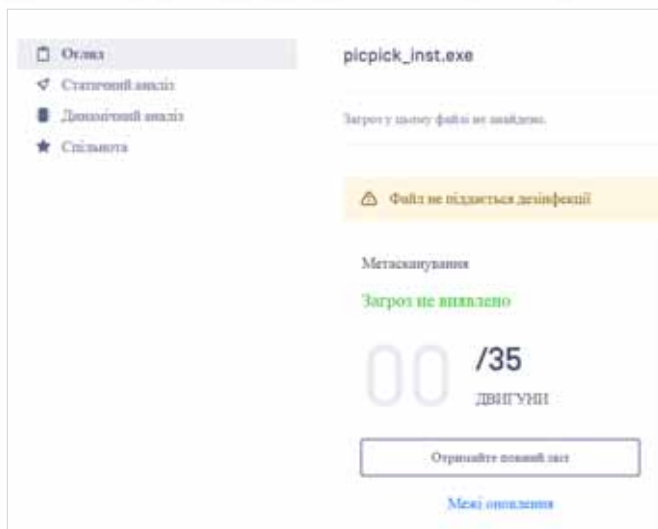
Завдання. Перевірити інсталяційну програму графічного редактора *PicPick* на наявність вірусів за допомогою онлайнного антивіруса *Metadefender Cloud*.

1. Відкрийте вікно браузера та введіть адресу www.metadefender.com.
2. У полі **Довіра №** натисніть кнопку *Select a File* (Обрати файл) та оберіть файл *picpick_inst.exe* з папки *Інформаційна безпека*.



Мал. 5.6

3. Проаналізуйте результат сканування (мал. 5.7). Про знайдені загрози сигналізуватиме червоний колір кільця та перше число в крузі, відмінне від нуля.



Мал. 5.7

5.3. Які засоби браузера призначено для уникнення загроз безпеці?

Сучасні браузери мають вбудовані засоби для уникнення загроз інформаційній безпеці. Адже безпосередньо через браузер ми переглядаємо вміст вебсайтів, заходимо на сайти інтернет-банків, оплачуємо товари та послуги, користуємося онлайновими сервісами або обмінюємося конфіденційною інформацією. Саме на браузер лягає первинна відповідальність за безпеку в інтернеті.

Обираючи браузер для роботи, зазвичай оцінюють його інтерфейс, швидкість і зручність роботи, наявність додаткових функцій. Проте одним з важливих критеріїв вибору браузера є його засоби для гарантування безпеки. Різні браузери пропонують різні рішення проблем безпеки.

Засоби *Microsoft Edge* забезпечують захист від стеження, фільтрацію підозрілого активного вмісту, протидію XSS-атакам, перегляд у приватному режимі *InPrivate*. Розробниками реалізовано також виділення домену другого рівня в адресному рядку браузера жирним кольором, що дає змогу легко визначити, чи перебуває користувач саме на тому сайті, на який хотів зайти, або ж на шахрайському, адреса якого дуже схожа на адресу цього сайту. Однією з функціональних особливостей безпеки є фільтр *SmartScreen*, який має змогу фільтрувати не тільки шкідливі сайти по URL, а й шкідливі файли за допомогою технології *Application Reputation*.

Екран *SmartScreen* у програмі *Microsoft Edge* допомагає захистити користувача від фішингової атаки та зловмисних програм, попереджаючи, що вебсайт або розташування для завантаження повідомлено як небезпечне.

ЦІКАВО

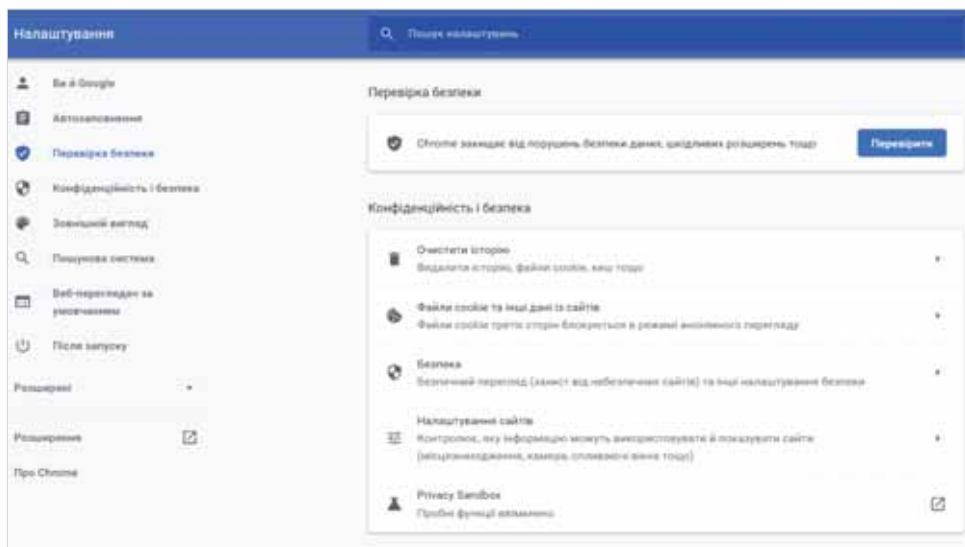
XSS-атака (англ. *Cross-Site Scripting* — міжсайтовий скриптинг) — тип атаки на вебсистеми, що полягає у вбудовуванні в сторінку, яка переглядається, шкідливого коду (що буде виконано на комп'ютері користувача під час перегляду цієї сторінки) та взаємодії цього коду з вебсервером зломисника.

До засобів захисту браузера *Mozilla Firefox* відносять захист від XSS-атак, інтеграцію з батьківським контролем *Windows*, функції «Приватний перегляд», інтеграцію з антивірусними продуктами, фільтр шкідливих сайтів, захист від стеження за діями користувача в інтернеті за допомогою спеціальних скриптів, що розміщуються на інтернет-сторінках, підтримки HTTPS-з'єднань.

У *Google Chrome* передбачено захист від шахрайських і фішингових сайтів, зосереджений у технології «Безпечний перегляд», реалізовано технологію забезпечення безперервності HTTPS-з'єднання та захисту його від компрометації, захист від XSS-атак та інші корисні функції.

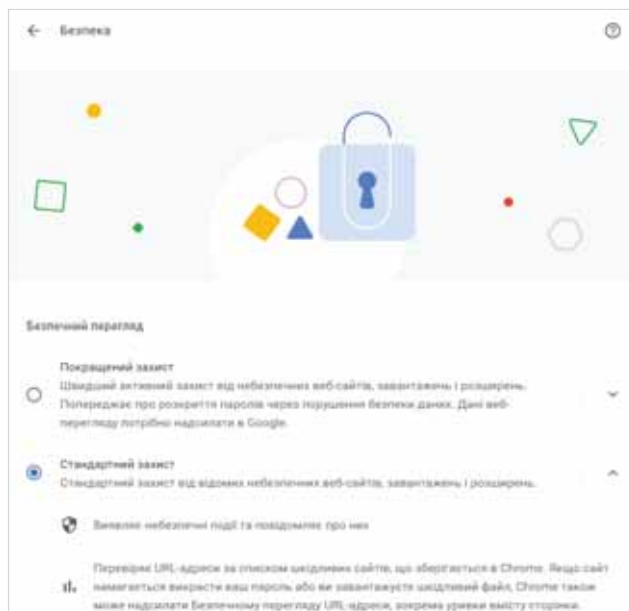
Крім вбудованих засобів, різні браузери передбачають розширені налаштування безпеки. Деякі вебсайти можуть намагатися використувати ваші особисті відомості для викрадення цільової реклами, шахрайства та викрадення особистих даних. Усі сучасні браузери мають параметри конфіденційності, які можна ввімкнути, щоб контролювати, яку інформацію можуть використовувати сайти.

Для налаштування засобів безпеки в браузері *Google Chrome* необхідно відкрити меню браузера за допомогою інструмента  в правій частині вікна, та обрати вказівку *Налаштування* (мал. 5.8).



Мал. 5.8

Додаткові параметри захисту можна встановити, якщо в розділі *Конфіденційність і безпека* обрати кожний розділ і встановити параметри безпеки. Наприклад, як на малюнку 5.9.



■ Мал. 5.9 ■



ДІЄМО

Вправа 2. Налаштування параметрів безпеки браузера.

Завдання. Перевірте налаштування та налаштуйте браузер за інструкцією (<https://cutt.ly/3EG3gzj>). Створіть фотозвіт про виконання завдання.

5.4. Як виконувати резервне копіювання та відновлення даних?

На жаль, дані неможливо абсолютно надійно зберігати в комп'ютері. Відмова апаратної частини (жорсткий диск), вірусна атака чи неакуратність самого користувача (випадкове видалення даних) можуть призвести до втрати важливих даних. Щоб запобігти цьому, необхідно періодично робити резервне копіювання даних.

Резервна копія — копія окремих файлів, групи файлів або всього диску, що збережена на іншому носії, для наступного відновлення у випадку руйнувань, псування або втрати даних на головному носії.

Розрізняють такі типи резервних копій.

- **Резервна копія операційної системи.** Дуже корисна річ, якою часто нехтують навіть досвідчені користувачі. Потрібно встановити операційну систему, драйвери й необхідні програми. Потім зробити резервну копію налагодженої операційної системи та, в разі потреби (вірусна

атака, або просто захаращення системи), відновити резервну копію. Це займає набагато менше часу, ніж нова установка й налаштування операційної системи. Як правило, для резервного копіювання операційної системи використовують спеціальні програми або засоби.

- **Резервна копія логічного диску (розділу)** — як правило, це дублікат диска, на якому встановлена операційна система. Дані упаковуються в єдиний файл-контейнер. Дозволяє швидко відновити систему та всі встановлені програми, які попали в утворений образ.

- **Резервна копія окремих файлів і папок** — найпоширеніший спосіб резервного копіювання.

Резервне копіювання та відновлення даних є різновидом операцій збереження даних зі своїми особливостями:

- При збереженні даних ми найчастіше маємо справу з одним або декількома файлами. При резервному копіюванні зазвичай об'єктом копіювання є набір великої кількості файлів, папки або диски.

- Резервне копіювання завжди проводять на інші носії — на DVD-диск, флеш-накопичувач, мережні ресурси, відмінні від тих, з яких копіюють. Не можна розміщувати резервну копію на диску з файлами, резервну копію яких створює користувач, або диску, де встановлена операційна система.

- При резервному копіюванні значення збережених даних для користувача комп'ютера більше, ніж у випадку окремого файлу чи файлів. Тому для таких випадків використовують спеціальні засоби — **програми для резервного копіювання даних**.

Під час резервного копіювання файлів ОС *Windows* створює на відповідному носії папку з іменем *Set [дата] [час]*. У разі першого створення резервної копії *Windows* виконує повне резервне копіювання всіх файлів визначеного типу. Після цього здійснюється оновлення резервної копії файлами, що створені чи змінені з моменту запису останньої резервної копії. Можна періодично створювати нову, повну резервну копію, щоб починати роботу «з чистого аркуша».

Резервне копіювання та відновлення файлів і дисків виконується за допомогою вказівок вікна *Відновлення системи*, яке відкривають з вікна  Панель керування > Усі елементи Панелі керування > Система у вкладці *Захист системи* (мал. 5.10).

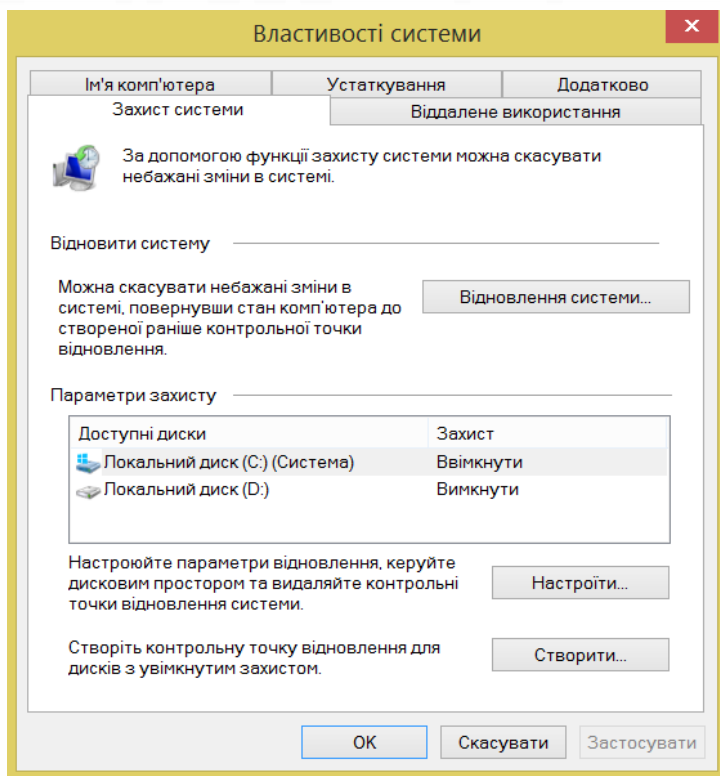
ЦІКАВО

Backup – від англ. резервна копія.

КОРИСНЕ ПОСИЛАННЯ

Перелік програм, з допомогою яких можна створювати образ диска
<https://uk.soringpcrepair.com/program-to-create-a-disk-image/>.

Доцільно дотримуватися таких рекомендацій щодо резервного копіювання даних.



■ Мал. 5.10 ■

- *Робити резервні копії періодично.* Залежно від типу даних кожного дня, тижня, місяця тощо або хоча б після істотного оновлення даних. Інакше в разі втрати даних можна відновити лише застарілу версію резервної копії, в якій не вистачатиме нещодавно змінених даних.
- *Тиражувати копії.* Зробивши резервну копію важливих даних, розмножити цю копію на фізично різних носіях — флеш-накопичувачах, зовнішньому жорсткому диску, CD/DVD-диску, хмарному сховищу тощо. Що більше копій, то більша ймовірність не втратити потрібні дані.
- *Захистити резервну копію від сторонніх.* Краще за все — поєднати кілька способів захисту. Наприклад, захист даних паролем і шифрування.
- *Зберігати резервні копії у різних місцях,* навіть якщо їх зроблено на різних носіях. Інакше у разі крадіжки, пожежі чи стихійного лиха усі копії можна втратити.

КОРИСНЕ ПОСИЛАННЯ

Як створити резервну копію даних пристрою *Android* і відновити їх:
<https://support.google.com/android/answer/2819582?hl=uk>

5.5. Як безпечно видаляти дані?

Файли, які видаляються в *Кошик*, за необхідності можуть бути відновлені. Навіть після очищення *Кошика* видалені файли все одно залишаються на диску, поки на їх місце не буде записано інші файли. При видаленні особистих та конфіденційних даних слід передбачити захист від небажаного відновлення видалених файлів іншими особами.

Для безпечного видалення даних без можливості відновлення використовують спеціальні програми. Наприклад, програми *Eraser* і *File Shredder* призначені для безповоротного видалення даних з дисків шляхом неодноразового перезапису.

Популярною програмою для безпечного видалення та очистки комп'ютера від неживих і тимчасових файлів є *CCleaner* (www.piriform.com/ccleaner). Програма має базову безкоштовну версію та платні версії з додатковими можливостями.

За допомогою *CCleaner* можна швидко й безпечно видаляти:

- тимчасові файли, історію відвідувань браузерів;
- вміст *Кошика*, буфера обміну, тимчасових файлів, файлів журналів, переліку нещодавно відкритих документів в операційній системі *Windows*;
- тимчасові та інші непотрібні файли багатьох програм.

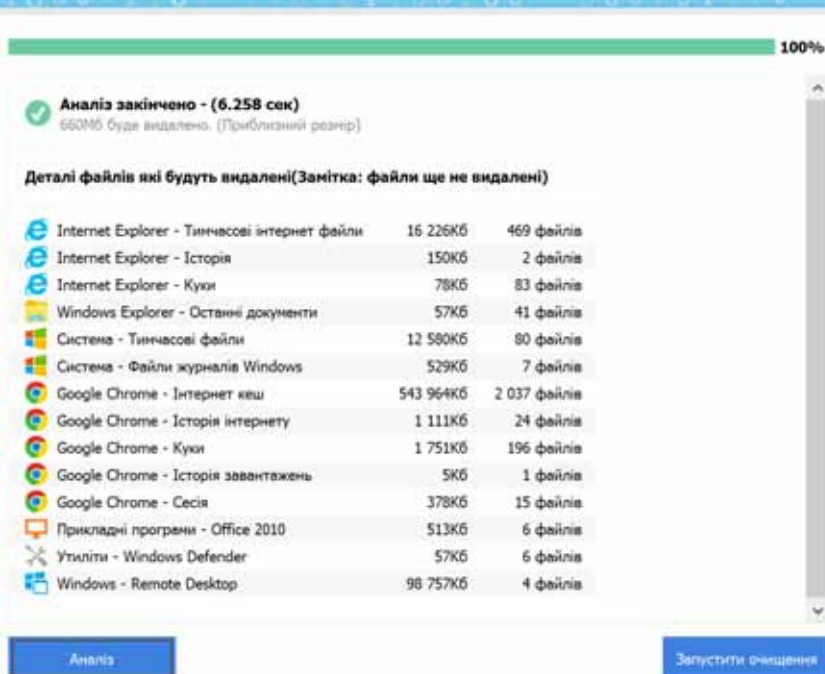
Видалення непотрібних файлів за допомогою *CCleaner* звільняє місце на жорстких дисках та забезпечує швидшу роботу операційної системи. Після вибору типів даних, що підлягають очищенню (мал. 5.11), слід натиснути кнопку *Аналіз*. У вікні буде відображено деталі про файли, які будуть видалені, зокрема вказано їхній обсяг (мал. 5.12). Для їх видалення натискають кнопку *Запустити очищення*.



Мал. 5.11

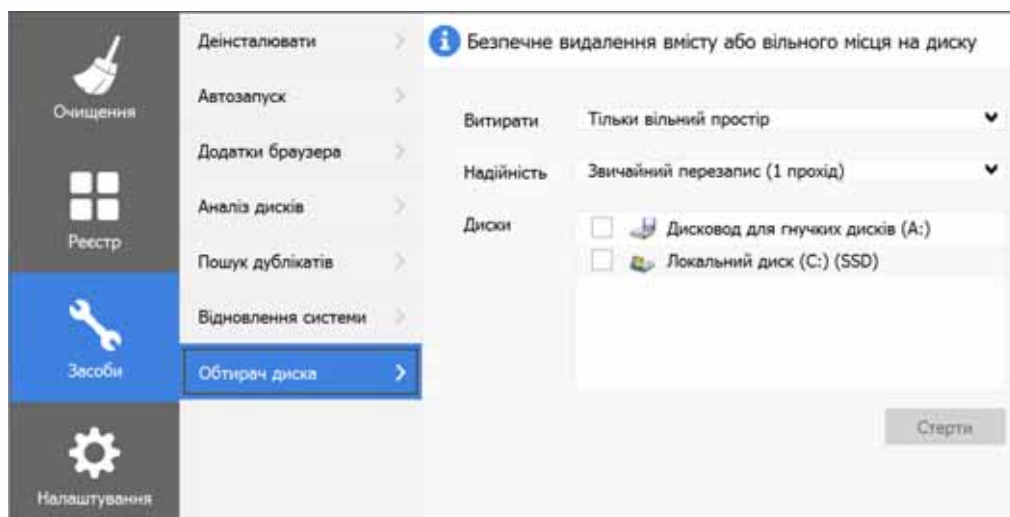
ЦІКАВО

Видалені фацли можна відновити. Для цього є спеціальне програмне забезпечення. Наприклад, відновити видалений чи втрачений файл можна за допомогою програм для відновлення даних: *Recuva*, *Hetman Partition Recovery*, *EaseUS Data Recovery Wizard*, *UndeletePlus*, *R-Studio*, *Ontrack EasyRecovery*.



Мал. 5.12

Безпечне видалення файлів з диска шляхом неодноразового перезапису забезпечують додаткові засоби, зокрема *Обтирач диска* (мал. 5.13).



Мал. 5.13

ЦІКАВО

Близько 2,5 млн користувачів з усього світу використовують *CCleaner* для безпечного видалення даних. Щомісяця за допомогою цієї програми видаляється 35 000 000 Гбайт даних.

КОРИСНЕ ПОСИЛАННЯ

Безкоштовні програми для видалення даних: <http://freesoftware.in.ua/148-krashha-bezkoshtovna-programa-dlya-znishhennya-danix.html>.

5.6. Яких правил доцільно дотримуватися для безпечної роботи в інтернеті?

Ніякі найнадійніші засоби не зможуть гарантувати стовідсотковий захист від комп'ютерних вірусів і троянських програм, але дотримуючись певних правил, можна істотно знизити вірогідність вірусної атаки та ступінь можливих збитків. Одним з основних методів боротьби з вірусами є, як і в медицині, своєчасна профілактика. Комп'ютерна профілактика складається з невеликої кількості правил, дотримання яких значно знижує вірогідність зараження вірусом і втрати даних.

З метою профілактики зараження комп'ютерними вірусами користувачеві слід дотримуватися певних рекомендацій. А саме:

- необхідно використовувати надійні джерела програмного забезпечення для свого комп'ютера, купувати його лише в офіційних продавців;
- перевіряти за допомогою антивірусних програм файли, які надходять ззовні (з дисків, інтернету) до комп'ютера;
- обмежити доступ сторонніх осіб до комп'ютера;
- регулярно створювати резервні копії важливих даних;
- періодично перевіряти комп'ютер на наявність вірусів з використанням нових версій антивірусних програм;
- не відкривати вкладення електронної пошти, які були надіслані від невідомих вам адресатів;
- не завантажувати з інтернету файли з розширенням *exe* з неперевірених сайтів.

Часто шкідливі програми можуть потрапляти на комп'ютер через спам. Найнадійніший спосіб боротьби зі спамом — не дозволити спамерам довідатися вашу електронну адресу. Для цього доцільно дотримуватися таких правил.

- Не варто без необхідності публікувати адресу електронної пошти на вебсайтах чи в групах новин.
- Не потрібно реєструватися на підозрілих сайтах. Якщо деякий корисний сайт вимагає реєстрації, можна вказати спеціально для цього створену адресу.
- Ніколи не відповідати на спам і не переходити за посиланнями, які містяться в ньому. Цим ви тільки підтвердите, що користуєтеся своєю електронною адресою і будете одержувати ще більше спаму.
- Вибираючи собі ім'я електронної пошти варто, за можливості, обирати довге й незручне для вгадування ім'я.

Важливо розуміти також загрози, які спрямовані не тільки на інформаційну систему, а й на самих користувачів, зокрема кібербулінг.

Кібербулінг — це цькування із застосуванням цифрових технологій. Кібербулінг може відбуватися в соціальних мережах, платформах обміну повідомленнями (месенджерах), ігрових платформах і мобільних телефонах, зокрема через:

- поширення брехні чи розміщення фотографій, які компрометують когось у соціальних мережах;
- повідомлення чи погрози, які ображають когось або можуть завдати комусь шкоди.



ДІЕМО

Вправа 3. Кампанія проти кібербулінгу.

Завдання. Ознайомтесь із порадами щодо особистого захисту в інтернеті: <https://cutt.ly/6T1YsYZ>.

Складіть відповідну презентацію-порадник для своїх однокласників.



ДОСЛІДЖУЕМО

Вправа 4. Антивірус для ОС Linux.

Завдання. Проведіть дослідження чи існують антивірусні програми для комп'ютерів, які працюють під ОС *Linux*. Якщо ви отримаєте позитивну відповідь, то наведіть приклади таких програм. Якщо ні, то поясніть причину. Результати свого дослідження подайте у самостійно обраному вигляді.



ПРАЦЮЕМО В ПАРАХ

1. Які антивіруси вважаються в нашій країні найбільш використовуваними? Які вони мають особливості? Обговоріть у парах.
2. Визначте недоліки безкоштовних антивірусних програм. Потрібні відомості знайдіть в інтернеті. Обговоріть у парах.
3. Як діяти за наявності ознак зараження комп'ютера? Скористайтесь матеріалами з файлу *Дії при виявленні вірусу* (<https://cutt.ly/vEG46HK>). З якими рекомендаціями ви погоджуєтесь? Запропонуйте власні рекомендації.
4. Назвіть профілактичні заходи для уникнення загроз безпеці комп'ютера. Скористайтесь матеріалами з файлу *Профілактика заражень* (<https://cutt.ly/TEG7eQL>). З якими рекомендаціями ви погоджуєтесь? Запропонуйте власні рекомендації.

-  5. Попрацюйте в парах над мініпроектом «Наш порадник із захисту від небезпек в інтернеті». Оберіть одну із загроз, які можуть виникати для інформаційної системи чи користувачів та складіть презентацію, як розпізнати обрану загрозу та як варто вчинити, щоб її уникнути. Використайте презентацію *Приклад для проєкту* (<https://cutt.ly/1EG7ncG>). Презентуйте результат у класі.



ПРАЦЮЄМО САМОСТІЙНО

6. Завантажте антивірусну програму, що встановлена на вашому комп'ютері.
- За допомогою довідки визначте, які операції може виконувати ця програма. До якого типу антивірусних програм вона належить?
 - Виконайте антивірусну перевірку папки *Мої документи* вашого комп'ютера.
 - Виконайте перевірку жорсткого диска та зовнішньої пам'яті вашого комп'ютера на наявність вірусів.
 - Перегляньте звіти за результатами перевірки. Зробіть висновки.

-  7. Знайдіть в інтернеті відомості про історію створення вірусів й антивірусів. Створіть лінію часу, на якій відобразить основні етапи цієї історії. Чи гарантує безпеку роботи з комп'ютером індустрія антивірусних програм?

-  8. Перегляньте сайти розробників найбільш використовуваних антивірусних програм. Складіть порівняльну таблицю конкретних характеристик та властивостей різних антивірусних програм.

Для порівняння використайте критерії: наявність платних та безкоштовних версій та обмежень, пов'язаних з функціональністю; швидкість штатного сканування; швидкість визначення потенційних загроз та можливість їх видалення чи ізоляції в карантині за допомогою вбудованих алгоритмів; частота оновлення антивірусних баз; самозахист та надійність; наявність додаткових можливостей. Які з них належать до вільного програмного забезпечення? Які схожі послуги пропонують розробники антивірусних програм? Намалюйте схему таких послуг.

-  9. Проведіть опитування своїх знайомих, чи були в них випадки зараження їхнього комп'ютера вірусом. Яку антивірусну програму вони використовують? Як часто поновлюють базу антивірусної програми? Які способи оновлення обирають?

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Я знаю типи програмного забезпечення.
- ☐ Я можу пояснити відмінності між ними та навести приклади.
- ☐ Я вмію класифікувати операційні системи.
- ☐ Я можу пояснити призначення драйверів.
- ☐ Я можу пояснити поняття сумісності програмного забезпечення.
- ☐ Я можу назвати різні типи ліцензій на програмне забезпечення, пояснити відмінності між ними.
- ☐ Я можу пояснити принципи стиснення даних.
- ☐ Я можу назвати типи файлів архівів.
- ☐ Я можу назвати основні типи шкідливих програм та пояснити принцип їх дії.
- ☐ Я можу пояснити принципи й знаю методи захисту від інформаційних загроз.
- ☐ Я вмію стискати файли та розпаковувати архіви.
- ☐ Я вмію застосовувати антивірусну програму для захисту комп'ютерного пристрою від інформаційних загроз.
- ☐ Я вмію налаштувати параметри антивірусної програми.
- ☐ Я вмію добирати програмне забезпечення під конкретні задачі.

ОПРАЦЮВАННЯ ТАБЛИЧНИХ ДАНИХ

6

ЯК ЗДІЙСНЮЮТЬ ОБЧИСЛЕННЯ В ТАБЛИЧНОМУ ПРОЦЕСОРІ?

ПРИГАДАЙТЕ:

- з якими основними об'єктами можна працювати в середовищі табличного процесора;
- як застосовувати засіб автозаповнення для введення даних;
- як копіювати та переміщувати дані з клітинок і діапазонів клітинок;
- як змінювати форматування таблиці;
- як виконувати обчислення із числовими даними таблиці;
- як здійснювати обчислення за допомогою вбудованих функцій;
- що відбувається при копіюванні формул.

ВИ ДІЗНАЄТЕСЯ:

- чим відрізняються абсолютні та відносні посилання на клітинки;
- як додати до формули вбудовану функцію;
- які математичні та статистичні функції найчастіше використовуються в середовищі табличного процесора.



ВИВЧАЄМО

6.1. Чим відрізняються абсолютні та відносні посилання на клітинки?

Ви вже знаєте, що обчислення в електронних таблицях виконуються за допомогою формул. У формулах використовують посилання на клітинки або діапазони клітинок, вказуючи їхні адреси або ім'я клітинки чи діапазону, що надані користувачем.

Під час роботи з електронними таблицями часто однотипні розрахунки необхідно виконувати в кількох суміжних клітинках. У середовищі табличного процесора, зокрема *Microsoft Excel* або *LibreOffice Calc*, можна в цьому разі не вводити формули кілька разів із клавіатури, а скопіювати формулу з однієї клітинки в іншу за допомогою буфера обміну або скористатися засобом автозаповнення. При цьому також відбувається копіювання формули.

На відміну від копіювання текстових значень, при копіюванні формули, що містить посилання на клітинки, можна отримати формули, які

не дублюють вихідну. Це залежить від вигляду посилань на клітинки, які можуть бути **відносними**, **абсолютними** або **комбінованими (мішаними)**. Вигляд посилань має значення лише при копіюванні формули, при обчисленні значення за формулою в одній клітинці вигляд посилань не є суттєвим.

Ви вже створювали формули, які містять відносні посилання — такі посилання використовуються за замовчуванням. При копіюванні формули, що містить **відносні посилання**, такі посилання будуть коригуватися залежно від напрямку копіювання. Відносно посилання на клітинку складається лише з назви стовпця і номера рядка. Наприклад, якщо клітинка *F4* містить формулу $=D4 * E4$, то при копіюванні її до клітинки *F5* вона набуде вигляду $=D5 * E5$. Така формула з відносними посиланнями при копіюванні в нову клітинку інтерпретується так: знайти добуток значень двох сусідніх зліва клітинок, що в тому само рядку, що й клітинка, яка містить формулу.

Абсолютне посилання у формулі свідчить, що при копіюванні формули необхідно залишати посилання саме на ту клітинку, адреса якої зазначена; така адреса залишиться незмінною. Щоб відрізнити абсолютне посилання від відносного, у його записі перед назвою стовпця і номером рядка ставиться символ **\$**, наприклад, $\$C\1 (мал. 6.1).

	A	B	C	D	E
1	Відсоток знижки		5%		
2					
3	Найменування товару	Вартість	До сплати		
4	Навушники A4tech MK-610	109,3	103,84	$=B4*(1-\$C\$1)$	
5	Миша TRUST GXT 25 Gaming Mouse	299	284,05	$=B5*(1-\$C\$1)$	
6	Серветки ColorWay CW-1071	49	46,55	$=B6*(1-\$C\$1)$	
7					

Мал. 6.1

У **комбінованому**, або **мішаному**, **посиланні** назва стовпця є абсолютною, а номер рядка — відносним, або навпаки. Наприклад, $B\$4$, $\$B4$ (мал. 6.2). Коригується при копіюванні лише відносна частина адреси.

	A	B	C	D	E	F	G	H	I	J
1	Стрибки в довжину									
2	Спортсмен	Рекорд	Спроба			Відхилення				
3			1	2	3	1	2	3		
4	Андрієвський	466	460	468	463	-6	2	-3		
5	Васильковець	485	482	485	480	-3	0	-5		
6										
7										

Мал. 6.2

Алгоритм виконання обчислень у середовищі табличного процесора з однотипними розрахунками можна подати словесно так.

1. Внести в клітинки таблиці дані, які будуть необхідні для виконання обчислень.

2. Визначити, які клітинки в таблиці мають містити однотипні розрахунки, які посилання слід використати для обчислень — абсолютні, відносні чи комбіновані.

3. Створити формулу з посиланнями на клітинки, використовуючи виділення в таблиці потрібних клітинок або діапазонів.

4. Вказати за потреби ті посилання, які є абсолютними чи комбінованими, додавши знак \$ перед назвою стовпця та (або) номером рядка.

5. Скопіювати створену формулу до діапазону клітинок, які мають містити однотипні розрахунки.



ДІМО

Вправа 1. Розрахунки.

Завдання. Обчисліть вартість проданого товару за допомогою введення формул для даних таблиці, що містяться у файлі *Розрахунки*.

1. Створіть у власній структурі папок папку *Табличний процесор*.
2. Завантажте файл *Розрахунки*, що зберігається в папці *Електронні таблиці* (<https://cutt.ly/ZEKmrOr>). Збережіть його в папці *Табличний процесор*.
3. Виділіть клітинку *D3*, введіть до неї формулу $=C3*\$C\12 (мал. 6.3) та натисніть клавішу *Enter*. Ціна в гривнях розраховується як добуток ціни в доларах на курс долара, однак адресу клітинки *C12*, у якій введено курс долара, не потрібно змінювати при копіюванні формули в інші клітинки. Тому формула містить відносну адресу *C3*, яка коригуватиметься при копіюванні, та абсолютну адресу *\\$C\\$12*.

D3	fx =C3*\$C\$12						
	A	B	C	D	E	F	G
1	Розрахунки вартості проданого товару						
2	№	Товар	Ціна в дол.	Ціна в грн	Ціна з ПДВ	Кількість	Вартість
3	1	Фоторамка	\$6,00	141,24 грн		3	
4	2	Портативна колонка	\$10,60			2	
5	3	Електронна книга	\$65,00			2	
6	4	Карта пам'яті 16 ГБ	\$5,56			5	
7	5	Карта пам'яті 32 ГБ	\$12,24			4	
8							
9							
10		Дата випускування рахунку	12.01.16				
11		Дата оплати рахунку	17.01.16				
12		Курс дол.	23,54				
13							
14		Вартість покупки					

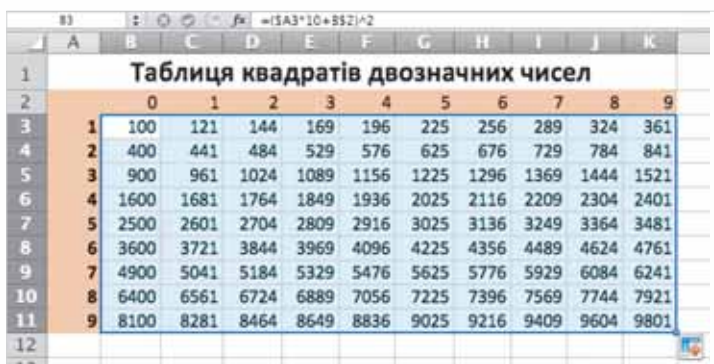
Мал. 6.3

4. Виділіть клітинку *D3* та виконайте протягування за маркер автозаповнення вниз до клітинки *D7*.
5. Для обчислення ціни товару з ПДВ (податок на додану вартість становить 20 % від вартості товару) необхідно до ціни товару додати ще 20 % його вартості, тому до клітинки *E3* введіть формулу $=D3+D3*0,2$ та натисніть клавішу *Enter*. Оскільки необхідно, щоб при подальшому копіюванні цієї формули до клітинок стовпця *E* адреса клітинки *D3* змінювалась відповідно на *D4*, потім на *D5*, далі на *D6* і *D7*, у формулі слід лишити відносну адресу клітинки *D3*.
6. За допомогою автозаповнення скопіюйте формулу з клітинки *E3* до клітинок діапазону *E4:E7*.
7. Виділіть клітинку *G3* та введіть формулу $=E3*F3$ для обчислення вартості вказаної кількості товару. Скопіюйте цю формулу до діапазону клітинок *G4:G7*.
8. Виділіть клітинку *G14*. За допомогою автосуми знайдіть суму значень діапазону клітинок *G3:G7*.
9. Збережіть результати у файлі з тим само ім'ям у папці *Електронні таблиці* своєї структури папок.

Вправа 2. Таблиця квадратів.

Завдання. Створіть таблицю квадратів двозначних чисел з використанням формул, що містять комбіновані посилання.

1. Відкрийте файл *Таблиця квадратів*, що зберігається в папці *Електронні таблиці* (<https://cutt.ly/bEKmYU3>).
2. Виділіть клітинку *B3* та введіть до неї формулу $=(\$A\$3*10+B\$2)^2$ (мал. 6.4).
3. Виділіть клітинку з формулою та виконайте протягування за маркер автозаповнення вправо до клітинки *K3*. Не знімаючи виділення з діапазону, виконайте протягування за маркер автозаповнення вниз до клітинки *K11*.
4. Поясніть формулу, що застосована для цієї таблиці.
5. Збережіть результати у файлі з тим само ім'ям у папці *Електронні таблиці* своєї структури папок.



	0	1	2	3	4	5	6	7	8	9
1	100	121	144	169	196	225	256	289	324	361
2	400	441	484	529	576	625	676	729	784	841
3	900	961	1024	1089	1156	1225	1296	1369	1444	1521
4	1600	1681	1764	1849	1936	2025	2116	2209	2304	2401
5	2500	2601	2704	2809	2916	3025	3136	3249	3364	3481
6	3600	3721	3844	3969	4096	4225	4356	4489	4624	4761
7	4900	5041	5184	5329	5476	5625	5776	5929	6084	6241
8	6400	6561	6724	6889	7056	7225	7396	7569	7744	7921
9	8100	8281	8464	8649	8836	9025	9216	9409	9604	9801

Мал. 6.4

6.2. Як додати до формули вбудовану функцію?

Ви вже вмієте використовувати для обчислень деякі вбудовані функції табличного процесора для знаходження суми значень діапазону клітинок, середнього значення, максимального чи мінімального значень.

Табличний процесор містить великий набір вбудованих функцій, які можна використати для обчислень та опрацювання даних. Кожна функція має своє ім'я, більшість функцій містить принаймні один необхідний для обчислення значення функції аргумент. Аргументи записуються в круглих дужках і відокремлюються один від одного крапкою з комою (;). Аргументом може бути число, текст, що записаний у лапках, посилання на клітинку чи діапазон клітинок, або вираз, що може також містити функції. Деякі аргументи є обов'язковими, деякі — ні.

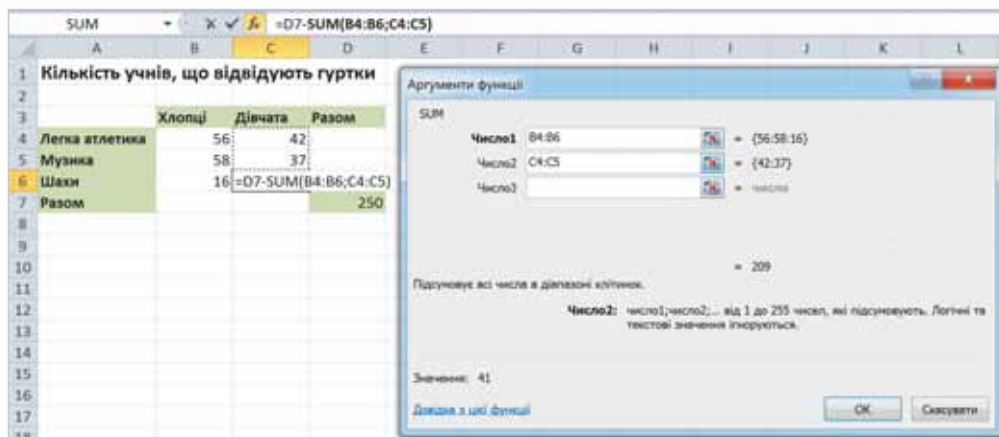
Функцію можна ввести з клавіатури, як і будь-який вміст клітинки. Для спрощення введення функцій до формули можна скористатися відповідними інструментами табличного процесора.

Для полегшення пошуку всі функції об'єднані в категорії: математичні, статистичні, логічні, фінансові, текстові тощо. Також виділена окрема категорія *Нещодавно використані в Microsoft Excel (Востаннє вжито в LibreOffice Calc, Усі в Таблицях Google)*, у якій можна переглянути та обрати функції, що використовувались останнім часом під час роботи на конкретному комп'ютері. Як правило, у цій категорії будуть відображені найчастіше вживані функції, тому в ній можна швидко знайти потрібну функцію. У табличному процесорі *Microsoft Excel* знайти та додати до формули функцію можна за допомогою інструментів із групи *Бібліотека функцій* на вкладці *Формули*.

У табличних процесорах ліворуч від рядка формул розташований інструмент *Вставити функцію* —  або , за допомогою якого відкривається вікно *Вставлення функції в Microsoft Excel* або *Помічник з функцій у LibreOffice Calc*.

У цьому вікні слід спочатку обрати категорію функції та ім'я потрібної функції зі списку доступних функцій обраної категорії. За потреби можна скористатися довідкою про використання кожної функції. На наступному кроці слід задати значення аргументів функції. Це можна зробити, виділивши клітинки на робочому столі за допомогою миші або ввівши адреси клітинок за допомогою клавіатури.

Аргументи функцій, як і посилання у формулах, можуть містити абсолютні, відносні або комбіновані посилання — залежно від того, чи має посилання змінюватися під час копіювання формули з функцією. Після завершення введення аргументів буде відображено значення функції, а в області *Значення* — результат обчислення формули (мал. 6.5).



Мал. 6.5



ДІЄМО

Вправа 3. Створення формули з використанням функції для обчислення.

Завдання. Учністу 8 та 9 класів запропонували обрати один з гуртків: легкої атлетики, музики та шахів. Серед 250 учнів 56 хлопців і 42 дівчини обрали легку атлетику, 58 хлопців і 37 дівчат — музику, 16 хлопців обрали шахи. В електронній таблиці *Гуртки*, що збережена в папці *Електронні таблиці*, створіть формули для обчислення кількості дівчат, які обрали шахи, та загальної кількості учнів та учениць, що обрали кожний гурток.

1. Завантажте файл *Гуртки*, що збережений у папці *Електронні таблиці* (<https://cutt.ly/jEKR6JP>).
2. Для обчислення кількості дівчат, що обрали шахи, слід від загальної кількості всіх учнів відняти суму всіх хлопців, а також дівчат, що обрали інші гуртки. Для створення такої формули виділіть клітинку *C6*, введіть символ «=», клацніть на клітинці *D7*, що містить значення 250, та введіть символ «-». Оберіть інструмент *Вставити функцію* ліворуч від рядка формул і перегляньте список функцій у категорії *Нещодавно використані*. Якщо серед них є функція *SUM*, оберіть її. В іншому разі оберіть категорію *Математичні* та оберіть цю функцію.
3. Для аргумента *Число 1* виділіть діапазон клітинок *B4:B6*, для аргумента *Число 2* — діапазон *C4:C5* (мал. 6.5). Натисніть кнопку *OK*.
4. Використовуючи інструмент *Автосума*, створіть формулу в клітинці *D4* для обчислення суми значень з діапазону клітинок *B4:C4*.
5. За допомогою автозаповнення скопіюйте створену формулу в клітинки *D5* і *D6*.

6. Збережіть результат у файлі з тим само іменем у папці *Електронні таблиці* своєї структури папок.

6.3. Які математичні та статистичні функції найчастіше використовуються в середовищі табличного процесора?

Найчастіше вживані функції, які можна додати до формули за допомогою засобу *Автосума*, можна знайти також у категоріях *Математичні* або *Статистичні*. Наприклад, функція SUM (СУММ) входить до категорії *Математичні*. Прикладами функцій з категорії *Статистичні* є AVERAGE (СРЗНАЧ) для знаходження середнього значення діапазону клітинок, COUNT (СЧЁТ) — кількості непорожніх клітинок у заданому діапазоні, MAX (МАКС) та MIN (МИН) відповідно — для знаходження найбільшого та найменшого значень. Для розв’язування задач з алгебри й геометрії можна використовувати вбудовані в табличний процесор математичні функції. Найбільш вживані з них наведено в таблиці 6.1.

Таблиця 6.1

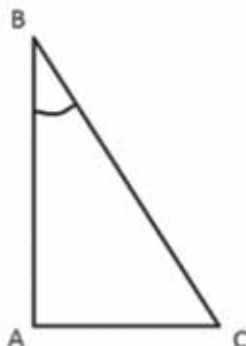
Функція	Функція (рос. інтерфейс)	Результат
ABS (число)	ABS (число)	Модуль (абсолютне значення) аргумента
COS (число)	COS (число)	Косинус аргумента, заданого в радіанах
DEGREES (число)	ГРАДУСЫ (число)	Перетворює значення кута, заданого в радіанах, у градуси
MOD (число; ділитель)	ОСТАТ (число; ділитель)	Остача від ділення заданого числа на вказаний дільник
SIN (число)	SIN (число)	Синус аргумента, заданого в радіанах
TAN (число)	TAN (число)	Тангенс аргумента, заданого в радіанах
PI()	ПИ ()	Значення числа π з точністю до 15 знаків
POWER (число; ступінь)	СТЕПЕНЬ (число; ступінь)	Результат піднесення числа до вказаного степеня
RADIANS (число)	РАДИАНЫ (число)	Перетворює значення кута, заданого в градусах, у радіани
ROUND (число; кількість знаків)	ОКРУГЛ (число; кількість знаків)	Округлене число до вказаної кількості знаків після коми
SQRT (число)	КОРЕНЬ (число)	Значення арифметичного квадратного кореня аргумента

Вправа 4. Використання математичних функцій у формулах.

Завдання. Для прямокутного трикутника ABC обчислити довжину гіпотенузи та другого катета за заданими значеннями катета і протилежного до нього кута.

1. Завантажте файл *Прямокутний трикутник*, що збережений у папці *Електронні таблиці* (<https://cutt.ly/GEKTdwA>).
 2. Виділіть клітинку $B5$, на вкладці *Формули* оберіть категорію *Математичні* та функцію $RADIANS$, для введення аргумента клацніть на клітинці $B4$. Формула в клітинці $B5$ набуде вигляду: $=RADIANS(B4)$.
 3. Для обчислення довжини гіпотенузи слід знайти частку довжини катета й синуса протилежного кута. Для створення формули виділіть клітинку $B8$, введіть символ « $=$ », клацніть на клітинці $B3$ та введіть символ « $/$ ». Оберіть інструмент *Вставити функцію* , у категорії *Математичні* знайдіть й оберіть функцію SIN . Зверніть увагу, що аргументом цієї функції має бути значення в радіанах, тому в поле *Число* слід ввести посилання на клітинку $B5$. Формула набуде вигляду $=B3/SIN(B5)$.
- П р и м і т к а. Формула може містити кілька функцій, наприклад, якщо не обчислювати значення кута B в радіанах у клітинці $B5$, то можна записати формулу в клітинці $B8$ так: $=B3/SIN(RADIANS(B3))$.
4. Аналогічно в клітинці $B9$ створіть формулу для обчислення довжини другого катета.
 5. Для клітинки $B9$ установіть числовий формат з округленням до десятих (мал. 6.6).

	A	B	C	D
1	У прямокутному трикутнику ABC знайди сторони AB та BC			
2				
3	Катет AC, см	4		
4	Кут B, у градусах	30		
5	Кут B, у радіанах	0,523599		
6				
7				
8	Гіпотенуза BC, см	8		
9	Катет AB, см	6,9		
10				
11				



Мал. 6.6

6. Збережіть результат у файлі з тим само ім'ям у папці *Електронні таблиці* своєї структури папок.



ПРАЦЮЄМО В ПАРАХ

1. Обговоріть, у чому полягає відмінність між абсолютними й відносними посиланнями на клітинки електронної таблиці. Чи має комбіноване посилання спільні ознаки абсолютного й відносного посилання?
2. Пограйте у гру *Аргументи функції*. Хтось називає ім'я функції з категорії *Математичні* чи *Статистичні*, а хтось — приклади можливих аргументів. Перевіряйте щоразу в середовищі табличного процесора та міняйтеть ролями.
3. Обговоріть, які функції з категорії *Математичні* б ви порадили використовувати учням 8 класу. Поясніть свій вибір.
4. Запропонуйте в парі функції із категорії *Статистичні*, які ви б порадили використовувати учням 9 класу, та наведіть приклад завдання.



ПРАЦЮЄМО САМОСТІЙНО

5. У чому відмінність між абсолютними, відносними та мішаними посиланнями у формулах в табличному процесорі? Що відбувається, коли формулу з однієї клітинки скопіювати в іншу? Як при цьому «спрацьовують» абсолютні та мішані посилання? Перевір себе у вправах:
 - 1) <https://learningapps.org/view22531728>;
 - 2) <https://learningapps.org/view22531611>.
6. Завантажте файл *Правило*, що збережений у папці *Електронні таблиці* (<https://cutt.ly/PEKTMSr>). Виконайте обчислення у файлі електронної таблиці (мал. 6.7). За допомогою табличного процесора перевірте істинність твердження $a \cdot b = \text{НСК}(a; b) \cdot \text{НСД}(a; b)$

Перше число	Друге число	Добуток	Найменше спільне кратне	Найбільший спільний дільник	Добуток НСК та НСД
a	b	ab	$\text{НСК}(a, b)$	$\text{НСД}(a, b)$	
48	12				

Мал. 6.7

7. Засобами табличного процесора створіть електронну таблицю степенів натуральних чисел першого десятка від першого степеня до п'ятого.

8. Засобами табличного процесора створіть електронну таблицю для визначення об'єму газу за нормальних умов (мал. 6.8).

	A	B	C	D	E	F
1	Молярний об'єм 1 моля газу за нормальних умов, л/моль					22,4
2	Газ	Маса газу, г	Молярна маса, г/моль	Об'єм, л	=B3/C3*\$F\$1	
3	Кисень	0,6	32			
4	Водень	0,4	2			
5	Вуглекислий газ	1,8	44			
6						

Мал. 6.8

9. У файлі *Каса*, що збережений у папці *Електронні таблиці* (<https://cutt.ly/2EKYuZq>), розміщено зведені відомості про продаж квитків касами автовокзалу. Завантажте файл і знайдіть вартість квитків по кожному напрямку та загалом.
10. Створіть електронну таблицю для обчислення вартості поїздки автомобілем *Nissan Note* від Києва до вказаних на малюнку 6.9 міст. Відстань можна дізнатися з інтернету чи розрахувати за картою автомобільних доріг України, враховуючи відповідний її масштаб. Вартість 1 л пального та витрати палива на 100 км для даного автомобіля знайдіть в інтернеті.

	A	B	C	D
			Кількість	
		Відстань до	необхідного	Вартість,
1	Міста	Києва, км	пального	грн.
2	Львів			
3	Одеса			
4	Хмельницький			
5	Вінниця			
6	Луцьк			
7	Умань			
8				
9	Вартість 1 літра пального (А-95)			

Мал. 6.9

11. Попрацюйте в мініпроекті «Мій вплив на енергозбереження». Засобами табличного процесора створіть електронну таблицю для планування економії споживання електричної енергії. Скористайтесь файлом *Енергоспоживання* (<https://cutt.ly/WEKPuZ5>) для отримання відомостей про потужність побутового пристрою. Обчисліть обсяг споживання електроенергії за тиждень у вашій родині та її вартість. Сплануйте, де в підрахунках мають бути абсолютні, а де — відносні посилання.

ЯК РЕАЛІЗУВАТИ ВИБІР В СЕРЕДОВИЩІ ТАБЛИЧНОГО ПРОЦЕСОРА?

7

ПРИГАДАЙТЕ:

- як описують алгоритмічну структуру розгалуження мовою програмування;
- якими є таблиці істинності для логічних елементів AND, OR, NOT;
- як змінювати форматування таблиці.

