

ВИДАВНИЦТВО
РАНОК

Олена Бондаренко
Василь Ластовецький
Олександр Пилипчук
Євген Шестопапов



**НАВЧАННЯ
БЕЗ МЕЖ**

ІНФОРМАТИКА



ОБ'ЄКТИ

Таблиця

Об'єкт, призначений для зберігання даних

Клієнти		
	Номер	Компанія
+	2	Компанія А
+	3	Компанія В
+	4	Компанія С

Запис

Поле

Запит

Засіб відбору та/або опрацювання даних

Упорядкований список відмінників: запит-вибір

Учень	Успішність
Порядкови	Порядкови
Прізвище	18 січня
Ім'я	20 січня

Поле:	Прізвище	Тематична
Таблиця:	Учень	Успішність
Сортування:	☒	☐
Відображення:	☐	☐
Критерії:		12
Або:		

Форма

Об'єкт для зручного перегляду та редагування даних

Форма 1

Відомості про товари і замовлення



САДИБА
ТОРГОВА КОМПАНІЯ

Типи

Постачальник

Товари

Замовлення

Друк звітів

Вихід з Microsoft Access

Вікно бази даних

Перелік полів

Умова відбору

Зв'язок

Елемент керування

Емблема

Заголовок

СКБД ACCESS

Звіт

Документ, створений на основі даних



Облік продажів автомобілів

Марка	I	II	III	IV	всього
КрАЗ	2	4	7	9	22
ЗАЗ	8	12	5	15	40
Богдан	9	21	10	9	49
Еталон	1	5	9	8	23
Електрон	17	9	11	10	47

Заголовок

Колонтитули

Дані

Макрос

Засіб автоматизації виконання завдань



Макрос1

Вкладений макрос: Підпрограма 1

Клацніть тут, щоб ввести примітку

Вибрати об'єкт

Тип об'єкта Таблиця

Ім'я об'єкта Успішність

В області переходів Ні

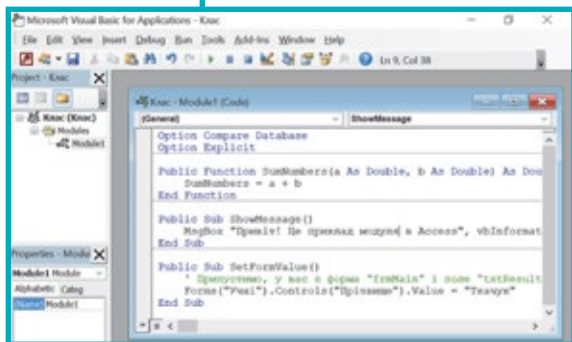
Відновити вікно

+ Додати нову дію

Завершити вкладений макрос

Модуль

Набір підпрограм мовою Visual Basic for Applications



Команда

Сполучення клавіш

Процедури

Функції

Оператори



Олена Бондаренко
Василь Ластовецький
Олександр Пилипчук
Євген Шестопалов

ІНФОРМАТИКА

Підручник для 9 класу
закладів загальної середньої освіти

Рекомендовано Міністерством освіти і науки України



Інтерактивний
електронний
додаток
rnk.com.ua/111374

Київ • Харків
Видавництво «Ранок»
2026



УДК 004:37.016(075.3)

I-74

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

Видано за рахунок державних коштів.

Продаж заборонено

Підручник створено відповідно до модельної навчальної програми
«Інформатика. 7–9 класи» для закладів загальної середньої освіти
(автори Бондаренко О. О., Ластовецький В. В.,
Пилипчук О. П., Шестопапов Є. А.)

Підручник успішно пройшов апробацію у 2025/2026 навчальному році
(наказ Міністерства освіти і науки України від 02.02.2026 № 142)

Інформатика : підруч. для 9 кл. закл. загал.
I-74 серед. освіти / О. О. Бондаренко, В. В. Ластовецький,
О. П. Пилипчук, Є. А. Шестопапов. — Харків : Вид-во
«Ранок», 2026. — 240 с. : іл.

ISBN 978-617-8775-57-5

УДК 004:37.016(075.3)



Інтерактивний
електронний додаток
доступний
за QR-кодом
або посиланням
rnk.com.ua/111374



**ЗРОБЛЕНО
В УКРАЇНІ**

ISBN 978-617-8775-57-5

© Бондаренко О. О., Ластовецький В. В.,
Пилипчук О. П., Шестопапов Є. А., 2026
© ТОВ Видавництво «Ранок», 2026



ДОРОГІ ДЕВ'ЯТИКЛАСНИКИ ТА ДЕВ'ЯТИКЛАСНИЦІ!

Ви тримаєте в руках підручник з інформатики, призначений саме для вас, учнів і учениць 9 класу. Ви вже набули навичок створення різних інформаційних продуктів за допомогою інструментів програмних середовищ, із якими ознайомились у 8 класі. Триває вивчення основ програмування мовою Python: ви навчилися розробляти програми з графічним інтерфейсом, знаєте, як задавати властивості найуживаніших об'єктів графічного інтерфейсу, як програмувати обробку подій.

Ви також розширили знання про апаратне забезпечення комп'ютера; з'ясували особливості опрацювання текстових документів на хмарному диску під час спільної роботи; навчилися створювати й опрацьовувати такі об'єкти мультимедіа, як звук і відео, опанували процес створення відеофільму та способи його розміщення в інтернеті.