ВИ ДІЗНАЄТЕСЯ:

- як у формулах використовують логічні функції;
- як скористатися засобом умовного форматування даних.



ВИВЧАЄМО

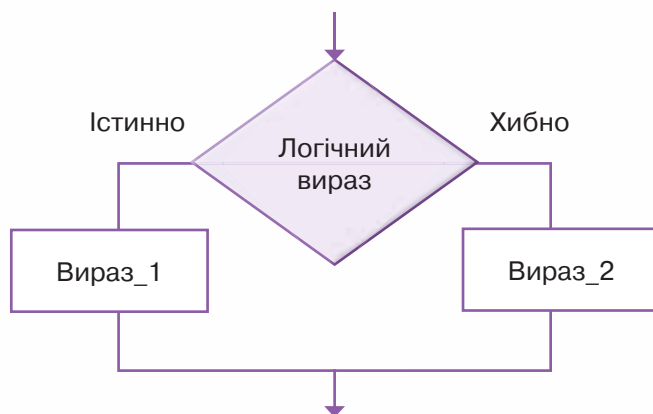
7.1. Як у формулах використовують логічні функції?

У середовищі табличного процесора використовуються також формули, що мають аргументом логічний вираз. Логічний вираз містить знак порівняння та може набувати одного з двох значень: *TRUE* (істина) та *FALSE* (хибність) залежно від конкретних значень, що зберігаються у клітинках, на адреси яких існує посилання у виразі. Наприклад, логічні вирази наведено в таблиці 7.1.

Таблиця 7.1

Логічний вираз	Пояснення умови, за якої логічний вираз набуває значення <i>TRUE</i> (істина)
$A1 > 1$	Числове значення в клітинці <i>A1</i> більше за 1
$F2 * A4 = \text{SUM}(B2:B13)$	Добуток значень у клітинках <i>F2</i> та <i>A4</i> дорівнює сумі значень діапазону <i>B2:B13</i>
$B13 = \text{«Петрик»}$	Вмістом клітинки <i>B13</i> є текстове значення «Петрик»
$B3 - C3 \geq 12$	Різниця значень у клітинках <i>B3</i> та <i>C3</i> більша або дорівнює 12

Якщо результат обчислення за формулою залежить від виконання деякої умови — істинності логічного виразу, то використовують логічні функції. Під час складання алгоритмів у цьому разі ви використовуєте алгоритмічну структуру розгалуження. До категорії логічних функцій належить функція *IF* (ЕСЛИ), за допомогою якої можна організувати розгалуження в середовищі табличного процесора (мал. 7.1).



■ Мал. 7.1 ■

Функція *IF* має три аргументи:

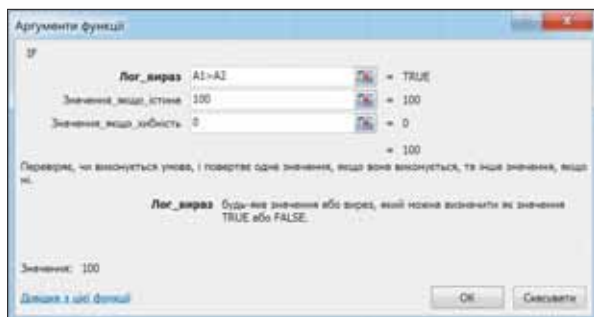
IF(логічний_вираз; значення_якщо_істина; значення_якщо_хибність).

При використанні функції *IF* перевіряється істинність вказаного логічного виразу, і до тієї клітинки, де міститься формула, *заноситься значення_якщо_істина*, якщо логічний вираз істинний, або *значення_якщо_хибність*, якщо логічний вираз хибний. Аргументами *значення_якщо_істина* та *значення_якщо_хибність* можуть бути число, текстове значення, яке має бути записано в лапках, або вираз для обчислення. Вираз для обчислення може містити інші функції, зокрема і вкладену функцію *IF*, якщо слід сформулювати більш складну умову. Приклади формул, що містять логічну функцію *IF*, подано в таблиці 7.2.

Таблиця 7.2

Формула	Значення в клітинках	Результат обчислення за формулою
= IF(A1*B2>0; SQRT(A1*B2); POWER(A1,B2))	A1 = 2, B2 = 18	6
	A1 = 5, B2 = -1	0,2
	A1 = -1, B2 = 4	1
= IF(B2>C2; «Перевищення бюджету»; «OK»)	B2 = 450, C2 = 500	OK
	B2 = 300, C2 = 250	Перевищення бюджету
= IF(A10=100; SUM(B5:B7); "")	A10 = 100, B5 = 32, B6 = 45, B7 = 18	95
	A10 = 50, B5 = 100, B6 = 210, B7 = 180	Порожня клітинка
= IF(A1>A2; 100; 0)	A1 = 20, A2 = 25	0
	A1 = 20, A2 = 15	100

Під час використання майстра функцій кожний з аргументів функції *IF* записують в окремому полі (мал. 7.2, а, б).



■ Мал. 7.2, а ■



■ Мал. 7.2, б ■

Для створення складеного логічного виразу використовують інші функції з категорії *Логічні*: *AND*(логзнач1; логзнач2;...) (*И*(логзнач1; логзнач2;...)), *OR*(логзнач1; логзнач2;...) (*ИЛИ*(логзнач1; логзнач2;...)), *NOT*(логзнач) (*НЕ*(логзнач)). Аргументами логічних функцій є логічні вирази, що можуть набувати одне з двох значень — істина чи хибність. Їх використання аналогічне відповідним командам, які ви використовуєте для запису складених умов під час складання алгоритмів — програм. Найчастіше логічні функції *AND*, *OR*, *NOT* використовують для запису складеної умови як аргументу логічної функції *IF*.



ДІЄМО

Вправа 1. Використання логічної функції *IF*.

Завдання. Для даних зросту учнів 9 класу, заданих у стовпчику *B* електронної таблиці *Зріст* (<https://cutt.ly/mEKGut3>), у стовпчику *C* створіть формули для визначення кольору маркування шкільних парт за умови, що для учнів, які мають зріст понад 160 см, маркування має бути зеленим, а для інших — червоним.

1. Завантажте файл *Зріст*, що збережений у папці *Електронні таблиці* (<https://cutt.ly/mEKGut3>).
2. Виділіть клітинку *C2*. Оберіть інструмент *Вставити функцію*, у категорії *Логічні* знайдіть і оберіть функцію *IF*.
3. Виділіть аргумент *Логічний вираз*, клацніть на клітинці *B2*, введіть із клавіатури ≤ 160 .
4. Виділіть аргумент *Значення_якщо_істина*, введіть текст *Червоне*. Зверніть увагу, що текстове значення автоматично буде взято в лапки.
5. Виділіть аргумент *Значення_якщо_хибність*, введіть текст *Зелене*. Натисніть кнопку *OK*. Переконайтеся, що формула має вигляд: $=IF(B2 \leq 160; \text{«Червоне»}; \text{«Зелене»})$.

- Виділіть клітинку C2, скористайтесь автозаповненням для копіювання формули в діапазон клітинок C3:C6. Перевірте результат у діапазоні C3:C6 (мал. 7.3).

	A	B	C
1	Прізвище учня	Зріст	Маркування
2	Антоненко	145	Червоне
3	Байтало	160	Червоне
4	Вівчар	162	Зелене
5	Гнатишин	149	Червоне
6	Едгар	170	Зелене

Мал. 7.3

Вправа 2. Лінійне рівняння.

Завдання. Засобами табличного процесора створіть електронну таблицю, в якій при введенні коефіцієнтів лінійного рівняння відображатиметься значення його кореня або довідка про кількість його розв'язків.

- Проаналізуємо, скільки розв'язків може мати лінійне рівняння $ax = b$:
 - Безліч розв'язків, якщо $a = 0$ та $b = 0$. Таку умову можна записати за допомогою логічної функції $AND(a=0;b=0)$.
 - Жодного розв'язку, якщо $a = 0$ та $b \neq 0$. Умову запишемо за допомогою функції $AND(a=0;b \neq 0)$.
 - Єдиний розв'язок $x = b/a$, якщо $a \neq 0$.
- Створіть файл *Лінійне рівняння*, збережіть його в папці *Електронні таблиці* (мал. 7.4).

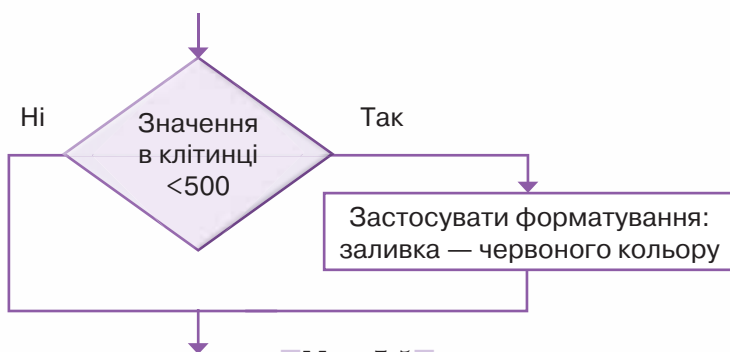
	A	B	C	D
1	Розв'язування лінійного рівняння $ax=b$			
2				
3	a=	b=		Розв'язок:
4		4	28	
5				

Мал. 7.4

- Виділіть клітинку D4 та введіть до неї формулу, що містить логічні функції: $=IF(AND(A4=0;B4=0); \text{«Безліч розв'язків»}; IF(AND(A4=0;B4 \neq 0); \text{«Жодного розв'язку»}; B4/A4))$.
- Змініть значення коефіцієнтів: введіть у клітинку A4 число 0. Перевірте розв'язок у клітинці D4. Введіть у клітинку B4 число 0. Як змінився результат?
- Збережіть результати роботи у файлі з тим само ім'ям у папці *Електронні таблиці* своєї структури папок.

7.2. Як скористатися засобом умовного форматування даних?

Змінити в електронній таблиці форматування даних, що відповідають певним умовам, можна за допомогою умовного форматування. Цей засіб використовують, щоб створювати правила форматування окремих клітинок таблиці залежно від їхнього значення. Наприклад, можна застосувати умовне форматування до клітинок так, щоб будь-яке числове значення, що менше ніж 500, відображалось у відповідних клітинках на червоному фоні. Дія такого умовного форматування відповідає команді розгалуження (мал. 7.5): *Якщо значення в клітинці, що входить до виділеного діапазону, < 500, то застосувати до клітинки форматування: заливка — червоного кольору.*



Мал. 7.5

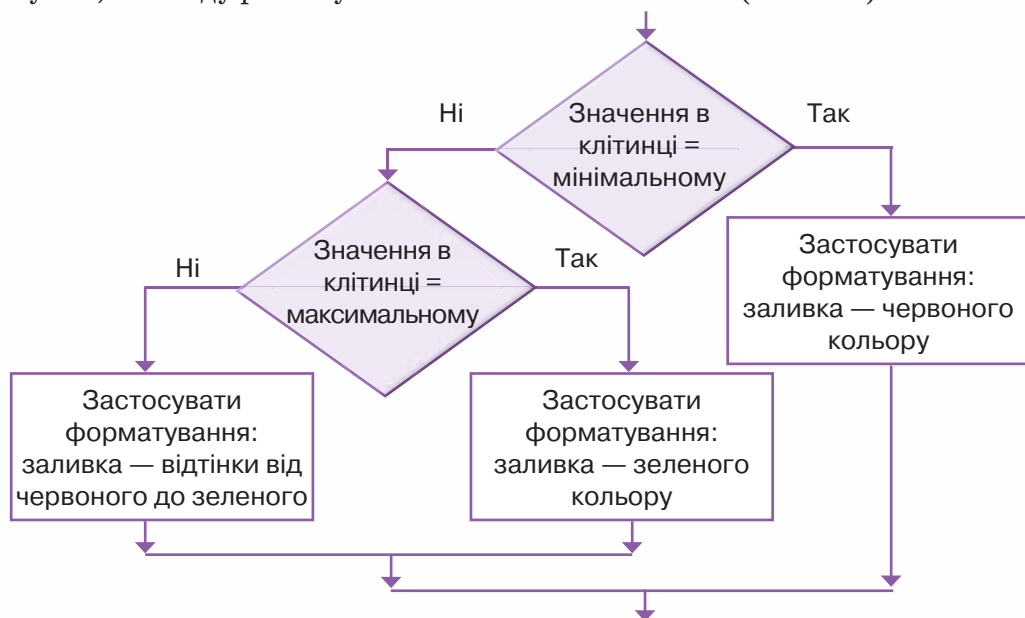
Табличні процесори також дають змогу застосовувати умовне форматування з використанням гістограм, кольорових шкал та наборів піктограм. Наприклад, за допомогою кольорової шкали всі найменші значення виділеного діапазону можуть мати червоний колір заливки, найбільші значення — зелений, а всі проміжні значення — інші відтінки від червоного до зеленого (мал. 7.6).

	A	B	C	D	E	F	G	H
1	Навчальні досягнення учнів							
2								
3	Прізвище	Алгебра	Геометрія	Фізика	Інформатика	Укр.літ.	Заруб.літ.	Біологія
4	Бондар В.	11	11	10	12	10	11	11
5	Гапон С.	9	8	9	8	9	10	10
6	Стецюк К.	8	9	9	9	8	9	10
7	Савицька О.	4	4	3	4	4	3	5
8	Ткаченко В.	9	9	10	11	11	11	10

Мал. 7.6

Дія умовного форматування з використанням гістограм, кольорової шкали та набору піктограм відповідає команді розгалуження з вкладеннями. Наприклад, для умовного форматування з використанням кольо-

рової шкали за шаблоном , результат якого відображено на малюнку 7.6, команду розгалуження можна записати так (мал. 7.7):

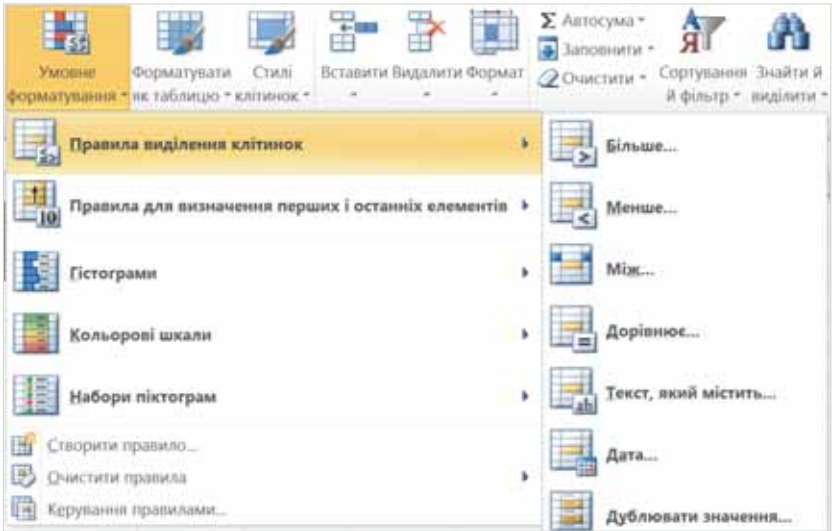


■ Мал. 7.7 ■

Якщо значення в клітинці збігається з мінімальним значенням діапазону, то застосувати до клітинки форматування: заливка — червоного кольору; інакше, якщо значення в клітинці співпадає з максимальним значенням діапазону, то застосувати до клітинки форматування: заливка — зеленого кольору; інакше застосувати форматування: заливка — відтінки від червоного до зеленого.

Для умовного форматування вмісту клітинок необхідно виділити клітинки, до яких застосовуватиметься таке форматування: на вкладці *Основне* в групі *Стили* обрати інструмент *Умове форматування* (для *Microsoft Excel*) або вибрати в меню *Формат* вказівку *Умове форматування* (для *LibreOffice Calc*) (мал. 7.8, а, б). Після цього слід обрати тип умовного форматування для створення нового правила виділення клітинок або гістограми, кольорові шкали чи набори піктограм. Під час створення нового правила виділення клітинок можна обрати умову для форматування (мал. 7.8, а), задати значення, з якими слід порівнювати значення з діапазону клітинок, та вказати формат, який необхідно застосувати (мал. 7.9). Якщо серед запропонованих форматів жоден не підходить, можна обрати варіант *Настроюваний формат*, при цьому буде відображено вікно *Формат клітинки*, в якому можна задати будь-які параметри форматування символів, заливки, меж тощо. Щоб скасувати умовне форматування в *Microsoft Excel*, достатньо виділити

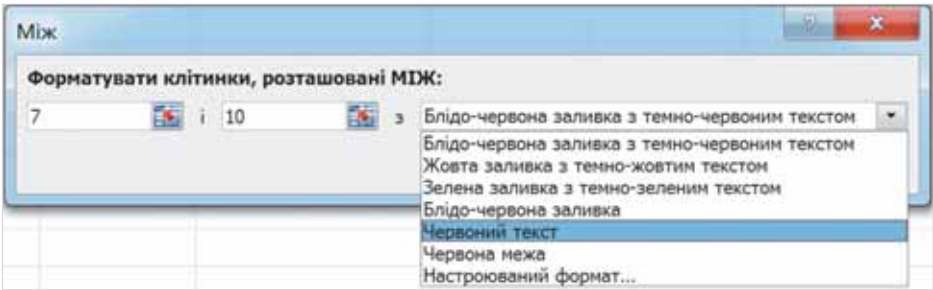
діапазон, до якого застосували умовне форматування, та обрати вказівку *Очистити* правила зі списку вказівок інструмента *Умове форматування* (мал. 7.8, а). У табличному процесорі *LibreOffice Calc* для цього після виділення діапазону з умовним форматуванням слід обрати вказівку *Формат/Умове форматування/Керування*, й у вікні *Управління умовним форматуванням* обрати кнопки *Видалити* та *Гаразд*.



Мал. 7.8,а



Мал. 7.8,б



Мал. 7.9



ДІМО

Вправа 2. Комп'ютерна вишивка.

Завдання. Застосуйте умовне форматування до таблиці *Вишиванка* так, щоб дані були відображені з використанням кольорової шкали: найменші значення — на червоному фоні, найбільші — на зеленому, всі проміжні — у відтінках від червоного до зеленого.

1. Відкрийте файл *Вишиванка*, що зберігається в папці *Електронні таблиці* (<https://cutt.ly/tT1DI1t>).
2. Виділіть діапазон клітинок *B2:K11*, що містить остачу від ділення добутку номера рядка та номера стовпця на деяке число, записане в клітинці *M1*.
3. Оберіть інструмент *Умовне форматування/Кольорові шкали* на вкладці *Основне* (вказівку *Формат/Умовне форматування/Колірна шкала*).
4. Серед шаблонів кольірних шкал оберіть такий, щоб найменше значення відображалось червоним, найбільше — зеленим, середнє — жовтим (мал. 7.10).
5. Декілька разів змініть значення у клітинці *M1* та дістаньте результат, який вам найбільше до вподоби.
6. Збережіть результати роботи у файлі з тим само іменем у папці *Електронні таблиці* своєї структури папок.

	A	B	C	D	E	F	G	H	I	J	K	L
		1	2	3	4	5	6	7	8	9	10	
1		Yellow	Green	Green	Red	Yellow	Green	Green	Red	Yellow	Green	
2		Green	Red	Green	Red	Green	Red	Green	Red	Green	Red	
3		Green	Green	Yellow	Red	Green	Green	Yellow	Red	Green	Green	
4		Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	
5		Yellow	Green	Green	Red	Yellow	Green	Green	Red	Yellow	Green	
6		Green	Red	Green	Red	Green	Red	Green	Red	Green	Red	
7		Green	Green	Yellow	Red	Green	Green	Yellow	Red	Green	Green	
8		Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	
9		Yellow	Green	Green	Red	Yellow	Green	Green	Red	Yellow	Green	
10		Green	Red	Green	Red	Green	Red	Green	Red	Green	Red	

Мал. 7.10



ДОСЛІДЖУЕМО

1. Дослідіть, які набори піктограм можна використовувати для умовного форматування даних у таблиці. Для цього введіть у таблиці значення 10, 20, 30, 40, виділіть діапазон клітинок із цими даними, оберіть інструмент *Умове форматування* на вкладці *Основне* та *Набори піктограм* (для *Microsoft Excel*) або вказівку *Формат/Умове форматування/Набір піктограм* (для *LibreOffice Calc*). Обирайте різні набори піктограм та визначте, як змінюється відображення даних для кожного з наборів.

	10		20		30		40
--	----	--	----	--	----	--	----

2. Дослідіть, яку кількість умов можна використовувати одночасно при застосуванні умовного форматування.
3. Дослідіть, правила яких типів можна створювати для умовного форматування даних таблиці.



ПРАЦЮЕМО В ПАРАХ

1. Обговоріть ситуації, в яких доцільно застосовувати умовне форматування електронної таблиці.
2. Пограйте в гру *Аргументи функції*. Хтось називає ім'я функції з категорії *Логічні*, а хтось наводить приклади можливих її аргументів. Поміняйтесь ролями.



3. Клітинка *B2* містить значення — 20, клітинка *C2* — значення 10. У клітинку *D2* введено формулу з використанням логічної функції *IF*. По черзі визначаєте, яке значення буде відображено в клітинці *D2* та обґрунтовуйте свою відповідь.

- 1) $=IF(B2<0; -B2; B2)$
- 2) $=IF(B2>C2; "більше"; "")$
- 3) $=IF(C2>0; B2+C2; B2-C2)$

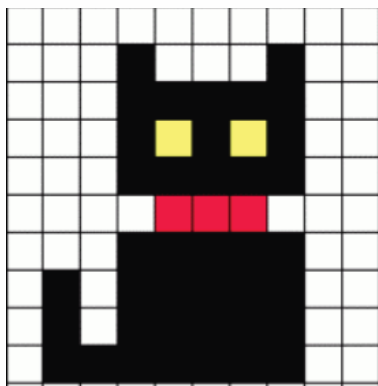


4. У клітинці *A1* введено число 100, а в клітинці *B1* — число 10. По черзі називаєте значення логічних функцій та обґрунтовуйте свою відповідь:

- 1) $AND(A10)$
- 2) $OR(A10)$
- 3) $AND(A1>20; B1<=50)$
- 4) $NOT(A1<>0)$
- 5) $AND(A1>50; A1<>70; B1=10)$
- 6) $OR(A1<>0; B1<20)$



5. Обговоріть, які засоби табличного процесора можна використати, щоб отримати зображення кішки, у якої колір бантика та «шерсті» може мінятись залежно від значення деякої клітинки.



ПРАЦЮЄМО САМОСТІЙНО

6. Відкрийте файл *Квадратний корінь*, збережений у папці *Електронні таблиці* (<https://cutt.ly/REKH7rl>). Створіть формулу для обчислення значень арифметичного квадратного кореня для невід'ємних чисел. Для від'ємних значень передбачте текстовий коментар «Не існує». Скористайтесь логічною функцією *IF* та функцією *SQRT* з категорії *Математичні*.
7. Створіть електронну таблицю, за допомогою якої можна визначити час у Сіднеї, Пекіні, Кейптауні, Парижі, Нью-Йорку, якщо буде задано час у Києві. Скористайтесь картою годинних поясів. Застосуйте до даних таблиці умовне форматування з кольоровою шкалою.
8. Попрацюйте в мініпроекті «Е-школа». Створіть електронну таблицю — аналог шкільного журналу. Передбачте, що один аркуш електронної таблиці буде відповідати одному шкільному предмету. На аркуші на кожному десятому стовпці після стовпця прізвищ автоматично обчислюється тематичне оцінювання (середнє арифметичне всіх оцінок з теми). До клітинок тематичного оцінювання застосоване умовне форматування за правилом: високі результати (10–12 балів) відображаються на зеленому тлі, достатні (7–9 балів) — на жовтому, усі інші — на рожевому. Додайте інші (на власний розсуд) підсумкові оцінки чи форматування. Представте результат проекту в класі.

ПРИГАДАЙТЕ:

- що відносять до інфорграфіки;
- як друкувати електронні документи, створені в текстовому процесорі.

ВИ ДІЗНАЄТЕСЯ:

- як наочно подати числові дані за допомогою діаграм;
- з яких об'єктів складається діаграма;
- як створити діаграму в середовищі табличного процесора;
- як «читати» діаграму, побудовану на основі даних таблиці;
- який тип діаграми обрати.



ВИВЧАЄМО

8.1. Як наочно подати числові дані за допомогою діаграм?

Для наочного подання числових даних та їх аналізу використовують *діаграми*. Діаграма є однією з форм подання інформаційної моделі та засобом для її дослідження.



Діаграма — графічне зображення, що наочно відображає лінійними відрізками чи геометричними фігурами співвідношення між числовими даними.

ЦІКАВО

Діаграма — від грецьк. *diagramma* — зображення, малюнок, креслення.

На уроках математики ви вже будували кругові та стовпчасті діаграми за допомогою креслярських інструментів. Їх можна створювати в середовищі табличного процесора.

ЦІКАВО

Кругові діаграми, створені в середовищі табличного процесора, називають секторними, а стовпчасті, створені в середовищі табличного процесора, — гістограмами.

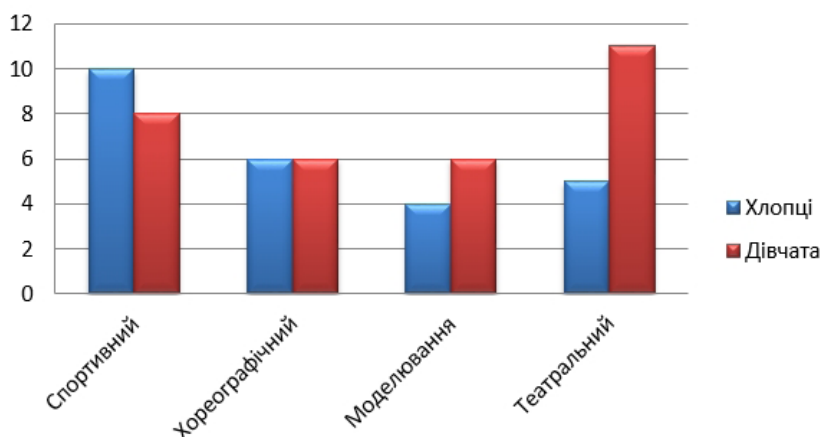
Для прикладу розглянемо таблицю з даними про участь учнів 9 класів у шкільних гуртках (мал. 8.1). На основі даних цієї таблиці можна побудувати стовпчасту (мал. 8.2) та кругову (мал. 8.3) діаграми.

	A	B	C	D
1	Участь учнів 9 класу у гуртках			
2	Назва гуртка	Хлопці	Дівчата	
3	Спортивний	10	8	
4	Хореографічний	6	6	
5	Моделювання	4	6	
6	Театральний	5	11	

Мал. 8.1

Стовпчаста діаграма демонструє зміну даних за певний період часу або дає змогу порівнювати дані, що належать до однакової категорії, наприклад, порівняти кількість дівчат та хлопців, які займаються у різних гуртках (мал. 8.2).

Участь учнів 9 класу у гуртках



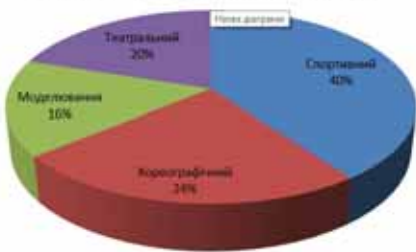
Мал. 8.2

Значення, що відповідають даним таблиці (мал. 8.1), відображені на вертикальній осі. Дані діапазону клітинок $B3:C11$ називають **рядами даних**.

На горизонтальній осі вказані назви заголовків таблиці для даних, на основі яких створена діаграма, — **поля даних**.

На основі даних, упорядкованих лише в один рядок або стовпець, можна побудувати **секторну діаграму** (мал. 8.3). Секторні діаграми відображають величину кожного елемента в одному ряду даних, пропорційно до суми елементів, числом чи відсотковим значенням. Наприклад, на малюнку 8.3, *а* ряд даних — це діапазон клітинок $B3:B6$ (мал. 8.1), а на малюнку 8.3, *б* — $B3:C3$.

Участь хлопців 9 класу у гуртках



Мал. 8.3, а

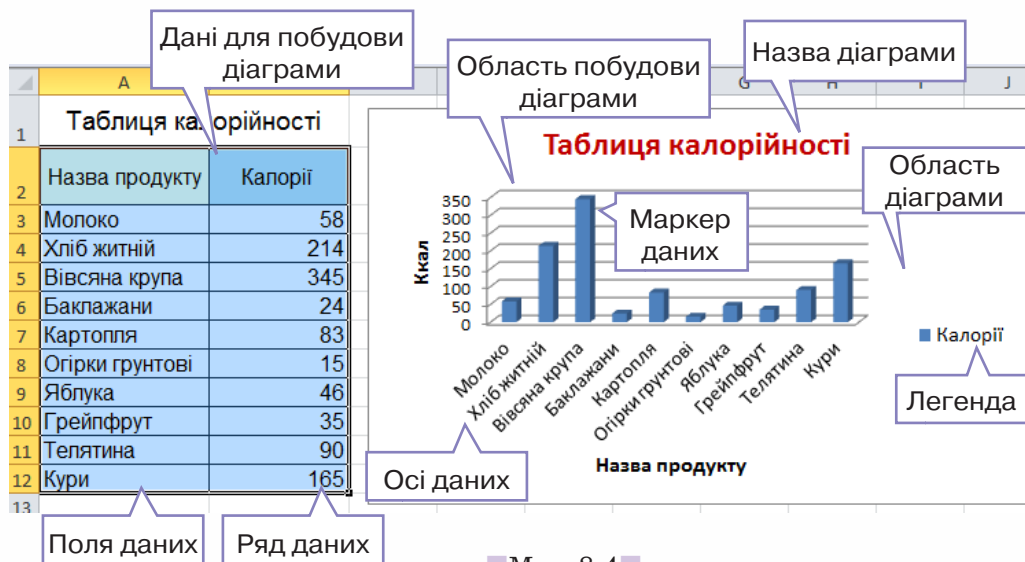
Спортивний гурток



Мал. 8.3, б

8.2. З яких об'єктів складається діаграма?

У середовищі табличного процесора діаграма та дані, на основі яких вона побудована, зв'язані між собою. Це означає, що діаграма автоматично оновлюється, якщо вносяться зміни в таблицю з даними.



Мал. 8.4

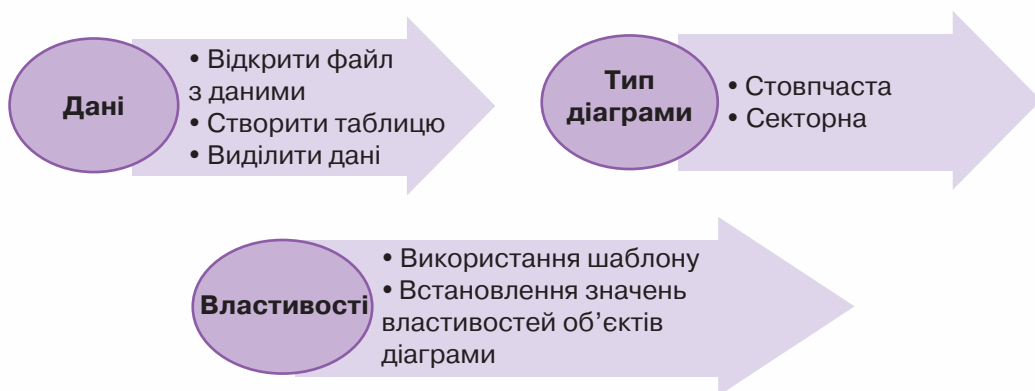
Діаграма складається з різних об'єктів (мал. 8.4):

- **заголовки:** назва діаграми, горизонтальної осі, вертикальної осі. Основний заголовок може, наприклад, відображати назву таблиці чи пояснювати дані діаграми; заголовки осей відповідають назвам стовпців і рядків виділеного діапазону даних;
- **осі:** горизонтальна та вертикальна, вздовж яких будується діаграма;
- **маркер даних** — це смуга, область, точка, сегмент або інший елемент на діаграмі, що відповідає одному значенню однієї клітинки аркуша; маркери даних одного кольору на діаграмі утворюють ряд даних; діаграми всіх типів, крім кругової, можуть містити кілька рядів даних;

- **легенда** — призначена для допомоги користувачеві при аналізі діаграми зрозуміти відповідність кольорів певним даним виділеного діапазону; у легенді відображаються імена рядів даних;
- **область діаграми** — містить кілька об'єктів, основним з яких є **область побудови діаграми**.

8.3. Як створити діаграму в середовищі табличного процесора?

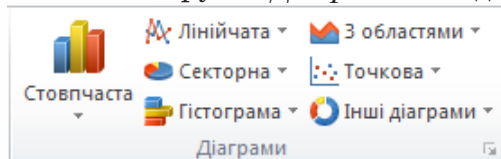
Діаграма створюється на основі даних таблиці. Тому перш ніж створювати діаграму (мал. 8.5), спочатку потрібно створити таблицю чи відкрити в середовищі табличного процесора файл — електронну таблицю, що містить потрібні дані.



Мал. 8.5

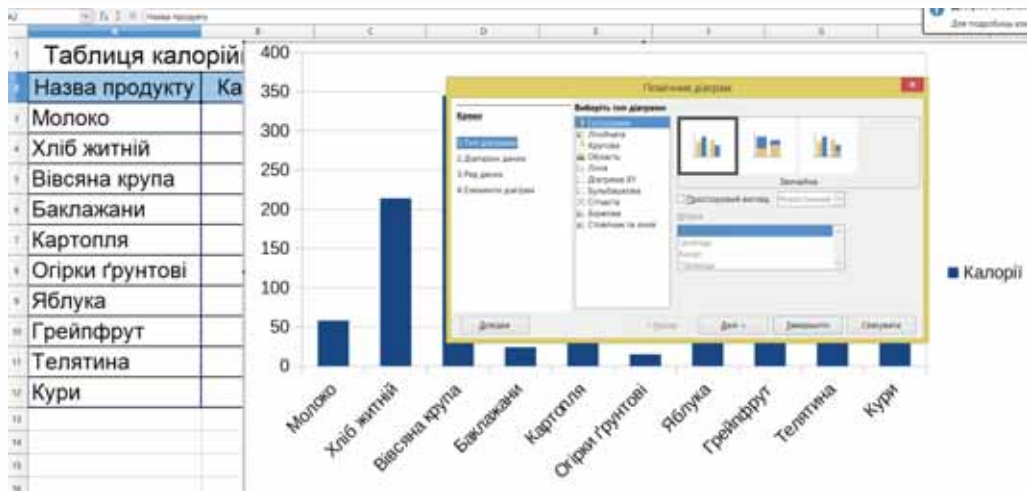
Для побудови діаграми виділяють діапазон клітинок, що містить необхідні дані: ряди та поля даних для побудови стовпчастої діаграми, поля даних — для секторної. При цьому слід виділяти назви стовпців чи/та рядків і числові дані (мал. 8.6).

Далі в табличному процесорі *Microsoft Excel* на вкладці *Вставка* в групі *Діаграми* слід обрати потрібний тип діаграми



та встановити властивості діаграми.

Для побудови діаграми за даними виділеного діапазону у середовищі табличного процесора *LibreOffice Calc* на панелі інструментів *Стандартна* обирають інструмент *Вставити діаграму* . При цьому на екрані з'являється стовпчаста діаграма — *гістограма* та вікно *Помічник діаграм* (мал. 8.6). Користувач може слідувати кроками помічника та на кожному з кроків задавати властивості обраних об'єктів діаграми або завершити роботу та прийняти усі властивості, які задані за замовчуванням.



Мал. 8.6

Для зміни властивостей діаграм та їхніх об'єктів можна використати інструменти вкладки *Знаряддя для діаграм* або



інструменти . Можливість доступу до інструментів зміни з'являється при виділенні діаграми.

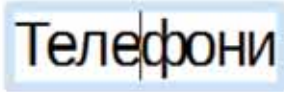


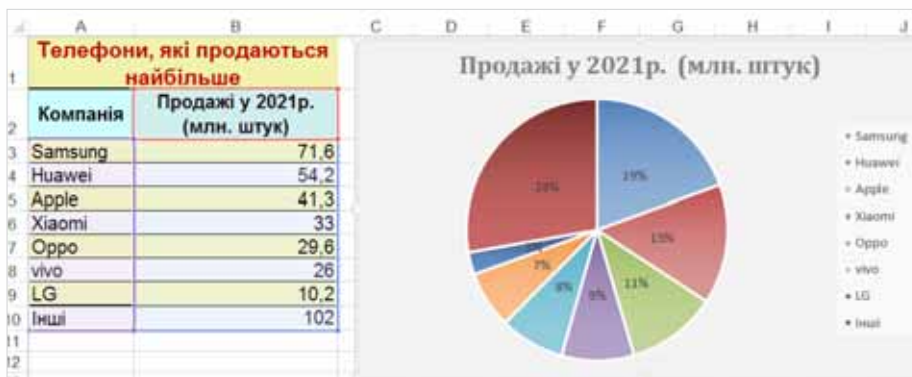
ДІМО

Вправа 1. Мобільні телефони.

Завдання. Змініть назву секторної діаграми *Телефони*, що зберігається у файлі *Продаж*, задайте розмір символів назви — 16. Змініть місце розташування легенди.

1. Відкрийте файл *Продаж*, що зберігається в папці *Електронні таблиці* (<https://cutt.ly/zECIFND>).
2. Виділіть назву діаграми.

Microsoft Excel	LibreOffice Calc
У контекстному меню оберіть вказівку <i>Змінити текст</i> (мал. 8.7).	Двічі клацніть, щоб перейти до режиму правки 



Мал. 8.7

- У текстовому написі введіть новий заголовок *Продаж мобільних телефонів*. Клацніть мишею у вільному від об'єктів місці в області діаграми.
- Встановіть розмір літер — 16, а колір тексту — синій.

Microsoft Excel	LibreOffice Calc
Клацніть правою клавішею миші на заголовку діаграми й за допомогою інструментів, розташованих на міні-панелі форматування, встановіть потрібні параметри форматування	У контекстному меню заголовка оберіть вказівку <i>Формат заголовка</i> й у вкладці <i>Шрифт</i> оберіть потрібні параметри форматування

- Змініть місце розташування легенди за зразком (мал. 8.8).

Microsoft Excel	LibreOffice Calc
Клацніть правою клавішею миші на діаграмі за допомогою інструментів групи <i>Стили діаграм</i> вкладки <i>Знаряддя для діаграм</i> розмістіть легенду знизу	У контекстному меню легенди оберіть вказівку <i>Формат легенди</i> й у вкладці <i>Позиція</i> оберіть <i>Внизу</i>



Мал. 8.8

- Збережіть результати у файлі з тим само іменем у папці *Електронні таблиці* вашої структури папок.

Вправа 2. Продаж телефонів.

Завдання. Побудуйте стовпчасту діаграму для даних таблиці *Продаж* для порівняння кількості проданих мобільних телефонів у 2020 та 2021 роках.

- Відкрийте файл *Продаж*, що зберігається в папці *Електронні таблиці* вашої структури папок, та оберіть аркуш *Продаж_порівняння* (мал. 8.9).

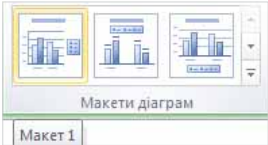
	A	B	C
1	Телефони, які продаються найбільше		
2	Компанія	Продажі у 2021 р. (млн. штук)	Продажі у 2020 р. (млн. штук)
3	Samsung	71,6	80,4
4	Huawei	54,2	38,5
5	Apple	41,3	41
6	Xiaomi	33	23,1
7	Oppo	29,6	30,5
8	vivo	26	25,8
9	LG	10,2	13,3
10	Інші	102	94
11			

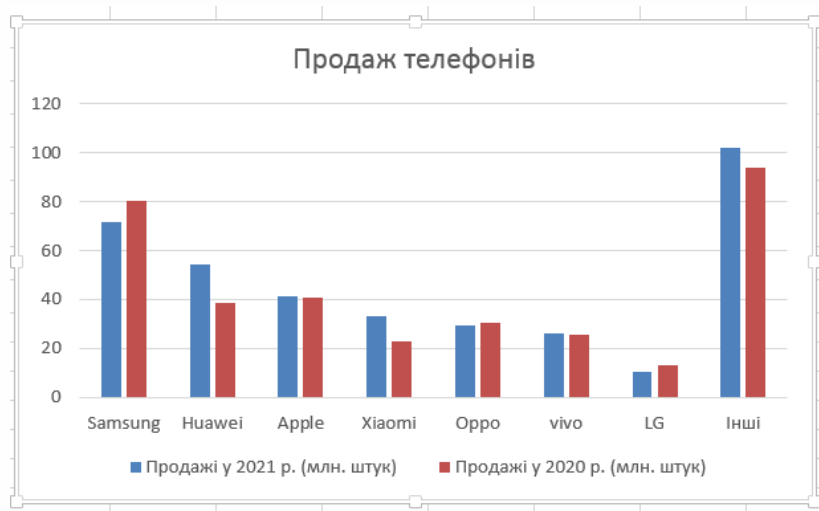
Мал. 8.9

- Виділіть діапазон клітинок A2:C10 і побудуйте стовпчасту діаграму.

<i>Microsoft Excel</i>	<i>LibreOffice Calc</i>
На вкладці <i>Вставлення</i> виберіть за допомогою інструментів групи <i>Діаграми</i> стовпчасту діаграму	На панелі інструментів <i>Стандартна</i> оберіть інструмент <i>Вставити діаграму</i>

- У полі заголовка введіть назву діаграми *Продаж телефонів* як показано на малюнку 8.10.

<i>Microsoft Excel</i>	<i>LibreOffice Calc</i>
За допомогою інструментів з групи <i>Макети діаграм</i> виберіть потрібний  макет	На четвертому кроці помічника діаграм введіть заголовок



■ Мал. 8.10 ■

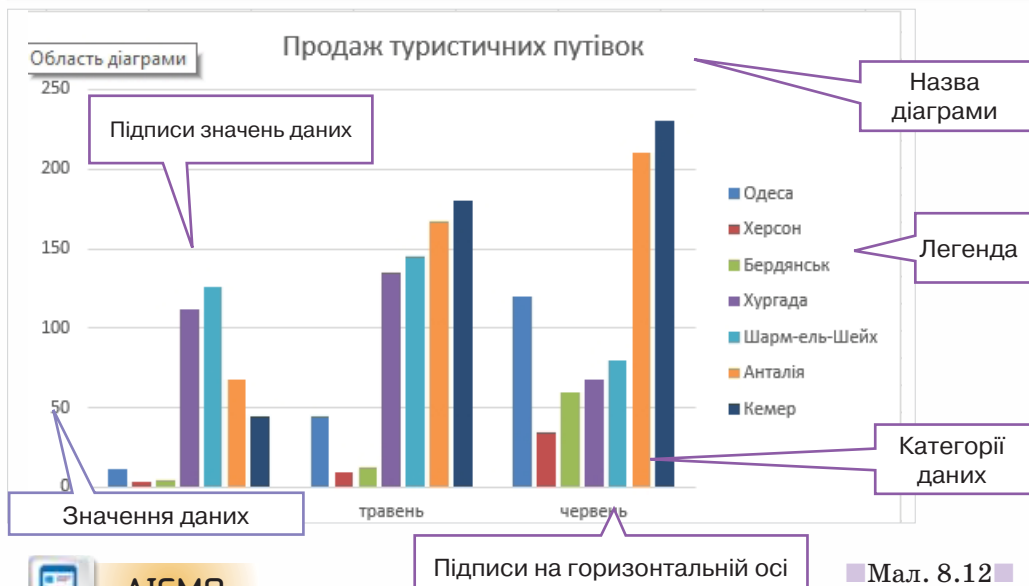
4. Збережіть результати у тому само файлі.

8.4. Як «читати» діаграму, побудовану на основі даних таблиці?

Після побудови діаграми важливо навчитись інтерпретувати її, тобто «читати» діаграму та пояснювати, що на ній відображено. Щоб діаграму правильно інтерпретувати, вона має містити всі необхідні складові: назву діаграми, підписи на осях, легенду та підписи даних. Якщо деякі із цих об'єктів відсутні, діаграму може бути складно аналізувати. Кругова діаграма може бути побудована лише для одного ряду даних — коли дані є складовою одного цілого; за допомогою інших типів, як правило, можна відображати декілька рядів даних. Для таблиці, що міститься у файлі *Путівки* (мал. 8.11), за допомогою стовпчастої діаграми можна порівнювати попит на тури до конкретних міст за кількістю проданих путівок за три місяці, або порівнювати кількість проданих путівок у кожному місяці в конкретному напрямку (місто) (мал. 8.12).

	A	B	C	D	E	F
1	Продаж туристичних путівок					
2						
3	Країна	Місто	квітень	травень	червень	
4	Україна	Одеса	12	45	120	
5	Україна	Херсон	3	10	35	
6	Україна	Бердянськ	4	13	60	
7	Єгипет	Хургада	112	135	68	
8	Єгипет	Шарм-ель-Шейх	126	145	80	
9	Туреччина	Анталія	68	167	210	
10	Туреччина	Кемер	45	180	230	
11		Всього	370	695	803	

■ Мал. 8.11 ■



ДІЄМО

Вправа 3. Аналіз даних, відображених на діаграмі.

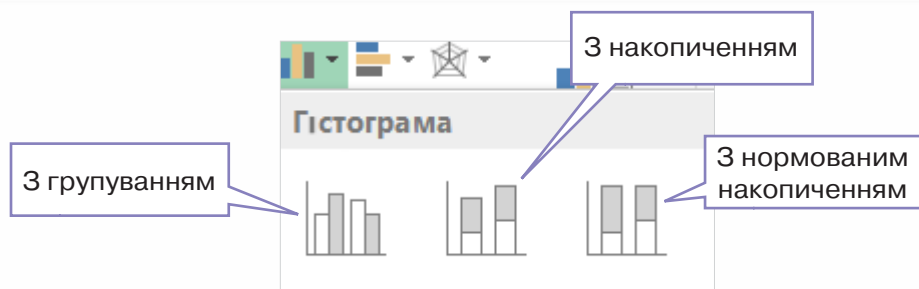
Завдання. Проаналізуйте дані, відображені на діаграмі, що міститься у файлі Путівки (<https://cutt.ly/VECOyY3>), та поясніть їх.

За даними діаграми, зображеної на малюнку 8.12, можна зробити такі висновки:

- на початку туристичного сезону, у квітні, найбільшою була кількість проданих путівок до м. Хургада і м. Шарм-ель-Шейх (Єгипет), а найменшою — до м. Херсон та м. Бердянськ (Україна);
- у червні ситуація дещо змінилася — найбільше було продано путівок до м. Кемер і м. Анталія (Туреччина), найменше — також до м. Бердянськ (Україна);
- путівки до м. Кемер стають популярнішими поступово від квітня (45) до червня (230);
- навпаки, путівки до м. Хургада більше продаються у квітні (112), ніж у червні (68);
- у жодному з місяців, що аналізувалися, не вдалося продати понад 230 путівок в одне місто, найменша кількість проданих путівок за всі місяці — 3.

8.5. Який тип діаграми обрати?

За допомогою табличного процесора можна будувати діаграми різних типів. Кожний тип містить кілька видів діаграм — плоскі та об'ємні, звичайні, з накопиченням чи нормовані з накопиченням. Наприклад, стовпчаста діаграма може бути одного з видів, зображених на малюнку 8.13 (с. 88).



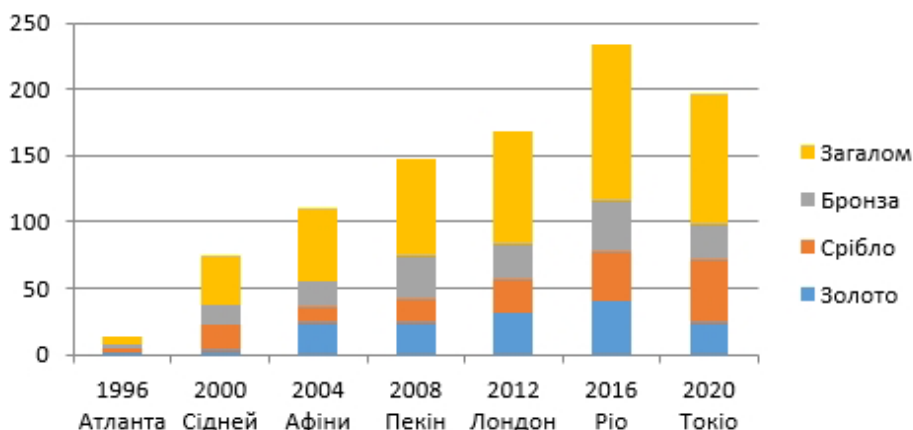
Мал. 8.13

Перш ніж обрати тип діаграми, слід виділити джерело даних — діапазон клітинок таблиці, що містить дані, на основі яких буде створюватися діаграма. Для побудови діаграми джерело даних має містити числові дані. Назви типів діаграм у різних табличних процесорах можуть дещо відрізнятися — подаватися синонімами, але їх призначення є однаковим.

Різні типи діаграм мають свої особливості щодо відображення даних. Для правильного вибору типу діаграм слід розуміти призначення кожного з них.

Стовпчаста діаграма (вертикальна гістограма) (мал. 8.14) демонструє зміну даних за певний період часу й ілюструє співвідношення окремих значень даних. Категорії розташовуються вздовж горизонталі, а значення — вздовж вертикалі. Таким чином, приділяється більша увага змінам у часі. За допомогою гістограми з накопиченням демонструється вклад окремих елементів у загальну суму.

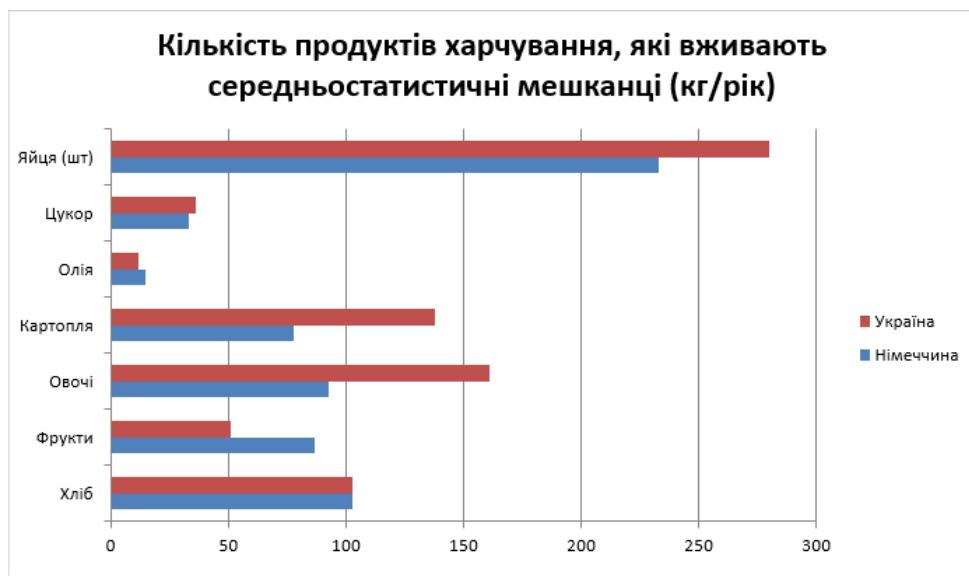
Участь параолімпійської збірної України



Мал. 8.14

Наприклад, за діаграмою з накопиченням на малюнку 8.14 можна зробити висновок, що найбільш успішною була участь української команди у Ріо в 2016 р.

Лінійчата діаграма (горизонтальна гістограма) (мал. 8.15) відображає співвідношення окремих компонентів.



■ Мал. 8.15 ■

Категорії розташовані вздовж горизонталі, а значення — вздовж вертикалі. Таким чином, приділяється більша увага співставленню значень, і менша — змінам у часі. Наприклад, на діаграмі на малюнку 8.15 відображено дані по двом країнам, середньостатистичні мешканці яких вживають певну кількість продуктів на рік. На діаграмі можна побачити, що українці більше за німців вживають овочів, картоплі та яєць, але менше фруктів.

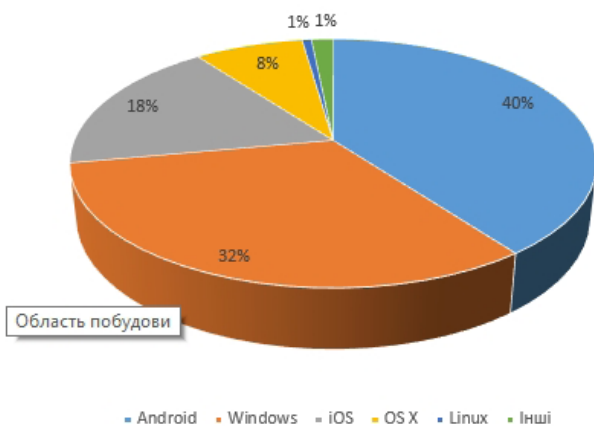
Лінійчата діаграма з накопиченням відображає внесок окремих елементів у загальну суму.

ЦІКАВО

В Таблицях Google також можна створювати діаграми. Для цього натиснуть *Вставити > Діаграма*, щоб створити свою діаграму та відкрити інструмент *Редактор діаграм*. За замовчуванням базова лінійна діаграма створюється з використанням заданих даних, а інструмент *Редактор діаграм* відкривається праворуч, щоб дозволити його подальше налаштування.

За допомогою кругової, або секторної, діаграми (мал. 8.16) ілюструють як абсолютну величину кожного елемента ряду даних, так і його внесок у загальну суму.

Популярність різних ОС у світі (квітень, 2020)

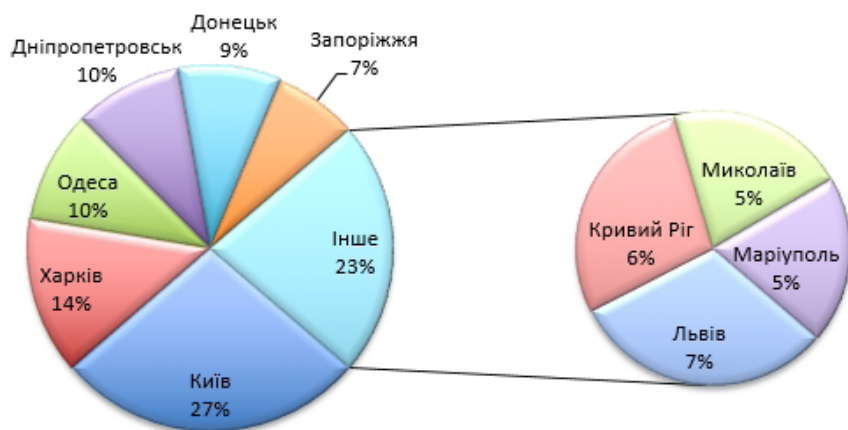


Мал. 8.16

На **круговій діаграмі** може бути представлений лише один ряд даних. Таку діаграму рекомендується використовувати, коли необхідно підкреслити деякий значний елемент. Наприклад, на малюнку 8.16 наведено діаграму щодо популярності різних операційних систем у світі. Із цієї діаграми можна побачити, що найбільш популярною зараз є операційна система *Android*, за нею йде операційна система *Windows*.

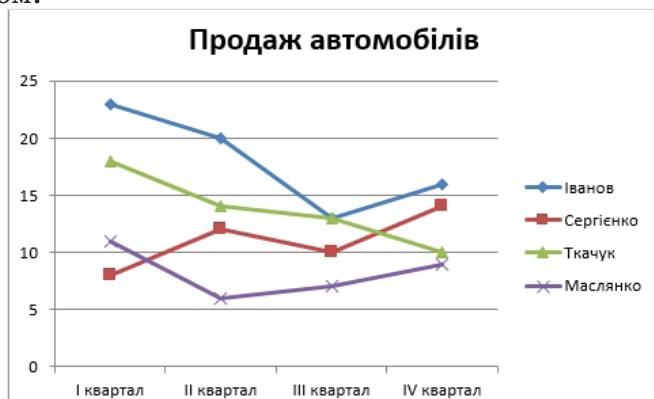
Для полегшення роботи з маленькими частками в основній діаграмі їх можна об'єднати в єдиний елемент на круговій діаграмі, а потім виділити в окрему діаграму поруч із основною (мал. 8.17).

Кількість населення найбільших міст України



Мал. 8.17

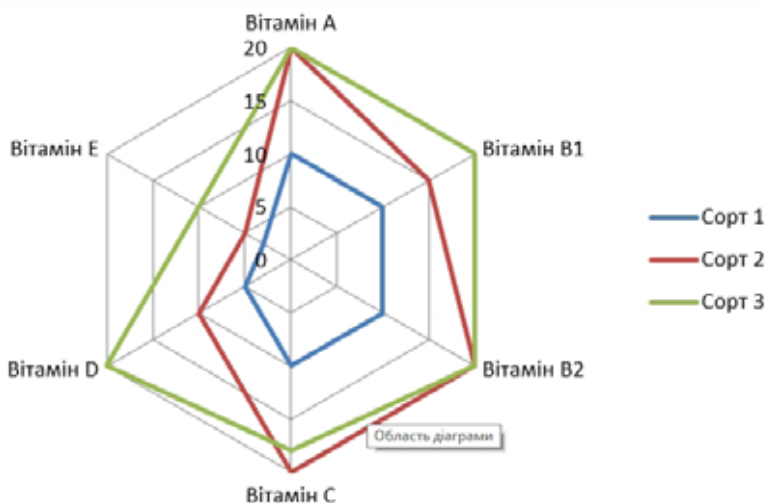
На **графіку** — діаграмі типу *Лінія* — точки, що відповідають даним, з'єднуються лініями (мал. 8.18). На діаграмах такого типу відображаються тенденції зміни даних за однакові проміжки часу. Наприклад, з діаграми на малюнку 8.18 видно, що продажі автомобілів консультантом Ткачуком зменшувалися протягом року, особливо у 2-му кварталі, а інші консультанти збільшили продажі в 4-му кварталі в порівнянні з 3-м кварталом.



Мал. 8.18

У **пелюстковій**, або **сітчастій**, діаграмі (мал. 8.19) кожна категорія має власну вісь координат, що виходить із початку координат. Лініями з'єднуються всі значення з певної серії.

За допомогою пелюсткової діаграми можна порівняти загальні значення з кількох наборів даних. На цій діаграмі ряд даних, що охоплює найбільшу площу (сорт 3), характеризує сорт овочів з найбільшою кількістю вітамінів, а найменшу площу (сорт 1) — з найменшою кількістю.



Мал. 8.19



ДІМО

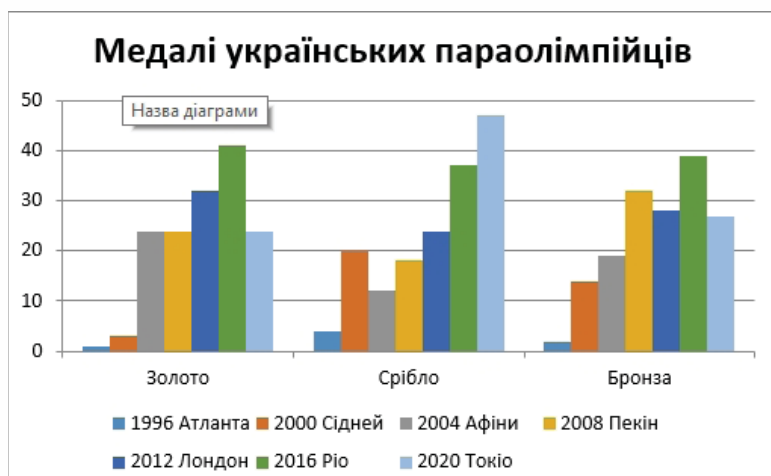
Вправа 5. Аналіз даних, відображених на діаграмах різних типів, побудованих на основі одних і тих само даних.

Завдання. Проаналізуйте дані, відображені на діаграмах різних типів, побудованих для даних таблиці *Олімпійські ігри* (мал. 8.20).

	A	B	C	D	E	F
1	Результати участі в лініх олімпіадах параолімпійської збірної України					
2						
3	Олімпіада	Золото	Срібло	Бронза	Загалом	Місце
4	1996 Атланта	1	4	2	7	44
5	2000 Сідней	3	20	14	37	35
6	2004 Афіни	24	12	19	55	6
7	2008 Пекін	24	18	32	74	4
8	2012 Лондон	32	24	28	84	4
9	2016 Ріо	41	37	39	117	3
10	2020 Токіо	24	47	27	98	6

Мал. 8.20

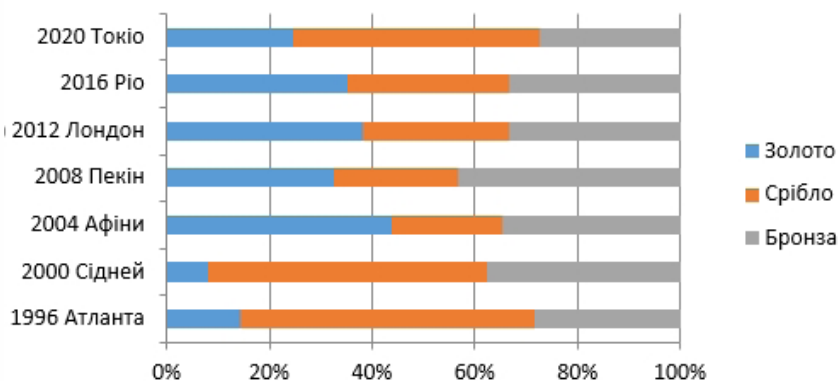
На малюнку 8.21 побудовано **стовпчасту діаграму** — вертикальну гістограму, на якій показано співвідношення золотих, срібних та бронзових медалей, що вибороли спортсмени та спортсменки на кожній олімпіаді. Аналізуючи дані за допомогою цієї діаграми, можна зробити висновок, що найбільше срібних медалей параолімпійська збірна виборола в Токіо, найбільше золотих і бронзових — в Ріо, однакову кількість золотих медалей — в Афінах і Пекіні.



Мал. 8.21

На малюнку 8.22 зображено **нормовану лінійчату діаграму**, або **горизонтальну гістограму**.

Результати виступу параолімпійців на літніх олімпіадах

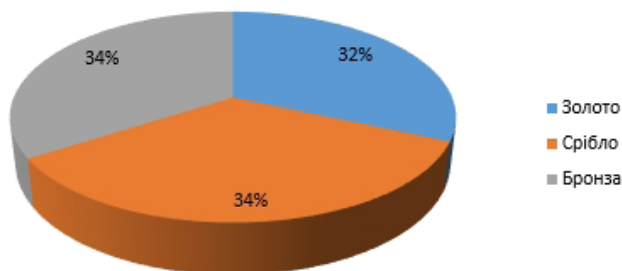


Мал. 8.22

На нормованій діаграмі загальна сума значень по кожному ряду даних відображається як 100 %. Із цієї діаграми можна побачити, що найбільша кількість усіх медалей на всіх олімпіадах — це срібні, на олімпіадах в Атланті та в Сідней вони становлять більше половини всіх медалей. Також видно, що на олімпіаді в Афінах більше 40 % усіх медалей — золоті, а на олімпіадах в Пекіні та Ріо спортсмени та спортсменки виборювали більше золотих медалей, ніж срібних.

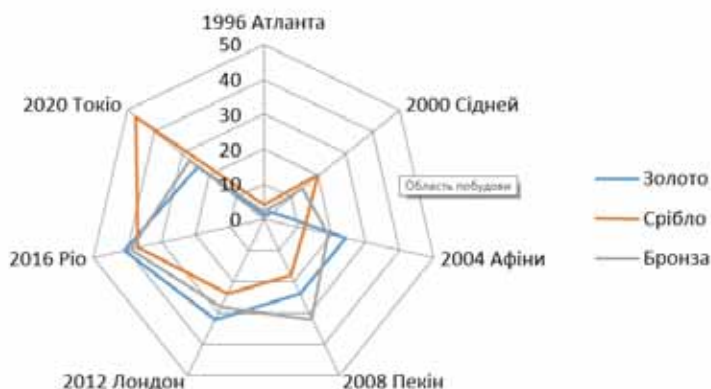
Малюнок 8.23 містить **кругову**, або **секторну**, діаграму, на якій демонструється частка медалей кожного виду в загальній сумі.

Кількість медалей параолімпійської збірної на олімпіадах з 1996 по 2020 рр.



Мал. 8.23

Зображена на малюнку 8.24 **пелюсткова**, або **сітчаста**, діаграма відображає тенденцію кількості медалей різних видів на літніх олімпіадах з 1996 по 2020 р., кожній проведеній олімпіаді відповідає вісь, на якій позначено конкретне числове значення для кожного виду медалей.



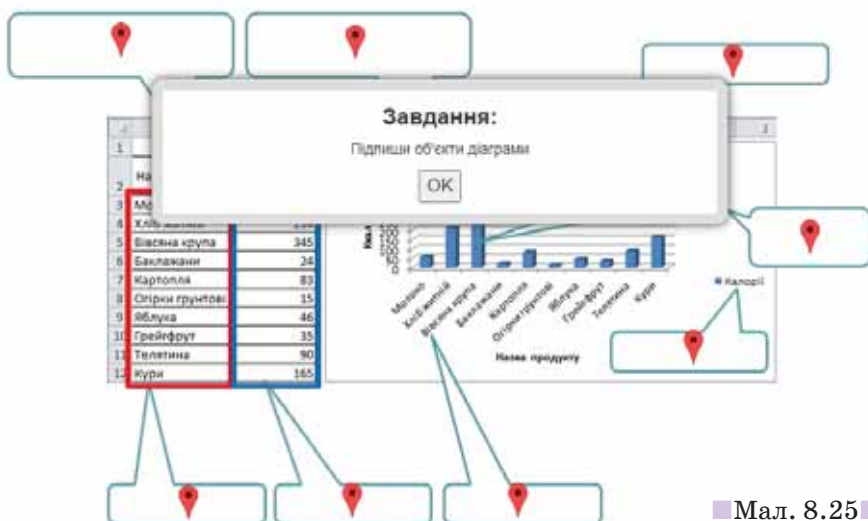
■ Мал. 8.24 ■

Бачимо, що найбільше на олімпіадах у Токіо, Атланті та Сідней було здобуто бронзових медалей. Водночас золото переважало в Афінах, Лондоні та Ріо.



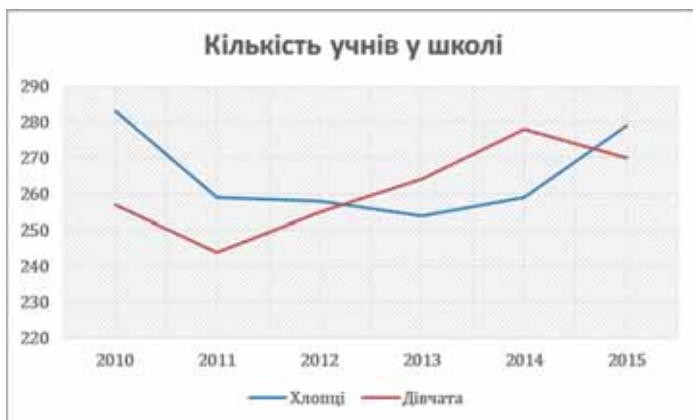
ПРАЦЮЄМО В ПАРАХ

1. Розгляньте об'єкти діаграми та таблиці-джерела її побудови. По черзі називайте об'єкти. Перевірте себе у вправі за посиланням <https://learningapps.org/view2168977> (мал. 8.25).



■ Мал. 8.25 ■

2. Обговоріть, що означає, що діаграма зв'язана з відповідними даними аркуша електронної таблиці, на основі яких вона створена. Наведіть приклади. Обговоріть у парі.
3. Проаналізуйте, діаграми яких типів використовуються у ваших підручниках з географії, історії, біології. Побудуйте відповідну таблицю частотності використання діаграм певного типу. Проаналізуйте й обговоріть у парі, для відображення яких даних використовуються в підручниках діаграми.
4. Запитайте в батьків або знайомих, який тип діаграм вони використовують найчастіше. Обговоріть причини та проведіть узагальнення.
5. За ключовим словом графік знайдіть в інтернеті різні зображення. Обговоріть аргументи щодо правильності добору відповідних типів діаграм. Одним реченням опишіть призначення знайденої діаграми та вкажіть тип діаграми, яким можна замінити знайдений.
6. Обговоріть, які складнощі можуть виникнути при читанні діаграми, якщо діаграма не містить: назви діаграми; легенди; підписів даних.
7. Для оформлення шкільної газети ви вирішили побудувати діаграми. Обговоріть, який тип діаграми використати для даних, що відображають: 1) кількість звернень учнями школи до шкільної бібліотеки; 2) показники зміни висоти рослини протягом тижня в досліді про вплив світла на ріст рослин; 3) відсоток зайнятості учнів у роботі шкільних гуртків; 4) дані про середній час використання інтернету учнями різних класів; 5) тенденцію зміни ставлення учнів різних класів до екологічних проблем до перегляду виступу шкільного гурту «За чисте довкілля» та після нього; 6) внесок кожного класу в загальну суму коштів, зібраних під час добродійного ярмарку.
8. Поставте одне одному запитання за даними діаграми (мал. 8.26). Оцініть відповідь кожного на поставлені запитання.



Мал. 8.26



ПРАЦЮЄМО САМОСТІЙНО

9. Перегляньте діаграму, зображену на малюнку 8.27. Проаналізуйте діаграму за такою схемою:

- Чому виділені комікси вважаються найуспішнішими?
- На виробництво якого з коміксів було витрачено за бюджетом найбільшу кількість грошей; найменшу кількість?
- Чи відповідає дійсності гіпотеза (аргументуй свою думку, спираючись на діаграму):

- найновітніші комікси отримали найбільший бюджет;
- бюджет, витрачений на виробництво коміксів, збільшується кожного року;
- з кожним роком збори від екранізації коміксів збільшуються;
- бюджет на виготовлення другої версії найуспішніших коміксів завжди більший (менший);
- збори від екранізації других версій найуспішніших коміксів завжди більші (менші).

г) У скільки разів грошові збори від екранізації коміксів перевищують витрачений на них бюджет?

д) Від екранізації якого коміксу отримано найбільші грошові збори?

10. За даними діаграми (мал. 8.27) створіть у табличному процесорі таблицю із заголовком *Найуспішніші екранізації коміксів*, яку збережіть у файлі *Комікси* в папці *Електронні таблиці* власної структури папок. За даними створеної таблиці побудуйте діаграму, як на малюнку 8.27, розмістіть її на окремому аркуші книги *Комікси* та збережіть результати роботи.



11. Відомо, що повітря має такий хімічний склад: азот — 78,08 %, кисень — 20,94 %, інертні гази — 0,94 %, діоксид вуглецю — 0,04 %. Зобразіть графічно дані про хімічний склад повітря, обери́ть доцільний тип і вид діаграми.

12. У таблиці наведено відомості про рівень зайнятості (% від загальної кількості) населення відповідної вікової групи за місцем проживання.

Вікова група	15–24	25–29	30–39	40–49	50–59	60–70
Міське населення	34,8	77,7	82,4	80,4	59,2	13,7
Сільське населення	43,1	72,7	79,3	78,6	65,5	37,3

Побудуйте діаграму, за якою можна відслідкувати тенденцію зміни частки зайнятого населення відповідної вікової групи за місцем проживання.

13. Створіть алгоритм для друкування трьох копій електронної таблиці, подайте його графічно.



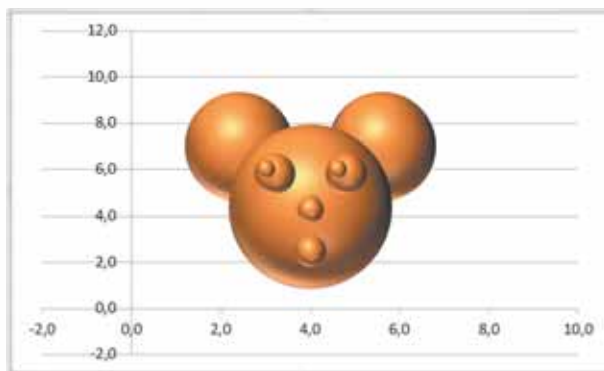
ДОСЛІДЖУЄМО

14. Дослідіть, які властивості для ряду даних бульбашкової діаграми в табличному процесорі *Microsoft Excel* можна змінювати. Для цього відкрийте файл *Бульбашкова діаграма*, що зберігається в папці *Електронні таблиці* (<https://cutt.ly/IECOIPP>), виділіть лише числові дані для побудови діаграми (без заголовків рядків і стовпчиків), побудуйте бульбашкову об'ємну діаграму. Встановіть формат ряду даних: значенням відповідає: площа бульбашок, масштаб бульбашок: 300. За потреби оберіть на вкладці *Конструк-*

тор інструмент *Перехід рядок/стовпець*



, щоб отримати результат, як на малюнку 8.28.



Мал. 8.28

ЯК ВИКОРИСТОВУВАТИ ЕЛЕКТРОННУ ТАБЛИЦЮ ДЛЯ ОПРАЦЮВАННЯ ОДНОТИПНИХ ДАНИХ?



ПРИГАДАЙТЕ:

- дані яких типів можуть містити клітинки електронної таблиці;
- як обчислити суму, максимальне, мінімальне та середнє значення для діапазону клітинок електронної таблиці.

ВИ ДІЗНАЄТЕСЯ:

- як впорядковувати дані в електронній таблиці;
- як і для чого в електронних таблицях використовують фільтри;
- за якими правилами створюють умови для розширеного фільтра;
- що таке проміжні підсумки та як ними користуватися.



ВИВЧАЄМО

9.1. Як впорядковувати дані в електронній таблиці?

Для аналізу та пошуку потрібних даних у таблицях, які містять їх велику кількість, використовують вбудовані засоби, одним з яких є **впорядкування**, або **сортування** даних.



Сортування — це розміщення у визначеному порядку елементів даних.

Упорядкування будь-яких даних застосовується для прискорення пошуку. Наприклад, якщо дані в стовпці, що містить прізвища, впорядковано за алфавітом, то в ньому шукати дані про людину з конкретним прізвищем легше, ніж у невпорядкованому списку. Але коли список даних достатньо великий, наприклад, у ньому багато родичів або тих, хто має однакові прізвища, відшукати відомості складніше.

Дані зв'язного діапазону в табличному процесорі можна впорядковувати за значенням вмісту одного або кількох стовпців за зростанням або за спаданням. У процесі впорядкування будуть змінювати своє місце в таблиці не лише клітинки того стовпця, за яким відбувається впорядкування, а й переставлятимуться записи, що містять дані про кожний об'єкт таблиці.

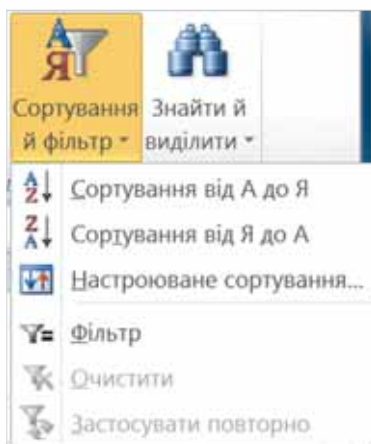
Клітинки з даними у стовпці називають полем, а рядки — записами. Кожний запис містить дані про один об'єкт таблиці, наприклад, про землетрус: коли він стався, у якому місті й країні це відбулося та до якої кількості жертв призвело. Верхній рядок таблиці при цьому містить назви полів (мал. 9.1).

Землетруси			
Дата	Місцевість	Країна	Кількість жертв
1138	Алеппо	Сирія	230000
1970	Анкаш	Перу	66000
1964	Анкоридж	США	131
893	Ардебіль	Іран	150000
856	Дамган	Іран	200000
1737	Калькутта	Індія	300000

Мал. 9.1

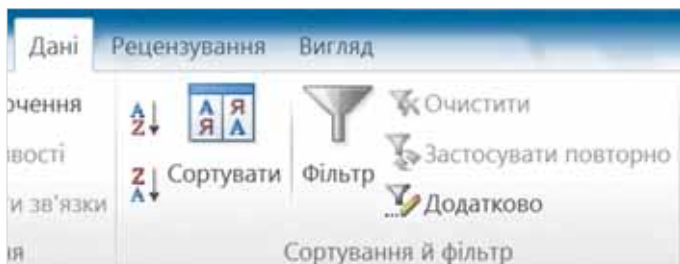
Процес впорядкування відбувається у два етапи: виділення поля таблиці для впорядкування та безпосередньо впорядкування. Під час роботи з даними всієї електронної таблиці або з окремими її полями необов'язково виділяти всю таблицю чи весь стовпець. Достатньо виділити будь-яку клітинку поля, за значеннями якого насамперед впорядковуватимуться дані всієї таблиці. Далі в табличному процесорі *Microsoft Excel* слід скористатись одним зі способів доступу до інструментів впорядкування:

1) на вкладці *Основне* в групі *Редагування* відкрити список інструмента *Сортування й фільтр* (мал. 9.2);



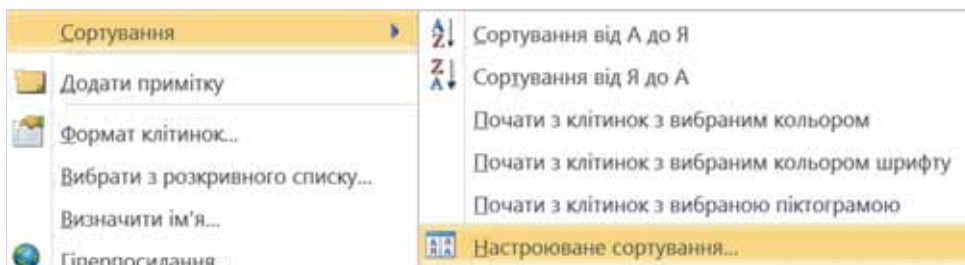
Мал. 9.2

2) відповідні інструменти розміщені також на вкладці *Дані* в групі *Сортування й фільтр* (мал. 9.3);



■ Мал. 9.3 ■

3) у контекстному меню, що з'являється, якщо клацнути на будь-якій клітинці поля, за яким необхідно впорядкувати дані, обрати вказівку *Сортування* (мал. 9.4).



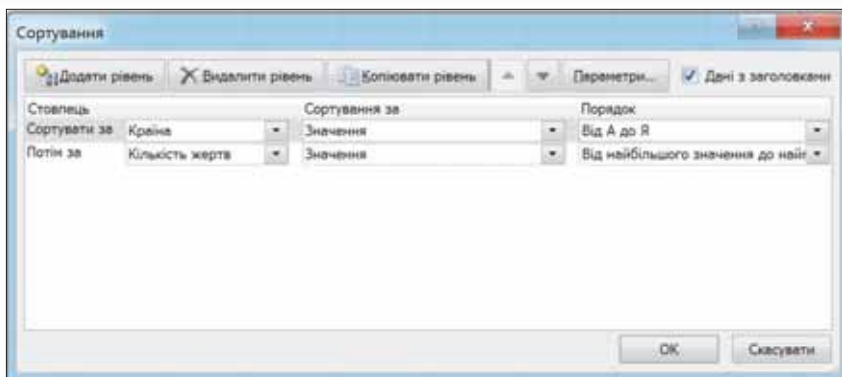
■ Мал. 9.4 ■

Потім слід скористатись одним з інструментів сортування за одним полем за зростанням чи спаданням, або обрати *Настроюване сортування*, якщо слід впорядкувати дані таблиці за кількома полями одночасно.

Інструменти  та  змінюють свою назву залежно від типу значень у полі, за яким відбувається впорядкування:

- текстові значення — *Сортування від А до Я* та *Сортування від Я до А*, при цьому впорядкування здійснюється за алфавітом або в оберненому порядку;
- числові значення — *Сортування від найменшого до найбільшого* та *Сортування від найбільшого до найменшого*;
- значення типу дата й час — *Сортування від наймолодшого до найстаршого* та *Сортування від найстаршого до наймолодшого*.

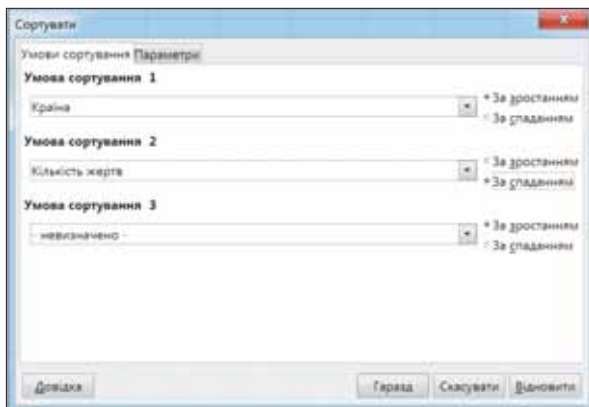
Якщо обрати вказівку *Настроюване сортування*, відкривається вікно *Сортування* (мал. 9.5, с. 101), у якому користувач може послідовно додати кілька полів для впорядкування за допомогою кнопки *Додати рівень*.



■ Мал. 9.5 ■

Вибрати поля для впорядкування можна за допомогою списків у вікні *Сортування*. Додатково для кожного поля, за яким відбудуватиметься впорядкування, слід зазначити порядок впорядкування — за зростанням чи за спаданням. Упорядкування за кількома полями використовують, якщо одне з полів містить групи однакових значень, тоді в межах кожної такої групи дані будуть сортуватися за другим полем. Наприклад, якщо в таблиці про землетруси виконати впорядкування за двома полями — *Країна* та *Кількість жертв*, то для кожної групи записів, для яких назва країни однакова, дані будуть відсортовані за кількістю жертв.

У табличному процесорі *LibreOffice Calc* упорядкування за одним полем здійснюється за допомогою інструментів *Сортування за зростанням*  або *Сортування за спаданням*  з панелі інструментів *Стандартна*. Для впорядкування за кількома полями одночасно використовують вказівку меню *Дані/Сортування*   або інструмент *Сортування* на панелі інструментів. При цьому відкривається вікно *Сортувати*, у якому можна обрати назви до трьох полів та вказати порядок сортування за кожним з них (мал. 9.6).



■ Мал. 9.6 ■



ДІМО

Вправа 1. Упорядкування даних списку.

Завдання. У таблиці *Землетруси*, що зберігається в папці *Електронні таблиці*, впорядкуйте дані за датою за зростанням; потім за кількістю жертв; за спаданням; за двома полями: за країною в алфавітному порядку та за кількістю жертв за спаданням. Дайте відповіді на поставленню запитання.

1. Завантажте файл *Землетруси*, що зберігається в папці *Електронні таблиці* (<https://cutt.ly/3EViHOv>). Назвіть імена полів таблиці.
2. Виділіть будь-яку клітинку поля *Дата* й оберіть інструмент *Сортування за зростанням*. Дайте відповіді на запитання: Які потужні землетруси зафіксовано до н. е., які — у XVIII ст., а які — у XX ст.?
3. Виділіть будь-яку клітинку поля *Кількість жертв* і оберіть інструмент *Сортування за спаданням* (мал. 9.7). Дайте відповідь на запитання: У яких країнах потужні землетруси спричинили найбільшу кількість жертв?

	A	B	C	D
1	Землетруси			
2				
3	Дата	Місцевість	Країна	Кількість жертв
4	1556	Шаньсі	Китай	800000
5	1976	Тянь-Шань	Китай	655000
6	1737	Калькутта	Індія	300000
7	1138	Алеппо	Сирія	230000
8	856	Дамган	Іран	200000

■ Мал. 9.7 ■

4. Виділіть будь-яку клітинку поля *Країна* та оберіть інструмент *Настроюване сортування (Сортування)*. У вікні *Сортування* в першому списку вже встановлено ім'я потрібного поля, натисніть кнопку *Додати рівень* й у другому списку виберіть поле *Кількість жертв*. Оберіть для нього порядок сортування *Від найбільшого до найменшого (За спаданням)*. Підтвердьте виконання операції, натиснувши кнопку *ОК (Гаразд)*. Проаналізуйте результати й дайте відповіді на запитання:
 1. У яких країнах відбувалося кілька землетрусів?
 2. Який землетрус призвів до найбільшої кількості жертв у Китаї?
 3. Який землетрус призвів до найбільшої кількості жертв у Японії?
5. Збережіть файл із тим самим іменем у папці *Табличний процесор* своєї структури папок.

9.2. Як і для чого в електронних таблицях використовують фільтри?

Крім упорядкування даних, для швидкого пошуку в таблиці даних, що відповідають деяким умовам, використовують фільтри.



Фільтри є засобом швидкого виділення зі списку набору даних, які відповідають заданим умовам.

Після застосування фільтра залишаються відображеними на екрані тільки записи, що відповідають заданим умовам, інші рядки приховуються. **Фільтрування** — процес застосування до електронної таблиці правил добору даних для відображення на екрані. Умови фільтрування можуть бути простими та складеними. Після опрацювання результатів фільтрування можна відобразити на екрані усі дані вихідної таблиці.

Під час роботи з табличним процесором для фільтрування даних можна скористатися засобами *Автофільтр* і *Розширений фільтр*.

Автофільтр використовують у випадках, коли необхідно вибрати зі списку дані за певними значеннями клітинок або сформулювати умови, які підтримує цей засіб. Автофільтр можна застосувати послідовно до кількох полів, проте умови, що створюються за різними полями, будуть пов'язуватися лише логічною операцією *І*. Складена умова за одним полем може бути створена з використанням логічних операцій *І* чи *АБО*, але кількість умов, які можна поєднати, є обмеженою. Для використання всіх інструментів для роботи з фільтрами в *Microsoft Excel* обирають вкладку *Дані* в групі *Сортування й фільтр* (мал. 9.3). Щоб скористатись *Автофільтром*, необхідно виділити будь-яку клітинку таблиці, яку по-

трібно фільтрувати, та обрати інструмент *Фільтр* . Після включення режиму *Автофільтр* (він являє собою вбудований засіб відбору) на екрані в усіх полях таблиці відображаються кнопки списків, що розкриваються (мал. 9.8).

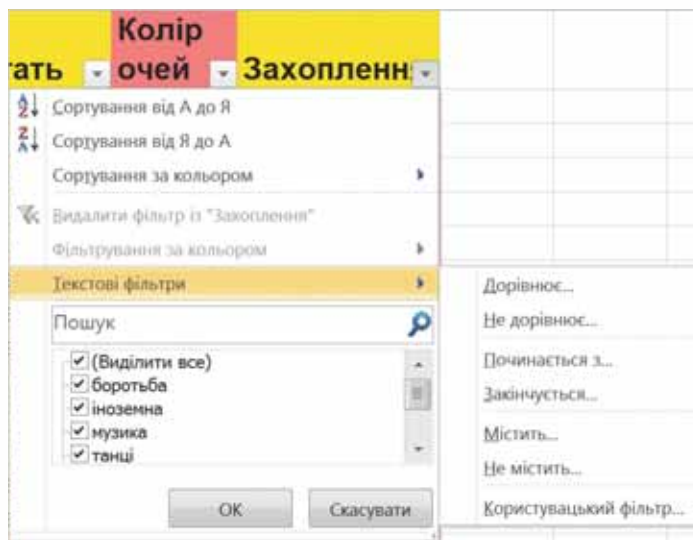
Списки, що розкриваються

А	В	С	Д	Е	F	Г	Н
Прізвище	Ім'я	Дата народження	Вік	Зріс	Стать	Колір очей	Захопленн
Павленко	Олена	09.05.2002	14	118	дівчинка	голубі	танці
Палажченко	Клим	03.04.2003	13	131	хлопчик	карі	футбол
Петрененко	Володимир	10.01.2004	12	134	хлопчик	голубі	боротьба

Мал. 9.8

При натисненні кнопки зі стрілкою в кожному полі розкривається список можливих значень та вказівки для впорядкування даних у цьо-

му полі й формування умов фільтрування (мал. 9.9). Залежно від типу даних у кожному полі вказівки для створення умов фільтрування можуть відрізнятися. За допомогою текстових фільтрів можна створювати умови для пошуку текстових значень, що починаються чи закінчуються на певний символ чи набір символів, містять чи не містять деякий текстовий фрагмент (мал. 9.9).



Мал. 9.9

Числові фільтри використовують для створення умов з операторами порівняння — більше, менше тощо, а також для знаходження значень, що більші чи менші середнього, відображення вказаної кількості найбільших або найменших значень тощо.

За допомогою фільтрів дат можна знайти дані, дата яких — перед або після заданої дати, міститься в заданому діапазоні тощо.

Для вибору даних, що збігаються з певним значенням, слід у списку унікальних значень позначити прапорцями значення, які необхідно шукати (мал. 9.9). За замовчуванням обрано режим *Виділити все* — його можна вимкнути й обирати лише потрібні значення. Якщо потрібно знайти записи, які задовольняють складеній умові, що складається з двох простих умов для значень одного поля текстового або числового типу, то необхідно вибрати зі списку фільтрів *Користувачський фільтр*. На екрані відображатиметься вікно *Користувачський автофільтр*, у якому можна зазначити:

- одну або дві прості умови;
- для кожної простої умови вибрати операцію (дорівнює, більше, менше тощо) та значення для порівняння;
- логічну операцію, яка використовується для складеної умови з визначених простих, — *І*, *АБО* (мал. 9.10).

Користувачський автофільтр

Показати лише ті рядки, значення яких:

Прізвище

починається з П

закінчується на ко

Знак питання "?" позначає один будь-який знак.
Символ "*" позначає послідовність будь-яких символів.

OK Скасувати

■ Мал. 9.10 ■

Результати відбору відразу відображаються на екрані, що визначається за трьома ознаками (мал. 9.11):

Зміна позначки

	A	B	C	D	E	F	G	H
1	Прізвище	Ім'я	Дата народження	Вік	Зріс	Стать	Колір очей	Захопленн
58	Павленко	Олена	09.05.2002	14	118	дівчинка	голубі	танці
59	Палажченко	Клим	03.04.2003	13	131	хлопчик	карі	футбол
60	Петренко	Володимир	10.01.2004	12	134	хлопчик	голубі	боротьба
61	Петренко	Наталія	17.05.2004	12	156	дівчинка	сірі	боротьба
68	Поноженко	Лідія	12.10.2003	13	119	дівчинка	голубі	танці
69	Приходько	Станіслав	08.10.2003	13	134	хлопчик	голубі	боротьба
70	Приходько	Тарас	09.01.2001	15	156	хлопчик	зелені	футбол

Номери рядків

Повідомлення про знайдену кількість записів

■ Мал. 9.11 ■

- номерами рядків записів, що увійшли до відфільтрованих даних — деякі номери пропускаються, а номери решти відображаються синім кольором;
- вмістом рядка стану — відображається повідомлення про знайдену кількість записів після виконання фільтрації;
- зміною позначки  на кнопці в полі, за значеннями якого здійснювався добір даних.

Для відображення всіх даних списку на екрані необхідно обрати інструмент *Очистити* в групі *Сортування й фільтрація* на вкладці *Дані*. Таким чином, алгоритм використання засобу *Автофільтр* можна сформулювати так:

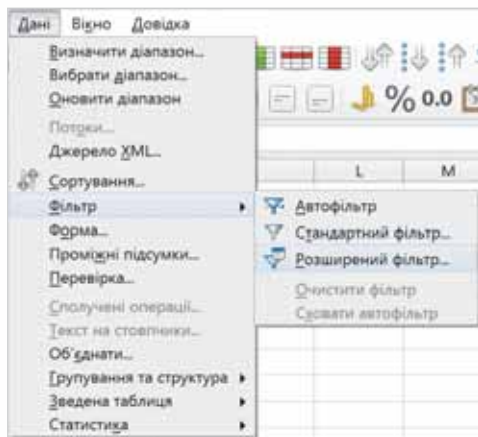
1. Виділити будь-яку клітинку таблиці, що містить дані.
2. Викликати засіб *Автофільтр* одним зі способів.

3. Сформувані умови пошуку даних за допомогою вбудованих засобів. Для створення складених умов скористатися вказівкою *Користувачський фільтр*.

4. Проаналізувати відображені дані.

5. Після аналізу даних скасувати дію фільтра — використати інструмент *Очистити*.

У *LibreOffice Calc* для роботи з фільтрами використовують вказівки в меню *Дані/Фільтр* (мал. 9.12).



Мал. 9.12

Автофільтр можна також викликати за допомогою однойменного ін-

струмента на  панелі інструментів *Стандартна*.

У *LibreOffice Calc* засоби створення умов під час використання автофільтру є однаковими для даних усіх типів, за їхньою допомогою можна обирати значення, що збігаються із заданими значеннями в обраному полі, впорядковувати дані, знаходити записи, що містять порожні або непорожні значення, відображати на екрані вказану кількість найбільших чи найменших значень за допомогою вказівки *10 перших* (мал. 9.13). Для створення більш складних умов можна обрати в цьому списку або в меню *Дані/Фільтр* вказівку *Стандартний фільтр*. Такі умови можна формувати у вікні *Стандартний фільтр* (мал. 9.14), що дає змогу поєднувати до чотирьох простих умов.

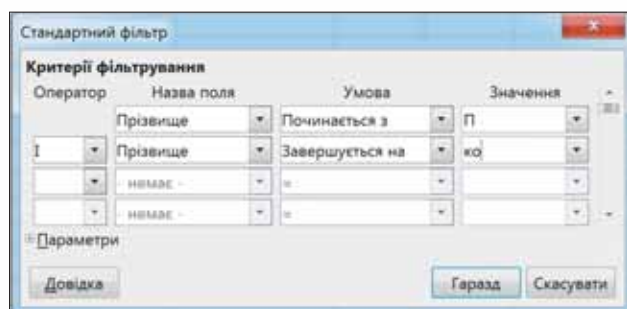


ДІЄМО

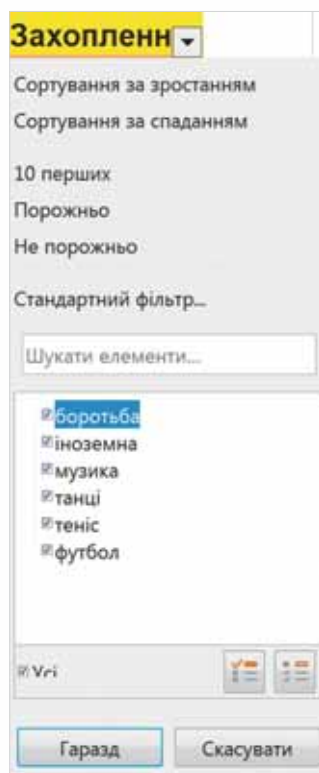
Вправа 2. Використання умови при застосуванні Автофільтра.

Завдання. За допомогою засобу *Автофільтр* знайдіть в електронній таблиці, що зберігається у файлі *Школа*, кількість і список хлопців, прізвища яких починаються з *П* й закінчуються на *-ко* та які займаються тенісом.

1. Відкрийте файл *Школа*, який зберігається в папці *Електронні таблиці* (<https://cutt.ly/nEVovYN>).
2. Виділіть будь-яку клітинку таблиці, що містить дані.
3. На вкладці *Дані* в групі *Сортування й фільтрація* оберіть інструмент *Фільтр*.
4. У списку в полі *Прізвище* оберіть вказівку *Текстові фільтри/ Користувацький фільтр*.
5. У вікні *Користувацький автофільтр* із переліку операцій оберіть *Починається з* (мал. 9.10, 9. 14), а поруч до поля введення введіть символ *П*.



Мал. 9.13



Мал. 9.14

Оберіть логічний операторі *I* для створення складеної умови та запишіть другу умову — *Закінчується на -ко*. Натисніть кнопку *ОК* (*Гаразд*). На екрані відобразатиметься відфільтрований список.

6. У списку в полі *Стать* зі списку значень виберіть *хлопчик*, а потім у списку в полі *Захоплення* виберіть *музика*.
7. Переконайтеся, що на екрані відображені записи, які відповідають умові завдання. Визначте їхню кількість.
8. Для відображення відомостей про всіх учнів на екрані виконайте вказівку *Дані/Фільтр/Відобразити всі*.

9.3. За якими правилами створюють умови для розширеного фільтра?

Розширений фільтр використовується за необхідності сформувати складні умови пошуку, які не підтримуються засобом *Автофільтр*. Це може бути у випадках, якщо умови за різними полями слід поєднати логічною операцією *АБО*, чи якщо для одного поля слід скласти складену умову, яка містить більше простих умов, ніж підтримує *Користувацький автофільтр* (дві для *Microsoft Excel* та чотири для *LibreOffice Calc*).

На відміну від автофільтра, який спочатку викликають, а потім формують умови фільтрування за допомогою вбудованих засобів, перш ніж викликати розширений фільтр, користувач має в деякому діапазоні таблиці сформулювати умови фільтрування за такими правилами:

1. Для того щоб відфільтровувати дані таблиці за допомогою розширеного фільтра, необхідно, щоб поля таблиці мали заголовки, тобто стовпці мали назви.

2. Між діапазоном, у якому будуть записуватись умови, і таблицею з даними має бути хоча б один порожній рядок або один порожній стовпець.

3. Слід скопіювати з таблиці заголовки стовпців, у яких здійснюється пошук, та вставити їх до першого порожнього рядка діапазону умов відбору. Діапазон умов не обов'язково має містити назви всіх полів таблиці з даними.

4. Ввести в рядки під заголовками діапазону умов необхідні умови відбору.

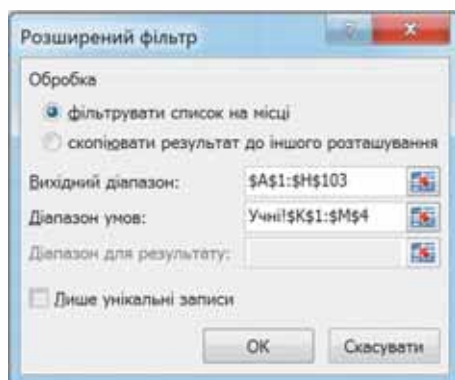
5. Складені умови, що мають бути пов'язані логічною операцією І, мають бути записані в одному рядку. Умови, записані в різних рядках, пов'язуються логічною операцією АБО.

6. Встановити курсор на будь-яку клітинку таблиці з даними.

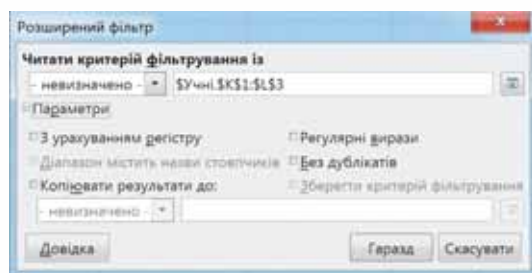
7. Викликати *Розширений фільтр* — у *Microsoft Excel* обрати інструмент  **Додатково** на вкладці *Дані* в групі *Сортування й фільтрація* (в *LibreOffice Calc* виконати вказівку меню *Дані/Фільтр/Розширений фільтр*).

8. Для того, щоб відобразити результати фільтрування, приховавши при цьому непотрібні рядки, слід у вікні *Розширений фільтр* у *Microsoft Excel* встановити перемикач у положення *Фільтрувати список на місці* (мал. 9.15, а), в *LibreOffice Calc* не вмикати прапорець *Копіювати результати до* (мал. 9.15, б).

9. Ввести в полі *Діапазон умов* (*Читати критерій фільтрування із*) посилання на діапазон із записаними умовами відбору, що включає заголовки стовпців.



Мал. 9.15, а



Мал. 9.15, б



ДІЄМО

Вправа 3. Використання розширеного фільтра.

Завдання. За допомогою розширеного фільтра у файлі *Школа* знайдіть записи, що містять відомості про хлопців, що займаються музикою, дівчат, захопленням яких є іноземна мова, а також усіх учнів віком 10 років — незалежно від їхнього захоплення.

1. Відкрийте файл *Учні*, що збережений у папці *Електронні таблиці* (<https://cutt.ly/nEVovYN>).
2. Починаючи з клітинки *K1*, створіть діапазон умов відповідно до завдання, зображено на малюнку 9.16.

K		L	M
Захоплення		Стать	Вік
музика		хлопчик	10
іноземна		дівчинка	

Мал. 9.16

3. Виділіть клітинку *A1*. Оберіть інструмент *Додатково* на вкладці *Дані* в групі *Сортування й фільтрація* (виконайте вказівку *Дані/ Фільтр/ Розширений фільтр*).
4. Переконайтеся, що в діалоговому вікні *Розширений фільтр* (мал. 9.15, а) в області *Вихідний діапазон* вказано *\$A\$1:\$H\$103* (в іншому разі виділіть цей діапазон клітинок).
5. У діалоговому вікні *Розширений фільтр* установіть текстовий курсор в область *Діапазон умов* та виділіть у таблиці діапазон, що містить створені умови: *K1:M4*. Адреса цього діапазону з абсолютними посиланнями та назвою аркуша буде відображена в області *Діапазон умов*. Натисніть кнопку *ОК*.
6. Визначте, скільки записів таблиці відповідає вказаним умовам.
7. Збережіть результати у файлі з іменем *Розширений фільтр* у папці *Електронні таблиці* своєї структури папок.

9.4. Що таке проміжні підсумки та як ними користуватися?

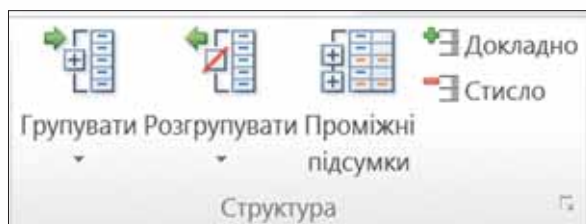
Ви вже вмієте застосовувати стандартні функції для обчислення суми, максимального, мінімального та середнього значень деяких діапазонів клітинок. Якщо таблиця містить дані, для яких за деяким полем можна утворити групи з однаковими значеннями, у середовищі табличного процесора можна скористатися вбудованим засобом для швидкого обчислення підсумкових значень у полях, що містять числові дані, для кожної групи без використання формул. Щоб утворити такі групи, не-

обхідно впорядкувати дані таблиці за тим полем, яке містить однакові значення. Після цього можна скористатися засобом *Проміжні підсумки* для додавання підсумкових значень для вказаних числових полів.

Наприклад, у таблиці, що містить дані про потужні землетруси, за допомогою такого засобу можна швидко обчислити загальну кількість жертв у кожній країні окремо. Якщо дані в полі *Країна* впорядковані, то ми маємо чіткі групи, після кожної з яких можна вставити підсумковий запис.