Цей навчальний рік буде надзвичайно пізнавальним і відповідальним, адже ви здобуваєте базову середню освіту. І, як завжди, підручник буде вашим надійним помічником. Пригадаймо, як із ним працювати.

Навчальний матеріал підручника налічує шість розділів. Кожен розділ складається з параграфів і практичних робіт. Кожен параграф містить теоретичні відомості за темою уроку, приклади практичного застосування отриманих знань, матеріали за рубриками «Питання для самоперевірки», «Вправи», «Комп'ютерне тестування».

«Питання для самоперевірки» дадуть змогу з'ясувати, чи зрозуміли ви вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт за комп'ютером. «Вправи» складаються із завдань теоретичного і практичного спрямування; вони призначені для самостійного та групового виконання.

У підручнику ви знайдете покроковий опис *практичних робіт*. Щоб виконати кожен з цих робіт, потрібно повторити матеріал, вивчений на попередніх уроках, — тоді ви зможете успішно застосувати свої знання й удосконалити дослідницькі навички.

Інтерактивний електронний додаток містить додаткові матеріали до вправ і практичних робіт, тестові завдання з автоматичною перевіркою відповідей, 3D-моделі, навчальні презентації й відеоролики, інтерактивні вправи, довідкові матеріали та ін.

Бажаємо натхнення та успіхів!

Авторський колектив

Умовні позначення



Запам'ятайте



Розгляньте приклад



Ознайомтеся із цікавою інформацією



Зверніть увагу



Дайте відповідь на питання



Виконайте завдання за комп'ютером



Виконайте завдання підвищеної складності



Виконайте завдання в парах



Знайдіть відповідь в інтернеті



Виконайте тестові завдання за комп'ютером



ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ІНФОРМАЦІЙНА БЕЗПЕКА

1

РОЗДІЛ

У цьому розділі на вас чекає докладне знайомство з програмним забезпеченням комп'ютера. Ви також більше дізнаєтеся про захист комп'ютера та даних від шкідливих програм і різних інтернет-загроз.

§1

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРА

Комп'ютер є основою будь-якої сучасної інформаційної системи. Проте комп'ютер не може працювати без програмного забезпечення (ПЗ) — сукупності програм, які створюють умови для його роботи. Розрізняють системне, прикладне та службове ПЗ.

Системне програмне забезпечення



Системне програмне забезпечення (системні програми) — програмне забезпечення, призначене для керування роботою складових комп'ютера та обміну даними між ними.

► Операційні системи

Операційна система (ОС) — комплект програм, призначених для керування ресурсами, обчислювальними процесами комп'ютера, а також для забезпечення взаємодії з користувачем / користувачкою (див. 5 клас, § 5).

Докладно ознайомитися з тим, як класифікують ОС за деякими ознаками, ви можете за QR-кодом або посиланням rnk.com.ua/114523.



До **основних функцій ОС** належить введення-виведення даних, запуск і зупинка програм; забезпечення роботи пристроїв введення-виведення; завантаження програм в оперативну пам'ять та їх виконання; керування доступом до даних на зовнішніх носіях; забезпечення інтерфейсу користувача; підтримка файлової системи тощо.

На ПК і мобільних пристроях використовують різні ОС (рис. 1.1).

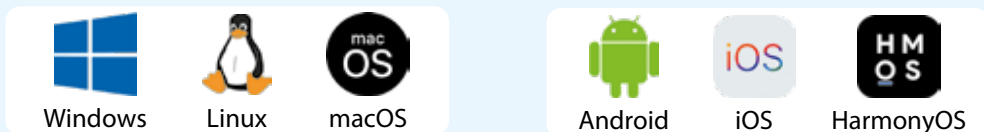


Рис. 1.1

Ознайоммося з найпопулярнішими **сімействами ОС** — групами, що мають спільні характеристики, архітектуру або походження. Це сімейства ОС Windows, Linux та ОС Android.

- Уже кілька десятиліть найпопулярнішими залишаються ОС сімейства Windows, які розробляє американська компанія Microsoft. Windows 11 — комерційна (див. далі) багатозадачна локальна 64-розрядна ОС із графічним інтерфейсом для персонального комп'ютера.
- Linux — ОС різноманітного призначення, побудовані на основі однойменного ядра. Використовуються на інтернет-серверах, суперкомп'ютерах, персональних та автомобільних комп'ютерах. ОС сімейства Linux постачаються у вигляді дистрибутивів, тобто збірок, які включають не лише ОС, а й набір застосунків. Більшість дистрибутивів Linux вільні, проте існують і комерційні.
- На ядрі Linux ґрунтується також ОС Android, розроблена компанією Google для мобільних телефонів і планшетів. Саме під керуванням різних версій ОС Android працює більшість смартфонів у світі.

Чи важлива сумісність при виборі програмного забезпечення?

ОС можуть значно відрізнятися одна від одної, тому здебільшого програма, розроблена для однієї ОС (приміром, Windows 11), не працюватиме або працюватиме неправильно в середовищі іншої ОС (приміром, Linux). У таких випадках кажуть про несумісність програми й ОС. Перед тим як купити і/або встановити програму, слід обов'язково звернути увагу на вимоги до комп'ютера та ОС.

► Драйвери

Взаємодію комп'ютера із зовнішніми пристроями забезпечують спеціальні програми — **драйвери**. Необхідно вибирати драйвер, призначений саме для встановленої на комп'ютері ОС. Драйвер для іншої ОС найпевніше працюватиме неправильно або не встановиться взагалі.



При підключенні, наприклад, нового принтера, потрібно, щоб особа із відповідними привілеями (наприклад, із правами адміністратора) установила на комп'ютер драйвер принтера потрібної моделі.

Прикладне програмне забезпечення



Прикладне програмне забезпечення (прикладні програми або застосунки) — програмне забезпечення, призначене для розв'язування конкретних завдань і потреб користувача / користувачки.

Прикладні програми поділяють за їх призначенням:

Прикладні програми	Призначення
Програми загального призначення	Різні галузі діяльності (наприклад, опрацювання текстів, малюнків, таблиць)
Програми спеціального призначення	Певні галузі діяльності або підприємства (наприклад, створення відеоефектів, моделювання інтер'єрів)
Інструментальні програми	Створення інших програм

Службове програмне забезпечення



Службове програмне забезпечення (службові програми) — програмне забезпечення, призначене для обслуговування апаратної і програмної складових комп'ютера.

До службових належать програми для архівування файлів, форматування та перевірки цілісності носіїв даних тощо.

Форматування носія даних (флешпам'ять, жорсткі (HDD), твердотільні (SSD) диски) — це процес розмічання простору зберігання даних. Воно є критично важливим для роботи й виконується перед першим використанням диска або виявленні серйозних збоїв у роботі. При цьому наявні на носії дані стають недоступними. Також можна перевірити цілісність носія.



Ліцензії на програмне забезпечення

Програми не можна використовувати без дозволу авторів (власників).



Ліцензія на програмне забезпечення — це угода, яка надає користувачу/користувачці право встановити його та використовувати.

Закон про авторське право забороняє виконувати над програмами дії, не дозволені ліцензією (модифікувати, розповсюджувати тощо).



GNU GPL (англ. *GNU General Public License* — Загальна громадська ліцензія проєкту GNU) — ліцензія на вільне програмне забезпечення, яка надає користувацькій спільноті широкі права.

Розгляньмо основні види ліцензій на програмне забезпечення:

Вид ліцензії	Особливості використання програм
Комерційна (commercial)	Власницькі програми (для їх використання необхідно купувати ліцензію)
Некомерційна (non-commercial use)	Програми, безкоштовні для некомерційного використання (версія комерційної програми з обмеженими функціями)
Умовно-безкоштовна (shareware)	Програми, якими можна користуватися безкоштовно за певних умов (низка комерційних програм надається для безкоштовного ознайомлення протягом певного періоду)
Вільна (free software)	Програми, які дозволено копіювати, модифікувати, розповсюджувати практично без обмежень

Встановлення програм

Перед початком використання програми потрібно налагодити її взаємодію з ОС, тобто **встановити програму** на комп'ютер. Цей процес називають також **інсталяцією**. Подивитися відеоролик про встановлення програми можна за QR-кодом або посиланням rnk.com.ua/114626.



Під час встановлення програми відбувається таке: в постійну пам'ять комп'ютера записуються її складники (виконувані файли, файли зображень, звуків, документації тощо); вносяться необхідні записи до файлів ОС; у головному меню ОС (в ОС Windows — у меню Пуск) створюються команди для запуску програми тощо.

Програми можна встановити, використовуючи встановлювальні файли, сховища програм, переносні програми. Розгляньмо ці способи.

▶ Встановлювальні файли

Розробник програми може поширювати її у вигляді встановлювального файлу. Це файл (виконуваний або спеціального формату), який потрібно відкрити на комп'ютері, після чого розпочнеться діалог щодо встановлення програми. В ОС Windows це може бути файл формату EXE або MSI, в ОС Linux — DEB або RPM тощо.

! Інсталювати програми за допомогою встановлювального файлу або зі сховища, видаляти програми (див. далі) у більшості ОС можуть лише особи, які мають права адміністратора комп'ютера.

З кількох варіантів отримання встановлювального файлу перевагу варто надавати тому, який пропонує розробник програми. Це допоможе уникнути файлів, змінених зловмисниками.

► Сховище програм

Розробники багатьох ОС пропонують не лише власне ОС, а й певний набір ПЗ, яке можна встановити додатково. Воно зберігається в інтернеті, доступ здійснюється за допомогою спеціальної програми.

- У сучасних версіях ОС Windows доступ до сховища забезпечує програма Microsoft Store, в ОС Linux Mint — інструмент Керування програмами, а в ОС Android — Play Маркет (рис. 1.2). Інтерфейси цих програм зазвичай відрізняються, проте призначення однакове: з'єднання зі сховищем і встановлення програм.

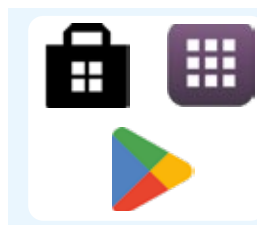


Рис. 1.2

Установлення програм із такого сховища має певну перевагу: вміщені в ньому програми розробники перевіряють на сумісність з ОС. Разом із тим, у сховищі часто містяться не найновіші версії програм, тому встановлювати нові версії доводиться в інший спосіб.