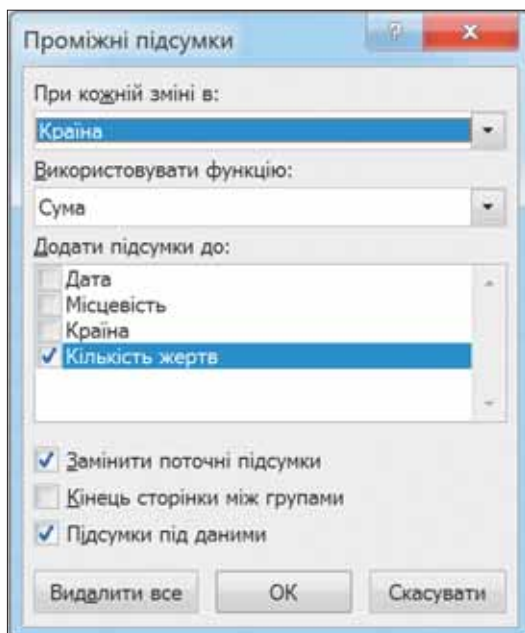
Отже, перша умова для автоматичного вставлення підсумків — це впорядкування даних у тому полі, за яким створюються групи. Після цього слід використати засіб додавання проміжних підсумків. У *Microsoft Excel* для цього необхідно:

- На вкладці *Дані* в групі *Структура* обрати інструмент *Проміжні підсумки* (мал. 9.17).



Мал. 9.17

- У вікні *Проміжні підсумки* (мал. 9.18) зі списку *При кожній зміні в:* обрати поле, за яким відбувалося впорядкування та утворені групи записів з однаковими значеннями.



Мал. 9.18

- У списку *Використовувати функцію* обрати функцію, що повинна використовуватися при обчисленні проміжних підсумків: сума, кількість, середнє значення, максимум, мінімум та ін.
- У полі *Додати підсумки до* вказати поля, за якими мають обчислюватися проміжні підсумки. Обирати слід лише поля, що містять числові дані.
- За потреби вимкнути прапорець *Підсумки під даними*, щоб відобразити рядки з підсумками над відповідними даними.
- Якщо окремі групи записів мають бути розташовані та виведені до друку на різних сторінках, то встановити прапорець параметра *Кінець сторінки між групами*.
- Закрити вікно, натиснувши кнопку *ОК*.

Аналогічно можна додати вкладені проміжні підсумки. Для цього потрібно спочатку впорядкувати дані в таблиці за декількома полями одночасно та послідовно використати інструмент *Проміжні підсумки*, створюючи по черзі підсумки кожного з рівнів. Після вставлення підсумків першого рівня на наступному кроці у вікні *Проміжні підсумки* слід вимкнути прапорець *Замінити поточні підсумки*. Після додавання проміжних підсумків ліворуч від таблиці з'являється структура (мал. 9.19), за допомогою якої можна приховати або відобразити рядки з даними для окремих проміжних підсумків.



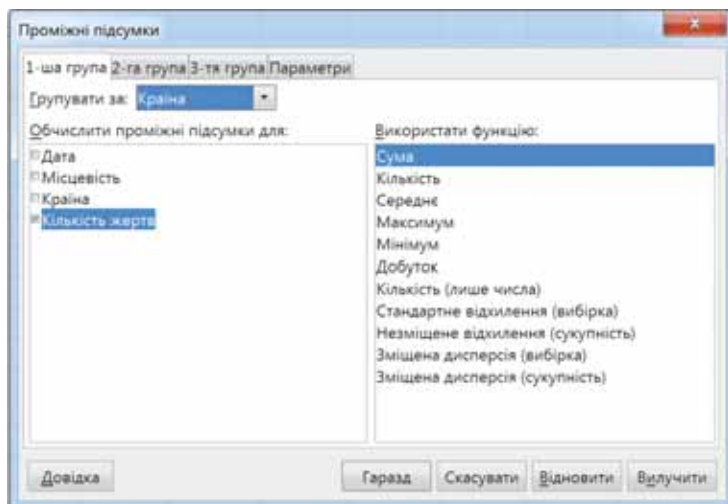
Дата	Місцевість	Країна	Кількість жертв
		Азербайджан Підсумок	80000
		Греція Підсумок	65000
		Індія Підсумок	300000
		Іран Підсумок	350000
		Італія Підсумок	160000
		Китай Підсумок	1855000
		Мексика Підсумок	9500
		Перу Підсумок	66000
		Португалія Підсумок	100000
		Сирія Підсумок	230000
		США Підсумок	3191
		Чилі Підсумок	5700
		Японія Підсумок	179200
		Загальний підсумок	3403591

■ Мал. 9.19 ■

Для цього можна скористатися позначками **-** та **+** для різних рівнів кожної з груп даних. Приховати або відобразити проміжні підсумки певного рівня можна також за допомогою кнопок з номерами рівнів **1 2 3** у верхній частині такої структури:

- 1** — виведення лише загальних підсумків;
- 2** — виведення загальних і проміжних підсумків;
- 3** — і нижче — виведення повного списку.

Для видалення підсумків, а разом з ними і структури, потрібно відкрити діалогове вікно *Проміжні підсумки* та натиснути кнопку *Видалити все*. Щоб замінити поточні підсумки новими, одержуваними за іншою формулою чи для інших полів, слід задати в цьому вікні потрібні параметри та встановити прапорець *Замінити поточні підсумки*. Проміжні підсумки будуть автоматично видалені при пересортуванні списку. При цьому на екран буде виведене попереджувальне повідомлення.



Мал. 9.20

У табличному процесорі *LibreOffice Calc* додавання проміжних підсумків відбувається за таким само алгоритмом. Після впорядкування даних таблиці слід виконати вказівку *Дані/ Проміжні підсумки* та задати всі необхідні властивості у вікні *Проміжні підсумки* (мал. 9.20). Для таблиці, що впорядкована за кількома полями, можна створити вкладені підсумки за допомогою вкладок, що визначають рівні: *1-ша група*, *2-га група*, *3-тя група*.



ДІЄМО

Вправа 4. Створення проміжних підсумків.

Завдання. У файлі *Землетруси* додайте проміжні підсумки, що відображають загальну кількість жертв потужних землетрусів у кожній країні.

1. Відкрийте файл *Землетруси*, що збережений у папці *Електронні таблиці*.
2. Виділіть клітинку C3, що містить заголовок поля *Країна*. На вкладці *Дані* в групі *Сортування й фільтр* оберіть інструмент *Сортування від А до Я* (на панелі інструментів *Стандартна* оберіть інструмент *Сортування за зростанням*).

- На вкладці *Дані* в групі *Структура* оберіть інструмент *Проміжні підсумки* (виконайте вказівку *Дані/Підсумки*). У діалоговому вікні *Проміжні підсумки* (мал. 9.19, 9.20) у списку *При кожній зміні в:* виберіть поле *Країна*, у списку *Використовувати функцію:* виберіть *Сума*, у списку *Додати підсумки до:* увімкніть прапорець для поля *Кількість жертв* (для інших полів прапорці мають бути вимкненими). Натисніть кнопку *ОК* (*Гаразд*). До таблиці після кожної групи буде додано підсумкові записи. У лівій частині вікна буде подано ієрархічну структуру, що дає змогу відображати всі записи разом з підсумками або лише підсумкові записи (мал. 9.21).

	A	B	C	D
3	Дата	Місцевість	Країна	Кількість жертв
4	1667	Шемаха	Азербайджан	80000
5			Азербайджан Підсумок	80000
6	-856	Коринф	Греція	45000
7	-464	Спарта	Греція	20000
8			Греція Підсумок	65000
9	1737	Калькутта	Індія	300000
10			Індія Підсумок	300000
11	893	Ардебіль	Іран	150000
12	856	Дамган	Іран	200000
13			Іран Підсумок	350000
14	1908	Мессіна	Італія	160000
15			Італія Підсумок	160000
16	1920	Каньшу	Китай	200000
17	1731	Пекін	Китай	100000
18	1976	Тянь-Шань	Китай	655000
19	1038	Чихлі	Китай	100000
20	1556	Шаньсі	Китай	800000
21			Китай Підсумок	1855000

Мал. 9.21

- Натисніть на кожній кнопці  в області структури, що дасть змогу приховати вихідні записи таблиці, а відображувати лише підсумкові записи (мал. 9.19 с. 111).
- Збережіть результати у файлі з іменем *Підсумки* в папці *Електронні таблиці* вашої структури папок.



ДОСЛІДЖУЕМО

- Дослідіть, яким буде результат додавання проміжних підсумків у разі, якщо дані електронної таблиці не впорядковані за жодним з полів. Зробіть висновок.



ПРАЦЮЕМО В ПАРАХ

- Коли й навіщо в житті застосовують впорядкування даних? По черзі називайте приклади — виграє той, хто назве приклад останнім.

2. Навіщо використовувати фільтри в побуті? Наведіть якомога більше прикладів. Назвіть три аргументи, які підтверджують, що дія використання побутових фільтрів подібна до дії фільтрування даних засобами табличного процесора.

 3. У яких випадках при створенні проміжних підсумків є доцільним, щоб прапорець *Замінити поточні підсумки* був увімкненим, а коли його доцільно вимкнути? Наведіть приклади. Обговоріть у парах.

4. Чи є щось спільне між автофільтрами та автозаповненням клітинок електронних таблиць? Відповідь аргументуйте. Обговоріть у парах.

 5. Відкрийте файл *Країни*, що зберігається в папці *Електронні таблиці* (<https://cutt.ly/OT8rBtO>) (мал. 9.22).

	A	B	C	D
1	Країна	Площа, тис. кв.	Населення, млн. чол.	Столиця
2	ФРН	356	80	Берлін
3	Франція	552	67	Париж
4	Великобританія	244	65	Лондон
5	Ірландія	70	4,8	Дублін
6	Нідерланди	41	17	Амстердам
7	Бельгія	31	11,1	Брюссель
8	Швейцарія	41	8	Берн
9	Австрія	84	8,4	Вена
10	Данія	43	5,6	Копенгаген
11	Норвегія	387	5	Осло
12	Швеція	450	9,6	Стокгольм
13	Фінляндія	338	5,4	Хельсінкі
14	Естонія	45	1,3	Таллін
15	Латвія	65	2	Рига
16	Литва	65	3	Вільнюс
17	Польща	313	38,4	Варшава
18	Україна	603,7	45,2	Київ
19	Чехія	128	10,5	Прага

■ Мал. 9.22 ■

а) Сформулюйте три запитання на впорядкування та відбір даних з таблиці, що задовольняють певним критеріям. Наприклад:

- У яких країнах назва столиць починається з літери С?
- Площа яких країн перевищує площу України?
- Населення яких країн менше від населення України?

б) Виконайте добір даних для відповіді на поставлені запитання.

в) Поставте сформульовані запитання іншій парі. Дайте відповіді на запропоновані запитання іншої пари, виконавши на основі заданої таблиці відповідне фільтрування даних.

г) Перевірте відповіді одне в одного. Обговоріть одержані результати.

д) Упорядкуйте дані стосовно країн за щільністю населення на квадратний кілометр. Для цього за необхідності вставте перед полем *Столиця* стовпець *Щільність населення* та проведіть необхідні обчислення.



6. Попрацюйте в четвірках у проєкті «Електронний магазин». Уявіть, що вам запропонували розробити зручну систему для консультантів і покупців магазину для отримання інформації про навні комплектуючі, переферію, мережеве обладнання та витратні матеріали. Для цього надали 4 електронні таблиці — прайс-листи (в архіві *pricelist* у папці *Електронні таблиці* (<https://cutt.ly/0EVuO6J>)). Запропонуйте аналітичні інструменти (фільтрування, підведення підсумків тощо) для організації такої системи. Презентуйте її у класі.



ПРАЦЮЄМО САМОСТІЙНО



7. Відкрийте файл *Розклад потягів* (мал. 9.23), що зберігається в папці *Електронні таблиці* (<https://cutt.ly/ZEVuVCh>).

	A	B	C	D	E	F	G
1	№ потяга	Пункт призначення	Час відправлення	Періодичність	Час прибуття до станції	Наявність квитків	Тариф
2	17	Адлер	16:00	Н	1:00	є	420-00
3	198	Акмол	12:00	****	17:34	є	328-00
4	31	Баку	0:46	Ср, сб	1:00	є	560-00
5	225	Бердянськ	22:26	Н	1:00	немає	189-00
6	273	Брест	18:00	Н	1:00	є	256-00
7	29	Варшава-Берлін	23:55	Щ	20:28	немає	680-00
8	67	Варшава	13:34	Щ	1:05	є	560-00
9	232	Вороніж	16:22	Ср, сб	12:00	є	299-00
10	51	Вроцлав	19:49	Щ	19:44	є	425-00
11	80	Дніпропетр.	21:00	Щ	1:00	є	98-00
12	38	Донецьк	19:25	Щ	9:20	немає	80-00
13	71	Запоріжжя	18:00	Щ	1:07	є	95-00
14	203	Ів.-Франків	17:00	Щ	19:40	є	76-00

Мал. 9.23

а) Упорядкуйте дані послідовно за напрямком, наявністю квитків, вартістю, часом відправлення.

б) За допомогою засобу *Автофільтр* знайдіть відомості про потяги:

- на які є квитки та які відправляються щодня;
- які відправляються до Запоріжжя та на які є квитки;
- які відправляються до Львова найпізніше;
- які відправляються до Варшави з найдешевшим тарифом.

в) Створіть діапазон з умовами та застосуйте розширений фільтр для пошуку відомостей про потяги, які відправляються щодня або час відправлення яких — пізніше ніж о 17:30.

8. Відкрийте файл *Турист* (мал. 9.24), що зберігається в папці *Електронні таблиці* (<https://cutt.ly/gEVieCt>).

	A	B	C	D	E	F	G
1	БАЗА ДАНИХ ТУРИСТИЧНИХ ФІРМ						
2							
3					Вартість номера за добу	Харчування	Авіаквитк и
4	Країна	Місто	Готель	Категорія			
5	Франція	Париж	Alexandra	1***	149	шведський стіл	окремо
6	Франція	Париж	Alexandra	1***	155	сніданок+вечеря	окремо
7	Франція	Париж	Alexandra	1***	160	харчування включено	окремо
8	Франція	Париж	Alexandra	2***	169	шведський стіл	окремо
9	Франція	Париж	Alexandra	2***	173	сніданок+вечеря	окремо
10	Франція	Париж	Alexandra	2***	180	харчування включено	окремо
11	Франція	Париж	Alexandra	люкс***	217	харчування включено	окремо
12	Франція	Париж	Du Louvre	1****	200	шведський стіл	окремо
13	Франція	Париж	Du Louvre	1****	370	сніданок+вечеря	окремо
14	Франція	Париж	Du Louvre	2****	365	шведський стіл	окремо

Мал. 9.24

а) Упорядкуйте дані таблиці одночасно за країнами, у них — за містами, у них — за вартістю номерів за добу.

б) За допомогою засобу *Автофільтр* знайдіть у таблиці відомості про:

- усі готелі Парижа, що мають категорію три зірочки (1*** чи 2***) та пропонують сніданок і вечерю;
- всі готелі для поїздки до Вашингтона (США), де включено харчування;
- найдешевші 10 готелів у Парижі;
- всі тури, до яких входять авіаквитки та шведський стіл для харчування;

в) За допомогою проміжних підсумків знайдіть середню вартість номера за добу в кожній країні та кожному місті.

9. Відкрийте файл *Школа*, що зберігається в папці *Електронні таблиці*.

а) За допомогою засобу *Автофільтр* знайдіть відомості про:

- усіх дівчат, які захоплюються тенісом, віком від 12 до 14 років. Результати фільтрування скопіюйте на новий аркуш;
- усіх хлопців, що займаються футболом, зріст яких більше 130 см. Результати фільтрування скопіюйте на новий аркуш;
- усіх учнів з карими очима віком 14 років, прізвище яких починається з літери *В*. Результати фільтрування скопіюйте на новий аркуш.

б) Виконайте впорядкування за зростанням за полем *Вік*. Додайте проміжні підсумки, які відображатимуть середній зріст учнів кожного віку.

ПІДГОТОВКА ЕЛЕКТРОННОЇ ТАБЛИЦІ ДЛЯ ПЕРЕДАВАННЯ

10

ПРИГАДАЙТЕ:

- які типи файлів можна створити в текстовому процесорі;
- в чому особливість роботи з текстовим редактором.

ВИ ДІЗНАЄТЕСЯ:

- коли варто скористатись імпортом/експортом даних під час роботи з електронною таблицею;
- як налаштувати значення параметрів сторінки перед друкуванням електронної таблиці;
- як надрукувати електронну таблицю.



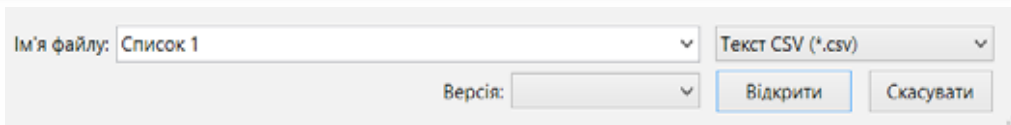
ВИВЧАЄМО

10.1. Коли варто скористатись імпортом/експортом даних під час роботи з електронною таблицею?

Ми вміємо копіювати, вставляти об'єкти з одних офісних додатків в інші. Але інколи пряме копіювання не дає бажаного результату. Розглянемо ситуацію. Нехай у текстовому документі міститься структурована інформація про товари, їхні ціни та кількість. Яким чином найпростіше для автоматизації опрацювання перенести ці дані в електронну таблицю? Ми знаємо, що можна поелементно копіювати та переносити ці дані з текстового документа, або перетворити текст на таблицю в текстовому редакторі з наступним копіюванням в електронну таблицю. Але є і інший спосіб. За допомогою *Microsoft Excel* можна імпортувати дані з текстового файлу до аркуша. *Майстер імпорту тексту* перевіряє текстовий файл, який імпортується, і забезпечує його належне імпортування. Для імпорту даних з текстового файлу потрібно використати команду: *Дані/Отримання зовнішніх даних/Із тексту*. Після вибору файлу запускається *Майстер імпорту тексту*. Після цього слід вибрати формат вихідних даних, формат файлу, інші початкові параметри. На другому кроці задають розділювачі для текстових даних та інші параметри. На третьому кроці — обирають формат даних стовпців. *Загальний формат* є найбільш універсальним, оскільки при його виборі числові дані подаються числами, текстові — текстом, дати — датами. Наприкінці вказують, як потрібно зберігати імпортовані дані (адреса клітинки, спосіб збереження тощо).

У табличному процесорі *LibreOffice Calc* для імпорту даних використовують файли з розширенням csv. Діють за таким алгоритмом.

1. Обрати команду *Файл/ Відкрити*.
2. Знайти потрібний файл CSV.



3. Натиснути кнопку *Відкрити*.

Відкриється діалогове вікно *Імпорт тексту*.

4. Вказати параметри для поділу тексту у файлі на стовпці.

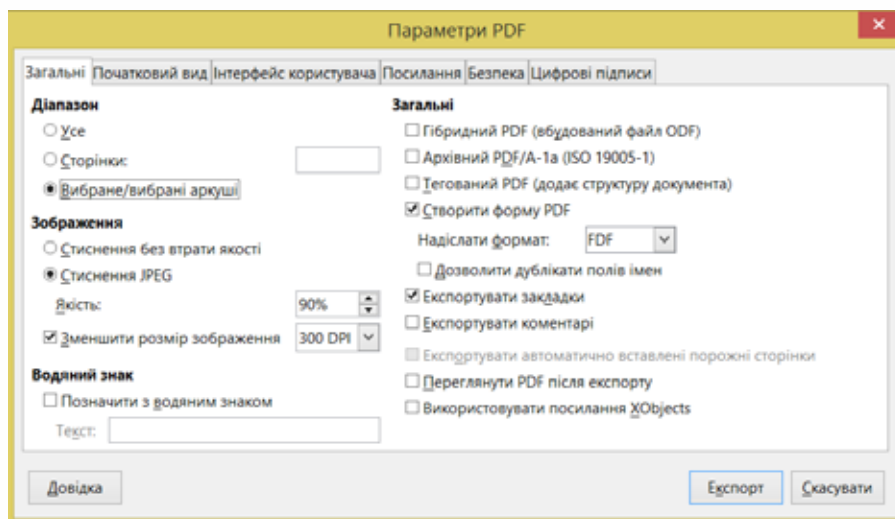
Можливий попередній перегляд розмітки імпортованих даних у нижній частині діалогового вікна *Імпорт тексту*.

Для налаштування формату чи приховування стовпця — клацнути правою кнопкою по стовпцю в режимі попереднього перегляду. Установити прапорець роздільника тексту, відповідний символу, що використовується у файлі як роздільник. Якщо роздільник відсутній у списку, то ввести цей символ у полі вводу.

5. Натиснути кнопку *Гаразд*.

У свою чергу, дані електронної таблиці можна експортувати, наприклад, у PDF-файл. Це потрібно, коли дані електронної таблиці готуються для передавання і є потреба захистити вміст від змін. Для експорту файлу у PDF формат використовують вказівку *Файл/Експорт*. Обирають *Створити документ pdf/xps* або інший формат (наприклад, текстовий).

У табличному процесорі *LibreOffice Calc* при експортуванні файлу у PDF-формат у вікні *Параметри pdf* вказують загальні параметри (мал. 10.1) та інші, які визначають особливість документа, що створюється. Якщо прийняти налаштування за замовчуванням, то у вікні натискають кнопку *Експорт*, після якої запускається процес перекодування файлу з формату електронної таблиці в обраний формат.

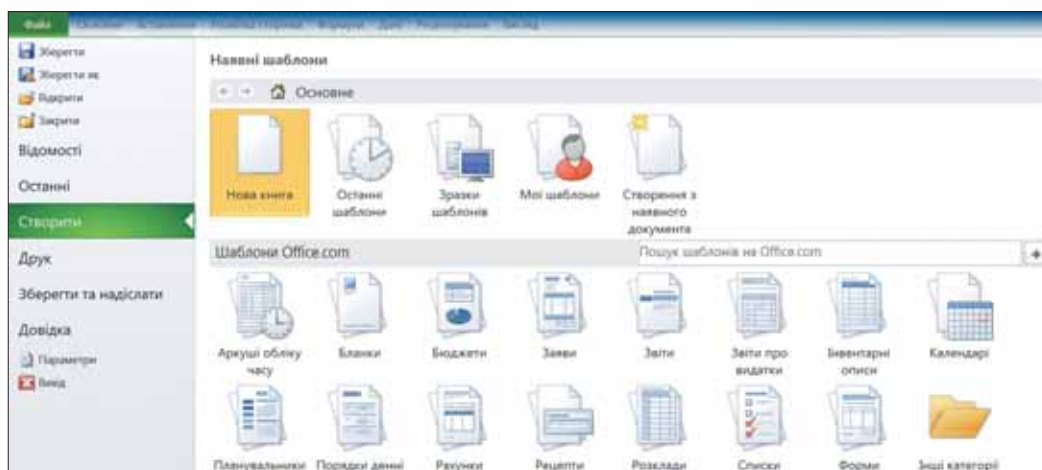


Мал. 10.1

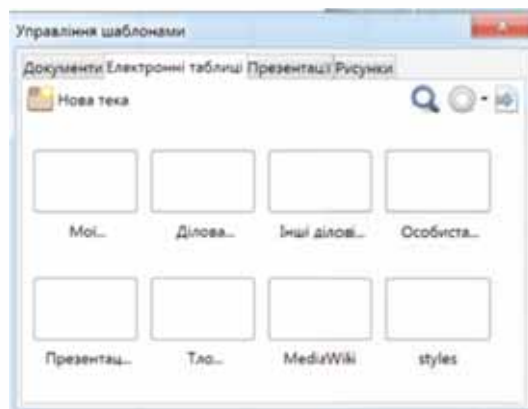
10.2. Як налаштувати значення параметрів сторінки перед друкуванням електронної таблиці?

Як і для текстових документів, під час створення електронної таблиці встановлюються параметри сторінки за замовчуванням: орієнтація сторінки — книжкова чи альбомна, розміри лівого, правого, верхнього та нижнього полів, розмір аркуша паперу тощо.

Для електронних таблиць, що створювалися на основі шаблонів, такі параметри можуть відрізнятися від тих, які застосовуються до нової порожньої книги. Як і в текстовому процесорі, у табличному процесорі *Microsoft Excel* за допомогою вказівки *Файл/Створити* (у табличному процесорі *LibreOffice Calc* за допомогою вказівки *Файл/Створити/Шаблони*) можна обрати шаблон для створення нової книги (мал. 10.2, а, б), що вже містить деякі дані, оформлення та параметри сторінки.



■ Мал. 10.2, а ■



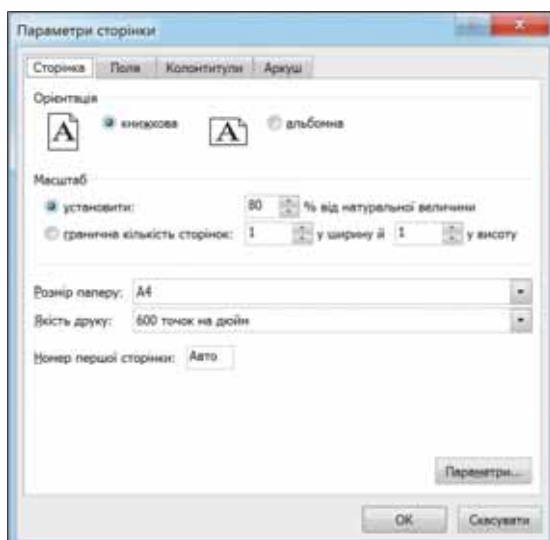
■ Мал. 10.2, б ■

Наприклад, нова книга, створена на основі шаблону *Журнал кров'яного тиску* (мал. 10.3), має такі параметри сторінки: орієнтація сторінки — книжкова, ліве, верхнє, праве та нижнє поля — по 1,3 см.

Журнал кров'яного тиску					
Ім'я:					
Орієнтовний кров'яний тиск*		Систолічний	Діастолічний		
		120	80		
Зателефонувати терапевту, якщо якщо		Систолічний	Діастолічний		
Номер телефону терапевта					
Дата	Час	Систолічний	Діастолічний	Серцевий пульс	Примітки
1.11.06	Вранці	129	79	72	
1.11.06	Ввечері	133	80	75	
2.11.06	Вранці	142	86	70	
2.11.06	Ввечері	141	84	68	
3.11.06	Вранці	137	84	70	
3.11.06	Ввечері	139	83	72	
4.11.06	Вранці	140	85	78	
4.11.06	Ввечері	138	85	69	
5.11.06	Вранці	135	79	75	
5.11.06	Ввечері	136	81	72	
6.11.06	Вранці	137	83	69	
6.11.06	Ввечері	138	85	66	
7.11.06	Вранці	135	80	67	
7.11.06	Ввечері	142	85	72	

■ Мал. 10.3 ■

Щоб змінити параметри сторінки в табличному процесорі *Microsoft Excel*, можна скористатись інструментами на вкладці *Розмітка сторінки* або у вікні *Параметри сторінки* (мал. 10.4).



■ Мал. 10.4 ■

Аналогічне вікно *Параметри сторінки* в табличному процесорі *LibreOffice Calc* (мал. 10.5, с. 121) викликають за допомогою вказівки *Формат/Сторінка*.



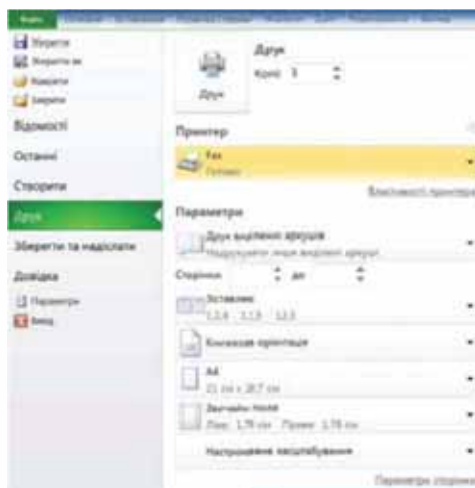
Мал. 10.5

Іноді вся таблиця може не поміститися на сторінку формату А4, на якій передбачається друкування. Наприклад, може не вміститися по ширині один стовпчик, а по висоті — два рядки. Тому під час підготовки до друку електронної таблиці слід враховувати масштаб відображення таблиці. Змінюючи масштаб, можна розмістити всю таблицю на одному аркуші.

10.3. Як надрукувати електронну таблицю?

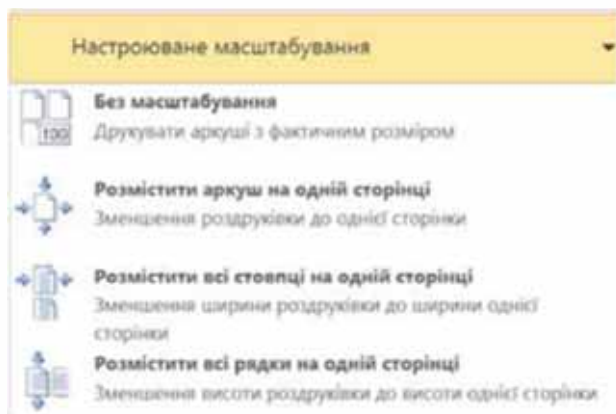
Після введення даних до таблиці, оформлення, виконання обчислень, створення діаграм та інших об'єктів, налаштування параметрів сторінки електронну таблицю можна надрукувати.

У табличному процесорі *Microsoft Excel*, як і в текстовому процесорі *Microsoft Word*, для цього виконують вказівку *Файл/Друк* та вказують значення параметрів друкування: обирають принтер, кількість копій для друкування, порядок друкування кількох аркушів і в разі необхідності змінюють значення параметрів сторінки (мал. 10. 6).



Мал. 10.6

Значення параметрів можна змінити, якщо відкрити відповідні списки (мал. 10.7). Крім того, можна обрати також один з режимів настроюваного масштабування, за якого автоматично буде обрано масштаб відображення, щоб уся таблиця розмістилася на одній сторінці чи всі стовпці або рядки розміщувалися на одній сторінці (мал. 10.7).

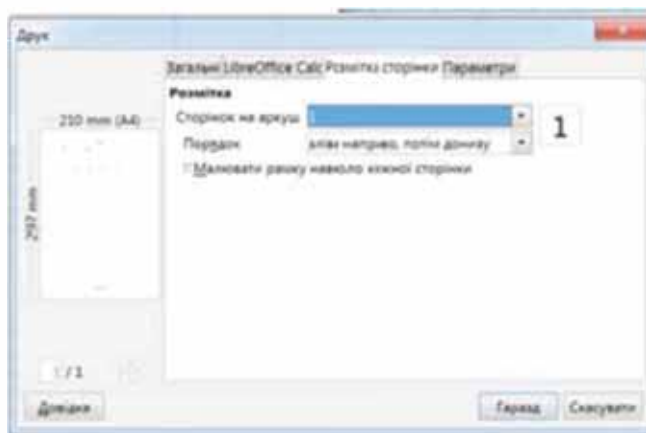


■ Мал. 10.7 ■

Після встановлення значень усіх необхідних параметрів слід натиснути кнопку  .

У табличному процесорі *LibreOffice Calc* параметри друкування та сторінки зібрано на різних вкладках вікна *Друк* (мал. 10.8), яке від-

кривають за допомогою вказівки *Файл/Друк* або кнопки *Друк*  на панелі інструментів. Щоб надрукувати таблицю, натискають кнопку *Гаразд*.



■ Мал. 10.8 ■



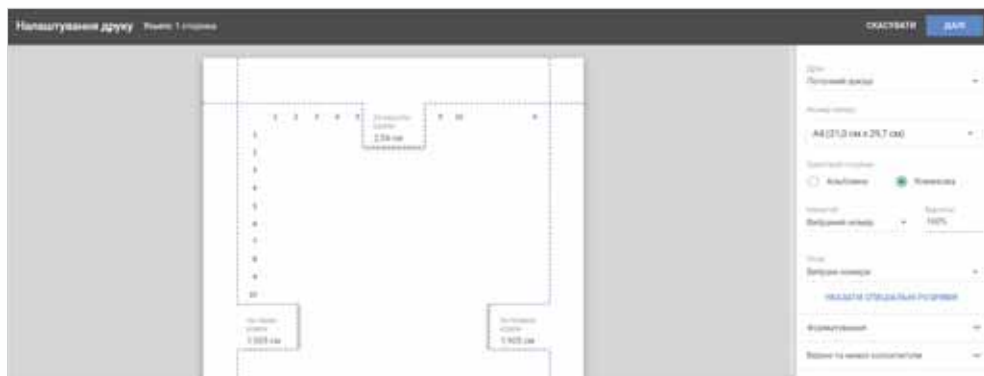
ПРАЦЮЄМО В ПАРАХ

1. Обговоріть, чи можна використати табличний процесор для друку плакатів або мініпублікацій (наприклад, візитівок). Підтвердьте своє припущення. Сформулюйте загальне правило.
2. Знайдіть в інтернеті відомості про файли форматів CSV та PDF. Поясніть одне одному, як їх використовують під час роботи з електронними таблицями.
3. Порівняйте, що спільного та відмінного мають табличний процесор і текстовий процесор для:
 - налаштування параметрів сторінки;
 - друкування документа.За результатами порівняння побудуйте діаграму Венна.



ПРАЦЮЄМО САМОСТІЙНО

4. Дослідіть, які шаблони доступні для роботи в табличному процесорі на вашому комп'ютері. Складіть список із трьох з них, які б ви порадили для використання своїм рідним.
5. Дослідіть, чим подібний друк електронної таблиці, відкритої в середовищі табличного процесора й *Таблицях Google*. Складіть презентацію-пояснення. Використайте вікно налаштування друку в *Таблицях Google* (мал. 10.9).



Мал. 10.9

6. Попрацюйте в мініпроєкті «Адресна книга». Створіть адресну книгу з власних контактів і роздрукуйте її на аркушах розміру *Карточка вказівника*. Доберіть параметри друку так, щоб утворити аркуші для саморобного мініблокноту. Продемонструйте свій виріб у класі.

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Я можу пояснити відмінність між посиланнями різних типів.
- ☐ Я можу назвати основні логічні, математичні й статистичні функції та пояснити їхнє призначення.
- ☐ Я можу інтерпретувати деякі види електронних таблиць як набори відомостей про однотипні об'єкти.
- ☐ Я можу пояснити призначення функцій і засобів табличного процесора для опрацювання наборів однотипних об'єктів.
- ☐ Я вмію добирати й застосовувати доцільну функцію або засіб табличного процесора для розв'язування певної задачі.
- ☐ Я вмію використовувати посилання різних типів для опрацювання рядів даних.
- ☐ Я вмію добирати тип діаграми, що є найдоречнішим для візуального подання набору даних.
- ☐ Я вмію будувати та інтерпретувати діаграми різних типів.
- ☐ Я вмію застосовувати умовне форматування для унаочнення даних, що задовольняють певні умови.
- ☐ Я вмію розв'язувати задачі, що вимагають сортування та обчислення проміжних і загальних підсумків, застосовувати прості й розширені фільтри для відбору об'єктів.
- ☐ Я вмію експортувати й імпортувати вміст електронних таблиць.
- ☐ Я усвідомлюю значення електронних таблиць як засобу для фінансових розрахунків і розв'язання задач із інших дисциплін.
- ☐ Я можу налаштувати друк електронної таблиці в потрібному вигляді.

11

ЯК ПРАЦЮЮТЬ З НАБОРОМ ДАНИХ В МОВІ ПРОГРАМУВАННЯ?

ПРИГАДАЙТЕ:

- як описати команду введення та виведення мовою *Python*;
- які операції виконують над рядками мовою *Python*.

ВИ ДІЗНАЄТЕСЯ:

- як у мові програмування *Python* описують послідовність даних;
- як звернутись до окремого елемента списку;
- які дії можна виконувати над списками в мові *Python*;
- як здійснювати введення та виведення елементів списку.



ВИВЧАЄМО

11.1. Як у мові програмування *Python* описують послідовність даних?

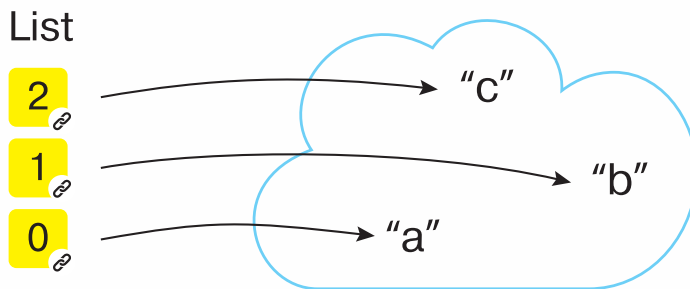
Ви вмієте створювати програми мовою програмування *Python* для опрацювання певної кількості змінних. Їхні значення вводять за допомогою команди введення *input()*, у вікнах повідомлень чи, наприклад, у текстовому полі. Але часто потрібно опрацьовувати велику кількість змінних або множину значень однієї змінної. Для цього в мові програмування *Python* використовують **списки** (*list*) — впорядковану послідовність із декількох значень об'єктів будь-яких типів: числа, рядки, списки тощо. В інших мовах програмування для опису сукупності однотипних даних, що об'єднані спільним іменем та яким можна надавати значення в процесі виконання програми, використовують **масиви**.

ЦІКАВО

Масив у програмуванні — це тип структури даних, що має складені значення. (З Оксфордського словника англійської мови, 1995 р.)

На відміну від рядків, списки складаються не із символів, а з різних об'єктів (значень, даних), і вкладаються не в лапки, а в квадратні дужки `[]`.

Взагалі *Python* розміщує елементи списку в пам'яті, а сам список містить вказівники на об'єкти (мал. 11.1, с. 126). Таким чином, список у *Python* — це масив вказівників.



Мал. 11.1

У мові *Python* елементи списку розділяються комами (табл. 11.1). Наприклад,

Таблиця 11.1

<code>empty_list = []</code>	порожній список з ім'ям <i>empty_list</i>
<code>weekdays = ['Monday', 'Friday', 'Wednesday', 'Thursday', 'Friday']</code>	список днів тижня з ім'ям <i>weekdays</i>
<code>animals = ['camels', 'bats', 'elephants', 'dolphins', 'bears']</code>	Список тварин з ім'ям <i>animals</i>
<code>mix=['Oksana', 15,165]</code>	Мішаний список, що містить текстові та числові дані з ім'ям <i>mix</i>
<code>bin=[[0, 0, 0], [0, 0, 1], [0, 1, 0]]</code>	Список, що складається зі списків з ім'ям <i>bin</i>

Якщо присвоїти один список більш ніж одній змінній, то зміни у списку в одному місці спричинять за собою його зміни в інших (табл. 11.2).

Таблиця 11.2

Фрагмент коду	Результат
<code>>>> a = [1, 2, 3]</code> <code>>>> a</code>	[1, 2, 3]
<code>>>> b = a</code> <code>>>> b</code>	[1, 2, 3]
<code>>>> a[0] = 'new'</code> <code>>>> a</code>	['new', 2, 3]
<code>>>> b</code>	['new', 2, 3]

Значення списку можна скопіювати в незалежний новий список за допомогою одного з таких способів:

- функції `copy()`;
- функції `list()`;
- розділенням списку за допомогою `[:]`.

Наприклад, оригінальний список буде присвоєний змінній *a*, а інші списки — *b*, *c*, *d* — будуть копіями списку *a*. Зауважимо, що *b*, *c*, *d* — це нові об’єкти, що мають свої значення, які не пов’язані з оригінальним списком елементів [1, 2, 3], на який посилається змінна *a*. Зміни в *a* не впливають на копії *b*, *c*, *d* (табл. 11.3).

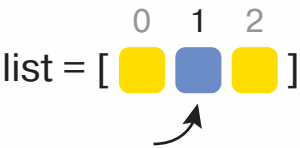
Таблиця 11.3

Фрагмент коду	Результат
>>> a = [1, 2, 3] >>> b = a.copy() >>> c = list(a) >>> d = a[:]	
>>> a	[1, 2, 3]
>>> b	[1, 2, 3]
>>> a = [4, 2, 3] >>> a >>> b >>> c >>> d	 [4, 2, 3] [1, 2, 3] [1, 2, 3] [1, 2, 3]

11.2. Як звернутись до окремого елемента списку?

Списки є впорядкованими наборами даних, тому для доступу до будь-якого елемента списку слід повідомити *Python* позицію (індекс) потрібного елемента.

Індекси приймають тільки цілочисельні значення. Щоб звернутися до елемента у списку, вказують ім’я списку, за яким слідує індекс елемента у квадратних дужках. Як і для рядків, зі списку можна отримати конкретне значення, вказавши його індекс (якщо вказати позицію, яка міститься перед списком або після нього, буде згенеровано помилку).



```
color_model = ['red', 'green', 'blue']  
print(color_model [1])
```

Результат: green

Індекс може бути як > 0, так і < 0. Останній символ має номер -1, передостанній -2, і т. д.

11.3. Які дії можна виконувати над списками в мові Python?

Над списками можна виконувати прості дії: об’єднання, дублювання списків, отримання фрагменту списку, визначення кількості елементів списку, додавання до списку нового елемента.

Подібні операції ми виконували над рядками. Але на відміну від рядків, списки — це змінювані послідовності. Якщо уявити рядок як об’єкт

у пам'яті, то коли над ним виконуються операції об'єднання і повторення, цей рядок не змінюється, а в результаті операції створюється інший рядок в іншому місці пам'яті. У рядок не можна додати новий символ або видалити існуючий, не створивши при цьому нового рядка. При виконанні операцій зі списками інші списки можуть не створюватися, а змінюється безпосередньо оригінал. Зі списків можна видаляти елементи, додавати нові. Списки можуть копіюватися, коли результат операції присвоюється іншій змінній.

Розглянемо дії над списками на прикладах.

Об'єднання (конкатенація) списків задається операндом «+». Особливості застосування цієї операції подано на прикладах у таблиці 11.4.

Таблиця 11.4

Фрагмент програми	Результат
<pre>1 a= [1, 2, 3] 2 b= [4, 5, 6] 3 print (a+b)</pre>	[1, 2, 3, 4, 5, 6]
<pre>1 a= [1, 2, 3] 2 b= [4, 5, 6] 3 print (b+a)</pre>	[4, 5, 6, 1, 2, 3]
<pre>1 a= [1, 2, 3] 2 b= [4] 3 print (a+b)</pre>	[1, 2, 3, 4]
<pre>1 a= [1, 2, 3] 2 b= 4 3 print (a+b)</pre>	<pre>print (a+b) TypeError: can only concatenate list (not "int") to list</pre>

Дублювання списку задають операндом «*» за правилами, які описано на прикладах таблиці 11.5.

Таблиця 11.5

Фрагмент програми	Результат
<pre>1 a= [1, 2, 3] 2 b= 3 3 print (a*b)</pre>	[1, 2, 3, 1, 2, 3, 1, 2, 3]
<pre>1 a= [1, 2, 3] 2 b= -3 3 print (a*b)</pre>	[]



Фрагмент програми	Результат
<pre> 1 a= [1, 2, 3] 2 b= -1 3 print (a*b) </pre>	[]
<pre> 1 a= [1, 2, 3] 2 b= 2.5 3 print (a*b) </pre>	<pre> print (a*b) TypeError: can't multiply sequence by non-int of type 'float' </pre>

Фрагмент списку має назву **зріз**. Щоб отримати потрібний фрагмент, вказують початковий і кінцевий індекс елемента за правилами, подібними до утворення зрізів у рядках (табл. 11.6).

Таблиця 11.6

Фрагмент програми	Результат
<pre> 1 a= [1, 2, 3, 4, 5, 6] 2 b= a[1:3] 3 print (b) </pre>	[2, 3]
<pre> 1 a= [1, 2, 3, 4, 5, 6] 2 b= a[:3] 3 print (b) </pre>	[1, 2, 3]
<pre> 1 a= [1, 2, 3, 4, 5, 6] 2 b= a[3:] 3 print (b) </pre>	[4, 5, 6]
<pre> 1 a= [1, 2, 3, 4, 5, 6] 2 b= a[3:10] 3 print (b) </pre>	[4, 5, 6]

Для визначення кількості елементів списку використовують функцію `len()`.

Наприклад, при виконанні фрагмента коду

```
a=[2, 4, 6, 8, 10]
```

```
print (len(a))
```

отримаємо значення 5.

Елементи списку можна **змінювати**. Для цього слід звернутись до індексу списку та за адресою цього індекса надати нове значення (мал. 11.2, с.130).

list = [   ]

```
1 a= ['a', 'b', 'c', 'd']  
2 a[2]='f'  
3 print (a)
```

Результат: ['a', 'b', 'f', 'd']

■ Мал. 11.2 ■

Щоб додати новий елемент у список, використовують метод `append()`:

`list.append(x)` , де `list` — список, `x` — потрібне значення (мал. 11.3).

list = [    ]

```
1 a= ['a', 'b', 'c', 'd']  
2 a.append('e')  
3 print (a)
```

Результат: ['a', 'b', 'c', 'd', 'e']

■ Мал. 11.3 ■

Щоб додати в список елемент на вказану позицію, використовують метод `insert`:

`list.insert(i, x)` , де `list` — список, `i` — позиція, `x` — потрібне значення (мал. 11.4).

list = [   ]

```
1 a= ['a', 'c', 'd']  
2 a.insert(2, 'b')  
3 print (a)
```

Результат: ['a', 'c', 'b', 'd']

■ Мал. 11.4 ■

Для **видалення** елемента зі списку використовують оператор `del`:
`del list[i]`, де `list` — список, `i` — індекс (позиція) елемента у списку (мал. 11.5).



```
1 a= ['a', 'b', 'c', 'd']
2 del a[2]
3 print (a)
```

Результат: ['a', 'b', 'd']

■ Мал. 11.5 ■

Примітка. Видаляти можна не тільки окремі елементи, а й цілі діапазони, вказуючи зрізи списку.

Для **видалення конкретного значення** зі списку використовують метод `list.remove(x)`, де `list` — список, `x` — значення, яке видаляють (мал. 11.6).

```
1 a= ['a', 'b', 'c', 'd']
2 a.remove('b')
3 print (a)
```

Результат: ['a', 'c', 'd']

■ Мал. 11.6 ■

11.4. Як здійснювати введення та виведення елементів списку?

Для введення елементів списку в процесі виконання програми використовують функцію `input()` і метод `split()` (мал. 11.7).

`list=input().split()`

Наприклад,

```
1 a= input().split()
2 print (a)
```

Результат: ^{1 2 3 4 5} ['1', '2', '3', '4', '5']

■ Мал. 11.7 ■

Для того щоб отримати список із числових значень, використовують конструкцію, як показано на прикладі (мал.11.8, с. 132):


```
1 a = [int(i) for i in input().split()]
2 print(a)
```

Результат: 1 2 3 4 5
 [1, 2, 3, 4, 5]

Мал. 11.8

У мові програмування *Python* вивести всі елементи списку *a* можна за допомогою команди `print(a)`, при цьому буде виведено значення елементів списку, поміщені в квадратні дужки, через кому (мал. 11.9).

```
a = [1, 2, 3]
b = [4, 5]
c = a + b
d = b * 3
print(c)
print(d)
print([7, 8] + [9])
print([0, 1] * 3)
```

Програмний код

```
C:\Users\Ольга\AppData
[1, 2, 3, 4, 5]
[4, 5, 4, 5, 4, 5]
[7, 8, 9]
[0, 1, 0, 1, 0, 1]
```

Результат виконання програми

Мал. 11.9

Якщо потрібно вивести елементи списку в стовпець або рядок, використовують поелементне виведення. Вказівка повторення `for` може організувати виведення перебором індексів або значень елементів, тобто цикл `for` надає зручний спосіб перебрати саме елементи послідовності, а не індекси елементів (мал. 11.10).

Програмний код	Виведення	Програмний код	Виведення
<pre>a = [5, 2, 7, 4, 3] for i in range(len(a)): print(a[i])</pre>		<pre>a = [5, 2, 7, 4, 3] for elem in a: print(elem, end=' ')</pre>	<pre>C:\Users\Ольга\AppData 5 2 7 4 3 Process finished with</pre>

Мал. 11.10



ДІЄМО

Вправа 1. Введення масиву дійсних чисел.

Завдання. Розробіть проєкт для введення масиву *x*, що складається з *n* ($n < 20$) дійсних чисел, де значення кількості елементів масиву вводять у текстове поле, а самі елементи — у вікні введення.

1. Відкрийте середовище програмування. Створіть новий файл програми мовою програмування *Python* з іменем *Введення масиву* в папці *Навчальні проєкти* своєї структури папок.

2. У вікні редактора коду введіть команди для імпорту бібліотеки візуальних компонентів:

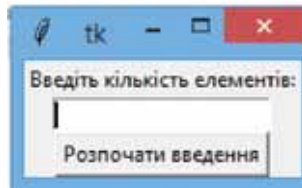
```
import tkinter
```

3. Запишіть програму за зразком, використавши метод *append* — додавання елементів списку (мал. 11.11).

```
window = tkinter.Tk()
# створення об'єкта для отримання значення кількості
# елементів з текстового поля
s = tkinter.StringVar()
# створення порожнього списку
x = []
# опрацювання події натиснення кнопки
def button_click():
    n=int(s.get())
    i=0
    for i in range(n):
        # додавання до списку чергового елемента
        x.append(int(input()))
```

■ Мал. 11.11 ■

4. Розмістіть елементи управління на екранній формі (мал. 11.12), доповнивши програму кодом.



■ Мал. 11.12 ■

```
# створення текстового напису та його розміщення на го-
# ловній формі
label=tkinter.Label(text='Введіть кількість елемен-
# тів:')
```

```
label.pack()
```

```
# створення текстового поля та його розміщення на голо-
# вній формі
```

```
edit=tkinter.Entry(main, textvariable=s)
```

```
edit.pack()
```

```
# створення кнопки та розміщення об'єкта на головній
# формі
```

```
button=tkinter.Button(window, text='Розпочати
# введення',command=button_click)
```

```
button.pack()
```

```
# запуск опрацювання подій програми
```

```
window.mainloop()
```

5. Запустіть проєкт на виконання. Перевірте, чи дії, пов'язані з об'єктами управління екранної форми, відповідають умові завдання. Якщо є помилки, виправте їх.
6. Завершіть роботу з проєктом і середовищем програмування, зберігши всі зміни.

Вправа 2. Виведення масиву цілих чисел.

Завдання. Розробіть проєкт для поелементного виведення масиву a з n ($n < 20$) цілих чисел від 0 до 10 у вікні повідомлення, де значення кількості елементів масиву вводять у текстове поле, а самі елементи добираються випадково після запуску програми.

1. Відкрийте середовище програмування. Створіть новий файл програми мовою програмування *Python* з іменем *Виведення масиву* в папці *Навчальні проєкти* своєї структури папок.
2. У вікні редактора коду введіть команди оголошення об'єктів екранної форми проєкту (мал. 11.13).

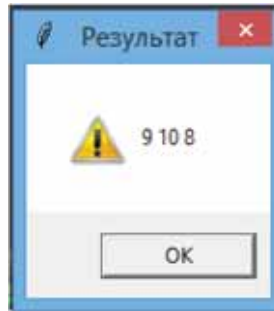
```
import tkinter
from tkinter import ttk
from tkinter import messagebox
import random
window = tkinter.Tk()
# створення об'єкта для отримання значення кількості
# елементів з текстового поля
s = tkinter.StringVar()
# створення порожнього списку
x = []
```

■ Мал. 11.13 ■

3. Продовжте програму за зразком (мал. 11.14), створюючи рядкову величину *sl*, у якій формується результуючий список значень, розділених пропусками. Наприклад, для $n = 3$ одним з можливих результатів буде список, поданий на малюнку 11.15.

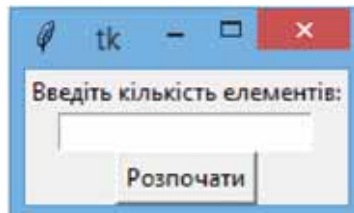
```
# опрацювання події натиснення кнопки
def button_click():
    # отримання значення кількості елементів списку
    n=int(s.get())
    # задання початкового значення рядка результату
    sl=''
    for i in range(n):
        # додавання до списку чергового елемента
        x.append(random.randint(0,10))
        # створення списку елементів у рядку
        sl=sl+str(x[i])+' '
    # виведення вікна повідомлення із результатом
    messagebox.showwarning('Результат', sl)
```

■ Мал. 11.14 ■



Мал. 11.15

4. Розмістіть елементи управління на екранній формі (мал. 11.16), доповнивши програму кодом (мал. 11.17).



Мал. 11.16

```
# створення текстового напису та його розміщення на головній формі
label=tkinter.Label(text='Введіть кількість елементів:')
label.pack()
# створення текстового поля та його розміщення на головній формі
edit=tkinter.Entry(main, textvariable=s)
edit.pack()
# створення кнопки та розміщення об'єкта на головній формі
button=tkinter.Button(window,
text='Розпочати',command=button_click)
button.pack()
# запуск опрацювання подій програми
window.mainloop()
```

Мал. 11.17

5. Запустіть проект на виконання. Перевірте, чи дії, пов'язані з об'єктами управління екранної форми, відповідають умові завдання. Якщо є помилки, виправте їх.
6. Завершіть роботу з проектом і середовищем програмування, зберігши всі зміни.



ПРАЦЮЄМО В ПАРАХ

1. Запропонуйте одне одному приклади проектів, у яких доцільно вводити значення списку, використовуючи:

- а) текстові поля чи написи;
- б) генерування випадкових чисел;
- в) вікно повідомлення.

2. Запропонуйте одне одному правила виконання дій над списками. Хтось проговорює правило, а хтось — перевіряє його на конкретному прикладі в середовищі програмування. Щоразу міняйтеся ролями.

 3. Поясніть на прикладах, як додають елементи до списку, видаляють їх та змінюють елемент списку.

 4. Запропонуйте одне одному приклади, що ілюструють спосіб, який варто застосовувати для введення даних у список.

 5. Поясніть одне одному особливості виведення даних списку в командному та графічному інтерфейсі.



ПРАЦЮЄМО САМОСТІЙНО

6. Придумайте ім'я списку. Запишіть значення другого елемента списку. Визначте кількість елементів у списку

а)

-3.0	4.0	12.0	...	9
-3.0	4.0	12.0	...	-2.0

б)

0	1	2	...	19
32	45	89	...	112

в)

0	1		...	5
Алг.	Мова		...	БЖД

г)

0	1	2	...	8
1.1	1.2	1.3	...	1.9

7. Поставте у відповідність опис і відповідну таблицю значень:

- 1) список із 20 текстових величин;
- 2) список з 10 дійсних чисел;
- 3) список із 7 символічних величин;
- 4) список з послідовності 9 чисел, що змінюються через 0,1;
- 5) список з 9 символів;

 8. Засобами середовища програмування створіть проект *Розклад уроків*, у якому назви уроків формують список з іменем *pazva* за допомогою вікна введення, а результат виводиться у вигляді стовпця.

 9. Засобами обраного середовища програмування створіть проект *Гравітаційна стала*, в якому формуються значення елементів списку з 5 дійсних чисел з діапазону від 6.0 до 7.0, а результат виводиться в написі, повторюючи послідовність отриманих значень двічі.



ПРИГАДАЙТЕ:

- які об'єкти використовують за блок-схемного опису алгоритмів;
- як описати команду розгалуження мовою програмування *Python*;
- які операції виконують над рядками мовою *Python*.

ВИ ДІЗНАЄТЕСЯ:

- як можна опрацьовувати табличні величини;
- як описати програму зміни значення даних списку;
- які методи мови програмування *Python* можна використати для аналізу даних списку;
- як виконувати дії над даними у списку.



ВИВЧАЄМО

12.1. Як можна опрацьовувати табличні величини?

Практичні завдання, для розв'язування яких можна застосовувати алгоритми опрацювання табличних величин, можуть мати різне формулювання. Наприклад, найпростіше завдання може розпочинатись зі слів: «Дано n чисел...», і далі вказується, що потрібно зробити з цими числами. Але, зазвичай, трапляються складніші завдання, у яких спосіб організації даних задано неявно. Наприклад, сформувати таблицю квадратів двоцифрових чисел. Під час написання програм для розв'язування подібних завдань слід виконати такі дії:

- 1) за умовою завдання встановити, чи можна подати вхідні дані у вигляді табличної величини — списку;
- 2) надати табличній величині одне ім'я (його у програмі буде використано для роботи з окремими елементами табличних даних);
- 3) задати порожній список, у якому буде зберігатись набір даних;
- 4) у програмному коді записати команди для введення елементів масиву в пам'ять комп'ютера.

Подальше опрацювання табличних величин зводиться до використання алгоритмів:

- 1) зміни значень елементів масиву;
- 2) знаходження підсумків за набором даних, які передбачають використання простих арифметичних операцій чи виразів над елементами масиву;
- 3) обчислення суми або кількості значень елементів, що задовольняють заданим умовам;
- 4) пошуку елементів табличної величини, що задовольняють заданим умовам;
- 5) упорядкування масиву за деякою ознакою.

12.2. Як описати програму зміни значення даних списку?

Зміну значення даних у списку виконують за алгоритмом, який описано блок-схемою (мал. 12.1).



■ Мал. 12.1 ■

Для перегляду елементів списку в програмах мовою *Python* використовують конструкцію

```
for i in range (n):  
    команди
```

де i — індекс елемента списку, $n-1$ — індекс останнього елемента.

Якщо змінювати потрібно не всі значення у списку, а ті, які задовольняють певній умові, то блок-схему доповнюють блоком перевірки умов.



ДІЄМО

Вправа 1. Квадрати значень.

Завдання. Створіть проєкт *Квадрати значень*, у якому елементи табличної величини a , яка містить 10 цілих чисел, змінюються їх квадратами.

1. Відкрийте середовище програмування.

2. Створіть новий файл програми мовою програмування *Python* з іменем *Квадрати значень* у папці *Навчальні проекти* своєї структури папок.
3. У вікні редактора коду введіть команди.

```
import math
# створюємо порожній список
a = []
for i in range (10):
    # додаємо введені елементи до списку в області
    виконання проекту
    a.append(int(input('Введіть елемент списку')))
    # змінюємо елементи списку
    a[i]=a[i]**2
# виводимо результат
print(a)
```

4. Запустіть проєкт на виконання. Введіть значення елементів списку для тестового прикладу — числа від 1 до 10 у вікні виконання проєкту після вказівки

Введіть елемент списку

5. Перевірте, чи відповідають отримані результати умові завдання. Якщо є помилки, виправте їх.
6. Завершіть роботу з проєктом та середовищем програмування, зберігши всі зміни.

Вправа 2. Результати тренувань.

Завдання. На тренуванні у стрибках у висоту тренер усі дані, які менші, ніж особистий рекорд вихованця чи вихованки, обнуляє. Виведіть список можливих результатів, якщо спортсмен або спортсменка має особистий рекорд у стрибках у висоту a см. Зауважте, що рекорд у юнацьких змаганнях з результатом 194 см установила в 2018 р. бронзова призерка Олімпійських ігор 2020 Ярослава Могучіх.

Спортсмен або спортсменка робить 10 спроб, формуючи список результатів (дані отримуються випадковим чином з деякого діапазону).

Створіть проєкт *Результати тренувань*, у якому елементи списку *list*, які менші від заданого значення, будуть замінюватись нулями.

1. Відкрийте середовище програмування.
2. Створіть новий файл програми мовою програмування *Python* з іменем *Результати тренувань* у папці *Навчальні проекти* своєї структури папок.
3. У вікні редактора коду введіть команди (мал. 12.2).


```

1 import random
2 # задаємо контрольне значення
3 a=int(input('Введіть значення особистого рекорду'))
4 # створюємо порожній список
5 b = []
6 for i in range (10):
7     # додаємо введені елементи до списку в області виконання проекту
8     b.append (random.randint((a-10), 194))
9     # змінюємо елементи списку
10    if b[i] < a:
11        b[i]=0
12 # виводимо результат
13 print(b)
14

```

■ Мал. 12.2 ■

4. Поміркуйте, чому у 8-му рядку програми саме так задано діапазон даних для формування списку. Чи можна задати по-іншому цей діапазон і як саме?
5. Поясніть, які результати могли би бути в початковому списку висоти стрибків, якщо після запуску програми на виконання отримали:

```

Введіть значення особистого рекорду150
[162, 172, 181, 157, 193, 185, 155, 0, 0, 184]

```

6. Завершіть роботу з проектом та середовищем програмування, зберігши всі зміни.

12.3. Які методи мови програмування *Python* можна використати для аналізу даних списку?

У мові програмування *Python* передбачено декілька методів для аналізу списку.

Для **визначення індексу елемента** в списку використовують метод `index ()`.

Метод `index ()` дає змогу отримати значення індексу (позиції) заданого елемента списку. Метод отримує 1 параметр, який є шуканим елементом. Значення індексу, що відповідає першому елементу списку, дорівнює 0. Наприклад,

```

# Заданий список
A = [ 'a', 'b', 'c', 'd', 'e', 'f' ]
t = A.index('c')           # t = 'c'
print( "t =", t)

```

Результат виконання програми:

```
t = 2
```

Якщо елемента немає в списку, то видається повідомлення про помилку. Наприклад, якщо в наведеному вище коді рядок

```
t = A.index('c')
```

замінити рядком

```
t = A.index ('g')          # t = 'c'
```

то отримаємо повідомлення про помилку:

```
ValueError: 'g' is not in list
```

Метод *count()* повертає **кількість входжень** заданого елемента в список. Метод отримує один параметр.

```
# Заданий список
```

```
A = [ 'a', 'b', 'c', 'd', 'e', 'f']
```

```
na = A.count('d')          # na = 1
```

```
B = [1, 3, 5, 3, 2, 4]
```

```
nb = B.count(3)            # nb = 2
```

```
print("na =", na)
```

```
print("nb =", nb)
```

Результат виконання програми

```
na = 1
```

```
nb = 2
```



ДІЄМО

Вправа 3. Оцінка якості.

Завдання. На підприємстві планують створити автоматизовану систему перевірки якості пакування продукції. Кожний укладальник має свій код — двоцифрове число. Визначте, на якій лінії перебуває пакувальник за введеним кодом.

Використайте список кодів [12, 23, 34, 45, 56, 67, 78, 89, 90].

1. Відкрийте середовище програмування.
2. Створіть новий файл програми мовою програмування *Python* з іменем *Оцінка якості* в папці *Навчальні проекти* своєї структури папок.
3. У вікні редактора коду введіть команди (мал. 12.3).

```

1 import random
2 # задаємо список
3 a=[12, 23, 34, 45, 56, 67, 78, 89, 90]
4 # вибираємо випадкове число із списку
5 vib=random.choice(a)
6 # обираємо індекс числа
7 t=a.index(vib)
8 # виводимо результат
9 print(vib)
10 print(t)
11

```

■ Мал. 12.3 ■

4. Запустіть проєкт на виконання. Поясніть, чому у вас щоразу виходять різні відповіді.
5. Завершіть роботу з проєктом та середовищем програмування, зберігши усі зміни.



ДОСЛІДЖУЄМО

Вправа 4. Код.

Завдання. Кожний товар на складі маркується десятизначним числовим кодом, для запису якого використовують цифри від 1 до 5 в довільному порядку та кількості через пропуск. Визначте, скільки разів у деякому коді, введеному користувачем, трапляється деяка цифра, яка генерується випадково.

1. Відкрийте середовище програмування.
2. Створіть новий файл програми мовою програмування *Python* з іменем *Код* у папці *Навчальні проєкти* своєї структури папок.
3. У вікні редактора коду введіть команди (мал. 12.4).

```

1 import random
2 # задаємо список
3 a= input().split()
4 # перетворюємо дані із тексту в цілі числа
5 for i in range(len(a)):
6     a[i] = int(a[i])
7 # вибираємо випадкове ціле число із діапазону чисел від 1 до 5
8 vib=random.randint(1,5)
9 # обираємо індекс числа
10 b=a.count(vib)
11 # виводимо результат
12 print(vib)
13 print(b)
14

```

■ Мал. 12.4 ■

4. Запустіть проєкт на виконання. Дослідіть, що означає отриманий результат (мал. 12.5).

```
1 1 1 2 3 4 5
3
1
>>>
```

Мал. 12.5

5. Завершіть роботу з проєктом і середовищем програмування, зберігши всі зміни.

12.4. Як виконувати дії над даними у списку?

Засобами мов програмування можна автоматизувати процес знаходження підсумкових даних: суми, добутку, середнього значення набору числових даних. Такі задачі передбачають, що табличні елементи змінюватись не будуть, а результатом їх виконання буде одне підсумкове значення.

У програмах, де передбачені дії з елементами табличної величини, що задовольняють деяким умовам, використовують оператор розгалуження *if*. Опис умов здійснюють так само, як і з простими змінними. Для запису простих умов використовують оператори порівняння $:$, $<$, $<=$, $>=$, $=$, $<>$, а складені умови формують із простих з використанням логічних операторів *and*, *or*, *not*.

Для обчислення значення суми елементів деякого списку із числових даних можна використати функцію *sum(list)*.

Наприклад,

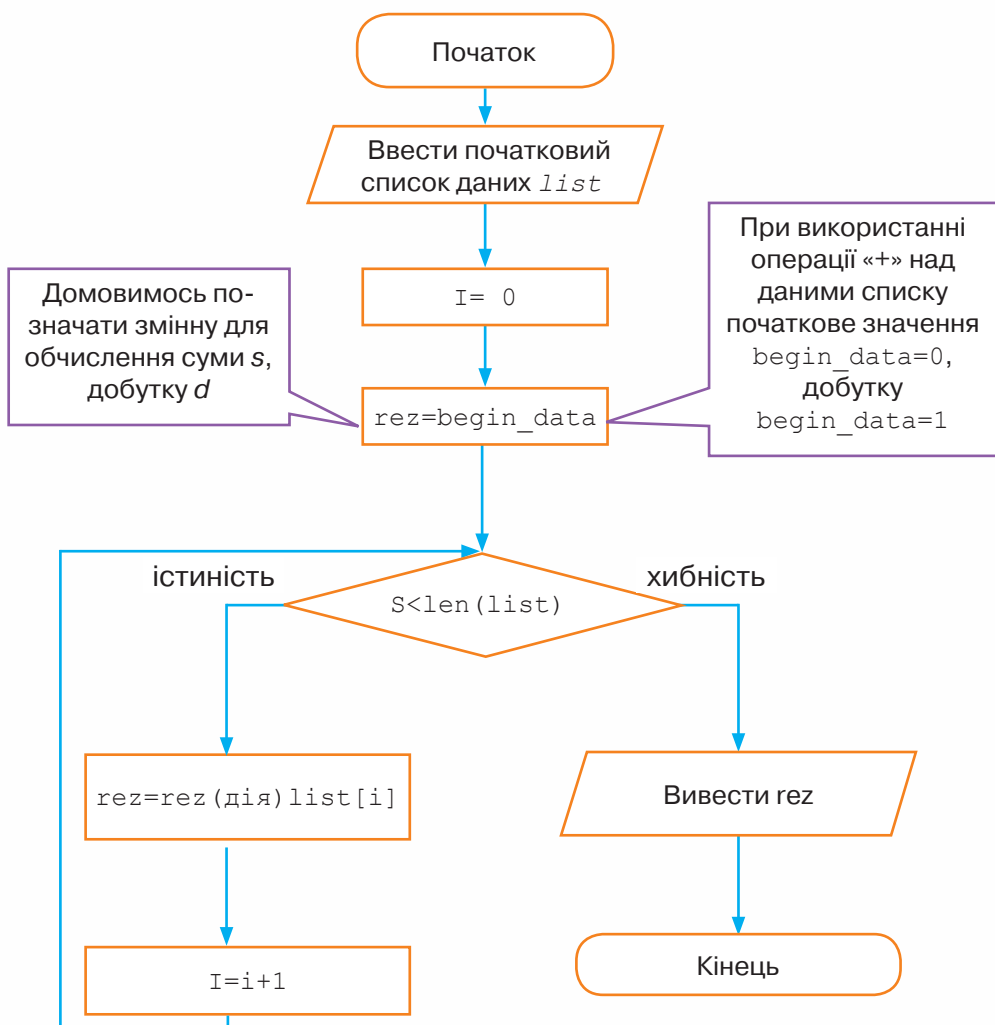
```
# Заданий список
A = [ 1, 2, 3, 4, 5, 6, 7, 8, 9]
s = sum(A)
print("Сума s =", s)
```

Отримаємо результат Сума s=45.

Для виконання інших дій над даними списку у програмі потрібно передбачити їх перебір (мал. 12. 6).

Каса 3	#
Чек 730429, Z 1872	#
Нектар 0,95	
Вишня тетра-пак	35,30 A
Хліб 350 г	
половинка нарізаний	13,90 A
Папір туалетний 1 шт	
2 X 7,70	15,40 A
Пластівці 400 г	10,50 A
Сметана 20% 350 г	26,99 A
Цибуля ріпчата жовта	
0,692 X 11,89	8,23 A

СУМА	110,32 грн



Мал. 12.6

Вправа 5. Середнє значення.

Завдання. При виконанні лабораторного експерименту з фізики учні отримали десять дійсних чисел з діапазону 9,0–10,0. Складіть проект *Середнє* для обчислення середнього значення результатів експерименту.

1. Відкрийте середовище програмування.
2. Створіть новий файл програми мовою програмування *Python* з іменем *Середнє* в папці *Навчальні проекти* своєї структури папок.
3. У вікні редактора коду введіть команди (мал. 12.7). Врахуйте, що для округлення дійсного числа можна застосувати команду `round(<величина>, <порядок>)`, яка не потребує підключення бібліотеки *math*.

```

# імпорт модуля генерування випадкових чисел
import random
# створюємо порожній список
a = []
# надаємо змінній для обчислення середнього початкового значення
s=0
for i in range (10):
    # додаємо введені елементи до списку в області виконання проекту
    a.append(round((random.random()+9.0),2))
    # знаходимо суму елементів списку
    s=s+a[i]
# знаходимо середнє значення елементів списку
s=s/10
# виводимо згенерований список
print(a)
# виводимо середнє значення
print(round(s,2))

```

■ Мал. 12.7 ■

4. Запустіть проєкт на виконання. Перевірте, чи відповідають отримані результати умові завдання. Наприклад, при одному із запусків програми отримано такий результат (мал. 12.8).

```

[9.02, 9.36, 9.13, 9.91, 9.54, 9.82, 9.75, 9.76, 9.75, 9.79]
9.58

```

Process finished with exit code 0

■ Мал. 12.8 ■

5. Якщо є помилки, виправте їх. Завершіть роботу з проєктом та середовищем програмування, зберігши всі зміни.

Вправа 6. Сума парних.

Завдання. Обчисліть суму парних елементів списку *a*, заповненого випадковими числами.

1. Відкрийте середовище програмування.
2. Створіть новий файл програми мовою програмування *Python* з іменем *Сума_з_умовою* в папці *Навчальні проєкти* своєї структури папок.
3. У вікні редактора коду введіть команди (мал. 12.9).

```

1 import random
2 n=9
3 amax=99
4 a=[]
5 for j in range(n):
6     a.append(random.randint(1,amax))
7 print(a)
8 print('Сума парних чисел з поданих дорівнює')
9 s=0
10 for j in range(n):
11     if a[j]%2==0:
12         s+=a[j]
13 print(s)
14

```

■ Мал. 12.9 ■

4. Запустіть проект на виконання. Поясніть призначення змінних n , $amax$, s (мал. 12.11).
5. Завершіть роботу з проектом і середовищем програмування, зберігши всі зміни.



ДОСЛІДЖУЄМО

Завдання. Дослідіть, як змінити код програми *Сума парних*, якщо потрібно у вікно повідомлення вивести список даних і результат.



ПРАЦЮЄМО В ПАРАХ

1. У середовищі програмування створіть проекти, у яких будуть реалізовані такі завдання:
 - а) п'ятому елементу списку A з 10 елементів рядкового типу надати значення "123";
 - б) першому елементу списку A з 10 елементів цілого типу надати значення восьмого елемента цього ж списку;
 - в) елементові з номером і списку A дійсного типу надати значення 15,5;
 - г) сьомому елементу списку $Nomer$ з 10 елементів цілого типу надати деяке конкретне значення;
 - д) другому елементу списку $Nomer$ з 10 елементів дійсного типу надати значення вмісту цієї ж клітини, збільшене вдвічі.
 Перевірте записи одне в одного. Обговоріть, чи можуть відрізнятися фрагменти програм.

-  2. Обговоріть, як зміниться програмний код у проєктах *Квадрати значень*, *Середнє значення* якщо:
- а) вхідні дані будуть подані величинами іншого типу;
 - б) вхідні дані будуть генеруватись випадково;
 - в) елементи управління з бібліотеки `tkinter`: інформаційне вікно, текстове поле, напис тощо.
- Щоб перевірити власні припущення, внесіть відповідні зміни до кожного з проєктів. Запустіть їх на виконання та збережіть з іменами *Квадрати значень* новий і *Середнє значення* новий у папці Проєкти своєї структури папок.
-  3. Обговоріть, як змінити програму в завданні на заходження середнього значення (мал. 12.7), щоб скористатись вбудованими функціями для роботи зі списками мови програмування *Python*.
-  4. Попрацюйте в мініпроєкті «Обираємо майбутню професію». Кожен обирає певний професійний напрям, наприклад, «людина-техніка». У середовищі програмування створіть проєкт *Вибір*, у якому задано список якостей, що необхідні для роботи за обраним напрямом. Користувач чи користувачка проєкту має отримати відповідь, чи є вказана якість важливою для даної професії. Обговоріть, як із частин проєкту, підготовлених його учасниками й учасницями створити програму, яку можна використовувати для визначення свого вибору майбутнього.



ПРАЦЮЄМО САМОСТІЙНО

-  5. У середовищі програмування створіть проєкт *Надбавка*, у якому заробітна плата 10 працівників деякого підприємства, подана списком з 10 дійсних чисел, збільшується на 15 %.
-  6. У середовищі програмування створіть проєкт *Продаж товару*, у якому для введеної кількості проданих одиниць товару, що фіксується кожну годину за зміну (8 год), знаходять підсумкове значення.
-  7. У вибраному середовищі програмування створіть проєкт *Підрахунок*, у якому знаходять суму додатних та кількість від'ємних елементів лінійної таблиці дійсного типу.
-  8. У вибраному середовищі програмування створіть проєкт *Виробництво*, за допомогою якого можна знайти сумарне навантаження на його стрічки за умови: на конвеєрі працюють по 5 робітників на трьох лініях, на стрічку поступають фрукти, маса яких задається випадково.



ПРИГАДАЙТЕ:

- як описати команду повторення мовою програмування *Python*;
- як використовують елемент керування «прапорець» у програмах із графічним інтерфейсом.

ВИ ДІЗНАЄТЕСЯ:

- як описати програму пошуку заданого елемента;
- як описати програму пошуку елемента з найбільшим/найменшим значенням.



ВИВЧАЄМО

13.1. Як описати програму пошуку заданого елемента?

При опрацюванні наборів даних часто розв'язують завдання пошуку елемента, який відповідає деякій умові. Ми вже вміємо знаходити індекс елемента у списку та кількість його входжень. Але якщо елементів у списку декілька, то метод *index()* дає змогу знайти індекс тільки першого.

Для того щоб перевірити наявність заданого елемента в списку, можна скористатись операцією *in* (мал. 13.1).

```
1 # Операція in - визначення наявності елемента в списку
2 A = [ -1.5, 7, 8.5, -100 ]
3 # Використання операції in
4 item = 8.5 # шуканий елемент
5 b = item in A # b = true
6 print("b = ",b)
7
```

Мал. 13.1

Результат виконання програми

b = True

Для виконання програм пошуку заданого елемента можна використати цикл та виконати такі дії.

1. Уводять спеціальну логічну величину — **прапорець**, призначення якої полягає в тому, що її значення зберігатиме результат наявності чи відсутності такої величини в списку. Початкове значення цієї величини — *false*, яке означає, що шукане значення поки що не траплялося у списку.

2. Переглядають усі елементи списку, починаючи з першого, доти, доки не буде переглянуто всі елементи або знайдено елемент із потрібним значенням.

3. Якщо елемент знайдено, то змінюють значення прапорця на *true* та переривають виконання циклу.

4. Здійснюють виведення результату залежно від значення прапорця.

Якщо потрібно не просто визначити наявність шуканого елемента, а його номер у масиві чи списку даних, то вводять змінну *flag* для фіксування такого номера. Початкове значення *flag* присвоюємо 0.



ДІМО

Вправа 1. Пошук номера елемента.

Завдання. Створіть проєкт *Номер елемента*, у якому здійснюють пошук індексу елемента списку з 10 цілих чисел, значення якого дорівнює її першому елементу.

1. Відкрийте середовище програмування.
2. Створіть новий файл програми мовою програмування *Python* з іменем *Номер елемента* в папці *Навчальні проєкти* своєї структури папок.
3. У вікні редактора коду введіть команди введення значень елементів масиву, знаходження номера елемента та виведення його значення або повідомлення, що такого елемента не існує (мал. 13.2).

```
a=[]
flag=0;
for i in range(10):
    a.append(int(input('Введіть елемент списку')) )
for i in range(9):
    if a[i+1]==a[0]:
        flag=i+1
        break
print(a)
if flag>0:
    print(flag)
else:
    print('Елемент не існує')
```

Мал. 13.2

4. Запустіть проєкт на виконання. Перевірте на контрольному прикладі, дібраному самостійно, чи відповідають отримані результати умові завдання. Якщо є помилки — виправте їх.
5. Завершіть роботу з проєктом та середовищем програмування, зберігши всі зміни.

Вправа 2. Температура.

Завдання. У шкільній метеостанції ведеться спостереження за температурою повітря упродовж дня. Датчики фіксують температуру та надсилають для програмного опрацювання у вигляді послідовного списку значень. Створіть програму для знаходження кількості отриманих значень, які відповідають першому значенню у списку.

1. Відкрийте середовище програмування.
2. Створіть новий файл програми мовою програмування *Python* з іменем *Температура* в папці *Навчальні проекти* своєї структури папок.
3. У вікні редактора коду введіть команди (мал. 13. 3).

```
1 a=[]
2 k=0
3 for i in range (24):
4     a.append(float(input('Отримана температура')))
5 for i in range (1,24):
6     if a[i]==a[0]:
7         k+=1
8 print (k)
```

■ Мал. 13.3 ■

4. Запустіть проект на виконання. Перевірте на контрольному прикладі, дібраному самостійно, чи відповідають отримані результати умові завдання. Якщо є помилки, виправте їх.
5. Завершіть роботу з проектом і середовищем програмування, зберігши всі зміни.

13.2. Як описати програму пошуку елемента з найбільшим/найменшим значенням?

У мові програмування *Python* для пошуку елемента з найбільшим значенням використовують функцію *max(list)*, з найменшим — *min(list)*.

Наприклад, результатом виконання програми (мал. 13.4) будуть значення: найбільше значення у списку 15, а найменше — 9, що легко перевірити із вказаного набору даних.

■ Мал. 13.4 ■

```
1 a=[10, 12, 14, 9, 15, 11]
2 a_max=max(a)
3 a_min=min(a)
4 print ('Найбільше значення у списку ', a_max)
5 print ('Найменше значення у списку ', a_min)
```

Для того щоб зрозуміти, як відбувається пошук найбільшого елемента деякого набору даних, уявімо себе на місці виконавця алгоритму. Елементи списку записано в пам'яті, доступ до комірок якої здійснюється за номерами елементів. Виконавець може одночасно відкривати доступ до однієї комірки. Нехай задано список з *n* цілих чисел (мал. 13.5, с.151).

Крок	Елементи					
	$a[1]$	$a[2]$	$a[3]$	...	$a[n]$	max
1	5					5
6 > 5						
2		6				6
4 < 6						
3			4			6
...	9 > 6					...
n					9	9

Мал. 13.5

Виконаємо такий алгоритм:

1) прочитаємо з пам'яті перший елемент списку. Його значення дорівнює 5. Запам'ятаємо його як максимальне — надамо його значення величині max ;

2) прочитаємо другий елемент списку. Його значення більше за max , тому «забудемо» про попереднє значення і запам'ятаємо значення $max=6$;

3) прочитаємо третій елемент списку. Його значення менше за max , тому можна приступати до наступного кроку без зміни значення max .

Продовжимо таким чином кроки алгоритму, допоки не будуть переглянуті усі елементи списку до n включно. На кожній ітерації циклу у змінній max міститиметься найбільше значення з пройденої частини списку, а по завершенні циклу змінна max зберігатиме максимальне значення в усьому списку.

Пошук найменшого елемента списку здійснюють за аналогічним алгоритмом, знаходячи відповідно елемент (min), який менший від усіх переглянутих елементів.



ДІЄМО

Вправа 3. Найменший елемент списку.

Завдання. Складіть програму знаходження найменшого елемента списку без використання функції min .

1. Скористайтесь запропонованою ідеєю: змінній a_min — найменшому значенню з проглянутих — спочатку надають значення елемента списку з найменшим номером. Послідовно переглядаючи значення наступних елементів списку при виявленні значення, меншого від a_min , надаємо змінній a_min цього значення.
2. Проаналізуйте, чи подана програма (мал. 13.6) реалізовує описану ідею.


```

1 import random
2 n=9
3 amax=99
4 a=[]
5 for j in range(n):
6     a.append(random.randint(1,amax))
7 print(a)
8 print('Найменше з поданих чисел дорівнює')
9 a_min=a[0]
10 for j in range(n):
11     if a[j]<a_min:
12         a_min=a[j]
13 print(a_min)

```

Мал. 13.6

3. Відкрийте середовище програмування.
4. Створіть новий файл програми мовою програмування *Python* з іменем *Найменший_елемент* у папці *Навчальні проекти* своєї структури папок.
5. У вікні редактора коду введіть обрані команди з програми (мал. 13.6).
6. Запустіть проект на виконання. Перевірте на контрольному прикладі, дібраному самостійно, чи відповідають отримані результати умові завдання. Якщо є помилки — виправте їх.
7. Завершіть роботу з проектом та середовищем програмування, зберігши всі зміни.

Вправа 4. Робот.

Завдання. Робот еколог рухається населеним пунктом та фіксує стан забруднення території важкими металами. Дані на обробку подаються списком. На першому місці списку вказують, якою стороною вулиці рухався робот: з парними номерами (0) чи з непарними (-1). Далі — покази замірів. Складіть програму, за допомогою якої можна визначити номер прибудинкової території, де зафіксовано найбільше забруднення.

1. Проаналізуйте умову задачі. Серед прикладів списку оберіть той, який можна використати для тестування своєї програми:
 $A=[\text{'0'}, 5.5, 21.5, 60.5, 4.3, 45.1]$
 $A=[0, 5.5, 21.5, 60.5, 4.3, 45.1]$
 $A=[-1, 5.5, 21.5, 60.5, 4.3, 45.1]$
2. Міркуйте так: якщо робот рухається по парній стороні, то елемент списку з індексом 1 має номер 2, з індексом 2 — номер 4, з індексом 3 — відповідно 6... Отже, номер будинку на парній стороні визнача-

ється, як подвоєний індекс елемента списку. Якщо робот рухається по непарній стороні, то номери будуть визначатись за правилом: 1-й індекс — номер 1, 2-й індекс — номер 3, 3-й індекс — номер 5. Отже, номер будинку на непарній стороні визначається, як подвоєний індекс елемента списку без 1.

3. Відкрийте середовище програмування.
4. Створіть новий файл програми мовою програмування *Python* з іменем *Робот* у папці *Навчальні проекти* своєї структури папок.
5. У вікні редактора коду введіть команди (мал. 13. 7). Поясніть призначення команд у кожному рядку програми.

```
1 A=[0, 5.5, 21.5, 60.5, 4.3, 45.1]
2 a_max=max(A)
3 n = A.index(a_max)
4 if A[0]==0:
5     k=n*2
6 if A[0]==-1:
7     k=n*2-1
8 print ('Номер будинку ', k)
9 |
```

■ Мал. 13.7 ■

6. Запустіть проєкт на виконання. Перевірте на контрольному прикладі, дібраному самостійно, чи відповідають отримані результати умові задання. Якщо є помилки — виправте їх.
7. Завершіть роботу з проєктом та середовищем програмування, зберігши всі зміни.



ПРАЦЮЄМО В ПАРАХ

1. Обговоріть, чим відрізняється алгоритм пошуку найменшого елемента лінійної таблиці від алгоритму пошуку найбільшого.



2. Обговоріть, як зміниться програмний код у проєкті *Номер елемента*, якщо:

- а) розглядати список дійсних чисел та здійснювати пошук елемента, який дорівнює останньому елементу;
- б) елементи списку генеруються випадково.

Внесіть відповідні зміни в проєкт *Номер елемента*, збережіть зміни в проєкт з новим іменем у папку *Проекти* власної структури папок, запустіть проєкт на виконання та перевірте своє припущення.



3. Обговоріть, як зміниться програмний код у проєкті *Випробування*, якщо:

- а) знаходити різницю між найбільшим елементом списку та середнім значенням усіх елементів;
- б) знаходити найменший елемент серед елементів списку з парними індексами.

Внесіть відповідні зміни в проєкт *Випробування*, збережіть зміни в проєкт із новим іменем у папку *Проєкти* своєї структури папок, запустіть проєкт на виконання та перевірте своє припущення.



ПРАЦЮЄМО САМОСТІЙНО

- 4. У середовищі програмування створіть проєкт *Спортивні результати*, у якому за поданим переліком результатів випробувань на тренуванні зі стрибків у довжину знаходять номер першої спроби, яка повторює контрольний результат.
- 5. У вибраному середовищі програмування створіть проєкт *Прогноз погоди*, у якому за поданою таблицею температури протягом тижня знаходять найбільший і найменший показники температури.
- 6. У середовищі програмування створіть проєкт *Швидкий пошук* для знаходження у списку з 10 дійсних чисел елемента, що відповідає заданій умові.
- 7. У фігурному катанні загальна оцінка якості виконання програми виступу обчислюється за такими правилами: відкидаються найвища й найнижча оцінки, а з решти обчислюється середнє арифметичне.

Складіть програму для визначення оцінки за цими правилами.

КОРИСНЕ ПОСИЛАННЯ

Інтерактивний підручник *Пітонтьютор* <https://cutt.ly/LT26SN7>

ЦІКАВО

Англійська вчена Ада Лавлейс (Ada Lovelace) відома створенням опису першої електронно-обчислювальної машини, проєкт якої було розроблено Чарльзом Беббіджем (Charles Babbage), і написанням першої програми для неї в 1840-х. Програмою став алгоритм обчислення чисел Бернуллі, саме завдяки йому дівчина вважається першою програмісткою в історії. Крім того, саме Ада ввела в ужиток терміни «цикл» і «робоча комірка».

Українка, що першою у світі створила мову програмування для комп'ютера, — Катерина Логвинівна Ющенко. Саме її розробка «Адресна мова програмування» випередила мову Фортран на 2 роки, Кобол — на 3 й Алгол — на 5 років.

ПРИГАДАЙТЕ:

- вилучити елемент у списку засобами мови *Python*;
- як перевести текст із нижнього регістру у верхній засобами мови *Python*.

ВИ ДІЗНАЄТЕСЯ:

- як упорядковувати дані в деякому наборі;
- які методи упорядкування можна використати в мові програмування *Python*;
- що розуміють під поняттям «складність алгоритму».



ВИВЧАЄМО

14.1. Як упорядковувати дані в деякому наборі?

Для розв'язування багатьох задач зручно спочатку впорядкувати дані за певною ознакою. Наприклад, пошук елемента в списку можна значно прискорити, якщо відповідні дані впорядковано. При цьому ознакою такого впорядкування може бути за зростанням (якщо значення елементів не повторюються), за неспаданням (якщо значення елементів можуть повторюватись), за спаданням, за незростанням.

Правило (ознака), за яким виконують впорядкування елементів, називають **ключем впорядкування**. У словниках ключами є самі слова, впорядковані в лексикографічному порядку (тобто у відповідності до порядку літер в алфавіті). Список учнів за ключем, що відповідає їх номеру в алфавітній книзі школярів. Дати, як правило, впорядковуються за ключем «rrrrr.мм.дд», де rrrrr — рік, мм — місяць, дд — день. Основним під час організації впорядкування є визначення відношення порядку на множині елементів, яка впорядковується, тобто для будь-яких двох елементів цієї множини важливо визначити, який з них слідує за іншим, передує іншому або що вони співпадають.

Існує багато різних методів впорядкування, які відрізняються один від одного ступенем ефективності. Ступінь ефективності враховує кількість порівнянь та кількість обмінів, які виконано під час впорядкування: що меншою є така кількість, то ефективнішим є метод впорядкування.

Розглянемо один з методів впорядкування списку — **метод вибору**. Уявімо, що дані містяться в таблиці. За таким методом спочатку з набору з довільним розташуванням елементів вибирають елемент із найменшим значенням і виконують його взаємозаміну із значенням в першій клітинці таблиці — таким чином у першій клітинці таблиці розташовується найменше значення вмістів клітинок таблиці. Далі знаходять еле-

мент із найменшим значенням з решти $n-1$ елементів і виконують його взаємозаміну з вмістом клітинки з номером 2, і т. д. Потім розглядаються елементи, що лишилися, серед яких знову знаходять найменший, який потім міняють місцями з вмістом третьої клітинки. Таким чином, для прикладу таблиці з 5 елементів, послідовно розглядають чотири різні набори даних (чотири таблиці, що мають різну довжину): у першому наборі було п'ять елементів, у другому — чотири, у третьому — три, у четвертому — два. З кожним набором елементів виконуються однакові дії:

- в наборі вибирається найменший елемент, запам'ятовується його номер у такому наборі (таблиці);
- знайдений найменший елемент міняють місцями з першим елементом набору, що розглядається.

Наприклад, упорядкування даних у таблиці з п'яти цілих чисел продемонстровано на малюнку 14.1, де жовтим кольором виділено найменший елемент серед елементів, що залишаються для перегляду на кожному кроці, стрілками — порядок обміну елементами.

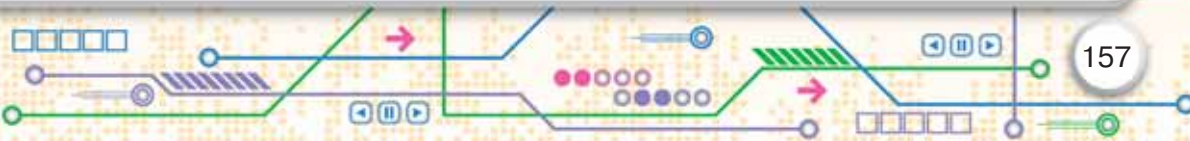
Елементи					
Кроки	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$
	12	8	10	2	6
1	2	8	10	12	6
2	2	6	10	12	8
3	2	3	8	12	10
4	2	3	8	10	12

Мал. 14.1

Зверніть увагу, що хоча таблиця має 5 елементів, достатньо 4 рази знайти найменше значення елементів з іще не впорядкованої частини лінійної таблиці та обміняти його місцями зі значенням першого зі ще не впорядкованої частини масиву елементів.

Для впорядкування набору даних від найменшого до найбільшого можна скористатись ідеєю створення нового списку, у який по черзі буде з вихідного списку вилучатись найменше та додаватись до нового.

Розглянемо алгоритм на прикладі, де a — вихідний список, b — новий, упорядкований:



1. `a=[5, 2, 4, 3]; b=[1]`
2. `a=[5, 4, 3]; b=[1, 2]`
3. `a=[5, 4]; b=[1, 2, 3]`
4. `a=[5]; b=[1, 2, 3, 4]`
5. `a=[]; b=[1, 2, 3, 4, 5]`

У результаті виконання відповідної програми (мал.14.2) отримуємо результат:

```
a = []
b = [1, 2, 2, 4, 5]
```

```
1 # Задано список
2 a = [5, 2, 1, 4, 2]
3 # Утворюємо порожній список
4 b = []
5 # Опрацьовуємо списки
6 for i in range(len(a)):
7     b=b+[min(a)] # У список b додаємо найменший
8     a.remove(min(a)) # Із списку a видаляємо найменший
9 # Виводимо результат
10 print("a =", a)
11 print("b =", b)
12
```

■ Мал. 14.2



ДІЄМО

Вправа 1. Упорядкування списку.

Завдання. Створіть проект *Упорядкування*, у якому елементи списку з 10 цілих чисел впорядковуються за зростанням.

1. Відкрийте середовище програмування.
2. Створіть новий файл програми мовою програмування *Python* з іменем *Упорядкування* в папці *Навчальні проекти* власної структури папок.
3. У вікні редактора коду запишіть команди введення значень елементів списку, його упорядкування та виведення списку у вікні виконання програми. Використайте у проекті змінні: *a* — список цілих чисел, *j* — номер ітерації пошуку мінімального елемента списку, *i* — номер елемента списку (мал. 14.3, с. 159).

```

1 import random
2 n=9
3 amax=99
4 a=[]
5 for j in range(n):
6     a.append(random.randint(1,amax))
7 print(a)
8 for j in range(n-1):
9     for k in range(j+1,n):
10        if a[j]>a[k]:
11            a[j],a[k]=a[k],a[j]
12 print('Результат впорядкування цього масиву такий')
13 print(a)
14

```

Мал. 14.3

4. Запустіть проект на виконання. Якщо є помилки, то виправте їх.
5. Завершіть роботу з проектом та середовищем програмування, зберігши всі зміни.

14.2. Які методи упорядкування можна використати в мові програмування *Python*?

У мові програмування *Python* для сортування списку використовують метод *sort* (). За замовчуванням метод сортує елементи списку в порядку зростання значень. Метод може змінити порядок сортування за допомогою таких іменованих аргументів:

- *key* — аргумент, який дає змогу визначити власну функцію порівняння при виклику методу *sort* () (мал. 14.4). Ця функція отримує один єдиний аргумент і повертає значення, яке буде використовуватися в операції порівняння;
- *reverse* — аргумент, який використовується для вказівки порядку сортування елементів. Якщо *reverse = True*, то елементи списку сортуються в порядку спадання.

```

1 # Заданий список
2 A = [ 'a', 'f', 'v', 'd', 'n', 'b' ]
3 # Сортування списку
4 A.sort()
5 B = [ 1, 3, 5, 10, 2, 8 ]
6 B.sort()
7 print("A =", A)
8 print("B =", B)
9

```

Мал. 14.4

Наприклад, результатом виконання програми з використанням методу `sort()` (мал. 14.4) є списки `A = ['a', 'b', 'd', 'f', 'n', 'v']` та `B = [1, 2, 3, 5, 8, 10]`.

Для того щоб використовувати метод `sort()`, всі елементи списку мають мати однаковий тип. Наприклад, такий код (мал. 14.5)

```
1 # Помилка!  
2 C = ["Hello", "ABC", 7]  
3 C.sort()
```

■ Мал. 14.5 ■

згенерує помилку.

`TypeError: '<' not supported between instances of 'int' and 'str'`

Для того, щоб список відсортувати в порядку спадання, додамо аргумент `reverse` (мал. 14.6).

```
1 # Заданий список  
2 A = [ 'a', 'f', 'v', 'd', 'n', 'b']  
3 # Сортування списку  
4 A.sort(reverse = True)  
5 B = [1, 3, 5, 10, 2, 8]  
6 B.sort(reverse = True)  
7 print("A =", A)  
8 print("B =", B)  
9
```

■ Мал. 14.6 ■

Отримаємо

```
A = ['v', 'n', 'f', 'd', 'b', 'a']  
B = [10, 8, 5, 3, 2, 1]
```

Для демонстрації використання ключа `key` в методі сортування розглянемо задачу. Нехай маємо список слів «мама», «тато», «Оксана», «Тарас». Відсортуємо список за зростанням та спаданням, використавши для порівняння функцію переведення символу з нижнього регістру до верхнього `upper()`:

`'aBc' => 'ABC'`

У результаті виконання програми (мал. 14.7) маємо списки:

```
S= ['мама', 'тато', 'Оксана', 'Тарас']  
S2= ['мама', 'Оксана', 'Тарас', 'тато']  
S3= ['тато', 'Тарас', 'Оксана', 'мама']
```



```

1 # Заданий список рядків
2 S = ["мама", "тато", "Оксана", "Тарас"]
3
4 S2 = list(S)           # створити новий список
5 S2.sort(key = str.upper) # відсортувати за ключем key
6
7 S3 = list(S)           # ще один список
8 S3.sort(key = str.upper, reverse = True) # відсортувати за аргументами key і reverse
9
10 print("S=", S)
11 print("S2=", S2)
12 print("S3=", S3)

```

Мал. 14.7

Для зміни порядку елементів списку на зворотній у вже відсортованому списку використовують метод реверсування списку *reverse()* (мал. 14.8).

```

1 # Задані два списки
2 A = [1, 2, 3, 4, 5]
3 B = ['f', 'e', 'd', 'c', 'b', 'a']
4
5 # Реверсування списків
6 A.reverse()
7 B.reverse()
8
9 print("A =", A)
10 print("B =", B)

```

Мал. 14.8

Результат виконання програми

```

A = [5, 4, 3, 2, 1]
B = ['a', 'b', 'c', 'd', 'e', 'f']

```



ДІЄМО

Вправа 2. Перевезення.

Завдання. Для перевезення посилок у сервісі доставки кожний пакунок маркують за відстанню. Транспортний засіб для перевезення завантажують за таким порядком: спочатку ті пакунки, які перевозять на більшу відстань, потім — на меншу. Складіть програму для автоматизації відбору пакунків для завантаження.

1. Проаналізуйте умову задачі. Побудуйте її модель. Міркуйте так: дано деякий список із значеннями відстаней для транспортування. Результат: потрібно отримати список, упорядкований за спаданням.
2. Відкрийте середовище програмування.
3. Створіть новий файл програми мовою програмування *Python* з іменем *Перевезення* в папці *Навчальні проекти* своєї структури папок.

4. У вікні редактора коду складіть програму, яка реалізує алгоритм упорядкування списку. Для перевірки роботи програми використайте список [24.5, 17.3, 45.2, 151.6, 127.5].
5. Запустіть проект на виконання. Якщо є помилки, то виправте їх.
6. Завершіть роботу з проектом та середовищем програмування, зберігши всі зміни.

14.3. Що розуміють під поняттям «складність алгоритму»?

Часто під час складання програм ставлять перед собою питання: *Який час потрібний для виконання програми, що реалізує певний алгоритм? Чи можна взагалі отримати результати обчислення за даним алгоритмом на комп'ютері?* На подібні питання відповідає **теорія алгоритмів** — розділ інформатики, що займається дослідженням складності алгоритмів для розв'язування задач на основі формально визначених моделей обчислювальних пристроїв.

Що таке складність алгоритму? Інтуїтивно можна виділити такі основні складові складності алгоритму:

1. **Логічна складність** — кількість людино-місяців, витрачених на створення алгоритму.
2. **Статична складність** — довжина опису алгоритмів (кількість операторів).
3. **Часова складність** — час виконання алгоритму.
4. **Ємнісна складність** — кількість умовних одиниць пам'яті, необхідних для роботи алгоритму.

Складність алгоритму дає змогу визначитися з вибором ефективного алгоритму серед тих, що побудовані для розв'язування конкретної проблеми.



Складність алгоритму — це кількісна характеристика, що відображує споживані алгоритмом ресурси під час свого виконання.

Складність алгоритмів зазвичай оцінюють за часом виконання або за використовуваною пам'яттю. В обох випадках складність залежить від розмірів вхідних даних: список з 100 елементів буде опрацьовано швидше, ніж аналогічний з 1000. При цьому мова йде не про точний час обчислень, який залежить від процесора, типу даних, мови програмування тощо. Оцінюється складність при прагненні розміру вхідних даних до нескінченності.



Часова складність алгоритму — характеристика продуктивності алгоритму, що визначається кількістю елементарних операцій, які потрібно виконати для реалізації алгоритму.

При цьому вважають, що кожна елементарна операція виконується за однаковий час. Часову складність оцінюють для найгіршого випадку й визначають як максимальний час, необхідний для опрацювання алгоритмом будь-якої множини з n елементів. Часова складність алгоритму зазвичай визначається виразом $O(f(n))$ (або так званої O — нотації). Вираз $O(f(n))$ означає, що час виконання алгоритму зростає з тією самою швидкістю, що й функція $f(n)$.

Якщо час роботи алгоритму **не залежить від обсягу вхідних даних**, то його часову складність позначають як $O(1)$; приклад — визначення значення третього елемента списку, для чого не потрібно ні запам'ятовувати елементи, ні проходити по ним декілька разів. Завжди потрібно просто дочекатися в потоці вхідних даних третій елемент і це буде результатом, на обчислення якого для будь-якої кількості даних потрібний один і той же час.

Лінійна складність $O(n)$: подвоєння розміру задачі подвоїть і необхідний час; приклади — алгоритм пошуку найбільшого елемента в невідсортованому списку, для чого потрібно переглянути всі n елементів списку; алгоритм додавання/віднімання чисел з n цифр.

Квадратична складність $O(n^2)$: час роботи алгоритму зростає пропорційно квадрату кількості оброблюваних елементів, подвоєння розміру задачі вчетверо збільшує необхідний час; приклад — алгоритм сортування, що виконує два вкладені цикли перебору списку.

Кубічна складність $O(n^3)$: подвоєння розміру задачі збільшує необхідний час у вісім разів. Припустимо, що певним алгоритмом потрібно виконати $4n^3 + 7n$ умовних операцій, щоб опрацювати n елементів вхідних даних. При збільшенні n на час роботи буде значно більше впливати піднесення n до кубу, ніж множення його на 4 або ж додавання $7n$.



ПРАЦЮЄМО В ПАРАХ

1. Обговоріть, які ідеї закладено в танцях, що демонструють методи упорядкування наборів даних за посиланнями (відео надані Youtube-каналом *AlgoRythmics*)
<https://cutt.ly/iE3JOz3>
<https://cutt.ly/5E3JYes>
2. Обговоріть, який спосіб упорядкування набору даних подано у 3D-візуалізації (автор *Luciano Vernacchia*), яку збережено на спільному ресурсі в папці *Алгоритми та програми* (<https://cutt.ly/1E3XQmA>).
3. Обговоріть, як можна змінити програму у задачі *Упорядкування списку* (мал. 14.3), щоб список упорядковувався за спаданням. Складіть відповідну програму та перевірте її.



4. Обговоріть, як дослідити ефективність використання методів мови програмування *Python* й алгоритмів упорядкування списків. Чи можна результати обговорення подати в термінах поняття «складність алгоритму»?