► Переносні програми

Якщо людина не планує користуватися програмою довго або не має прав адміністратора, потрібних, щоб її встановити, для ознайомлення з програмою можна скористатися її переносним варіантом.

- **Переносна (портативна) програма** (англ. *portable application*) — це програма, для запуску якої на комп'ютері не потрібне встановлення.

Для користування переносною програмою достатньо скопіювати на комп'ютер або флешнакопичувач папку з файлами програми, відшукати в ній потрібний виконуваний файл і запустити. Після цього вже можна створити ярлик на робочому столі для запуску програми.

Видалення програм

Програму, яка більше не потрібна на комп'ютері, варто видалити. Процес коректного видалення програми ще називають **деінсталяцією**.

Вирішивши видалити програму, не варто зразу видаляти папки, файли і ярлики, пов'язані з нею. Слід запустити системну програму для видалення інстальованих програм або програму деінсталяції певної програми. Тоді з ОС буде вилучено й решту змін (команди головного меню, ярлики тощо). Тільки для видалення переносної програми достатньо видалити файл або папку з її файлами.



Питання для самоперевірки

1. Що таке операційна система? Наведіть приклади.
2. Які бувають ліцензії на програмне забезпечення?
3. Що таке сумісність програм?
4. Що таке інсталяція й деінсталяція програми?
5. Які є способи встановлення програми на комп'ютер?



Вправа 1

- Установити на комп'ютер програму-архіватор PeaZip.
- 1. Відкрийте офіційну сторінку програми (<https://peazip.github.io/>)¹.
- 2. Залежно від установленної на комп'ютері ОС натисніть PeaZip Windows або PeaZip Linux.
- 3. Дізнайтесь у вчителя/вчительки, який саме варіант слід звантажити (32- чи 64-бітовий), клацніть його назву, натисніть посилання (наприклад, PeaZip 64 біт) і дочекайтесь завантаження файлу.
- 4. Запустіть установлювальний файл від імені адміністратора.
- 5. У діалоговому вікні підтвердьте намір установити програму.
- ⚙ 6. Перевірте, чи з'явилася команда для запуску програми в меню Пуск, запусіть програму. Покажіть результат учителю/вчительці.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111793



¹ PeaZip — вільна утиліта для архівування файлів. URL: <https://peazip.github.io/> (дата звернення 19.04.2026).

§2

АРХІВУВАННЯ ДАНИХ. АРХІВАТОРИ

Людина, яка користується смартфоном, час від часу стикається з проблемою нестачі пам'яті для програм і даних. Виявляється, у багатьох випадках можна зробити так, щоб файли займали обсяг пам'яті значно менший від їх звичного обсягу.



Архівування даних — процес стиснення даних, що зберігаються у файлах. Процес повернення заархівованих даних до початкового стану називають **розархівуванням (видобуванням) даних**.

Результатом архівування є **архів** — файл, який містить у стисненому вигляді один або більше файлів. Архівування даних дає змогу заощаджувати дисковий простір, швидше передавати файл за допомогою інтернету, захищати дані паролем тощо.

Принцип архівування даних

Поширеним методом стиснення даних є знаходження в них фрагментів, які повторюються, та їх кодування коротшими кодами. Такими фрагментами можуть бути пікселі растрового зображення, які мають однаковий колір, повторювані слова чи символи в тексті тощо.



Код растрового зображення (див. 8 клас, §3), яке має великі одноколірні ділянки, містить довгі послідовності однакових чисел:

5 4 **1 1 1 1 1 1 1 1** 6 6 **1 1** 6 6 6 6 6 6 6

Побудуємо на його основі послідовність пар чисел, у яких перше число (виділене нижче синім кольором) означає кількість повторень, а друге — число, що повторюється:

1 5 **1** 4 **8** **1** **2** 6 **2** 1 **8** 6

Цю послідовність можна прочитати так: одна п'ятірка, одна четвірка, вісім одиниць і т. д.

Отже, послідовність із 22 чисел ми замінили коротшою — із 12 чисел, за якою легко відновити початкову.



Описаний у прикладі 1 спосіб архівування легко реалізувати у вигляді програми мовою програмування Python, робота якої зрозуміла з коментарів, які наведено далі.

```

code = list(map(int,input().split())) # Вводимо список чисел
P = 0                                # Позиція у списку: починаємо з 0
while P<len(code)-1:                  # Цикл до кінця списку
    K = P                              # Допоміжна змінна
    # Шукаємо кінець чергової послідовності
    while code[P+1] == code[P]:        # Поки наступне число таке саме,...
        P = P+1                        # переходимо далі
        if P == len(code)-1:           # Якщо раптом список закінчився,...
            break                       # зупиняємо цикл
    # Опрацьовуємо знайдену послідовність
    K = P-K+1                          # Обчислюємо кількість однакових
    print(K, code[P], end = ' ')
    P = P+1                            # Переходимо до наступного числа
if P<len(code):                       # Якщо залишилось одне число,...
    print(1, code[P])                 # то опрацьовуємо його

Зразок вхідних і вихідних даних для перевірки наведено в при-
кладі 1.

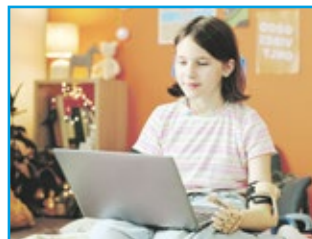
```

Як оцінити, наскільки зменшився обсяг даних після стиснення?