ПРАЦЮЄМО САМОСТІЙНО

5. Розгляньте матеріал *Великої української енциклопедії*: https://vue.gov.ua/Алгоритм_сортування. Визначте, який внесок зробили українські дослідники в розробку алгоритмів сортування. Складіть повідомлення на шкільний сайт чи канал новин.
6. Розгляньте візуалізацію алгоритмів упорядкування за посиланням <https://cutt.ly/ME3JxDX>. Складіть презентацію за запропонованим матеріалом.
7. Розгляньте візуалізацію алгоритмів сортування. Порівняйте часову складність запропонованих алгоритмів.
8. У середовищі програмування складіть проект *Словник*, у якому слова, що вводяться користувачем, утворюють список, записаний в алфавітному порядку.
9. У середовищі програмування складіть проект *Черга*, у якому дані номерів, що присвоєні об'єктам із черги, будуть впорядковуватися за принципом:
 - 1) перший прийшов — перший пішов;
 - 2) перший прийшов — останній пішов.Сформулюйте правило проходження черги в термінах упорядкування списку та реалізуйте їх обраними засобами мови програмування.
10. У середовищі програмування складіть проект *Функція*, у якому буде формуватися список, що міститиме дані з вихідного списку в порядку спадання, за якими йдуть дані вихідного списку за зростанням.

ЦІКАВЕ ДЖЕРЕЛО

Алгоритми та структури даних для початківців: сортування
<https://echo.lviv.ua/dev/6908>

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Я можу пояснити принцип організації даних за допомогою одновимірних масивів.
- ☐ Я можу пояснити поняття масиву, елемента масиву, індексу та значення елемента.
- ☐ Я можу описати алгоритми опрацювання елементів масиву, що задовольняють певній умові.
- ☐ Я можу описати алгоритм знаходження підсумкових величин у масиві.
- ☐ Я можу описати принаймні один алгоритм упорядкування масиву.
- ☐ Я умію складати й описувати мовою програмування алгоритми для опрацювання елементів масиву, що задовольняють певну умову, знаходження підсумкових величин у масиві та його впорядкування.
- ☐ Я розумію у чому суть часової та ємнісної складності алгоритмів.
- ☐ Я усвідомлюю важливість застосування ефективних методів для опрацювання великих наборів даних.

БАЗИ ДАНИХ. СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ

15

ЩО ТАКЕ БАЗИ ДАНИХ ТА ЯКІ ПРОГРАМИ ВИКОРИСТОВУЮТЬ ДЛЯ РОБОТИ З НИМИ?

ПРИГАДАЙТЕ:

- що таке модель;
- етапи побудови комп'ютерної моделі;
- що таке електронна таблиця;
- як здійснюється доступ до клітинок електронної таблиці.

ВИ ДІЗНАЄТЕСЯ:

- що таке база даних;
- що таке система керування базами даних;
- які особливості вікна СКБД *Microsoft Access*;
- які об'єкти може містити база даних в СКБД *Microsoft Access*.



ВИВЧАЄМО

15.1. Що таке база даних?



База даних — це засіб збирання та впорядкування інформації.



Бази даних можуть зберігати відомості про людей, продукти, замовлення, книги, транспорт або будь-що інше. Багато баз даних починаються зі списку в текстовому редакторі або електронній таблиці. Оскільки список збільшується, то в даних стає важко розібратися, а також є обмежені способи пошуку або витягування деяких даних для перегляду. Уявіть собі, що ви хочете придбати квиток на деякий потяг на кон-

кретну дату. Якщо дані про потяги та квитки подати у вигляді списку в таблиці, то користувачеві складно буде не тільки знайти потрібний, а й підтвердити своє замовлення та роздрукувати квиток. Складно буде й поповнювати таку базу її адміністраторам, оскільки є ймовірність повторень даних чи їх втрати.

Існують різні означення поняття бази даних.

Наприклад, базу даних можна розглядати як різновид архіву, за допомогою якого зберігають великі обсяги однотипних даних, управляють ними, сортуючи та відбираючи їх так само, як книги або великі таблиці.

Одним із простих прикладів бази даних може бути телефонний довідник. Дані в ньому структуровані та відсортовані (за алфавітом), складаються з багатьох записів, які відображають окремі відомості для кожної людини (підприємства тощо).



База даних (БД) — це поіменована, структурована сукупність взаємопов'язаних даних, які належать до певної предметної області.

Бази даних створюють із певною метою, а для зручності опрацювання даних використовуються комп'ютери та спеціальні програми. Це дає змогу за потреби швидко змінювати дані й багаторазово використовувати їх під час розв'язування задач певної предметної області. Це може бути супермаркет, бібліотека, туристична агенція, довідкова служба, школа, аптека тощо. З базою даних ми працюємо під час покупки в онлайн-магазині: обираємо виробника, товар та вказуємо його конкретні характеристики.





Вправа 1. Навчальні заклади.

Завдання. Розгляньте інформаційну систему *Конкурс* (<https://vstup.osvita.ua/>). Встановіть які об'єкти містить база даних інформаційної системи на прикладі одного із навчальних закладів області, в якій ви проживаєте.

1. У браузері відкрийте інформаційну систему *Конкурс* (мал. 15.1). Оберіть регіон.

Мал. 15.1

Вступ.ОСВІТА.УА

Пошук навчального закладу:

Пошук спеціальності:

Реклама

Кількість заяв вступників за регіонами

Регіон	Бакалавр та магістр	Магістр	Фаховий молодший бакалавр	Молодший бакалавр	Всього
Київ	291128	44309	40982	943	321111
Львівська область	50524	4939	13325	241	54973
Вінницька область	17602	4132	9406	0	34139
Львівська область	46352	10633	31073	312	112908
Львівська область	9930	3362	9908	20	24694
Львівська область	18441	3744	11241	343	36300
Львівська область	14121	2573	4742	96	23125
Львівська область	26672	8191	14825	50	54955
Львівська область	56456	4984	14875	0	83415

2. У списку навчальних закладів регіону оберіть заклад, куди можуть вступати випускники 9 класу, – коледж, училище. Визначте, які відомості містить база даних на час перегляду. Наприклад у коледжі, в якому готують майбутніх агроінженерів (мал. 15.2), маємо інформацію:

Список абітурієнтів

Інформація за:

2021

2020

2019

Фаховий молодший бакалавр (на основі базова загальна середня освіта)
 Галузь: Аграрні науки та продовольство
 Спеціальність: 201 Агроінженерія
 Освітня програма: Агроінженерія
 Тип пропозиції: Основа
 Термін навчання: 1 курс
 Термін навчання: 01.09.2021 - 30.06.2025
 Ліцензійний обсяг: 34
 Обсяг на контракт: 19
 Обсяг держ замовлення: 15

1. Українська мова (залив: балів: 100, k=1.00)
2. Математика (залив: балів: 100, k=1.00)
3. Середній бал документа про освіту (кандидатський показник, k=1.00)
4. Бал за успішне закінчення підготовчих курсів закладу освіти (кандидатський показник, k=1.00)

Детальніше

Мал. 15.2

3. Перейдіть у детальний список абітурієнтів. Порівняйте структуру таблиці з інформацією про спеціальність та таблиці із списком абітурієнтів. Чи є у них однакові дані?
4. Визначте за обраною спеціальністю, як змінювався середній бал вступника та які вступні іспити складались.

15.2. Що таке база даних?

Для введення, зберігання, пошуку, сортування, опрацювання, аналізу й виведення даних, що зберігаються в БД, використовуються спеціальні прикладні програми, які отримали назву системи керування базами даних (СКБД).



Програмне забезпечення, яке призначене для створення баз даних, оновлення даних, що зберігаються в них, забезпечення зручного доступу до баз даних з метою перегляду й пошуку даних, називається **системою керування базами даних (СКБД)**.

Іншими словами, ці програми призначено для структурування даних, розміщення їх у таблицях та маніпулювання ними. При цьому робота з базою даних проводиться не безпосередньо, а через СКБД (мал.15.3)



Мал. 15.3

Сучасні системи керування базами даних дають змогу:

- опрацьовувати дані: здійснювати введення, добір, вставляння, оновлення, видалення даних тощо;
- здійснювати пошук і формування запитів;
- забезпечувати цілісність даних;
- забезпечувати захист даних від несанкціонованого доступу та апаратних збоїв тощо.

До найбільш розповсюджених типів СКБД належать: *ORACLE, MS SQLServer, Informix, Sybase, DB2, MS Access* тощо.

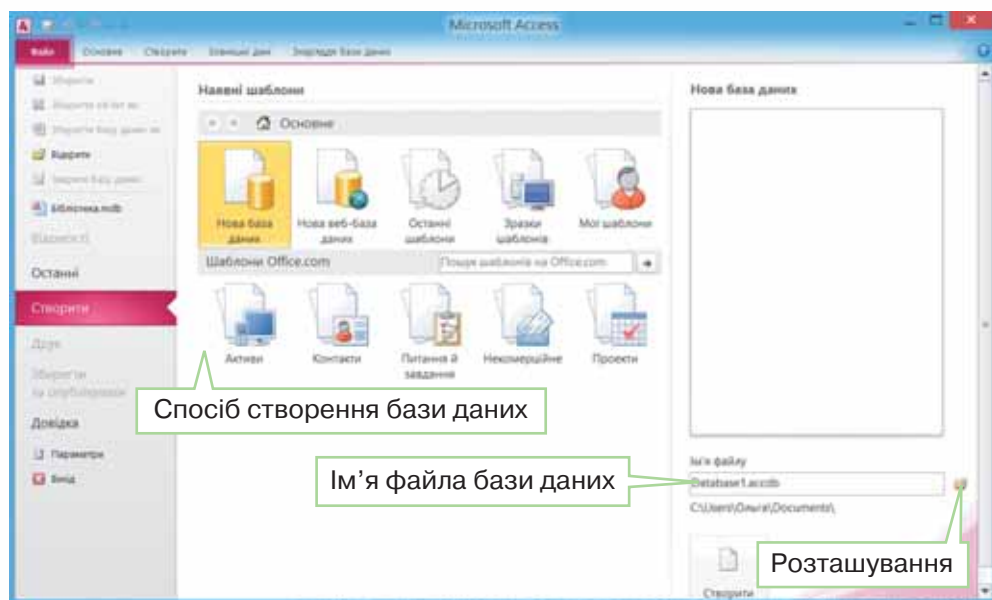
Розглянемо основи роботи з базами даних на прикладі СКБД *Microsoft Access*. *MS Access* — це система керування реляційними базами даних, яку призначено для роботи на автономному персональному комп'ютері або в локальній комп'ютерній мережі під управлінням операційної системи *Microsoft Windows*, і в якій збалансовано засоби й можливості, що є типовими для сучасних СКБД. СКБД *Microsoft Access* має потужні, зручні та гнучкі засоби візуального проектування об'єктів за допомогою майстрів, що дає змогу користувачеві швидко створити інформаційну систему на рівні таблиць, запитів, форм і звітів.

До основних функцій СКБД *Microsoft Access* належать такі:

- проектування базових об'єктів — двовимірні таблиці з полями, що містять дані різних типів;
- створення зв'язків між таблицями, з підтримкою цілісності даних, каскадного оновлення полів і каскадного видалення записів;
- введення, збереження, перегляд, упорядкування, зміна та добір даних з таблиць із використанням різних засобів контролю даних, індексування таблиць і засобів алгебри логіки;
- створення, модифікація та використання похідних об'єктів (запитів, форм і звітів).

15.3. Які особливості вікна СКБД *Microsoft Access*?

Інтерфейс користувача *Microsoft Access* — це комплекс програм, який реалізує діалог у процесі роботи користувача з програмою *Microsoft Access*. Після запуску програми *Microsoft Access* відкривається вікно програми з початкової сторінки *Файл/Створити* (мал. 15.4).



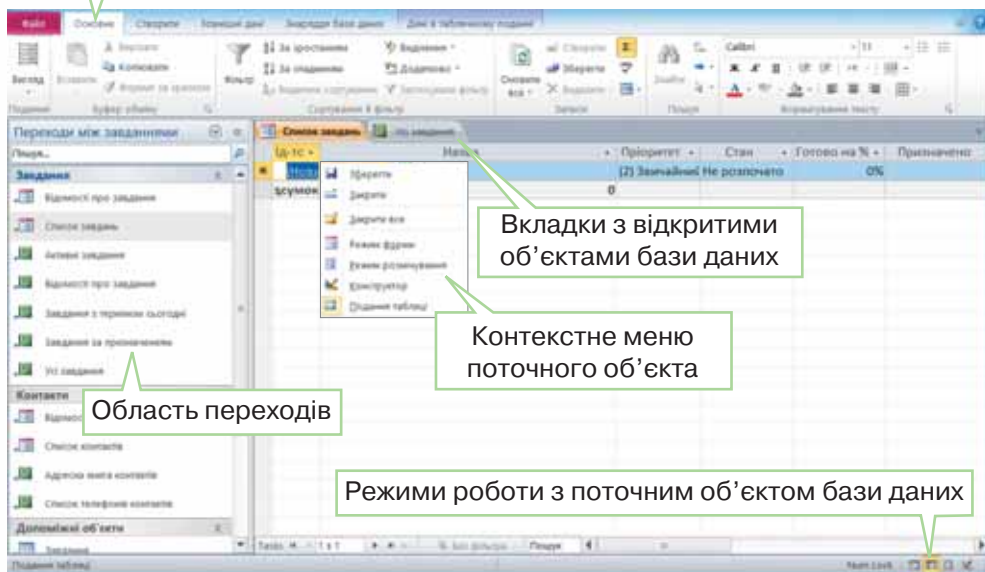
■ Мал. 15.4 ■

Тут вказують спосіб створення бази даних, ім'я бази даних та папку для зберігання.

Відкривають базу даних вказівкою *Файл/Відкрити* або безпосередньо з файла бази даних, який має розширення *accdb*.

Вікно системи керування базами даних має подібні елементи до вікон уже відомих вам програм з пакета *Microsoft Office* та відмінності, які пов'язані зі специфікою даної програми (мал. 15.5).

Стрічка, на якій відображаються команди активної вкладки СКБД



Вкладки з відкритими об'єктами бази даних

Контекстне меню поточного об'єкта

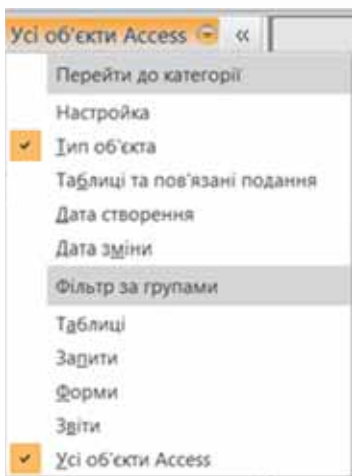
Область переходів

Режими роботи з поточним об'єктом бази даних

Мал. 15.5

15.4. Які об'єкти може містити база даних в СКБД Microsoft Access?

Бази даних, створені у форматі *MS Access*, мають розширення файла *accdb*. Така база даних може містити об'єкти різних категорій. У лівій частині вікна бази даних можна відобразити список категорій об'єктів (мал. 15.6). При виборі зі списку відповідної категорії у правій частині вікна бази даних відображаються всі доступні об'єкти цієї категорії, які можна відкрити для перегляду або редагування (мал. 15.7): таблиці, запити, звіти, форми, сторінки даних, макроси та модулі.



Мал. 15.6

Таблиця бази даних схожа на електронну таблицю — в обох дані зберігаються в рядках і стовпцях. Кожен рядок у таблиці називається записом. У записах зберігаються окремі елементи даних. Кожен запис складається з одного або кількох полів. Поля відповідають стовпцям у таблиці (мал. 15.7).

Ім'я поля 1	Ім'я поля 2	Ім'я поля 3	Ім'я поля 4
Запис		Поле	

Мал. 15.7

Наприклад, можна створити таблицю *Працівники*, де кожен запис (рядок) містить відомості про окремого працівника, а поля (стовпці) містять дані різного типу, наприклад ім'я, прізвище, адресу тощо. Полям має бути призначено певний тип даних (текст, дата або час, число чи інший тип). Щоб зрозуміти, що таке записи й поля, можна також уявити старий картковий каталог у бібліотеці. Кожна картка в шухляді відповідає запису в базі даних. Кожен елемент даних на окремій картці (автор, назва тощо) відповідає полю в базі даних.

Кожний запис має відрізнятися від інших значеннями **основного ключа** — певного поля або сукупності полів, що ідентифікують запис, роблять його унікальним.

Наприклад, у базі даних *Школа*, у таблиці *Список учнів* ключовим полем є шифр учня (табл. 15.1).

Таблиця 15.1

Шифр учня	Прізвище, ім'я по батькові	Рік вступу	Дата народження	Стать	Місце проживання	Батьки
Г-46	Гаврилишин Олександра Ярославівна	2015	31.01.2009	Ж	Вул. Миру, 21	Ірина Євгенівна, Ярослав Васильович
І-36	Ільчишин Михайло Зеновійович	2015	18.09.2009	Ч	Вул. Бо- кова, 18	Алла Василівна, Зеновій Михайло- вич

Запит — засіб для відбору даних, які задовольняють певним умовам.

За допомогою запитів можна дібрати з бази даних лише необхідні відомості або вносити зміни до таблиць бази даних в автоматичному режимі. Користувач може задавати запиту умови, відповідно до яких з бази вибираються дані, які задовольняють даним умовам, або формується нова таблиця, що містить дані з раніше створених таблиць відповідно до заданих умов (мал. 15.8).

Форми дають змогу створити інтерфейс користувача, у якому можна вводити й редагувати дані. Форми часто містять кнопки та інші елементи керування, використовуючи які, можна виконувати різні завдання.

Звіт — засіб, який дає можливість вибрати з бази потрібні відомості й подати їх у вигляді, зручному для сприйняття та перегляду. Звіт може бути надрукований або включений у документ іншої програми.

ЦІКАВО

При заповненні користувачем форми виконується заповнення даними однієї чи декількох таблиць бази. Коли користувач формує запит, тим само він також звертається до деякої сукупності таблиць.



Макрос — набір макровказівок, які створюються користувачем для автоматизації виконання конкретних операцій. Макроси запускаються натисненням декількох клавіш і швидко виконують найбільш уживані операції, наприклад, відкриття таблиць і форм, виконання вказівок меню, управління вікнами та ін.

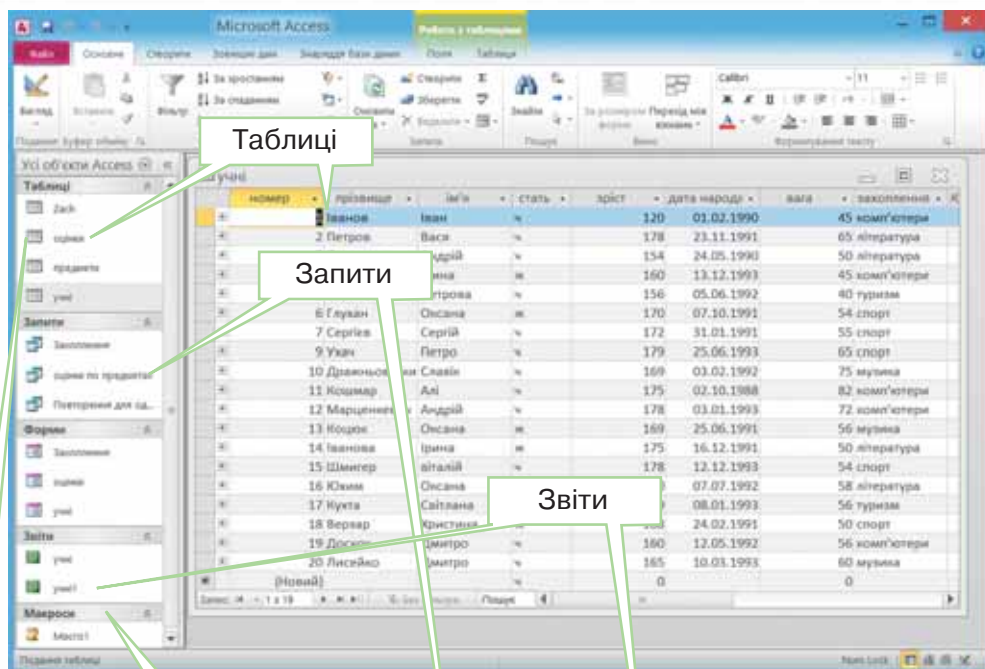
Наявність об'єктів усіх зазначених категорій у кожній базі даних не є обов'язковою, але більшість із них використовується майже в усіх базах даних. Кожний об'єкт, створений у базі користувачем, може бути змінений або видалений, що дає змогу швидко коригувати внутрішню будову бази відповідно до вимог користувача.

СКБД *MS Access* містить велику кількість вбудованих програм, що мають назву **майстри**: майстер таблиць, майстер кнопок, майстер форм тощо. За допомогою майстра здійснюється послідовне конструювання об'єкта відповідно до вказівок користувача.

ЦІКАВО

Для швидкого створення складних таблиць БД в *Access* можна використовувати вбудовану програму *Майстер створення бази даних*. *Майстер* для виконання завдання задає питання покроково в діалогових вікнах і використовує відповіді користувача.

В *MS Access* усі об'єкти бази даних мають візуальне відображення у вікні бази даних.



оцінки по предметах

код предмету	нікна	зніладан	код учня	оцінка
1	інформатика	Власенко Т.І.	Ульянов	12
1	інформатика	Власенко Т.І.	Іванов	8
1	інформатика	Власенко Т.І.	Петров	10
1	інформатика	Власенко Т.І.	Саранчук	10
1	інформатика	Власенко Т.І.	Медницька	8
1	інформатика	Власенко Т.І.	Глухан	8
1	інформатика	Власенко Т.І.	Сергієв	9
1	інформатика	Власенко Т.І.	Ухан	6
1	інформатика	Власенко Т.І.	Дражницька	9

оцінки

код оцінки:

код учня:

код предмету:

оцінка:

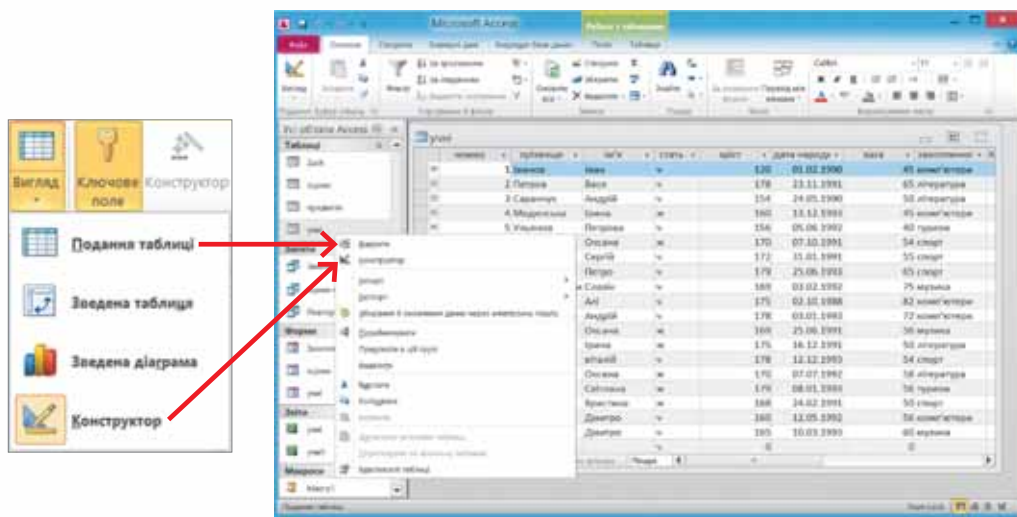
Запис: 1 з 154

учні

код учня	ім'я	прізвище	стать	вік	дата народи	вага	закорочення
1	Іванов	Іван	ч	120	01.02.1990	45	комп'ютери
2	Петров	Вася	ч	178	23.11.1991	65	література
3	Сидор	Олександр	ч	154	24.05.1990	50	література
4	Сидорова	Олена	ж	160	13.12.1993	45	комп'ютери
5	Сидорова	Олена	ж	156	05.06.1992	40	туризм
6	Глухан	Оксана	ж	170	07.10.1991	54	спорт
7	Сергієв	Сергій	ч	172	31.01.1991	55	спорт
8	Ухан	Петро	ч	179	25.06.1993	65	спорт
9	Дражницька	Славка	ж	169	03.02.1992	75	музика
10	Кочмар	Алі	ч	175	02.10.1988	82	комп'ютери
11	Марценко	Андрій	ч	178	03.01.1993	72	комп'ютери
12	Кордов	Оксана	ж	169	25.06.1991	56	музика
13	Іванова	Ірина	ж	175	16.12.1991	50	література
14	Шмидт	Віталій	ч	178	12.12.1993	54	спорт
15	Юхим	Оксана	ж	170	07.07.1992	58	література
16	Никита	Світлана	ж	168	08.01.1993	56	туризм
17	Вервар	Кристина	ж	170	24.02.1991	50	спорт
18	Доско	Дмитро	ч	160	12.05.1992	56	комп'ютери
19	Лисейко	Дмитро	ч	165	10.03.1993	60	музика
20	(Немає)			0		0	

Мал. 15.8

Майже всі об'єкти бази даних передбачають роботу з ними в одному з двох режимів — *подання та конструктора* (мал. 15.9).



Мал. 15.9

Більшість операцій із використання можливостей СКБД можна виконувати за допомогою пунктів меню програми та вказівок контекстного меню під час роботи з об'єктами, деякі з них — за допомогою кнопок панелі інструментів. Пункти меню можуть змінюватися залежно від об'єкта, який опрацьовується в даний момент, та режиму його відображення, наприклад, пункт *Робота з таблицями* з'являється в меню лише в тому разі, якщо відкрита таблиця бази даних.

У робочій області вікна *MS Access*, крім списку режимів створення об'єктів, відображаються створені об'єкти (наприклад, таблиці, форми тощо), які можна переглядати чи редагувати. Для цього необхідно виділити відповідний об'єкт, наприклад таблицю, та обрати вказівку *Вигляд* у вкладці *Файл*. Наприклад, коли виділена таблиця, то обрання у списку *Вигляд* вказівки *Подання таблиці* задає режим таблиці, у якому можна переглядати та редагувати дані у вибраній таблиці. Вказівка *Конструктор* відкриває таблицю в режимі конструктора, який призначено для перегляду та змін структури таблиці.



ДІЄМО

Вправа 2. Ознайомлення з об'єктами бази даних у різних режимах.

Завдання. Ознайомтеся з різними об'єктами бази даних *Магазини* в режимі перегляду та режимі конструктора.

1. Завайтайте базу даних *Магазини*, що міститься в папці *Бази даних* на спільному ресурсі (<https://cutt.ly/KE03T3P>). Збережіть файл у своїй структурі папок. Відкрийте файл бази даних.
2. Двічі клацніть на значку таблиці *Магазини*. Перегляньте вміст таблиці.



Натисніть кнопку  на панелі інструментів. Визначте кількість полів таблиці. Чи відображаються дані таблиці в режимі конструктора? Закрийте таблицю.



3. Виділіть таблицю *Підсумок* і натисніть кнопку .
4. Перейдіть у режим перегляду таблиці за допомогою кнопки . Закрийте таблицю.
5. Клацніть правою кнопкою миші на таблиці *Фільми* та виберіть у контекстному меню вказівку *Конструктор*. Закрийте таблицю.
6. Аналогічно перегляньте в режимах перегляду та конструктора інші об'єкти бази даних: запити, форми, звіти.
7. Визначте призначення даної бази.



ДОСЛІДЖУЄМО

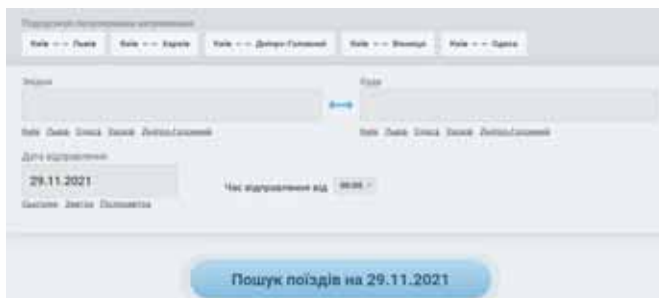
Завдання. Дослідіть Єдиний державний реєстр юридичних осіб, фізичних осіб-підприємців та громадських формувань (<https://usr.minjust.gov.ua/content/free-search>). Дайте відповідь на питання: з якими даними працює вказана база даних? Як їх подають? Для проведення дослідження дайте запит на пошук відомостей про фізичну особу-підприємця за прізвищем, іменем та по-батькові особи (особа відома вам або надає послуги вашому освітньому закладу, наприклад із організації харчування).



ПРАЦЮЄМО В ПАРАХ

1. Визначте спільне й відмінності таблиці в середовищі табличного процесора та таблиці бази даних. Обговоріть.
2. Хто може бути кінцевим користувачем баз даних? Наведіть приклади життєвих ситуацій. Як можуть класифікуватись користувачі БД? Обговоріть у парах.
3. Доведіть, що СКБД *MS Access* дійсно належить до стандартного пакета програм *MS Office*. Для цього порівняйте елементи вікна програми, функціональність, сумісність документів тощо. Хтось обирає СКБД *MS Access*, а хтось — одну з відомих програм з пакета *MS Office*, та спільно знаходять подібне.

4. Обговоріть, як можна організувати базу даних, якою будуть користуватись пасажери залізниці? Для прикладу візьміть інформаційну систему <https://booking.uz.gov.ua/> (мал. 15.10).



Мал. 15.10

5. Складіть опитувальник із 4–5 питань за матеріалами уроку, наприклад, засобами форм *Google*. Розмістіть посилання на опитувальник на спільному ресурсі або запропонуйте його трьом своїм однокласникам чи однокласницям. Перевірте отримані результати.
6. Обговоріть, які дані з навчальної діяльності, побуту або ваших захоплень можна подати у вигляді бази даних. У якому вигляді можна подати цю базу даних?



ПРАЦЮЄМО САМОСТІЙНО

7. Визначте, які основні недоліки має збереження даних у текстових документах у порівнянні з базами даних. Результати подайте у вигляді радіальної діаграми.



8. У бібліотеці запис про книжку Василя Барки «Жовтий князь» міститься у формулярі читача, бібліотечному каталозі, журналі реєстрації бібліотечних надходжень. Дослідіть, які документи найкраще переглянути, щоб дізнатись короткий зміст книжки, її ціну, отримати дані про характеристику героїв. Отримання яких даних можна було б автоматизувати?



9. Серед таблиць, поданих у файлі *Таблиці* в папці *Бази даних* на спільному ресурсі (<https://cutt.ly/oE031yB>), оберіть ту, яку можна використати для побудови бази даних. Поясніть свою відповідь.



10. Попрацюйте в мініпроекті «Ми — майбутні фермери». Відкрийте базу даних *Урожай*, що міститься в папці *Бази даних* на спільному ресурсі (<https://cutt.ly/AE08tAZ>). Визначте об'єкти, що складають базу даних, та їх призначення. Опишіть завдання, які можна розв'язувати, скориставшись даними цієї бази. Як можна змінити структуру цієї бази? Свої пропозиції надішліть електронним листом на поштову скриньку вчителя чи вчительки.



ПРИГАДАЙТЕ:

- як створюють таблиці в середовищі табличного процесора;
- які типи даних опрацьовують за допомогою комп'ютера.

ВИ ДІЗНАЄТЕСЯ:

- як редагують дані в таблицях у середовищі MS Access;
- які типи даних можна використувати для опису полів таблиці.



ВИВЧАЄМО

16.1. Як редагують дані в таблицях у середовищі MS Access?

У СКБД *MS Access* дані зберігаються у таблицях. Уже введені дані можна редагувати. Процес редагування доступний одразу після відкриття таблиці. Клацнувши на потрібному полі обраного запису, редагують вміст відповідної клітинки таблиці (мал. 16.1).

Номер	Прізвище	Ім'я	Стать	Зріст	Дата народи	Вага	Захоплення
1	Іванов	Іван	ч	120	01.02.2003		45 комп'ютери

Мал. 16.1

При виділенні числових чи текстових даних в обраному полі запису автоматично встановлюється курсор після останнього введенного символу. Видалити символи можна за допомогою клавіш редагування на клавіатурі чи просто ввести нові дані після подвійного клацання на клітинці таблиці. Якщо потрібно змінити дані дати, то можна застосувати просте редагування або скористатись кнопкою відкриття календаря.

Дата народи • В
01.02.2003

Важливо розуміти, що деякі поля можуть бути недоступними для редагування: поля, які задаються автоматично системою або заборонені для редагування розробниками бази даних. Наприклад, у таблиці *Учні* на малюнку 16.1 поле *Номер* змінити не можна.



ДІЄМО

Вправа 1. Редагування даних у базі даних.

Завдання. На основі бази даних *Школа* створіть базу власного класу, де у таблицях *Учні*, *Оцінки*, *Предмети* введіть дані про учнів класу, їхні

оцінки з вказаних предметів та прізвища вчителів, які навчають предметів у вашій школі.

1. Відкрийте базу даних *Школа* за посиланням. Скопіюйте її у папку *Бази даних* у своїй структурі папок.
2. Відкрийте таблицю *Учні* та відредагуйте запис про першого учня так, щоб отримати дані про учня вашого класу, який у журналі йде першим по списку. Для цього клацніть двічі на полі *Прізвище* запису, поле *Номер* якого має значення 1. Завершуйте редагування кожної клітинки натискуванням клавіші *Tab*.
3. Поміняйте оцінки для учня/учениці під номером 1 (мал. 16.2).

Номер	Прізвище	Ім'я	Стать	Зріст
1	Гевко	Аліна	ж	

код оцінки	код предмету	оцінка	Клацніть, щ
1	фізика	10	
4	математика	11	
5	географія	10	
6	інформатика	8	
7	хімія	9	
9	біологія	8	
*	(Новий)	0	

Мал. 16.2

Для цього натисніть та у таблиці, що відкриється, змініть оцінки із вказаних предметів.

4. Відкрийте таблицю *Оцінки*. Переконайтесь, що оцінки для першого учня чи учениці відображаються відповідно до ваших змін. Таблиця *Учні* та *Оцінки* пов'язані між собою полями *Номер* та *код оцінки*. Такий зв'язок не тільки забезпечує швидкий перехід від однієї таблиці до іншої, а й цілісність даних.
5. Продовжуйте змінювати дані у таблицях *Учні* та *Оцінки*.
6. Відкрийте таблицю *Предмети*. Змініть дані про вчителів відповідно до того, хто у вашій школі навчає вказаним предметам.
7. Завершіть роботу з базою даних. Чи вимагає у вас СКБД зберігати дані у базі даних? Ще раз відкрийте файл *Школа* та перевірте, чи збереглися дані. Зробіть висновок.

16.2. Якими можуть бути типи даних у таблицях бази даних?

За назвами полів у таблиці даних або їх вмістом можна зрозуміти дані яких типів вони містять. Взагалі, *СКБД MS Access* підтримує різні типи даних (табл. 16.1, с. 180).

Таблиця 16.1

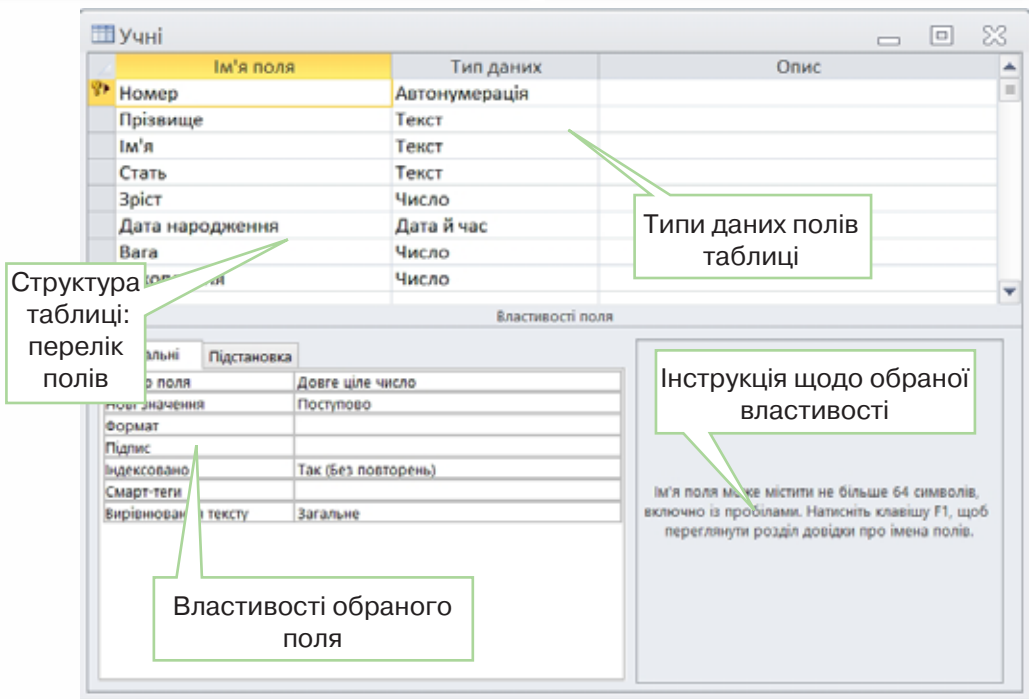
Тип	Значення
Текст	Тексти або набір цифр, що не потребують проведення розрахунків, наприклад, номери телефонів. Може містити до 255 символів
Примітка	Довгий текст або комбінація тексту й чисел, наприклад, примітки або опис. Може містити до 64 000 символів
Число	Числові дані, що використовуються для проведення розрахунків
Дата/час	Дати й час, що належать до років з 100 до 9999 включно
Грошова одиниця	Грошові значення й числові дані, які використовують у математичних розрахунках, що проводяться з точністю до 15 знаків у цілій і до 4 знаків у дробовій частинах. Використовується для запобігання округлень під час обчислень
Автонумерація	Унікальний послідовний (збільшений на 1) номер або випадкове число, призначене MS Access щоразу, коли до таблиці додається новий запис, не може бути оновлено
Логічний	Логічні вирази, а також поля, які можуть містити одне з двох можливих значень (<i>True/False, Так/Ні</i>)
Об'єкт OLE	Об'єкти (наприклад, документи MS Word, електронні таблиці MS Excel, малюнки, звуки та інші дані), які створені в інших програмах, що передбачають використання OLE-технології
Гіперпосилання	Поле, у якому зберігається гіперпосилання
Вкладення	Може містити зображення, файли електронних таблиць, документи, діаграми та інші типи підтримуваних файлів до записів у базі даних, подібно до того, як прикріплюють файли до повідомлень електронної пошти. Поля вкладення забезпечують більшу гнучкість, ніж поля OLE Object, і вони ефективніше використовують місце зберігання, оскільки не створюють растрового зображення
Обчислювальний	Містить формулу для обчислення значень на основі інших полів бази даних
Майстер підстановок	Створює поле, до якого дозволяється вибрати значення з іншої таблиці або зі списку значень, використовуючи поле зі списком

Для того, щоб пересвідчитись які типи даних можна вводити у поле таблиці бази даних, обирають у групі *Подання* режим конструктора:



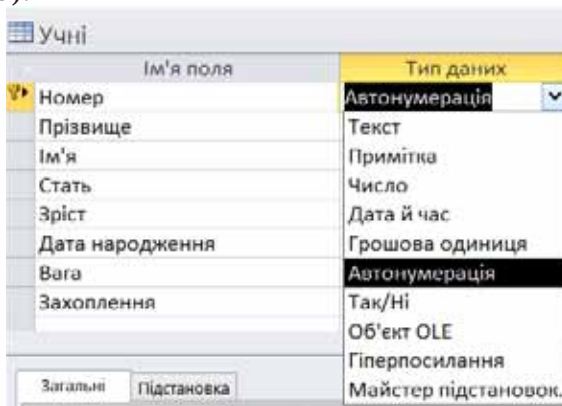
Вигляд: *Створення*.

Тоді отримують режим перегляду таблиці, у якому відображається її структура та властивості полів (мал. 16.2).



Мал. 16.2

Для того, щоб змінити тип даних, обирають потрібне зі списку (мал. 16.3).



Мал. 16.3

Зверніть увагу, що ключове поле, яке однозначно визначає кожний запис у таблиці бази даних в режимі конструктора позначається значком . У властивостях таких полів автоматично задається Індексовано Так (Без повторень).

Хоча в режимі конструктора є можливість задати позначку ключового поля іншому полю, користувачам не варто змінювати ключові поля у готовій базі даних, оскільки це може призвести до порушення структури даних бази даних та роботи з такою базою загалом.



ДОСЛІДЖУЕМО

Вправа 2. Типи даних.

Завдання. У таблиці *Учні* бази даних *Школа* дослідіть типи полів та значення перших трьох властивостей кожного поля. Визначте які значення можуть набувати такі властивості як *Розмір поля* та *Формат* (для поля *Дата народження*).

1. Відкрийте базу даних *Школа* із своєї структури папок.
2. Відкрийте таблицю *Учні* та перейдіть у режим конструктора *Подання*.
3. По черзі клацайте на назві кожного із полів та переглядайте зазначені в умові властивості поля.
4. Зробіть висновок.
5. Закрийте вікно бази даних, не зберігаючи зміни.



ПРАЦЮЕМО В ПАРАХ

1. Обговоріть спільне й відмінне в редагуванні даних електронної таблиці в середовищі табличного процесора й у таблиці бази даних. По черзі називайте свій доказ і демонструйте його.
2. Дані яких типів можна вводити до таблиці бази даних? Чим вони відрізняються від типів даних, які можна вводити в клітинки електронної таблиці у табличному процесорі. Обговоріть можливі причини.
3. Обговоріть, чи можна в таблиці *Художники* бази даних *Живопис*, що збережена в папці *Бази даних* на спільному ресурсі (<https://cutt.ly/SE2Maxy>), обрати інші типи даних. Якщо так, то які й чому?



ПРАЦЮЕМО САМОСТІЙНО

4. Відкрийте базу даних *Живопис*, що збережена в папці *Бази даних* на спільному ресурсі (<https://cutt.ly/SE2Maxy>), та виконайте вказані дії.
 - 1) Відкрийте таблицю *Художники* в режимі *Таблиці*. Знайдіть в інтернеті біографії вказаних у таблиці художників і додайте гіперпосилання у відповідні поля.
 - 2) Знайдіть в інтернеті відомості про життя і творчість Сальвадора Далі й доповніть базу даних відповідно до розробленої структури. Збережіть внесені зміни.
 - 3) Знайдіть відомості про українських художників чи художниць та додайте їх у базу даних за структурою таблиць.

**ПРИГАДАЙТЕ:**

- умовне форматування даних у середовищі табличного процесора;
- фільтрування та упорядкування даних у середовищі табличного процесора.

ВИ ДІЗНАЄТЕСЯ:

- які засоби системи керування базами даних дають змогу впорядковувати та знаходити в таблицях дані;
- які умови пошуку можна задати за допомогою *фільтра за виділеним фрагментом*;
- як формувати умови при використанні *фільтра для*;
- як задати складені умови за допомогою *фільтра Змінити фільтр*.

**ВИВЧАЄМО****17.1. Які засоби системи керування базами даних дають змогу впорядковувати та знаходити в таблицях дані?**

Таблиці баз даних заповнюються даними в процесі їх накопичення, й особливістю цих даних є невпорядкованість, тобто незалежно від того, які дані фіксуються, вони відсортовані лише за часом їх внесення в таблицю. На збереження даних це ніяк не впливає, але відшукати потрібні дані в такій таблиці дуже складно. Тому в СКБД для здійснення пошуку потрібних даних передбачено різні засоби. Одним з них є **запити**, при використанні яких у базі даних створюються нові об'єкти. Запити дають змогу зберігати результати пошуку. Крім того, за допомогою запитів можна знаходити дані з різних таблиць бази даних.

Якщо потрібно відшукати дані лише в одній таблиці бази даних, можна скористатись **фільтрами**. На відміну від запитів, при цьому в базі даних не створюються нові об'єкти й результати фільтрування не зберігаються.

У *MS Access* використовується чотири типи фільтрів. Виклик фільтрів здійснюється або за допомогою контекстного меню, або за допомогою використання вкладки *Головна* групи інструментів *Впорядкування та фільтр*. Але не всі фільтри можна викликати всіма названими способами, відповідні вказівки виклику деяких з них є лише в контекстному меню. При застосуванні будь-якого типу фільтра на екрані відображаються всі поля тих записів, що відповідають умові фільтрування. Обмежити кількість полів у фільтрах не можна. При цьому на кожному

полі, до якого застосовувався фільтр, відображається значок фільтра



(мал. 17.1).

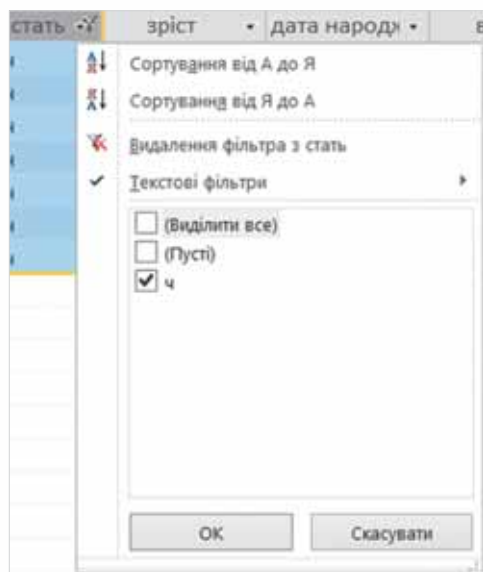
номер	прізвище	ім'я	стать	зріст	дата народж.	вага	захоплення	Класифікація
2	Петров	Василь	ч	178	23.11.2002		література	
9	Ухач	Петро	ч	179	25.06.2003		спорт	
10	Дранківський	Ярослав	ч	169	03.02.2003		75 музика	
11	Кошмар	Алі	ч	175	02.10.2002		82 комп'ютери	
12	Марценків	Андрій	ч	178	03.01.2003		72 комп'ютери	
20	Лисейко	Дмитро	ч	165	10.03.2003		60 музика	
(Новий)			ч	0			0	

Мал. 17.1

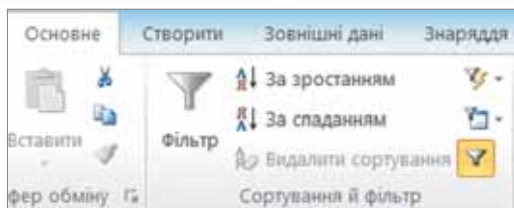
Скасувати результати фільтрування та повернутись до вихідного стану таблиці можна за допомогою вказівки *Видалення фільтра з...*, позначивши *Виділити все* (мал. 17.2), або в групі *Сортування й фільтр* вкладки *Основне* — обрати інструмент *Видалити фільтр* (мал. 17.3). Необхідний запис у таблиці можна знайти за значенням будь-якого її поля або за фрагментом його значення. Пошук записів здійснюється аналогічно до пошуку об'єктів засобами операційної системи чи, наприклад, табличного процесора *MS Excel*. Для швидкого пошуку першого входження потрібного зразка слід вибрати *Основне* й групу *Пошук*



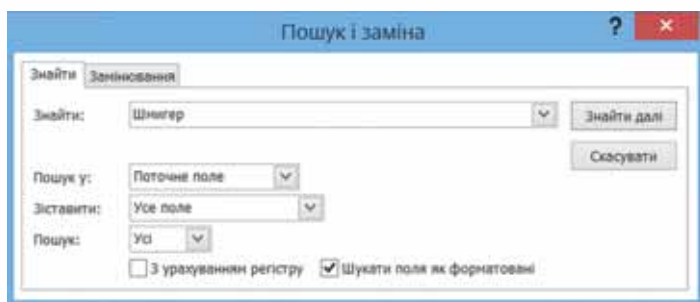
та задати шукане значення поля, наприклад, учня або учениці за вказаним прізвищем (мал. 17.4).



Мал. 17.2



Мал. 17.3

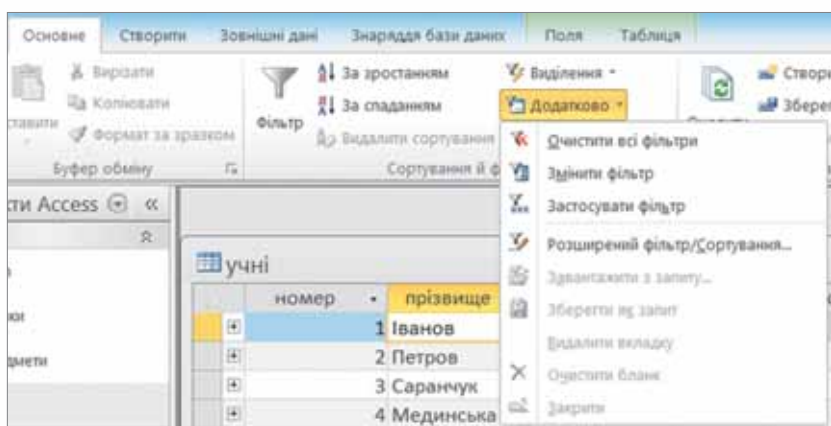


Мал. 17.4

ЦІКАВО

У деяких випадках виникає необхідність подання числових даних як текстових, і навпаки. Для цього перед сортуванням необхідно використати функцію перетворення типу даних.

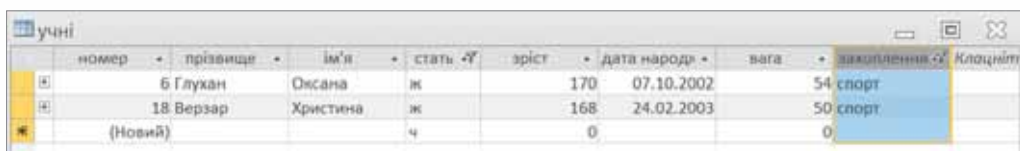
Знайдені записи можна змінювати, копіювати та видаляти. Здійснювати впорядкування даних у межах одного виділеного поля можна за допомогою вказівок *Сортування за зростанням* і *Сортування за спаданням* на вкладці *Основне* у групі інструментів *Сортування та фільтр*. Упорядковувати за декількома полями одночасно можна за допомогою розширеного фільтра (мал. 17.5) або запитів.



Мал. 17.5

17.2. Які умови пошуку можна задати за допомогою фільтра за виділеним фрагментом?

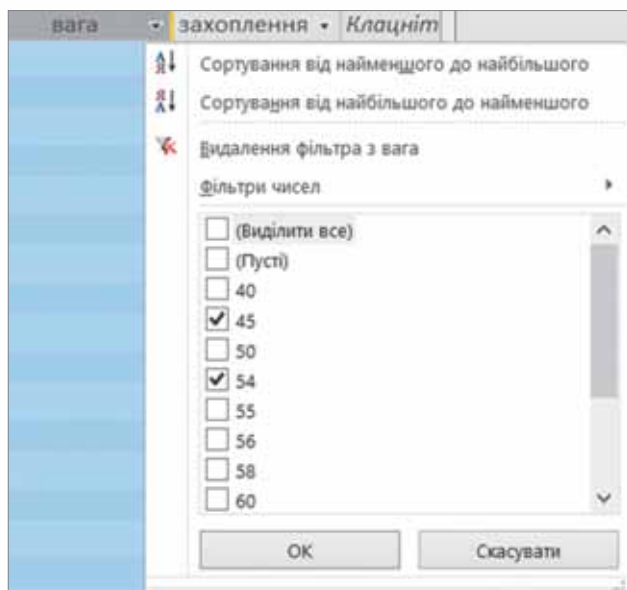
Фільтр за виділеним — це найпростіший тип фільтрування, який дає змогу знаходити в кожному полі лише одне конкретне значення, яке збігається з виділеним у цьому полі. Особливістю його застосування є те, що спочатку треба встановити текстовий курсор на потрібне значення в даному полі. Це можна зробити, вибравши інструмент *Виділення* у групі *Сортування та фільтр* або в потрібному полі — список унікальних значень й обравши лише потрібне значення. *Фільтр за виділеним* можна використовувати послідовно для декількох полів. При цьому умови між собою пов'язані логічною операцією *I*, тобто для вибору потрібного запису всі умови мають виконуватись одночасно. Наприклад, для того щоб відібрати дівчат, які займаються спортом, можна послідовно застосувати у відповідних полях фільтри за виділеним: у полі «стать» — фільтр *Дорівнює «ж»*, у полі «захоплення» — *Дорівнює «спорт»* (мал. 17.6).