Ступінь стиснення даних (K) — це відношення обсягу стиснених даних (V_a) до обсягу початкових (нестиснених) даних (V_0), яке зазвичай виражають у відсотках $K = (V_a / V_0) \cdot 100\%$. Так, у прикладі 1 ступінь стиснення становить $K = (12 / 22) \cdot 100\% \approx 55\%$.

Ступінь стиснення залежить від типу файлу й методу стиснення. Файли деяких форматів (як-от JPG, MP3) майже не можна стиснути, оскільки вміщені в них дані стискаються під час створення файлу.

Залежно від алгоритму, застосованого під час архівування, розрізняють такі форми архівів: ZIP, 7Z, RAR, GZ та ін.



Види архівів

Окрім **однофайлових** (звичайних) **архівів**, сучасні архіватори дають змогу створювати багатотомні архіви й архіви із саморозархівуванням.

Багатотомний архів — тобто розділений на кілька файлів (томів) — створюють тоді, коли виявляється, що з одним файлом великого обсягу складно працювати, приміром, не можна зберегти його на носіях меншого обсягу, надіслати електронною поштою тощо.

Архів із саморозархівуванням (SFX-архів, від англ. *self extracting* — самовидобування) — це виконуваний файл, який після запуску автоматично розархівовується. Його доцільно використовувати, якщо на комп'ютері отримувача не встановлено відповідного архіватора.

Архіватори



Архіватори — службові програми, за допомогою яких здійснюють операції архівування та розархівування.

Серед сучасних архіваторів можна назвати вільний PeaZip, умовно безплатний WinZIP, онлайнвий B1 Free Archiver. Ознайоммося з першим.

► Архіватор PeaZip

Архіватор PeaZip, логотип якого наведено на рисунку, випускається для ОС Windows, Linux, macOS та ін. Він дає змогу опрацьовувати архіви більш ніж 15 різних форматів, а також різних видів: однофайлові, багатотомні, із саморозархівуванням.



Для **створення архіву** потрібно:

- 1) запустити програму і відкрити папку з файлами для архівування;
- 2) вибрати один або більше файлів;
- 3) натиснути кнопку Додати — відкриється сторінка для налаштування параметрів архіву (папки, формату, типу тощо);
- 4) зробити потрібні налаштування й натиснути кнопку Так.

Після цього в зазначеній папці з'явиться архівний файл (або файли) із вибраною назвою та розширенням, відповідним формату.

Для **видобування файлів з архіву** потрібно:

- 1) запустити програму і в робочому полі вікна відкрити архів;
- 2) вибрати один або більше файлів (декілька файлів вибирають, утримуючи клавішу Ctrl);
- 3) натиснути кнопку Видобути — відкриється сторінка для налаштування параметрів (вибору папки для збереження файлів тощо);
- 4) зробити потрібні налаштування й натиснути кнопку Так — після цього в зазначеній папці з'являться файли, видобуті з архіву.

Окрім операцій архівування та розархівування, сучасні архіватори дають змогу виконувати й інші дії: наприклад, переглядати вміст архіву, додавати файли до архіву, вилучати окремі файли з архіву, перевіряти цілісність архівного файлу тощо.



Питання для самоперевірки

1. Що таке архівування даних; розархівування?
2. У який спосіб можна зменшити обсяг даних, не втративши дані?
3. Поясніть зв'язок між двома рядками чисел із прикладу 1.
4. Що називають ступенем стиснення даних?
5. Який ступінь стиснення має архів обсягом 51,4 KiB, якщо початковий обсяг файлів становить 5,9 MiB?
6. Як передати архів обсягом 50 MiB через службу миттєвих повідомлень, якщо вона дозволяє передавати файли обсягом до 10 MiB?



Вправа 2

- Випробувати й скласти програми, що ілюструють принципи архівування та розархівування даних.
1. Випробуйте програму з прикладу 2.
 2. Змініть програму з прикладу 2 так, щоб дані для архівування вводилися з текстового файлу, а результат виводився в інший текстовий файл (див. § 34 в підручнику для 8 класу).
- ⚙ 3. Складіть програму для розархівування даних із послідовності чисел, виведеної програмою з прикладу 2.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111794



- Debian — популярний вільний багатоцільовий дистрибутив Linux, який став основою для інших дистрибутивів: Ubuntu і Linux Mint — для ПК; SteamOS — для ігрової платформи Steam Box тощо. Rocks Cluster Distribution — дистрибутив Linux для суперкомп'ютерів. OpenWrt — ОС на ядрі Linux для бездротових Wi-Fi маршрутизаторів.





ПРАКТИЧНА РОБОТА 1

АРХІВУВАННЯ ТА РОЗАРХІВУВАННЯ ДАНИХ

Завдання: здійснити за допомогою програми-архіватора стиснення даних в архіви різних типів, виконати розархівування, порівняти ступінь стиснення даних.

Обладнання: комп'ютер із архіватором, папка Archive* із файлами для архівування.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть текстовий процесор. Визначте загальний обсяг файлів папки Archive*, наданої вчителем / вчителькою.
Створіть новий документ і запишіть отримані дані.
2. Створіть однофайловий архів папки. Визначте його обсяг і запишіть у документ.
3. Визначте ступінь стиснення архіву. Додайте результат у документ.
4. Створіть багатотомний архів папки із трьох томів приблизно однакового розміру. Знайдіть загальний обсяг трьох томів архіву й запишіть у документ.
5. Створіть саморозпаковуваний архів папки. Визначте обсяг архіву й запишіть у документ. Видобудьте з архіву папку, змінивши її назву.
6. Додайте в документ таблицю результатів і заповніть її даними, отриманими під час виконання попередніх пунктів.

Вид архіву	Обсяг файлу	Коефіцієнт стиснення, %
Однофайловий		
Багатотомний		
Саморозпаковуваний		

Збережіть файл із назвою Практична робота 1.

Зробіть висновок: як за допомогою архіватора створити однофайловий, багатотомний, саморозпаковуваний архіви; чому архівні файли різних типів мають різний обсяг.

Відтоді як комп'ютер із наукових лабораторій вийшов у широкий світ, пройшло багато років. Сьогодні він став доступним практично кожній людині. Але, на жаль, не всі люди використовують його можливості з добрими намірами. З'ясуємо, які чинники зумовили появу шкідливих програм.

Класифікація загроз

Залежно від мети зловмисника шкідливі програми мають різне призначення, а для їх проникнення на комп'ютер та маскування застосовуються різноманітні технології. Розгляньмо схему на рис. 3.1.



Рис. 3.1

Нині відомі різні типи шкідливих програм. Розгляньмо їх на прикладах. Існують програми, що дають змогу віддалено керувати комп'ютером і завдяки цьому використовувати його без відома власника/власниці зі зловмисною метою (розсилання спаму, проведення атак тощо). Такий комп'ютер іноді називають *зомбі*.



Для майнінгу (від англ. *mining* — видобування) криптовалюти потрібно виконувати багато обчислень. Якщо на комп'ютер потрапила програма-*криптомайнер*, то вона може запускатися й виконувати обчислення (коли на комп'ютері ніхто не працює). У цьому випадку комп'ютер споживатиме багато електроенергії.

2 Людина, яка керує багатьма комп'ютерами, може організувати атаку з метою порушення доступності певного вебресурсу. У ході такої атаки багато комп'ютерів — так званий *ботнет* — одночасно надсилають до атакованої системи (приміром, до сайту установи) багато запитів, які система не встигає опрацьовувати. Корисні запити «губляться» в потоці й можуть залишитися невиконаними.

3 *Програми-вандали* мають на меті пошкодити дані, що зберігаються на комп'ютері, або корисні програми, встановлені на ньому.

4 «*Клавіатурний шпигун*» (англ. *keylogger*, від *key* — клавіша і *logger* — реєструвальний пристрій) — це програма або пристрій, що реєструє всі натиснення клавіш на клавіатурі комп'ютера, зберігаючи зібрані дані у файлі або надсилаючи через мережу. Така програма корисна, якщо потрібно забезпечити посилений контроль за використанням комп'ютера. А зловмисник / зловмисниця завдяки їй може отримати дані, які вводилися з клавіатури.

Антишпигунські програми

Для видалення шпигунської програми із системи її потрібно спочатку виявити. Для цього застосовують так звані антишпигунські програми (англ. *antispyware*). Вони перевіряють різні місця в ОС і застосунках, де можуть «переховуватися» небажані програми.

Багато антивірусних програм мають антишпигунські компоненти.

5 Власницька програма SpyHunter (є обмежена безплатна версія) дає змогу знаходити й видаляти шпигунські та інші програми.

Комп'ютерні віруси

Як вам відомо, **комп'ютерний вірус** — це програма, яка копіює та поширює свій програмний код і може негативно впливати на роботу комп'ютера загалом (див. 5 клас, § 9).

У 1961 році співробітники фірми Bell Telephone Laboratories (США) створили ігрову програму «Дарвін». У пам'яті комп'ютера влаштували «змагання» між програмами-«організмами». Кожен «організм» намагався створювати власні копії й знищувати «організми-конкуренти». Нині їх називають комп'ютерними вірусами.

Програма, до якої приєднався вірус, інфікує інші програми. За деякий час ті перестають працювати або працюють неправильно, а згодом комп'ютер втрачає працездатність, пошкоджуються дані тощо.

Антивірусні програми

Як немає ліків від усіх хвороб, так немає й програмного засобу, який би гарантував виявлення і знищення всіх вірусів. Боротися з вірусами допомагають антивірусні програми, які класифікуються так:

Клас	Опис
Детектори	Виявляють файли, заражені вірусами
Лікарі (фаги)	«Лікують» заражені програми або диски, видаляючи із заражених програм вірусний код
Ревізори	Запам'ятовують стан програм і дисків та порівнюють їх поточний стан із попереднім і повідомляють про виявлені невідповідності
Фільтри	Перехоплюють ті звернення до системи, які віруси використовують для розмноження та заподіяння шкоди

Існує ціла низка популярних антивірусних програм. На рис. 3.2 наведено логотипи деяких із них.