номер	прізвище	ім'я	стать	вік	дата народи	вага	захоплення	Клацніт
6	Глухан	Оксана	ж	170	07.10.2002	54	спорт	
18	Верзар	Христина	ж	168	24.02.2003	50	спорт	
(Новий)			ч	0		0		

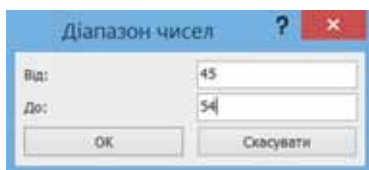
Мал. 17.6

У певному полі логічну операцію *I* можна реалізувати, вказавши потрібні значення в контекстному меню (мал. 17.7).



Мал. 17.7

Щоб обрати декілька числових значень із вказаного діапазону за допомогою *фільтра за виділенням*, можна використати вказівку *Фільтри чисел/Між*, після чого потрібно вказати діапазон чисел (мал. 17.8). Одним з різновидів фільтра за виділенням є фільтр *Виключити виділене* — у цьому разі в заданому полі відбираються всі записи, у яких значення не збігається з виділенням у даному полі. Режим *Виключити виділене* можна застосувати тільки з контекстного меню. Цей тип фільтра дає змогу задавати лише умови збігу та розбіжності, інші умови задавати не можна, для цього використовують фільтри інших типів.



■ Мал. 17.8 ■



ДІЄМО

Вправа 1. Пошук даних у таблиці за допомогою фільтра за виділенням.

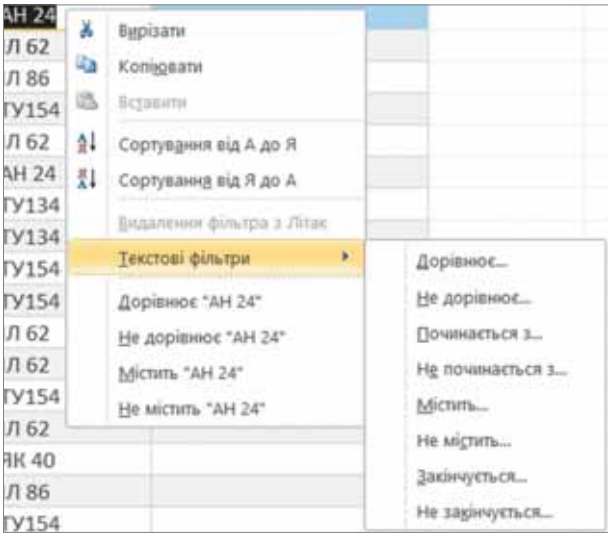
Завдання. У таблиці *Рейси* з бази даних *Розклад руху літаків* знайдіть записи, що містять час відправлення та прибуття літаків до Львова. Серед знайдених записів відобразить лише ті, у яких рейси здійснюються будь-якими літаками, окрім ІЛ-86.

1. Завантажте і відкрийте базу даних *Розклад руху літаків*, що збережена в папці *Бази даних* на спільному ресурсі (<https://cutt.ly/9E9tFwi>).
2. Відкрийте таблицю *Рейси* в режимі таблиці. Установіть текстовий курсор у клітинку таблиці, що в полі *Напрямок* має значення *Львів*. Натисніть на панелі інструментів кнопку *Виділення*
3. У полі *Літак* знайдіть значення *ІЛ-86* і клацніть на ньому правою клавішею миші. Виберіть у контекстному меню вказівку *Виключити виділене*.
4. Визначте, скільки рейсів відповідає заданим умовам, і проаналізуйте час їх відправлення та прибуття.
5. Закрийте таблицю *Рейси*.

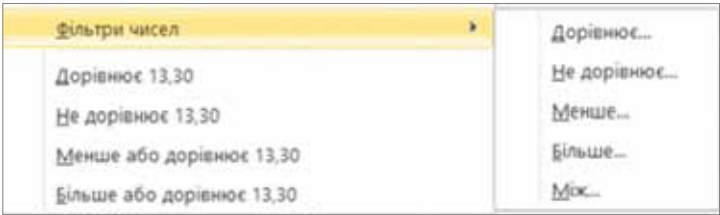
17.3. Як формувати умови при використанні фільтра для?

Фільтр для викликається лише з контекстного меню відповідного поля та вказівками *Текстові фільтри* або *Фільтри чисел* (мал. 17.9, мал. 17.10, с. 188). На відміну від фільтра за виділенням, можна задавати

умову, не встановлюючи курсор на потрібне значення, отже, немає необхідності потрібне значення спочатку знаходити в таблиці. Крім того, цей фільтр може використовувати групові операції з використанням метасимволів * та ?, які дають змогу задати умови типу «починається з...», «містить...» тощо. Для числових полів можна задати умови з використанням операторів відношення: >, >=, <= . Фільтр також можна використовувати для різних полів, у такому разі умови пов'язуються логічним І, логічне АБО не використовують.



■ Мал. 17.9 ■



■ Мал. 17.10 ■



ДІЄМО

Вправа 2. Створення умов пошуку з використанням фільтра для.

Завдання. У таблиці Квитки бази даних Розклад руху літаків знайдіть, на які рейси квитки коштують менше 800 грн і є в наявності. Упорядкуйте дані за збільшенням вартості квитків.

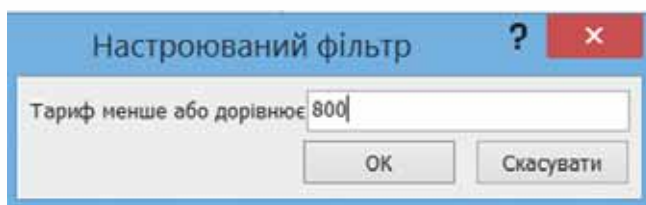
1. Відкрийте базу даних Розклад руху літаків, що збережена в папці Бази даних.

- Відкрийте таблицю *Квитки* в режимі таблиці. У контекстному меню поля *Тариф* оберіть *Фільтри чисел/Менше або дорівнює* та введіть умову 800 (мал. 17.11). Натисніть *ОК*.
- У контекстному меню будь-якої клітинки поля *Квитки* оберіть значення *Дорівнює «є»*.
- Виділіть поле *Тариф* і натисніть кнопку *Сортування за зростанням*



на панелі інструментів вікна програми.

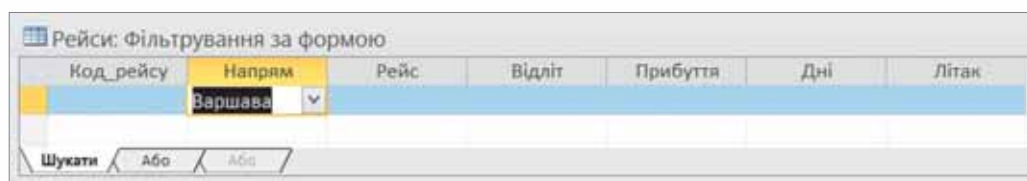
- Визначте кількість записів у таблиці, що відповідає вказаним умовам. Закрийте таблицю *Квитки*.



Мал. 17.11

17.4. Як задати складені умови за допомогою фільтра *Змінити фільтр?*

Фільтр, що викликається вказівкою *Додатково/Змінити фільтр*, дає можливість скористатися вкладками *Шукати* та *Або*. На відміну від фільтрів перших двох типів, дає змогу задавати умови, які пов'язані логічним *АБО*, тобто здійснювати операцію об'єднання значень вибраних полів (мал. 17.12).



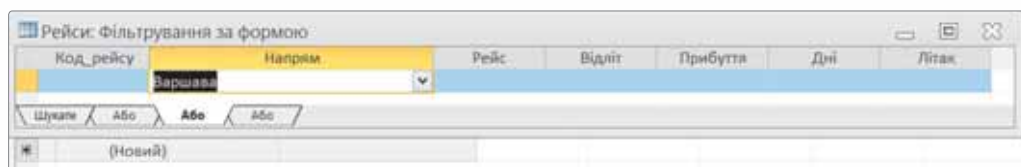
Мал. 17.12

При формуванні умови можна використовувати як дані зі списку можливих значень, так й інші операції, наприклад: M^* , >100 . Таким чином, цей тип фільтра є розширенням перших двох, але не дає змоги виконувати обчислення. За допомогою вибору списку () будь-якого поля таблиці можна задавати умови, які пов'язані логічним *АБО*. Для цього потрібно залишити обраними лише потрібні унікальні значення відповідного поля.

Вправа 3. Використання фільтра для створення складених умов, пов'язаних логічною операцією АБО.

Завдання. У таблиці *Рейси* бази даних *Розклад руху літаків* знайдіть записи про всі рейси, що прямують до міст Одеса, Брюссель або Варшава.

1. Відкрийте базу даних *Розклад руху літаків*, що збережена в папці *Бази даних*.
2. Відкрийте таблицю *Рейси* в режимі таблиці. Оберіть список у полі *Напря́м*, виберіть зі списку одне з можливих значень — *Одеса*.
3. У списку *Додатково* оберіть вказівку *Змінити фільтр*. У вкладках *Шукати й Або* послідовно введіть додаткові умови пошуку: *Напря́м — Брюссель*, *Напря́м — Варшава* (мал. 17.13).



Мал. 17.13

4. Визначте, скільки рейсів відповідають заданій умові (мал. 17.14).

Код_рейсу	Напря́м	Рейс	Відліт	Прибуття	Дні	Літак
3	Брюссель	586	22,55	23,00	1,3	ЯК 40
5	Одеса	M22	10,41	12,05	ЩДН	АН 24
18	Варшава	K154	23,39	2,05	2,4	ІЛ 62
26	Одеса	3044	21,40	22,55	ЩДН	АН 10
27	Брюссель	8206	17,10	18,55	2,4,6	ІЛ 86
(Новий)						

Мал. 17.14

5. Перевірте, яким чином буде записано умову пошуку у вікні *Рейси: Фільтрування за формою*, якщо після застосування фільтра ще раз обрати *Змінити фільтр*. Закрийте таблицю.



ПРАЦЮЄМО В ПАРАХ

1. Наведіть приклади ситуацій, коли доцільно використовувати пошук даних, фільтрування та сортування.
2. Знайдіть спільне й відмінне в застосуванні фільтрів під час опрацювання даних засобами табличного процесора та СКБД. Обговоріть у парах, чи можна здійснити фільтрування даних засобами СКБД *MS Access*, знаючи лише особливості використання фільтрів під час опрацювання даних засобами табличного процесора.

-  3. Назвіть можливі причини використання розширеного фільтра під час опрацювання баз даних. Знайдіть спільне й відмінності при застосуванні розширеного фільтра під час опрацювання даних електронних таблиць і баз даних. Обговоріть у парах.
-  4. Запропонуйте одне одному завдання на пошук даних з використанням різних фільтрів у базах даних, які використовували на попередніх заняттях. Перевірте отримані результати. Обговоріть причину вибору кожного з фільтрів для виконання завдань.



ПРАЦЮЄМО САМОСТІЙНО

5. Відкрийте базу даних *Живопис*, що міститься в папці *Бази даних*. Відкрийте таблицю *Художники* в режимі таблиці та виконайте вказані дії.

- 1) Упорядкуйте записи за збільшенням значень у полі *Рік народження*.
- 2) Знайдіть художників і художниць, які народились у Франції. Перегляньте результати фільтрування. Скасуйте фільтрування.
- 3) Відкрийте таблицю *Полотна* в режимі таблиці. Знайдіть картини, автором яких є Клод Моне.
- 4) Виведіть на екран дані про полотна, створені в XIX ст.

-  6. Створіть текстовий документ, де запишіть інструкцію для виконання завдання. Для ілюстрації використовуйте фрагменти копій екранів. Відкрийте базу даних *Магазини*, що збережена в папці *Бази даних*, та виконайте вказані дії над даними таблиці *Фільми*.

- 1) Упорядкуйте дані за зростанням ціни за одиницю.
- 2) Визначте, які фільми належать до жанру комедії.
- 3) Визначте фільми жанру Фантастика, ціна яких більша за 680 грн.

-  7. Попрацюйте в мініпроекті «Довідник з пошуку даних». Створіть довідник з основних засобів СКБД для пошуку даних у базі даних. Використайте для прикладу одну з баз даних, які збережено в папці *Бази даних*. Для створення довідника доберіть середовище самостійно. Презентуйте довідник у класі.

ЩО Є НАЙВАЖЛИВІШИМ ПІД ЧАС РОБОТИ З БАЗОЮ ДАНИХ?

ПРИГАДАЙТЕ:

- як здійснюють аналіз даних за-
собами табличного процесора;
- автообчислення в середовищі
табличного процесора;
- використання функцій таблично-
го процесора.

ВИ ДІЗНАЄТЕСЯ:

- як створити запит на вибірку
у базі даних;
- як можна змінити структуру
у базі даних;
- як надрукувати результати запи-
ту.



ВИВЧАЄМО

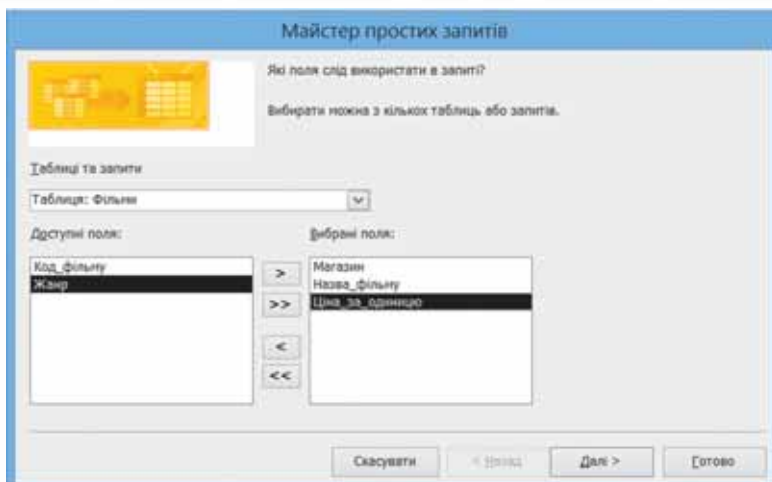
18.1. Як створити запит на вибірку в базі даних?

Крім фільтрів та інструмента *Пошук*, у СКБД завдання пошуку да-
них також допомагають розв'язати запити. Запит дає змогу вибрати
необхідні дані з однієї або кількох зв'язаних таблиць, провести обчис-
лення та одержати результат у вигляді таблиці, поновити дані в таблиці,
створити нову таблицю, додати або видалити записи.

Найпростішим запитом, який можна створювати у базі даних, є за-
пит на вибірку. Для побудови такого запиту необхідно у вікні бази да-
них на вкладці *Створити* вибрати групу *Запити* та клацнути на кнопці
Майстер запитів. У вікні, що відкрилось, потрібно вибрати один зі спо-
собів для створення нового запиту:

- *простий запит* — побудова запиту на вибірку за допомогою *Май-
стра*;
- *перехресний запит* — створюється запит, у якому потрібні дані по-
даються у вигляді таблиці;
- *записи, що повторюються*, — відбираються лише ті записи, у яких
значення вказаних полів повторюються;
- *незв'язані записи* — відбираються лише ті записи, які не мають
зв'язків із записами інших таблиць.

За допомогою *Майстра* можна створити лише прості запити на ви-
бірку, які не містять умов відбору та обчислювальних полів. При ство-
ренні простого запиту відображається діалогове вікно, у якому слід
з однієї чи кількох зв'язаних таблиць, назви яких можна вибрати зі
списку, що розкривається, вибрати поля, необхідні для формування за-
питу, та перенести їх до області *Вибрані поля* в потрібній послідовності
(мал. 18.1, с. 192).



■ Мал. 18.1 ■

На останньому кроці роботи *Майстра* залишається лише задати назву запиту. Усі інші параметри запитів можна налаштувати в режимі конструктора.



ДІЄМО

Вправа 1. Створення запиту за допомогою *Майстра*.

Завдання. У базі даних *Магазини* за допомогою *Майстра* створіть запит *Продажі*, що містить такі поля: *Магазин* з таблиці *Магазини*, *Назва_фільму* й *Ціна_за_одиноцю* з таблиці *Фільми* та *Кількість_продажів* з таблиці *Підсумок*.

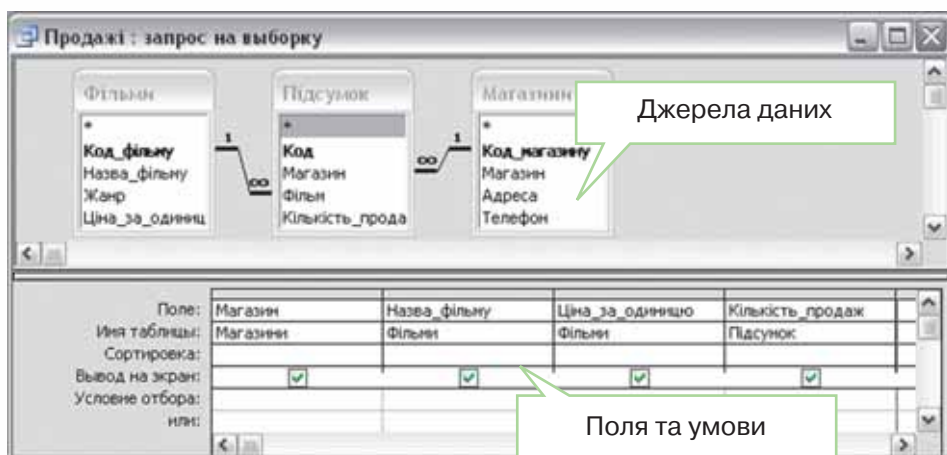
1. Відкрийте базу даних *Магазини*, що збережена в папці *Бази даних*.
2. Оберіть вкладку *Створення*, на ній групу *Запити* та інструмент *Майстер запитів*.
3. У вікні *Створення простих запитів* у списку *Таблиці й запити* виберіть таблицю *Магазини*.
4. В області *Доступні поля* виділіть поле *Магазин* і натисніть кнопку , щоб перемістити одне виділене поле до області *Вибрані поля*.
5. У списку *Таблиці й запити* оберіть таблицю *Фільми* та перемістіть поля *Назва_фільму* й *Ціна_за_одиноцю* до області *Вибрані поля*.
6. У списку *Таблиці й запити* оберіть таблицю *Підсумок* і перемістіть до області *Вибрані поля* поле *Кількість_продажів*. Натисніть кнопку *Далі*.
7. На наступному кроці майстра виберіть режим *Детальний* (виведення кожного поля для кожного запису). Натисніть кнопку *Далі*.
8. У рядку введення введіть назву нового запиту — *Продажі*. Натисніть кнопку *Готово*.

ЦІКАВО

Якщо передбачається використання кількох умов відбору, то їх слід поєднати за допомогою логічного *AND* (І) чи *OR* (АБО). Також логічне АБО можна задати, розмістивши умови в рядках, розташованих один під одним. За замовчуванням використовується умова — «дорівнює». Якщо потрібен не повний збіг, то використовують *Like* та службові символи * (будь-який набір довільних символів) і ? (один довільний символ). Якщо конкретне значення умови відбору потрібно ввести з клавіатури, у запиті використовується параметрична умова: у квадратних дужках записується звернення до користувача про необхідність введення з клавіатури конкретного значення, наприклад, [Введіть назву району].

18.2. Як можна змінити структуру запиту в режимі конструктора?

Відкрити запит у режимі конструктора можна тими само способами, що й таблиці бази даних. Бланк конструктора запитів поділено на дві частини: у верхній частині вказано таблиці, на основі яких створено запит, у нижній частині задаються всі параметри запиту, тобто його структура (мал. 18.2).



■ Мал. 18.2 ■

Під час редагування структури запиту в режимі конструктора можна видаляти деякі поля чи додавати інші з таблиць, що взяті за основу, змінювати послідовність розміщення полів, задавати порядок сортування за деякими полями, вказувати умови відбору записів, додавати нові поля, яких немає в таблицях бази даних, але які можуть бути створені в результаті обчислень, виконувати підсумкові обчислення з використанням групових операцій, що можна застосувати до полів, значення в яких повторюються.



ДІЕМО

Вправа 2. Зміна структури запиту в режимі конструктора.

Завдання. У режимі конструктора внесіть зміни в структуру запиту *Продажі* в базі даних *Магазини*, щоб відображались лише записи про фільми, продані в магазині *Стиль*, які упорядковані за зростанням за кількістю проданих копій.

1. Відкрийте вікно бази даних *Магазини*. В області навігації, що розташована в лівій частині робочої області, виберіть зі списку *Запити*. Клацніть правою кнопкою миші на запиті *Продажі* та виберіть з контекстного меню вказівку *Конструктор*.
2. На бланку запиту для поля *Магазин* у рядку *Умова відбору* введіть значення *Стиль*.
3. Для поля *Кількість_продажів* у рядку *Сортування* виберіть зі списку можливих значень за зростанням.

4. Натисніть кнопку , щоб відобразити результати виконання запиту на вибірку у вигляді таблиці. Переконайтеся, що знайдені записи відповідають умові.

5. Поверніться до режиму конструктора. На бланку запиту для поля *Магазин* у рядку *Умова відбору* замість слова *Стиль* введіть таке: [Введіть назву магазину].

6. Натисніть кнопку , у діалоговому вікні введіть назву *Стиль* і натисніть клавішу *Enter*. Порівняйте отриманий результат з попереднім.
7. Закрийте вікно запиту та збережіть внесені зміни.

18.3. Як надрукувати результати запиту?

Результати запиту можна не тільки переглядати у вікні СКБД, а й роздрукувати. Для цього у списку об'єктів бази даних виділяють створений запит і виконують одну з дій:

1) Обирають вказівку *Файл/Друк*. Тоді формується документ, у якому зазначаються не тільки дані запиту у формі таблиці, а й дата виконання операції.

2) У контекстному меню обирають вказівку *Експорт*. Обравши, наприклад, *Файл Word RTF*, отримаємо текстовий документ із таблицею результатів запиту, який можна редагувати та надрукувати засобами текстового процесора.

3) Створюють звіт, обравши вказівку *Створити/Звіт*. При цьому створюється звіт, до якого застосовано автоматичне форматування. Роздрукувати звіт можна за допомогою вказівки *Файл/Друк*.



ПРАЦЮЄМО В ПАРАХ

1. Наведіть приклади, коли в повсякденному житті ви користувалися запитом до бази даних. Обговоріть, чому важливо розробнику баз даних правильно організувати ведення бази даних за допомогою запитів.
2. Чи можна отримати такий само результат, як від застосування одного запиту для декількох таблиць, шляхом послідовного застосування декількох запитів до однієї таблиці (таблиці можуть бути різними)? Підтвердьте свої висновки на конкретній базі даних, виконуючи обрану послідовність дій.
3. Чи може редагування структури запитів призвести до втрати даних? Назвіть дії, які можна виконувати у процесі редагування структури запиту. Обговоріть у парах.
4. Знайдіть спільне та відмінне в застосуванні фільтрів і запитів під час опрацювання даних засобами *MS Access*. Результати подайте у вигляді діаграми Венна.



ПРАЦЮЄМО САМОСТІЙНО

5. Створіть презентацію з покроковим виконанням завдання з використанням бази даних *Підприємство*, що зберігається в папці *Бази даних* на спільному ресурсі (<https://cutt.ly/GE3TpvH>).

1) Перегляньте дані, що зберігаються в таблиці *Кадри* (мал. 18.3). Наведіть приклади завдань, які можна вирішувати за допомогою даних цієї бази.

Табельний №	№ п/п	Прізвище	Ім'я	По батькові	Посада	Дата народи	Оклад	Категорія	Наявність пільг
3	6	Лещук	Олег	Макарович	Директор	30.04.1985	7 180,00 грн.	1	✓
122	15	Шивал	Петро	Данилович	Художник	18.09.1978	5 080,00 грн.	1	
123	1	Антон	Павло	Якович	Робітник	01.05.1984	3 980,00 грн.	3	✓
136	14	Луць	Іван	Олекович	Менеджер	19.01.1985	6 030,00 грн.	2	
138	4	Шець	Роман	Іванович	Робітник	09.06.1981	3 980,00 грн.	3	
149	5	Карп	Лев	Йосипович	Робітник	15.12.1987	3 980,00 грн.	3	✓

Мал. 18.3

2) За допомогою *Майстра* створіть запит на вибірку, що містить поля *Прізвище*, *Посада*, *Оклад*, *Категорія*, *Наявність пільг*. Виконайте запит і переконайтесь у коректності його реалізації. Збережіть запит з назвою *Працівники*.

3) Створіть копію запиту *Працівники* з назвою *Оплата*.

4) У режимі конструктора змініть структуру запиту *Працівники* таким чином, щоб вибиралися записи лише про працівників, які працюють на посаді *Робітник* і не мають пільг. Значення поля *Наявність пільг* не відображайте на екрані, прізвища працівників

упорядкуйте за алфавітом. Відкрийте запит у режимі таблиці та проаналізуйте отриманий результат.

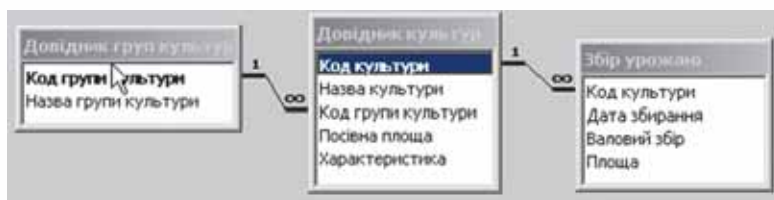
5) У режимі конструктора уточніть умови запиту *Оплата*: слід вибирати лише записи, значення поля *Оклад* у яких перевищує 5000 грн. Відкрийте запит у режимі таблиці та проаналізуйте отриманий результат.

6) У режимі конструктора уточніть умови запиту *Працівники*: слід вибирати записи про працівників, які народились у липні. Збережіть внесені зміни.

7) У режимі конструктора створіть запит *Категорія працівників*, за яким відбираються записи за такою параметричною умовою: [Введіть номер розряду]. Виконайте запит і проаналізуйте отриманий результат. Збережіть внесені зміни та закрийте файл.

8) Збережіть презентацію в персональному освітньому середовищі та надайте до неї доступ учителю або вчительці з правом коментування.

-  6. Створіть текстовий документ із покроковим виконанням завдання з використанням бази даних *Урожай* (мал. 18.4), що зберігається в папці *Бази даних* на спільному ресурсі (<https://cutt.ly/EE3TgEW>). Кроки виконання ілюструйте відповідними екранними копіями.



■ Мал. 18.4

1) Перегляньте запит *Дані про збір урожаю* в режимі конструктора та запишіть у текстовий документ можливе завдання, що передбачає створення такого запиту.

2) Відкрийте запит у режимі таблиці та проаналізуйте отриманий результат.

3) У режимі конструктора створіть запит на вибірку *Урожайність*, що містить поля *Назва культури* з таблиці *Довідник культур* і *Валовий збір* та *Площа* з таблиці *Збір урожаю*.

4) У запиті *Урожайність* створіть нове поле *Урожайність_нова*, значення якого будуть визначені шляхом обчислень як частка значень полів *Валовий збір* і *Площа*.

5) Виконайте запит і проаналізуйте отриманий результат. Збережіть внесені зміни.

6) Надішліть текстовий документ з іменем *Запит_Урожай_Прізвище* вкладенням до електронного листа на навчальну скриньку вчителя або вчительки.

7. Сформулюйте три запити до бази даних Підприємство та реалізуйте їх засобами *MS Access*.
8. Попрацюйте в мініпроєкті «Адмініструємо бази даних». Оберіть базу даних, яку використовували в класі на власний вибір. Підготуйте список можливих запитів користувачів до обраної бази даних. Реалізуйте їх у середовищі СКБД та передбачте можливість отримання результуючих документів для створених запитів. Продемонструйте результат у класі.

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Я розумію, що таке бази даних.
- ☐ Я можу пояснити відмінності подання даних у багатотабличних БД та в електронних таблицях.
- ☐ Я можу пояснити такі поняття: «таблиця», «поле», «запис», «ключ таблиці».
- ☐ Я можу пояснити призначення систем керування базами даних.
- ☐ Я вмію вводити дані в таблиці, усвідомлюючи обмеження, що накладаються структурою бази даних.
- ☐ Я вмію сортувати дані в таблицях бази за одним чи кількома полями.
- ☐ Я вмію фільтрувати дані в таблицях.
- ☐ Я вмію знаходити в базі дані за певними критеріями відбору, створюючи прості вибіркові запити в автоматизованому режимі.
- ☐ Я вмію редагувати дані в таблицях.

3D-ГРАФІКА

19

ЩО РОЗУМІЮТЬ
ПІД 3D-ГРАФІКОЮ?

ПРИГАДАЙТЕ:

- що таке комп'ютерна графіка;
- інструменти та пристрої для роботи з комп'ютерною графікою;
- що передбачає комп'ютерне моделювання.

ВИ ДІЗНАЄТЕСЯ:

- що таке тривимірна графіка;
- де застосовується 3D-графіка;
- як створюється тривимірне зображення;
- які прийоми дозволяють зробити 3D-зображення реалістичним;
- які програми використовують для роботи із 3D-графікою.



ВИВЧАЄМО

19.1. Що таке тривимірна графіка?



3D-графіка, або тривимірна графіка, — це один з розділів комп'ютерної графіки, комплекс прийомів й інструментів, які дають змогу створити об'ємні (3D) об'єкти за допомогою форми й кольору.

У результаті використання 3D-графіки об'єкт може бути поданий у вигляді «плоского» зображення — проєкції об'єкта, як 3D-анімація, як модель, яка може бути відтворена за допомогою інструментів доповненої реальності або надрукована на 3D-принтері.



Тигр в
натуральну
величину.



Перегляньте в 3D

Чим відрізняються 3D-зображення від двовимірного? Якщо при перегляді зображення можна описати ширину й висоту, але не спостерігається глибина, значить, це двовимірна графіка. Значки на робочому столі й покажчики на вулицях — все це належить до 2D-графіки (за деяким

винятком, коли художник використовує тінь або інші прийоми, щоб зробити картинку об'ємною). 3D-зображення обов'язково має глибину, тобто є об'ємним.

Наприклад, зображення квадрата — це приклад, двомірної графіки, а куба — тривимірної.

Від двомірних зображень 3D-зображення відрізняється тим, що передбачається побудова геометричної проєкції тривимірної моделі сцени (віртуального простору) на площину, робиться це за допомогою спеціалізованих програм. Отримана модель може відповідати об'єктам реального світу (наприклад, будівля, людина, автомобіль, астероїд) або бути цілком абстрактною.

Процес побудови такої тривимірної моделі отримав назву **3D-моделювання** і спрямований, перш за все, на створення візуального об'ємного образу об'єкта, що моделюється.

Сьогодні на основі тривимірної графіки можна створити високоточну копію реального об'єкта, створити щось нове, втілити в життя найбільш нереальні дизайнерські задумки.

На моніторі представляється проєкція тривимірної фігури, а об'ємне зображення формується в мозку людини. Просторове моделювання об'єктів передбачає, перш за все, побудову каркаса форми об'єкта, який має обсяг, вибір «віртуального» матеріалу для візуалізації поверхонь об'єкта, створення текстур. Завершенням роботи з тривимірною графікою є накладення поверхневих спецефектів, що підвищують реалістичність, якість сприйняття об'єкта.

19.2. Де застосовується тривимірна графіка?

3D-графіка — один з найбільш ефективних інструментів у рекламі, що дає змогу розширити вплив на потенційного клієнта й підвищити якість реклами як у реальному, так і у віртуальному світі.

Сьогодні 3D-технології графіки та технології 3D-друку проникли в різні сфери людської діяльності й приносять великий прибуток. Це:

- будівництво (візуалізація об'ємних архітектурних зображень будівель, об'єктів, інтер'єру, екстер'єру);



- виробництво (об'єктне моделювання);



- телебачення (модельовані фото в глянцеvih журналах, відеоролики,



- промисловість;



- ігрова індустрія (3D-анімація і віртуальні світи, розробка комп'ютерних ігор, розробка 3D-персонажів, 3D-об'єкти для ігор);



- поліграфія (створення поліграфічної продукції);



- міське планування;



- реклама (електронні презентації і каталоги, рекламні щити й ін.);



- кінематограф і мультиплікація — створення тривимірних персонажів і реалістичних спецефектів тощо;



- медицина — створення імплантів і протезів.



19.3. Як створюється тривимірне зображення?

Для того щоб отримати 3D-зображення об'єкта, необхідно виконати такі кроки:

1. Моделювання — побудова математичної 3D-моделі загальної сцени та її об'єктів. Моделювання передбачає створення віртуального простору й об'єктів усередині нього, включає створення різних геометрій, матеріалів, джерел світла, віртуальних камер, додаткових спецефектів.

На комп'ютері створюється модель, що точно передає форму об'єкта, який потрібно представити. Це може бути будь-який предмет, тварина або людина. Загалом — усе, що нас оточує. Існує кілька видів тривимірного моделювання, кожний з яких має свої особливості та принципи.

Завданням тривимірного моделювання — описати ці об'єкти й розмістити їх у сцені за допомогою геометричних перетворень відповідно до вимог до майбутнього зображення.

Використовують такі прийоми:

- вичавлювання;
- модифікатори;
- полігональне моделювання;
- обертання.

Сцена (віртуальний простір моделювання) включає в себе кілька категорій об'єктів:

- геометрія (побудована за допомогою різних технік модель, наприклад, будівля);
- матеріали (інформація про візуальні властивості моделі, наприклад, колір стін і здатність вікон відбивати чи заломлювати світло);
- джерела світла (налаштування напрямків, потужності, спектра освітлення);
- віртуальні камери (вибір точки та кута побудови проєкції);
- сили та дії (налаштування динамічних спотворень об'єктів, застосовується в основному в анімації);
- додаткові ефекти (об'єкти, що імітують атмосферні явища: світло в тумані, хмари, полум'я тощо).

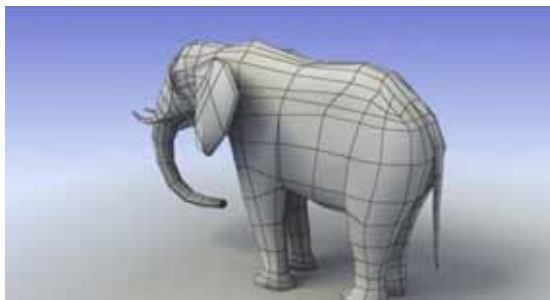
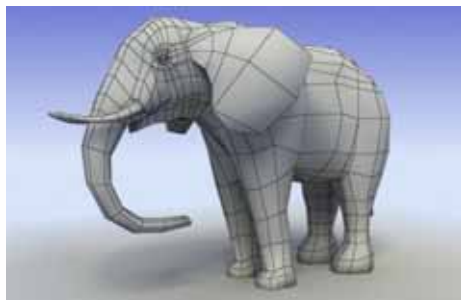
Існує кілька способів 3D-моделювання, які використовує 3D-редактор: *полігональне*, *сплайнове* та *NURBS-моделювання*. Вони можуть застосовуватися як окремо, так і комплексно.

Полігональне моделювання — це вид 3D-моделювання, яке передбачає побудову зображень за місцезнаходженням точки (за її координатами по осях X, Y, Z). Якщо три точки координат задати як вершини та з'єднати їх ребрами, то вийде трикутник, який у 3D-моделюванні називають полігоном.



Полігон — трикутник, що задається координатами трьох точок у тривимірному просторі.

Він є базовим геометричним примітивом у 3D-графіці. У більш широкому сенсі слова полігон — довільний плоский багатокутник, але в 3D-графіці це поняття звужують до трикутника, тобто до найбільш простої фігури, яка найлегше піддається розрахункам (за трьома точками задається площина).



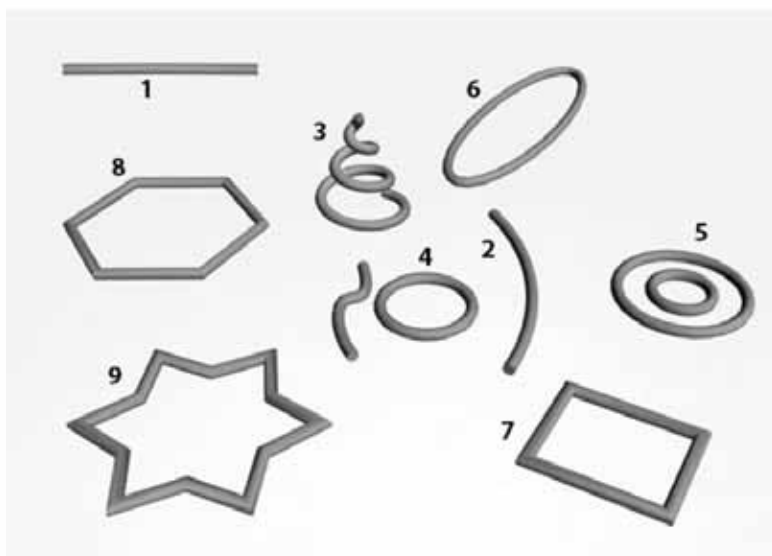
Кожний полігон може мати власну текстуру й колір, а об'єднавши кілька полігонів, можна отримати модель будь-якого об'єкта. Сполучені між собою полігони утворюють полігональну сітку, або полігональний об'єкт.

Незважаючи на те, що полігональне моделювання використовується досить часто, особливо у створенні тривимірних комп'ютерних ігор реального часу, останнім часом спостерігається перехід від моделювання полігонів до роботи зі сплайнами (моделювання сплайна).

Моделювання сплайна — це вид 3D-моделювання, за якого модель створюється за допомогою сплайнів (від англ. *spline* — гнучке лекало, у 3D — це тривимірна крива). Лінії сплайнів задаються тривимірним набором контрольних точок у просторі, які й визначають гладкість кривої. Усі сплайни зводяться до каркаса сплайна, на основі якого вже буде створюватися тривимірна геометрична поверхня.

Крім того, у моделюванні сплайна використовують примітиви сплайнів (параметричні об'єкти для моделювання об'єкта). Базовими сплайнів примітивами є:

- 1) лінія (*Line*);
- 2) дуга (*Arc*);
- 3) спіраль (*Helix*).
- 4) коло (*Circle*);
- 5) кільце (*Donut*);
- 6) еліпс (*Ellipse*);
- 7) прямокутник (*Rectangle*);
- 8) багатокутник (*NGon*);
- 9) багатокутник у вигляді зірки (*Star*).



Технологія NURBS ґрунтується на генерації кривих і поверхонь.

19.4. Які прийоми дають змогу зробити 3D-зображення реалістичним?

Для того щоб 3D-зображення мало реалістичний вигляд, використовують кілька прийомів.

1. Згладжування. Щоб краї моделі мали згладжений вигляд, необхідно, щоб полігони були малого розміру, а поверхня об'єкта складалася з маленьких площин.

2. Текстурування являє собою накладення на поверхню створеної тривимірної моделі реєстрового або векторного зображення, що дає змогу відобразити властивості й матеріал об'єкта.



ЦІКАВО

Текстура — плоске зображення, натягується на полігон або кілька полігонів. Процес заповнення полігону текстурою іноді називають обгортанням. Тексель — точка на поверхні текстури. З таких точок складається все зображення текстури. Піксель (елемент зображення) — одиночка точка, яка відображається на моніторі в конкретному місці.

3. Налаштування освітлення.

Освітлення — створення, установка напрямки та налаштування джерел освітлення в створеній сцені. Один з найскладніших етапів у створенні 3D-моделі. Адже саме від вибору тону світла, рівня яскравості, різкості та глибини тіней залежить реалістичність сприйняття зображення. Крім того, необхідно вибрати місце спостереження за об'єктом. Це може бути вид з висоти пташиного польоту або масштабування простору з досягненням ефекту присутності в ньому — завдяки вибору погляду на об'єкт із висоти людського зросту.

4. Створення анімації (рухомих об'єктів).

Моделі завжди розміщені на сцені та необхідні для виконання певної дії: переміщення, руйнування або передачі будь-якого іншого ефекту. Для розташування об'єктів на сцені та їх анімації може використовуватися та само програма, яка застосовувалася і для моделювання, але іноді розробники звертаються до іншої програми. Анімації теж бувають різними, наприклад, зараз особливо популярне захоплення руху (коли програма зчитує рух людини і передає їх на тривимірну фігуру).

Анімація передбачає створення рухомих об'єктів, а точніше імітації руху моделі. Сучасні 3D-редактори містять безліч інструментів для створення інструментів, існують і спеціалізовані програмні продукти для створення анімації з інструментами для побудова тривимірних моделей.

ЦІКАВО

Повноцінну появу 3D-елементів на екрані світ побачив у короткометражному фільмі «A Computer Animated Hand», що вийшов на екрани в 1972 р. Одним із перших фільмів, у якому глядач міг побачити анімацію людського обличчя, вважається «Futureworld», що вийшов у 1976 р.

19.5. Які програми використовують для роботи з 3D-графікою?

Для створення комп'ютерної графіки та 3D-моделей використовуються велика кількість різноманітних редакторів і цілих комплексів. Їх умовно можна згрупувати у чотири категорії:

1) універсальні 3D-редактори, які, як правило, підтримують усі методи створення 3D-зображень (*Cinema 4D, 3Ds Max, Maya, Blender*);

2) вузькоспеціалізовані редактори, які використовують для конкретних завдань (анімація рідин — *RealFlow*, створення текстур — *Mari*, створення *Lego* моделей *Lego Digital Designer*);

3) програми для створення цифрової скульптури й текстурного фарбування 3D-моделей (*Autodesk Mudbox*);

4) програми для створення 3D-ігор (*Unreal Engine 4, Unity 5*).

Найбільш поширеними програмними продуктами для 3D-моделювання є: *Autodesk 3D max, Pixologic Zbrush, Blender*.



ДІЄМО

Вправа 1. 3D-анімації.

Завдання. Розгляньте на сайті *Mozaik Education* за посиланням [https:// bit.ly/2JDBiol](https://bit.ly/2JDBiol) 3D-сцену, яка демонструє роботу чотиритактного двигуна внутрішнього згоряння (мал. 19.1). Зробіть висновок про те, чи можна назвати переглянуті анімації комп'ютерною моделлю.



Мал. 19.1

1. На сайті *Mozaik Education* відкрийте вкладку *3D*. У розділі *Фізика* оберіть сцену *Чотиритактний двигун Отто*.
2. Дочекайтеся завантаження 3D-сцени. За потреби встановіть пропонуєний сайтом переглядач або відкрийте файл за допомогою раніше встановленої програми *m3dViewer*.
3. Розгляньте різні анімації за вкладками (мал. 19.2)



Мал. 19.2



ПРАЦЮЄМО В ПАРАХ

1. Обговоріть, чому 3D-графіка дедалі більше використовується в наш час. Наведіть приклади й аргументи її переваг.
2. Наведіть приклади різних способів 3D-моделювання. Для ілюстрації прикладів знайдіть відповідні зображення в інтернеті.
-  3. Проаналізуйте матеріали статті «3D-моделювання: огляд кращих програм» із сайту <https://hi-news.pp.ua/kompyuteri/11369-3d-modelyuvannya-oglyad-kraschih-program.html>. Визначте, до якого типу програм для роботи із 3D можна віднести розглянуті приклади програм. Наведіть свої аргументи.
-  4. У мобільних пристроях з ОС *Android 7* або новішої версії можна переглядати результати пошуку в 3D. Для цього *Довідка Google* пропонує виконати таку інструкцію.

Як шукати та переглядати 3D-результати

1. Перейдіть на сторінку google.com або відкрийте додаток Google  на телефоні Android.
2. Введіть у рядку пошуку назву тварини, об'єкта або місця.
3. Якщо 3D-результат доступний, натисніть значок .
4. Щоб переглянути 3D-результат у режимі доповненої реальності, натисніть Побачити біля себе.
5. Виконайте вказівки на екрані.

У парі виконайте запропоновані кроки та знайдіть 3D бурого ведмедя, лабратора, морської черепахи. Обговоріть, які засоби використали розробники, щоб зробити зображення реалістичними.



ПРАЦЮЄМО САМОСТІЙНО

5. Ознайомтесь із публікацією «Як і навіщо винайшли 3D-друк» за посиланням <http://thefuture.news/3d-printing>. Використовуючи матеріал підручника та сайту, створіть карту знань про використання 3D-друку.
-  6. Ознайомтесь із технологією створення 3D-моделей з паперу за матеріалами сайту <http://www.supercoloring.com/ua/vyrobny-z-paperu/3d-papercraft>. Оберіть одну із моделей, розгляньте файл розгортки та сплануйте створення такої моделі.

Поясніть технологію паперкрафт в термінах 3D-моделювання.



Bear Head Trophy 3D
papercraft



Bear Wall Trophy 3D
papercraft



Flower Pot 3d papercraft



Fox Head Trophy 3D
papercraft



Grackle on the Wall Trophy
3D Papercraft



7. Дослідіть демонстраційний проект 3D-дизайну квартири за посиланням <https://planoplan.com/en/demo/>. Врахуйте, що макетом обертають за допомогою стрілок клавіатури.

Маркер зміни рівня фігури



Масштабування

Щоб змінити розташування об'єктів, двічі клацають об'єкт і переміщують чи повертають за допомогою кола обертання.





ПРИГАДАЙТЕ:

- які клавіші клавіатури належать до клавіш керування.



ВИВЧАЄМО

ВИ ДІЗНАЄТЕСЯ:

- які є режими роботи з середовищем *Tinkercad*;
- як змінювати властивості об'єктів побудови;
- як працювати з фігурами в робочій площині.

20.1. Які є режими роботи з середовищем *Tinkercad*?

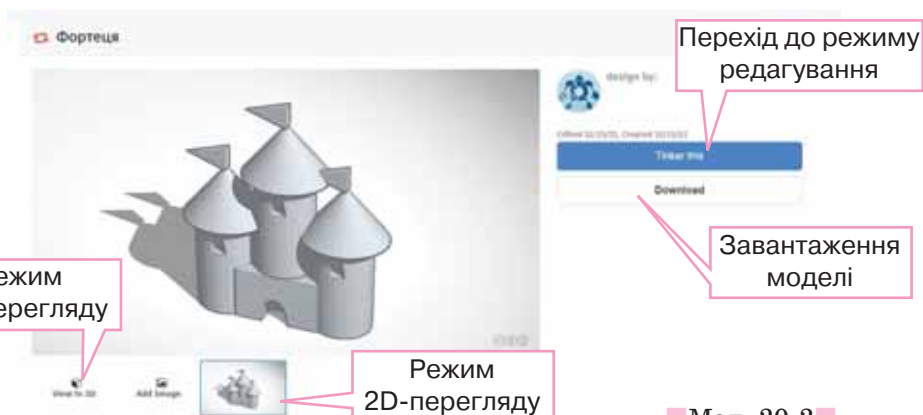
Однією з програм, в якій можна створювати та роздруковувати, завантажувати та змінювати 3D-моделі є середовище *Tinkercad* (<https://www.tinkercad.com/>). Після реєстрації у цьому середовищі користувачі мають змогу обрати різні режими роботи, серед яких є режим 3D-моделювання (мал. 20.1).

Є два способи створення нового проєкту.

- Редагування раніше створеного — для цього обирають проєкт із власного сховища або із галереї проєктів. Після цього модель можна переглядати в 2D, 3D та переходити до її редагування (мал. 20.2).



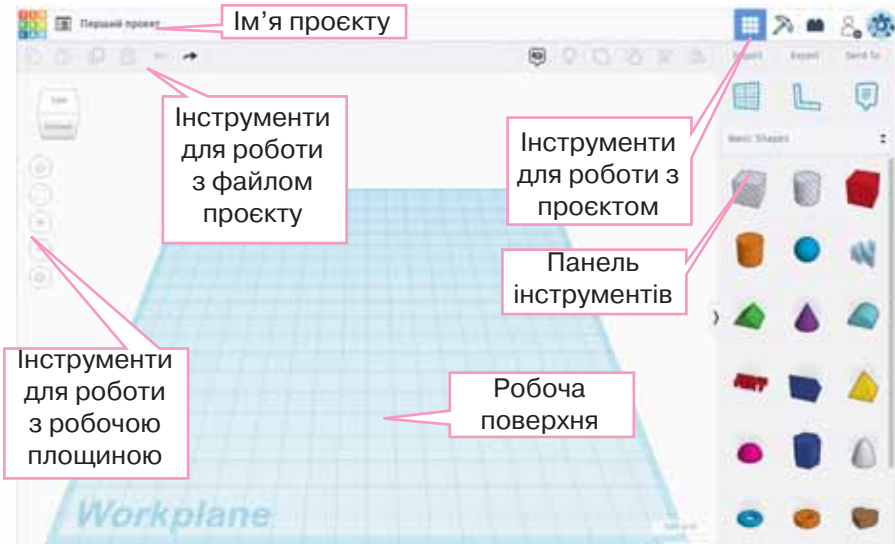
■ Мал. 20.1 ■



■ Мал. 20.2 ■

- Створення нового — з головного вікна переходять за допомогою кнопки **Create new design**.

Середовище моделювання складається із декількох областей (мал. 20.3).

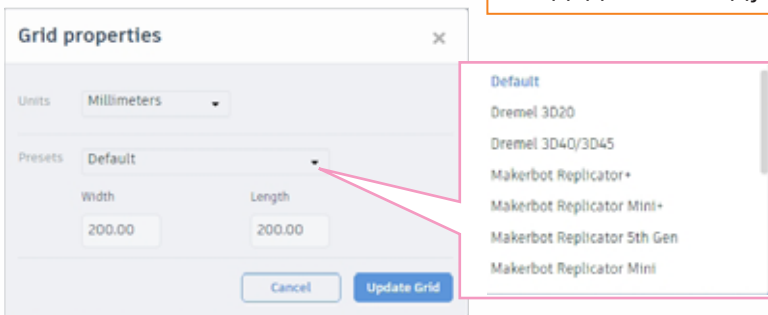


■ Мал. 20.3 ■

Щоб відредагувати сітку робочої поверхні, у правому нижньому куті натискають кнопку *Редагувати Сітку* (*Edit Grid*) та налаштовують одиницю вимірювання (можна перемикатися між дюймами та міліметрами), а також висоту та ширину сітки. Ця функція особливо зручна для тих, хто збирається друкувати створені моделі на 3D-принтері: можна задати сітці розміри друкованої поверхні принтера (крім осі Z), і тоді буде зрозуміло, якого розміру буде модель ще до друку (мал. 20.4).

ЦІКАВО

Автоматичне створення плоскої моделі із 3D-об'єкта має назву рендеринг. **Рендеринг** — це одна зі складових чи етапів візуалізації. Рендеринг дає змогу отримати готове зображення тримірної моделі в «плоскому» варіанті як відображення на площину перегляду. Залежно від складності об'єкта і якості зображень, цей процес може тривати від декількох хвилин до декількох годин. Професійні редактори 3D-графіки мають вбудовані засоби рендерингу або для такого проектування використовують додаткові модулі-програми.