Рис. 3.2

Під час установлення антивірусна програма здебільшого налаштовується на автоматичний запуск і постійно контролює підозрілі дії на комп'ютері. Надалі користувач/користувачка може самостійно вимикати та вмикати такий режим роботи програми. Примусово перевірити певний файл (папку) на віруси можна командою контекстного меню.

Антивірусна програма українського виробництва Zillya! захищає комп'ютер, на який її встановлено, від вірусів, троянських,

шпигунських та інших шкідливих програм. Його якість підтверджено сертифікатом Державної служби спеціального зв'язку та захисту інформації України.

Антивірус Zillya! (рис. 3.3) створила українська компанія «Лабораторія Zillya!». Вона розробляла антивірусну програму з 2009 року для ОС Windows, а з 2016 — і для ОС Android. Компанія також займається розробкою інноваційних технологій у сфері кіберзахисту.



Рис. 3.3

Запобігання зараженню вірусами

Ніколи не можна мати повну впевненість, що всі віруси на комп'ютері знешкоджено. Вірус спочатку створюється, розповсюджується і лише через деякий час для його знешкодження розробляється антивірусна програма. Крім того, багато вірусів мають «інкубаційний період» — спочатку непомітно поширюються й лише через деякий час дають про себе знати.

Ось кілька порад, щоб запобігти зараженню через носії даних.

- Остерігайтеся використовувати власні носії на чужих комп'ютерах.
- Після «контакту» з чужим комп'ютером перевірте свій носій.
- Уникайте неліцензованих програм, які часто є носіями вірусів.
- Не вимикайте захисні програми: антивірус і брандмауер.
- Періодично робіть антивірусну перевірку комп'ютера.
- Періодично оновлюйте версії ОС, антивірусу й вірусної бази.

Як бачимо, механізм передавання «інфекції» у комп'ютерів подібний до розповсюдження епідемії серед людей. Тому краще запобігати зараженню, ніж лікувати. Існують сайти, на яких можна перевірити підозрілий файл одночасно за допомогою кількох різних антивірусів.



Питання для самоперевірки

1. Які є способи проникнення шкідливих програм на комп'ютер?
2. Опишіть принцип роботи «клавіатурного шпигуна».
3. Як працюють антишпигунські програми?
4. Чим комп'ютерні віруси відрізняються від інших програм?
5. Як перевірити файл, якщо на комп'ютері немає антивірусу, але є доступ до інтернету?
6. Що слід робити, щоб уникнути зараження комп'ютера вірусом?



Вправа 3

■ Навчитися працювати з антивірусною програмою.

1. Запустіть текстовий процесор і створіть новий документ.
2. Визначте, яка антивірусна програма встановлена на вашому комп'ютері, введіть її назву в документ.
3. Визначте дату останнього оновлення антивірусної програми й запишіть у документ.
4. Визначте дату останньої перевірки комп'ютера антивірусною програмою й запишіть у документ.
- ⚙ 5. За допомогою контекстного меню або вікна антивірусної програми перевірте на віруси один із файлів (папок). Зробіть скриншот і вставте його в документ.
6. Збережіть файл із назвою Вправа 3 у відповідній папці.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111795



§ 4

ЗАХИСТ КОМП'ЮТЕРНИХ СИСТЕМ. БЕЗПЕКА В ІНТЕРНЕТІ

Із розвитком інформаційних технологій і їх проникненням в усі сфери людської діяльності зростає потреба в захисті інформації. Адже від надійності роботи комп'ютерної системи може залежати результат багатьох дій. Надійність залежить від низки чинників.

Загрози безпеці даних

Загрози безпеці даних в інформаційних системах можна класифікувати за чинниками, що їх спричиняють. Розгляньмо схему (рис. 4.1).

Основною причиною більшості загроз безпеці є людський чинник.



Рис. 4.1

Етичні й правові основи інформаційної безпеки

Нині в суспільстві інформаційна безпека відіграє провідну роль.



Інформаційна безпека — це захищеність даних та інформаційної системи від випадкового або навмисного пошкодження та несанкціонованого доступу.

Засоби й методи підтримки інформаційної безпеки мають різне призначення. Ознайомитися докладніше з їх класифікацією та призначенням можна за QR-кодом або посиланням rnk.com.ua/114524.



Яких правил інформаційної безпеки слід дотримуватися?

- Не використовуйте неліцензійне програмне забезпечення.
- Посилайтеся на використаний ресурс.
- Створюйте надійні паролі при реєстрації облікових записів.

- Не розголошуйте приватну інформацію стороннім особам.
- Використовуйте антивірусні програми.
- Обмежуйте доступ до комп'ютера сторонніх осіб.
- Будьте ввічливими під час інтернет-спілкування.
- Не надсилайте СМС-повідомлення для отримання «подарунків».

В Україні прийнято низку законів та постанов щодо забезпечення інформаційної безпеки: «Про захист інформації в інформаційно-телекомунікаційних системах», «Про державну таємницю», «Про захист персональних даних», «Про авторське право та суміжні права» та ін. Незаконне втручання в роботу комп'ютерів і мереж, розповсюдження вірусів тягне за собою кримінальну відповідальність (ст. 361 Кримінального кодексу України).

Загрози під час роботи в інтернеті

Розгляньмо типи шкідливих програм і загрози, які вони становлять.