■ Мал. 20.4 ■



ДОСЛІДЖУЕМО

Вправа 1. Середовище *Tinkercad*.

Завдання. Дослідіть призначення інструментів для роботи із робочою поверхнею середовища *Tinkercad*.

1. Завантажте середовище *Tinkercad*.
2. Створіть новий проєкт. Змініть назву проєкту у полі імені на *Перший проєкт*.
3. Пригадайте або визначте з допомогою перекладача переклад слів *TOP*, *FRONT*, *LEFT*, *RIGHT*, *BACK*, *BOTTOM*. За допомогою миші повертайте кубиком розміщення площини та переконайтесь, що назви на гранях куба відповідають напряму повороту робочої площини згідно із визначеним перекладом.



4. Дослідіть призначення інструментів . Який із інструментів повертає вигляд робочої площини в початковий? Як наблизити/віддалити робочу площину? Зробіть висновок про способи керування виглядом робочої площини.
5. Визначте, які події відбуваються в середовищі при:
 - переміщенням колеса миші вперед та назад;
 - утримуванні CTRL і лівої кнопки миші та переміщенні курсора по екрану;
 - утримуванні CTRL+SHIFT і лівої кнопки миші та переміщенні курсора по екрану.Зробіть висновки.

ЦІКАВО

Інструмент  задає ортогональний вид моделі — відключення вигляду перспективи. Такий режим дає змогу найбільш точно вирівняти окремі фігури одну відносно іншої і чітко їх розмістити на робочій площині.

20.2. Які є режими роботи з середовищем *Tinkercad*?

Об'єкти побудови моделей згруповані у бібліотеці форм. За замовчуванням доступний список основних форм. Переміщення 3D-примітиву з бібліотеки на робочу площину здійснюють простим перетягуванням. Властивості кожної з фігур доступні в режимі редагування (мал. 20.5).



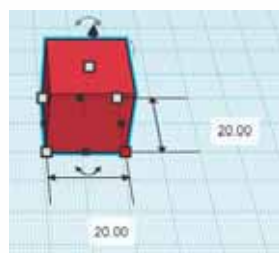
Мал. 20.5

Колір заповнення фігури обирають на палітрі кольорів після натиснення *Solid* (мал. 20.6).



Мал. 20.6

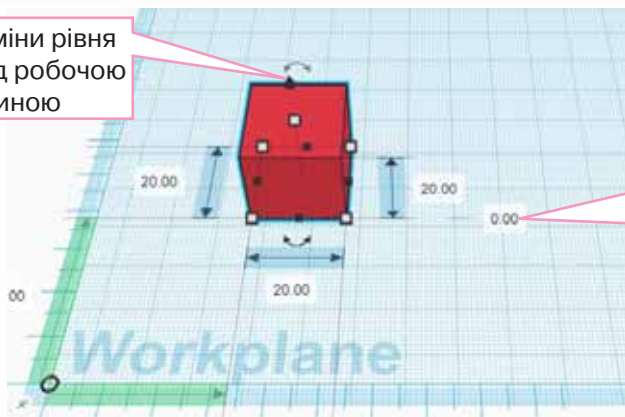
Для того, щоб змінити розміри фігури, переміщують маркери розмірів (квадратики). Якщо затримати вказівник миші біля одного з таких маркерів, то отримуємо значення відповідного параметра у вибраних одиницях вимірювання (мал. 20.7).



Мал. 20.7

Для визначення та встановлення розмірів об'єктів зручно використовувати інструмент *Лінійка*. Якщо обрати  та перемістити її у нижній лівий кут робочої площини, тоді розміри фігури будуть відображені як на малюнку 20.8.

Маркер зміни рівня
об'єкта над робочою
площиною



Значення
параметра
рівня об'єкта
над робочою
площиною

Мал. 20.8

Зверніть увагу на чорну стрілку вгорі зображення — це інструмент, яким можна піднімати об'єкт над рівнем площини.

20.3. Як працювати з фігурами в робочій площині?

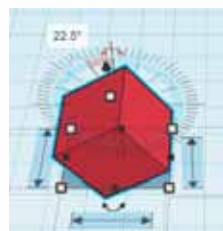
Фігури у робочій площині можна змінювати, переміщувати, видаляти та обертати. Перед виконанням кожної із дій фігуру або групу фігур потрібно вибрати.

Для **вибору** фігури, розташованої на робочій площині, натискають на неї мишею або, утримуючи мишу, обводять фігуру прямокутним виділенням. Для вибору кількох фігур можна обвести їх прямокутним виділенням або, утримуючи *SHIFT*, клацнути потрібні фігури. Кількість вибраних фігур відображається у дужках у шапці редактора фігур. Для зняття вибору фігури клацають у вільному місці робочої площини чи простору або на іншій фігурі.

Щоб **видалити** одну або кілька фігур, обирають потрібні та використовують клавіші *DELETE/BACKSPACE* або іконку сміттового бака. Для скасування дії використовують комбінації клавіш *Ctrl+Z*, для повтору *Ctrl+Y* (або іконки стрілок у верхньому лівому куті).

Щоб **перемістити** фігуру, наводять курсор на фігуру і, утримуючи ліву кнопку миші, переміщують об'єкт у бажану позицію. Виділений об'єкт також можна переміщувати за допомогою стрілок на клавіатурі. Якщо є потреба рухати об'єкти прямими лініями, то переміщення здійснюють при натиснутій клавіші **SHIFT**.

Для **обертання** фігури використовують вигнуті стрілки вздовж осей *X*, *Y*, *Z*, які для куба задають довжину, ширину та висоту. Для цього затискають вигнуту стрілку — навколо фігури з'явиться шкала градусів. Якщо пересувати мишу всередині шкали, то об'єкт обертатиметься на $22,5^\circ$, якщо ззовні — то на 1° . Також можна встановити конкретне значення градусів у невеликому білому полі поруч із шкалою (мал. 20.9).



Мал. 20.9

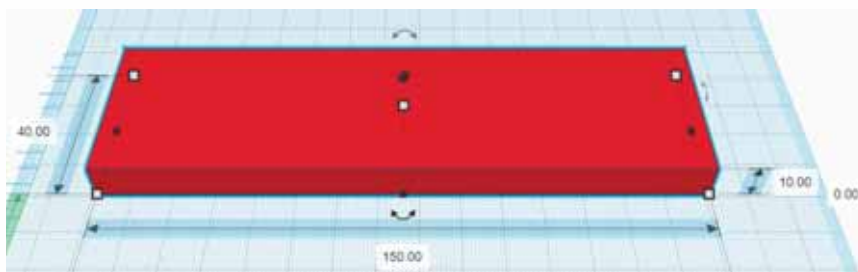
Для того, щоб продублювати об'єкт використовують комбінацію клавіш *Ctrl+D*. Тоді залишиться тільки розділити копію та оригінал (стрілками перемістити).



Вправа 1. Шрифт Брайля.

Завдання. Учні 9 класу вирішили розробити іменні пластинки із написами для слабозорих дітей, які уміють розпізнавати на дотик шрифт Брайля. Створіть таку пластинку для імені Ірина.

1. У середовищі *Tinkercad* створіть проєкт з назвою *Новий зір*.
2. Розмістіть пластинку з параметрами, які вказані на малюнку 20.10.

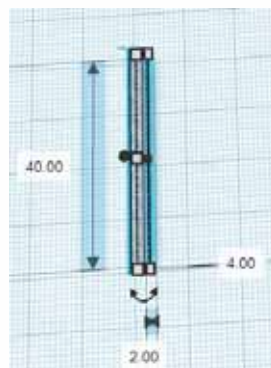


Мал. 20.10

3. Перейдіть до ортогональної проекції.
4. Відкрийте файл для літер українського алфавіту з файлу *Шрифт Брайля* (<https://cutt.ly/bYMNE5n>). Сплануйте, як за допомогою цього шифру закодувати потрібне слово. Перевірте, чи вийшло у вас таке:



5. Розділіть основу на 5 частин — за кількістю літер. Використайте прозорий прямокутник (мал. 20.11). Продублюйте його ще 4 рази (*Ctrl+D*). Розподіліть рівномірно розділювачі.
6. Для створення крапки використайте кулю (мал. 20.12).

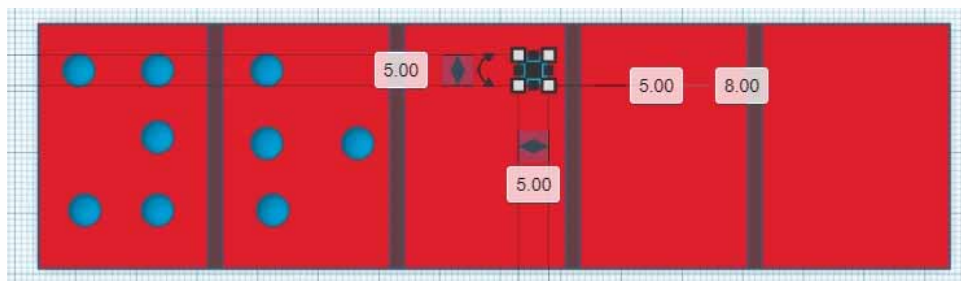


Мал. 20.11



Мал. 20.12

7. Створюйте код. Використайте вигляд зверху. Для правильного розміщення крапок користуйтеся напрямними лініями (мал. 20.13).



Мал. 20.13

8. Завершіть створення проєкту. Клацніть на іконку *Tinkercad* . Перегляньте проєкт у списку (мал. 20.14)



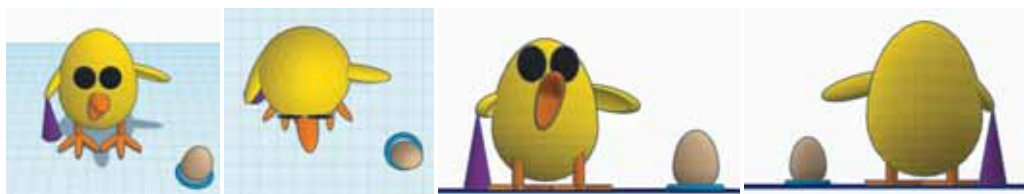
Другий зір

Мал. 20.14



ПРАЦЮЄМО В ПАРАХ

1. Обговоріть, які режими перегляду обрано для однієї і тієї само моделі для кожного із зображень. За якими ознаками ви це визначили?



2. Обговоріть, які геометричні фігури використано для створення 3D-зображення замкової вежі на малюнок 20.2. Обговоріть, які геометричні фігури використано для її створення та чи маєте ви достатньо засобів чи умінь, щоб створити подібний малюнок.

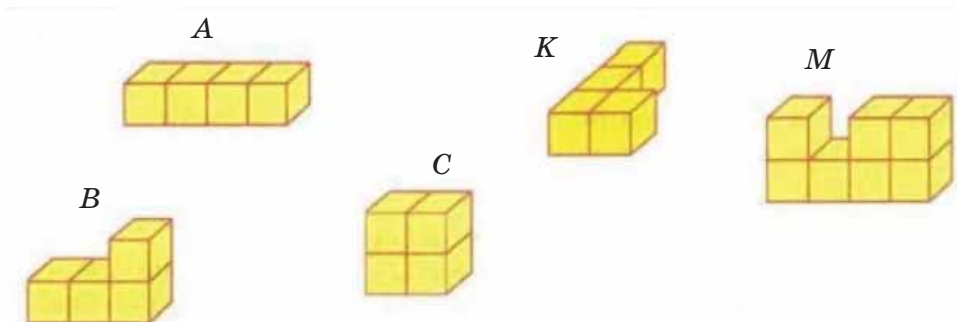
-  3. Обговоріть, які засоби середовища *Tinkercad* можна використати, щоб створити зображення. Поміркуйте, як можна використати такі моделі в житті. Оберіть одну з них та побудуйте. Оберіть одне з них та створіть. Продемонструйте свою роботу в класі.



ПРАЦЮЄМО САМОСТІЙНО

4. Розгляньте приклади об'єктів, створених в середовищі *Tinkercad*. Визначте, із яких геометричних примітивів вони складаються. Створіть обране зображення.

-  5. Продовжте роботу над створенням засобів орієнтування для слабозорих дітей. Розробіть 3D-модель, яка допоможе дітям в орієнтуванні на місцевості, у школі, на транспорті (наприклад, вказівні стрілки).



-  6. Придумайте ідею власного 3D-зображення та реалізуйте її в середовищі *Tinkercad*. Продемонструйте модель у класі.

ПРИГАДАЙТЕ:

- інструменти малювання у графічних редакторах;
- властивості символів тексту в електронному документі.

ВИ ДІЗНАЄТЕСЯ:

- як змінювати форму графічних 3D-примітивів;
- як створити власну фігуру;
- як додавати текстові об'єкти.



ВИВЧАЄМО

21.1. Як змінювати форму графічних 3D-примітивів?

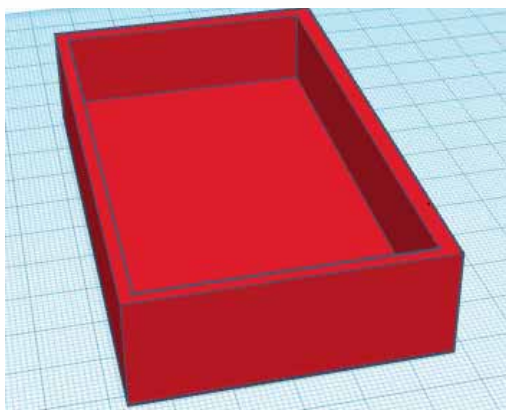
Наявні 3D-примітиви можна змінювати за рахунок зміни кількості граней, згладжування вершин чи кількості сторін (табл. 21.1).

Таблиця 21.1

Початкова форма	Змінена форма	Властивість
		Радіус 500
		Сторони 12
		Кроки 5
		Сторони 3
		Верхній радіус 10 Нижній радіус 01
		Кроки 3

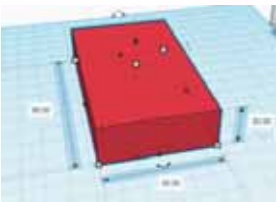
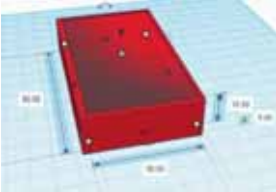
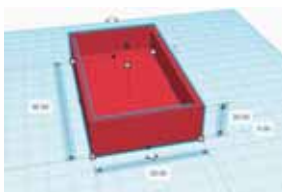
Зміна форми може відбуватись також за рахунок відсікання фрагментів об'єктів.

Щоб утворити об'єкт як на малюнку 21.1. виконують декілька кроків (табл.21.2).



Мал. 21.1

Таблиця 21.2

Крок 1. Побудувати основну фігуру		Крок 2. Продублювати основну фігуру, зменшити розміри. Змінити стиль заповнення на прозорий	
Крок 3. Перемістити допоміжну фігуру в основну		Крок 4. Виділити обидві фігури. Згрупувати	

ЦІКАВО

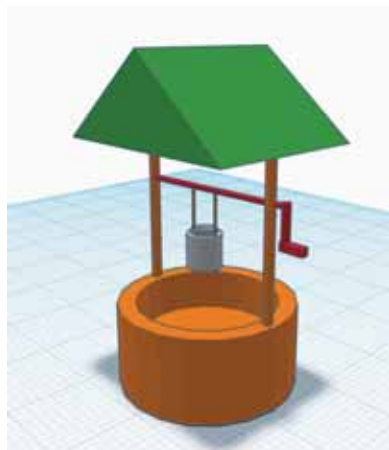
В професійних редакторах 3D-графіки використовують інструмент *Екструдкування* (видавлювання), який дає змогу змінювати геометрію об'єктів у режимі редагування за рахунок створення копій ребер і граней шляхом свого роду видавлювання – переміщення та зміни розмірів.



ДІЄМО

Вправа 1. Декорації.

Завдання. Для лялькової вистави активне учнівство розробляє декорації. Створіть 3D-модель криниці за зразком (мал. 21.2).

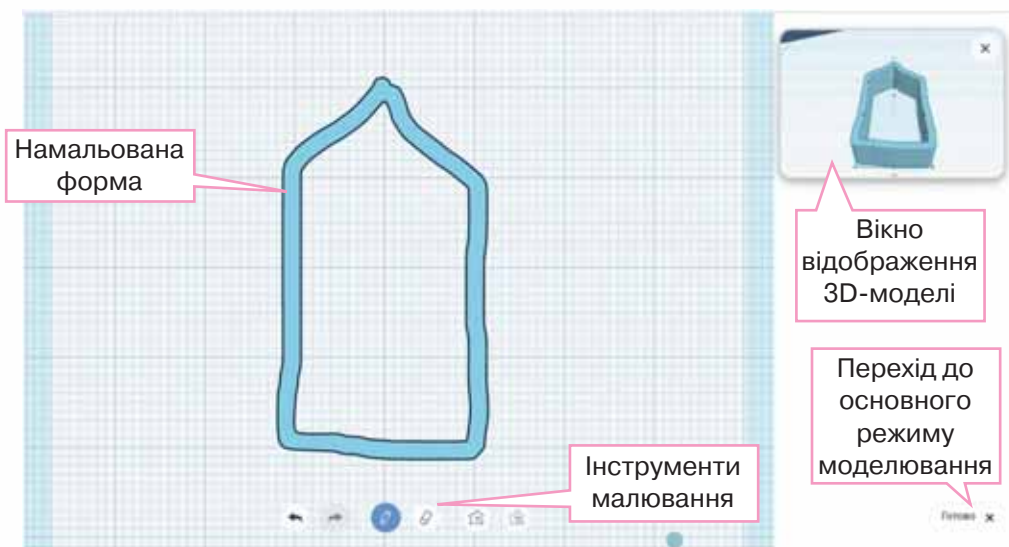


Мал. 21.2

21.2. Як створити власну фігуру?

Для створення власної фігури можна використати інструмент вільного малювання *Scribble* .

Коли клацнути інструментом *Scribble* на робочій області, ми переходимо в режим промальовування фігури (мал. 21.3).



Мал. 21.3

ЦІКАВО

В Україні є унікальний замок-корабель. Це Сидорівський замок на Тернопільщині (мал. 21.4). Форма малюнка 21.2 нагадує обриси фундаменту цього замку. Технологія створення фігур промальовуванням за давніми кресленнями або аерозйомками будівель допомагає дослідникам готувати їх 3D-реконструкції (<http://stem.tnpu.edu.ua/gallery/sidorivskij-zamok#gallery-10>).



■ Мал. 21.4 ■

Власні фігури часто використовують для створення персонажів 3D-ігор чи мультфільмів. За рахунок зміни фрагментів об'єкта, які розміщуються на декількох картинках підряд, при швидкій зміні таких кадрів (не менше 16 кадрів на секунду) досягається ефект анімації 3D-моделей.



ДІЕМО

Вправа 2. Ландшафтний дизайн.

Завдання. Намалуйте ескіз форми прибудинкової території за допомогою інструмента малювання власних форм. Розмістіть на ній вазони та інші об'єкти, які можна обрати із бібліотеки *Featured Collections* (мал. 21.5).

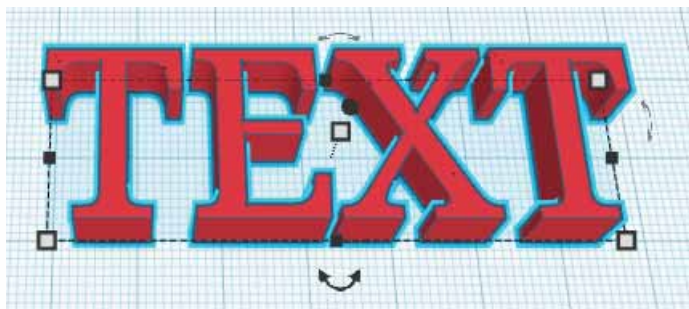


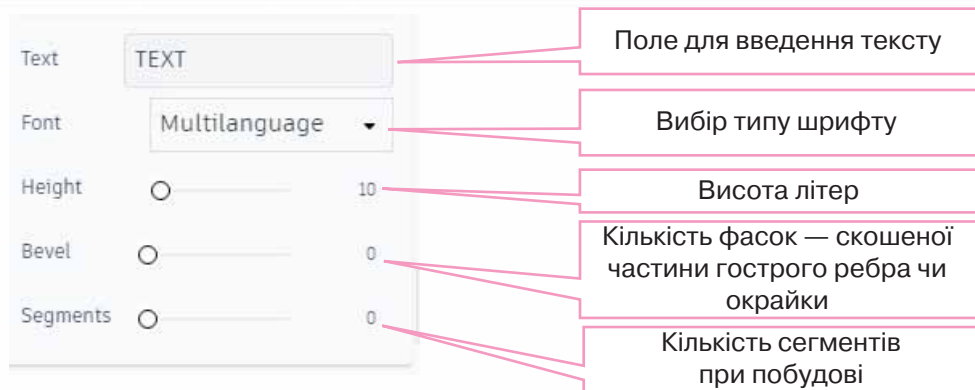
■ Мал. 21.5 ■

21.3. Як створити власну фігуру?

Для того, щоб додати текстові об'єкти до 3D-моделей, використовують інструмент **TEXT**.

На робочій площині отримаємо напис із вказаними властивостями (мал. 21.6).

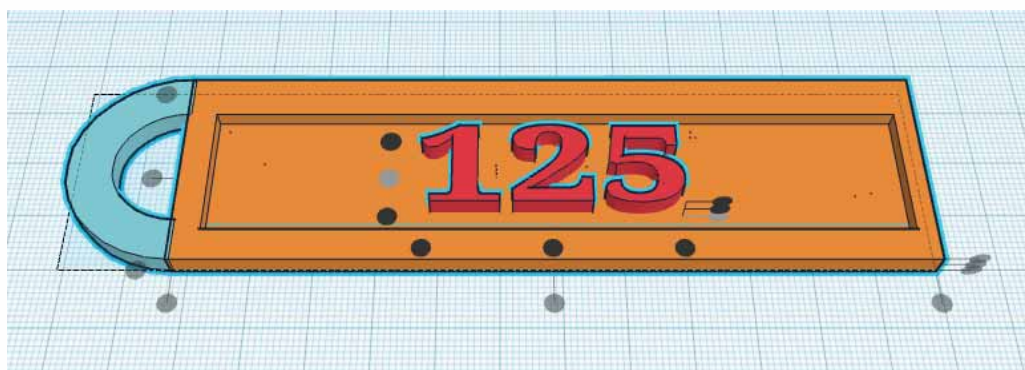




■ Мал. 21.6 ■

Можна використати також фігурний текст. У бібліотеці *Shape Generation* можна обрати фігурний текст: довкола , по колу . А в бібліотеці *Design Starters* можна обрати окремі літери чи цифри, наприклад,  .

Щоб вирівняти (відцентрувати) один об'єкт відносно іншого, виділяють об'єкти та використовують інструмент . Тоді за допомогою маркерів вирівнювання легко розмістити потрібні об'єкти (мал. 21.7).



■ Мал. 21.7 ■



ДІЄМО

Вправа 3. Брелок.

Завдання. Створіть брелок для театрального гардероба, як на малюнку 21.7. Прикрасьте його елементами декору на власний розсуд.



ПРАЦЮЄМО В ПАРАХ

1. Обговоріть як інструменти середовища *Tinkercad* використали для створення кафе (мал. 21.8). Поміркуйте, за скільки кроків можна отримати подібне зображення. Перевірте свої здогадки під час побудови моделі. Порівняйте створені моделі та кількість кроків із сусідом чи сусідкою по парті.



Мал. 21.8



2. Обговоріть як інструменти середовища *Tinkercad* використали для створення 3D-моделей (мал. 21.9). Сформулюйте план створення однієї з моделей. Створіть модель за планом щоразу чергуючись: хтось називає пункт плану, а хтось – виконує дію.



Мал. 21.9



3. У середовищі *Tinkercad* створено 3D-модель замку (мал. 21.10). Визначте, які геометричні фігури використані для його побудови. Розподіліть обов'язки між собою, хто яку частину замку створюватиме. Обговоріть, які вказівки меню роботи з об'єктом (*Експорт/Імпорт*) можуть допомогти вам обмінятися створеними фрагментами.



Мал. 21.10



ПРАЦЮЄМО САМОСТІЙНО

4. Визначте, якими можуть бути значення властивостей текстового об'єкта, створеного в середовищі *Tinkercad* (мал. 21.11). Створіть напис за прикладом та визначеними значеннями властивостей.



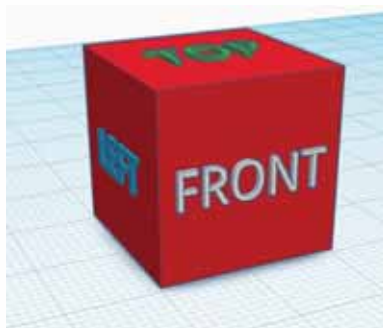
■ Мал. 21.11 ■

5. Використайте заготовку горнятка із бібліотеки *Everyday Objects*. Додайте вушко (мал. 21.12). Оформіть його зовнішню стінку написом чи малюнком, щоб створити іменне горнячко.



■ Мал. 21.12 ■

6. Створіть кубик, який використовують в середовищі *Tinkercad* для обертання робочою площиною (мал. 21.13).



■ Мал. 21.13 ■

7. Спроектуйте та створіть відповідну 3D-модель медалі для шкільного чемпіонату із ігрових видів спорту.
8. Спроектуйте та створіть відповідну 3D-модель об'єкта, який вивчається на уроках фізики, хімії, біології.



ПРИГАДАЙТЕ:

- сучасні пристрої виведення інформації;
- засоби для створення та відтворення відео.

ВИ ДІЗНАЄТЕСЯ:

- де використовують технології 3D-друку;
- які пристрої та технології використовують у 3D-друці;
- з яких етапів складається 3D-друк.



ВИВЧАЄМО

22.1. Де використовують технології 3D-друку?

3D-друк — це методика виготовлення об'ємних виробів на основі цифрових моделей. Незалежно від конкретної технології, суть процесу полягає в поступовому пошаровому відтворенні об'єктів.

Застосування технології 3D-друку сьогодні дуже поширене, також і сфера її застосування постійно розширюється. Окрім таких специфічних галузей, як біодрук, кулінарія та розробка дизайнерського одягу й аксесуарів, звичайний користувач за допомогою настільних 3D-принтерів може виготовляти дослідні зразки. Створення таких зразків за допомогою 3D-друку значно скорочує час і витрати виробництва. А завдяки можливостям 3D-моделювання спектр проєктованих деталей практично не обмежений. Це дає змогу наочно оцінити можливі недоліки виробу ще на етапі проєктування і внести істотні зміни в конструкцію деталі ще до її остаточного затвердження.

Ефективним застосування 3D-друку є в дрібносерійному виробництві. Властивості багатьох матеріалів дають змогу виробляти готові компоненти з мінімальними витратами. Порівняно з традиційними методами виробництва, дрібносерійне виробництво за допомогою 3D-друку дуже вигідне з фінансової точки зору. Виготовлення, наприклад, ливарних форм, являє собою тривалий і дорогий процес. До того ж саме лиття під тиском займає чимало часу. На 3D-принтері ж надрукувати партію необхідних виробів можна за лічені години. Це застосування 3D-друку вкрай актуальне вразі частих замовлень на невеликі партії деталей.

Ще одне застосування 3D-друку — ремонт і відновлення пошкоджених деталей. Проводити таку процедуру можна як самостійно, за наявності відповідних навичок й устаткування, так і в спеціалізованих сервісах 3D-друку. Спершу на основі пошкодженого виробу будується правильна 3D-модель. Для спрощення проєктування також може бути використане 3D-сканування. Далі готова модель відправляється в друк

і відтворюється на 3D-принтері в потрібній кількості примірників. Ремонт і відновлення пошкоджених деталей за допомогою 3D-друку відбувається швидко, а наявність цифрової моделі компонента дає змогу заново віддрукувати його в будь-який час.

Один з різновидів промислового застосування 3D-друку — виробництво функціональних моделей і готових компонентів. Виготовлення виробів на 3D-принтері з прозорого матеріалу дає змогу побачити роботу функціональної деталі «зсередини», що дуже корисно під час розробки різних інженерних зразків.

На 3D-принтері можна надрукувати іграшки, сувеніри, фігурки будь-яких персонажів й атрибутів комп'ютерних ігор і фільмів. А кольоровий 3D-друк дасть змогу виготовити ексклюзивні повнокольорові сувеніри — мініатюрні фігурки реальних людей. Для цього цифрова модель людини формується на основі даних 3D-сканування. До того ж всі текстури й дані про колір зберігаються.

Цікавим є застосування 3D-друку для відтворення моделей утрачених об'єктів, особливо предметів, які мають історичну чи культурну цінність.

Наприклад, 3D-модель Скалатського замку, що на Тернопільщині, у масштабі 1:200 відновлена за кресленнями та фото з історичних архівів студентами Тернопільського національного педагогічного університету імені Володимира Гнатюка (<http://stem.tnpu.edu.ua/gallery/skalatskij-zamok>) (мал. 22.1).



■ Мал. 22.1 ■

21.2. Які пристрої та технології використовують у 3D-друці?

Для створення 3D-моделей застосовується спеціальний електронний пристрій — 3D-принтер (мал. 22.2, с. 227).

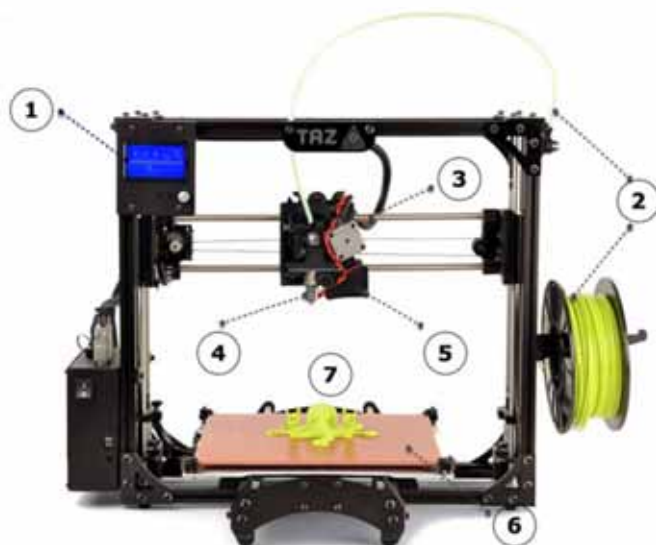


■ Мал. 22.2 ■

ЦІКАВО

Термін «3D-друк» придумали нещодавно — у 1995 р. в Массачусетському технологічному інституті. І відтоді всі машини, що забезпечують 3D-друк, ми називаємо 3D-принтерами.

3D-принтери мають різний вигляд. Але зазвичай використовуються однакові складові (мал. 22.3).



■ Мал. 22.3 ■

1. Панель управління 3D-принтера.
2. Пластик для 3D-друку (філамент).
3. Екструдер.
4. Хотенд (гарячий кінець).
5. Вентилятор (кулер).
6. Друкована платформа (стіл).
7. 3D-об'єкт.

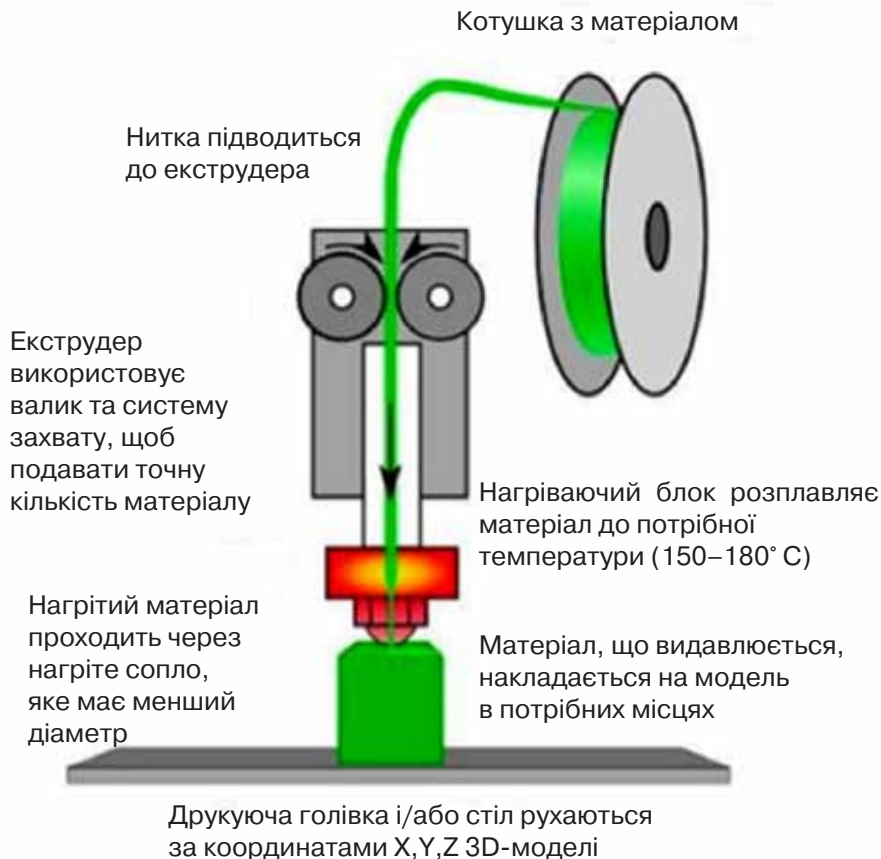
У сучасних 3D-принтерах матеріалами можуть бути фотополімерна смола, порошок, силікон, різні метали, віск, гіпс, різні види пластику, шоколад, цемент й інші.

Для відтворення різних об'єктів використовують різні технології 3D-друку. Вони відрізняються як застосовуваними витратними матеріалами, так і швидкістю і точністю друку.

1) *Моделювання методом наплавлення (Fused deposition modeling, FDM)*. Це одна з найпоширеніших технологій 3D-друку. Застосовується в більшості настільних 3D-принтерів. Друк відбувається за допомогою пошарової подачі нитки розплавленого пластику (мал. 22.4).

ЦІКАВО

Технологія пошарового наплавлення (FDM) (струменевий 3D-друк), винайдена ученим на ім'я Скотт Крамп. Ця технологія працює за досить простим принципом, що нагадує роботу машинки для шиття. Саме тому 95 % усіх сучасних 3D-принтерів використовують саме цю технологію.



Мал. 22.4

2) *Стереолітографія (Laser stereolithography, SLA)*. Формування об'єкта відбувається за рахунок пошарового засвічування лазером рідкої фотополімерної смоли, яка твердіє під впливом випромінювання. Одна з варіацій даної технології — DLP 3D-друк. У ній замість лазера застосовується спеціальний проєктор. Обидва методи 3D-друку застосовуються для створення об'єктів з високим ступенем деталізації. У разі DLP-друку додатковою перевагою є також швидкість.

3) *Селективне лазерне спікання (Selective laser sintering, SLS)*. Відтворення виконується за рахунок пошарового плавлення спеціального порошку під дією лазерного випромінювання. Цей метод 3D-друку широко застосовується в промисловості для виготовлення міцних металевих елементів.

У процесі 3D-друку задіяні такі складові 3D-принтера: інтерфейс управління, платформа друку, екструдер, друкувальна голівка.

Інтерфейс управління: деякі сучасні 3D-принтери мають сенсорний екран, який використовується для управління 3D-принтером. На старих принтерах замість сенсорного інтерфейсу може бути простий рідкокристалічний дисплей з фізичною прокруткою і колесом управління. Залежно від моделі також можуть бути наявні слот для SD-карти та USB-порт.

Платформа, або стіл 3D-принтера — це, по суті, поверхня, на якій виготовляються деталі. Платформи найчастіше виготовляють з підігрівом, щоб поліпшити адгезію деталі, але про це пізніше.

Екструдер — це компонент, відповідальний за плавлення і поступове осадження пластикової нитки для побудови моделі.

Насправді екструдер складається з двох підкомпонентів: гарячого й холодного. **Гарячий кінець**, або **хотенд**, містить нагрівач і сопло, які фактично розплавляють матеріал, у той час як холодний кінець складається з двигуна, приводних шестерень та інших дрібних компонентів, які проштовхують нитку в хотенд для розплавлення. Між гарячим і холодним кінцями розташовані радіатор і вентилятор, оскільки необхідно, щоб холодний кінець залишався холодним, аби уникнути заклинювання. На додаток до вентилятора радіатора зазвичай є, принаймні, ще один вентилятор, призначений для охолодження розплавленої нитки після того, як вона виходить з екструдера, — він зазвичай називається вентилятором охолодження деталей.

Друкувальна голівка: на друкувальній голівці встановлено один або кілька екструдерів (стандартні 3D-принтери мають один екструдер). У верхній частині друкувальної головки міститься трубка, по якій нитка подається в друкувальну голівку.

Процес відбувається так. Після запуску завдання на друк сопло починає нагріватися. Коли сопло досягає температури, необхідної для плавлення нитки, екструдер втягує нитку в гарячий кінець для підготовки



до моделювання методом наплавлення. Тепер принтер готовий приступити до 3D-друку деталі. Друкувальна голівка опускається на робочу поверхню (платформу) і починає наплавляти філамент, який охолоджується і твердне незабаром після виходу із сопла завдяки вентиляторам охолодження деталі. Пластик наноситься по одному шару за раз, і після того, як один шар буде завершено, друкувальна голівка переміщається вгору по осі Z на невелику відстань, і процес повторюється доти, доки деталь не буде завершена.



Вправа 1. 3D-друк.

Завдання. Розгляньте відеозапис роботи 3D-принтера, що збережений у папці 3D-друк за посиланням: <https://cutt.ly/eE8Ducl>. Обговоріть, яку технологію продемонстровано на цьому відео.

22.3. З яких етапів складається 3D-друк?

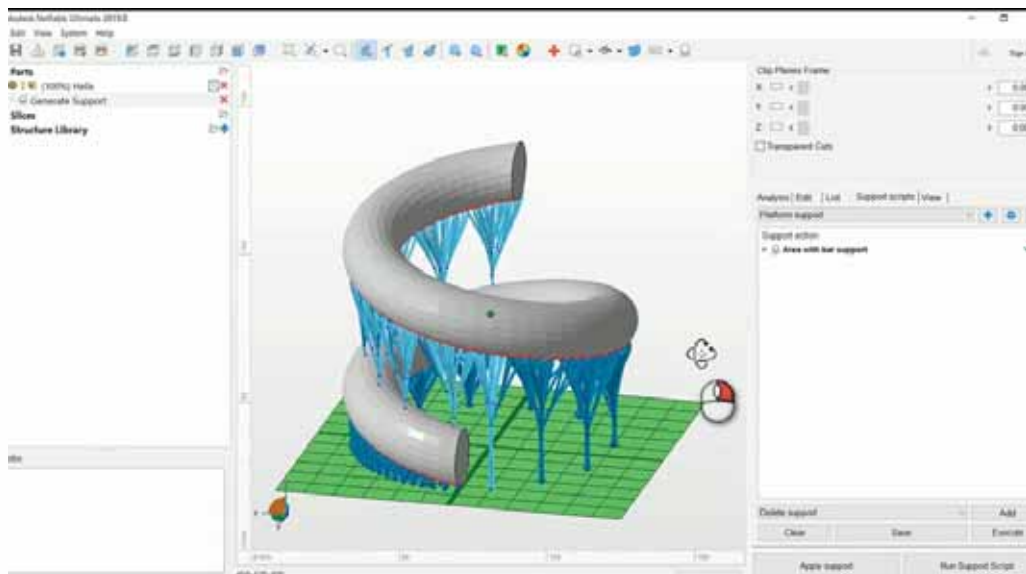
Перш ніж приступити до друку об'єкта на 3D-принтері, виконують ряд підготовчих кроків.

- 1) Виконується 3D-моделювання необхідного об'єкта за певними правилами.
- 2) Файл із цифровою моделлю завантажується в програму-слайсер, у якій генерується керуючий код для 3D-принтера.
- 3) Встановлюються необхідні параметри 3D-друку.
- 4) Код записується на знімний носій пам'яті, який підключається до 3D-принтера.
- 5) 3D-модель відтворюється.

Відтворення об'єктів відбувається поступово. Для набуття необхідної форми шар за шаром наноситься обраний матеріал, формуючи готовий виріб. Варто відзначити, що можливості 3D-друку практично безмежні, тобто виготовити можна все, що завгодно. У деяких технологіях для дуже тонких нависаючих елементів передбачено наявність підтримок, завдяки яким можна уникнути їх провисання.

Розглянемо детальніше інформацію про програму, яка є проміжною між програмою створення 3D-моделі та програмою керування принтером, яка запускає на друк підготовлені файли. Використовують програми-слайсери, такі як *Cura*, *Simplify3D*, *Astrosprint*, *3DPrinterOS* та інші. Основне призначення програми-слайсера — «розрізати» модель у файлі на шари й задати координати пересування для екструдера принтера на кожному шарі. У вікні програми можна налаштувати товщину

шару, масштаб, щільність заповнення, швидкість друку в різних місцях моделі, створення спеціальних підставок для тисяч елементів моделі й інші сценарії поведінки принтера (мал 22. 5).



Мал. 22.5

Після задання параметрів друку файл із цією інформацією експортується на комп'ютер у форматі *.gcode*, після чого його можна завантажити в принтер через SD-карту чи безпосередньо від комп'ютера. Останнє варто робити дуже рідко, оскільки 3D-друк — це тривалий процес, комп'ютер може перейти в режим сну, і вся робота пропаде.



ПРАЦЮЄМО В ПАРАХ

1. Обговоріть статтю «Чи варто купувати 3D-принтери»: <https://3d4u.com.ua/uk/blog/post/94-chi-varto-kupuvati-3d-printer-za-i-proti-plyusi-i-minusi>. Висловіть свої судження з приводу «за» і «проти», які описано у статті.
2. Знайдіть в інтернеті інформацію про види пластику для 3D-принтерів. Запропонуйте одне одному найкращий вибір «ціна-якість» та обговоріть.
3. За малюнком 22.6 (с. 232) поясніть складові 3D-принтера й обговоріть призначення кожної складової.



Мал. 22.6



ПРАЦЮЄМО САМОСТІЙНО

4. За матеріалами сайту <http://thefuture.news/3d-printing> створіть лінію часу про виникнення 3D-принтерів.
5. Складіть презентацію про застосування 3D-друку в обраній сфері. Розмістіть презентацію на спільному ресурсі.
6. Попрацюйте в мініпроекті «Технології майбутнього». Знайдіть в інтернеті матеріали й підготуйте відеопредставлення технології 3D-біопринтингу.

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Я можу пояснити призначення тривимірного моделювання об'єктів реального світу.
- ☐ Я знаю основні принципи тривимірного моделювання.
- ☐ Я можу пояснити принцип отримання тривимірного анімованого зображення.
- ☐ Я вмію створювати просторові моделі з використанням тривимірних примітивів.
- ☐ Я вмію редагувати форму й вигляд тривимірних об'єктів, змінюючи властивості вершин, ребер, граней і поверхонь.
- ☐ Я вмію створювати анімаційні ефекти.



Антивіруси — програми, призначені для виявлення і лікування програм і файлів, що заражені комп'ютерним вірусом, а також для запобігання зараження файлу вірусом.

База даних (БД) — це поіменована, структурована сукупність взаємопов'язаних даних, які належать до певної предметної області.

Брандмауер (або міжмережевий екран) — окремий пристрій або спеціальна програма, який для захисту мережевого під'єднання здійснює створюють бар'єр між комп'ютером і мережею.

Діаграма — графічне зображення, що наочно відображає лінійними відрізками чи геометричними фігурами співвідношення між числовими даними.

Власницька, або пропрієтарна, ліцензія (від англ. *proprietary* — власницький) — передбачає, що розробник ПЗ дає дозвіл користувачу використовувати одну або декілька копій програми, але при цьому сам залишається правовласником усіх цих копій.

Драйвери — програми, призначені для взаємодії з різними пристроями, вони дозволяють операційній системі правильно використовувати будь-який пристрій, що входить до складу комп'ютера.

Загроза — це потенційна можливість певним чином порушити інформаційну безпеку.

Інструментальне програмне забезпечення — спеціальні програми для розробки прикладного та системного програмного забезпечення комп'ютера.

Кібербулінг — цькування із застосуванням цифрових технологій. Кібербулінг може відбуватися в соціальних мережах, платформах обміну повідомленнями (месенджерах), ігрових платформах і мобільних телефонах тощо.

Комп'ютерні віруси — це спеціальні програми в машинних кодах або фрагменти програм, здатні без відома та згоди користувача розмножуватися та розповсюджуватися на інші програми шляхом копіювання свого коду у файли, що зберігаються в системі.

Комп'ютерний вірус (англ. *computer virus*) — комп'ютерна програма, яка має здатність до прихованого самопоширення та спрямовані на знищення, пошкодження, викрадання даних, зниження або унеможливлення подальшої працездатності операційної системи комп'ютера.

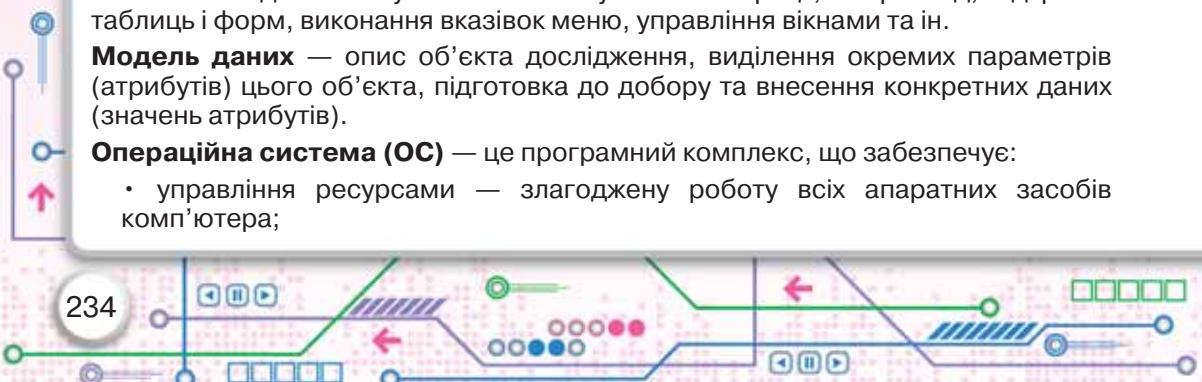
Ліцензія на програмне забезпечення — правовий документ, що визначає правила використання та поширення програмного забезпечення.

Макрос — набір макровказівок, які створюються користувачем для автоматизації виконання конкретних операцій. Макроси запускаються натисненням декількох клавіш і швидко виконують найбільш уживані операції, наприклад, відкриття таблиць і форм, виконання вказівок меню, управління вікнами та ін.

Модель даних — опис об'єкта дослідження, виділення окремих параметрів (атрибутів) цього об'єкта, підготовка до добору та внесення конкретних даних (значень атрибутів).

Операційна система (ОС) — це програмний комплекс, що забезпечує:

- управління ресурсами — злагоджену роботу всіх апаратних засобів комп'ютера;



- управління процесами — виконання всіх програм та їх взаємодію з пристроями комп'ютера та даними;
- взаємодію (обмін відомостями та даними) між користувачем і комп'ютером.

Полігон — трикутник, що задається координатами трьох точок у тривимірному просторі.

Програмне забезпечення (англ. *software*) — сукупність програм і правил, а також документації щодо управління роботою комп'ютера для опрацювання даних.

Програмне забезпечення прикладного рівня — комплекс прикладних програм, за допомогою яких виконуються конкретні завдання (від виробничих до творчих, розважальних і навчальних).

Предметна область — сфера застосування конкретної бази даних.

Резервна копія — копія окремих файлів, групи файлів або всього диску, що збережена на іншому носії, для наступного відновлення у випадку руйнувань, псування або втрати даних на головному носії.

Система керування базами даних (СКБД) — програмне забезпечення, яке призначено для створення баз даних, оновлення даних, що зберігаються в них, забезпечення зручного доступу до баз даних з метою перегляду й пошуку даних.

Складність алгоритму — це кількісна характеристика, що відображує споживані алгоритмом ресурси під час свого виконання.

Фільтри — засіб швидкого виділення зі списку набору даних, які відповідають заданим умовам.

Фішинг — один з найпопулярніших і прибуткових (для тих, хто його реалізує) видів атак.

Форматування (англ. *formatting*) — процедура створення структур порожньої файлової системи вказаного типу — розподіл доріжок магнітного диска (дискети, твердого диска) чи іншого носія даних (наприклад, флеш-накопичувача чи карти флеш-пам'яті) на фізичні чи логічні записи, що виконується перед першим використанням диска.

Хакер — кваліфікований ІТ-фахівець, який розбирається в роботі комп'ютерних систем та здійснює втручання до комп'ютера, аби без відома власника дізнатися деякі особисті відомості, а потім ними скористатися, чи пошкодити дані, що зберігаються в комп'ютері. Їхні мотиви можуть бути різними: помста, самовираження (декто робить це задля розваги, інші — щоб показати свою кваліфікацію), винагорода.

Часова складність алгоритму — характеристика продуктивності алгоритму, що визначається кількістю елементарних операцій, які потрібно виконати для реалізації алгоритму.

Шкідлива програма — комп'ютерна програма або переносний код, призначений для реалізації загроз даним, що зберігаються в інформаційній системі, або для прихованого нецільового використання ресурсів системи, або іншої дії, що перешкоджає нормальному функціонуванню інформаційної системи.

3D-графіка, або тривимірна графіка, — це один з розділів комп'ютерної графіки, комплекс прийомів та інструментів, які дозволяють створити об'ємні (3D) об'єкти за допомогою форми й кольору.



Розділ 1. Програмне забезпечення та інформаційна безпека

Тема 1. Для чого потрібне програмне забезпечення?	4
Тема 2. Для чого використовується операційна система?	13
Тема 3. Які програми відносять до службових програмних засобів?	23
Тема 4. Які існують загрози безпеці при роботі за комп'ютером?	32
Тема 5. Які засоби допоможуть захистити дані та безпечно працювати на комп'ютері?	43

Розділ 2. Опрацювання табличних даних

Тема 6. Як здійснюють обчислення в табличному процесорі?	60
Тема 7. Як реалізувати вибір у середовищі табличного процесора?	70
Тема 8. Як побудувати та використовувати діаграми різних типів?	80
Тема 9. Як використовувати електронну таблицю для опрацювання однотипних даних?	99
Тема 10. Як готувати електронну таблицю для передавання?	118

Розділ 3. Алгоритми та програми

Тема 11. Як працюють з набором даних у мові програмування?	126
Тема 12. Як описують алгоритми опрацювання наборів даних мовою програмування <i>Python</i> ?	138
Тема 13. Як описують алгоритми аналізу наборів даних мовою програмування <i>Python</i> ?	149
Тема 14. Як описують алгоритми упорядкування наборів даних мовою програмування <i>Python</i> ?	156

Розділ 4. Бази даних. Системи керування базами даних

Тема 15. Що таке бази даних та які програми використовують для роботи з ними?	166
Тема 16. Як створюють базу даних?	178
Тема 17. Як отримати дані з бази даних?	183
Тема 18. Що є найважливішим під час роботи з базою даних?	192

Розділ 5. 3D-графіка

Тема 19. Що розуміють під 3D-графікою?	199
Тема 20. Як працювати з тривимірними примітивами в програмах для створення 3D об'єктів?	210
Тема 21. Як змінювати вигляд 3D-об'єктів?	218
Тема 22. Що таке 3D-друк?	225
Глосарій	234