Загроза	Опис
Хробаки	Програми, які самостійно поширюються мережею, не інфікуючи інші файли
Троянські програми	Програми, що поширюються під виглядом нешкідливих програм та виконують несанкціоновані дії: викрадають інформацію (паролі, рахунки тощо), передають її злочинцям через інтернет, самостійно відкривають сайти для зміни рейтингів, хакерських атак тощо
Скрипт-віруси	Програми, що потрапляють на комп'ютер через електронну пошту, маскуючись під документи
Дропери	Виконувані файли, які самі не є вірусами, але призначені для встановлення шкідливих програм
Боти	Програми, які дають змогу зловмиснику таємно керувати вашим комп'ютером
Шпигунські й рекламні програми	Програми, що зазвичай устанавлюються на комп'ютер разом із неліцензованими програмами й збирають конфіденційну інформацію або демонструють нав'язливу рекламу
Фішинг	Вид інтернет-шахрайства: виманювання конфіденційної інформації через підробні сайти, що схожі на сайти відомих банків, інтернет-магазинів тощо, або за допомогою спаму

Брандмауери

Для запобігання інтернет-загрозам між комп'ютером і мережею встановлюють перешкоду — *міжмережевий екран*, або брандмауер (від нім. *Brandmauer*), файрвол (від англ. *Firewall* — «вогняна стіна»).

 **Брандмауер** — це технічний пристрій або програмний засіб для контролю даних, що надходять до комп'ютера через мережу.

Брандмауери допомагають запобігти проникненню на комп'ютер осіб, що мають злий намір, або шкідливих програм, але не запобігають витоку персональної інформації та завантаженню вірусів.

 ОС Windows 10 має вбудований брандмауер. Щоб увімкнути й налаштувати його, потрібно вибрати команду:

Пуск →  → Оновлення та захист → Безпека у Windows → Брандмауер і захист мережі.

Засоби браузера для гарантування безпеки

Вибираючи браузер для роботи, крім низки різних характеристик, зазвичай оцінюють його засоби для гарантування безпеки. Сучасні браузери містять вбудовані засоби захисту від низки інтернет-загроз.

 Існує ціла низка популярних нині браузерів (рис. 4.2).



Рис. 4.2

На багатьох комп'ютерах, телефонах і планшетах встановлюють браузер Google Chrome, який попереджає про відкриття сайту із загрозою фішингу або шкідливих програм; ізолювано відкриває вебсторінки, що в разі загрози призводить до закриття лише однієї шкідливої сторінки; дає змогу скасувати збереження конфіденційних даних; створює можливість керувати показом спливних вікон.

Можна налаштувати й інші дозволи або обмеження для інформаційного наповнення вебсторінок (анімація, малюнки тощо) або посилок на сайти (місцезнаходження тощо).

Захист даних може забезпечуватися *електронним цифровим підписом*, який ідентифікує автора електронного документа. Це особливий код, цифровий аналог звичайного підпису, яким можна скріпити електронні документи, що гарантуватиме їхню цілісність і секретність. Для створення такого підпису потрібен підтверджувальний сертифікат.

Захищені сайти

Для забезпечення інформаційної безпеки недостатньо захистити дані на комп'ютері-клієнті або сервері — особа зі злими намірами може перехопити дані під час обміну по каналах зв'язку. Захист даних під час передавання забезпечує протокол захищеного зв'язку https.

 **Захищений сайт** — це сайт, який використовує для обміну даними протоколи захищеного зв'язку.

Підтримка захищених каналів зв'язку потрібна сайтам, які працюють із персональною інформацією користувачів/користувачок. Це можуть бути урядові сайти, сайти банків, інтернет-магазинів. Адреса захищеного сайту починається з позначки одного з протоколів захищеного зв'язку, наприклад: <https://mon.gov.ua/> — сайт Міністерства освіти і науки України; <https://uk.wikipedia.org/wiki/Атом> — перша сторінка, створена в Українській вікіпедії².

Захист від спаму

Існують також непрямі загрози, скажімо, **спам** — це небажані повідомлення переважно рекламного характеру. Він може містити посилання на небезпечні сайти, привабливі пропозиції переказати гроші на певні рахунки тощо.

Спамери збирають адреси потенційних «клієнтів» спеціальними програмами-роботами. Найнадійніший спосіб захисту — не дати роботам дізнатися вашу електронну адресу.



² Атом // Вікіпедія: Вільна енциклопедія. URL: <https://uk.wikipedia.org/wiki/Атом> (дата звернення 19.04.2026).

Щоб захиститися від спаму, не публікуйте свою електронну адресу й номер телефону на сайтах. Або ж закодуйте їх, або подайте у вигляді картинки, щоб розпізнати їх могла лише людина.

Програмні засоби боротьби зі спамом (**спам-фільтри**) можуть бути складовими антивірусних програм або послугою поштових серверів (дозволяють налаштовувати власні фільтри, правила опрацювання поштових надходжень на основі набору символів з адреси відправника).



Питання для самоперевірки

1. Які загрози існують під час роботи в інтернеті?
2. Яке призначення брандмауерів?
3. Які засоби захисту від інтернет-загроз мають браузері?
4. Який сайт називають захищеним?
5. Що таке фішинг; спам?
6. Назвіть відомі вам способи захисту від спаму.



Вправа 4

- Навчитися працювати з брандмауером та із захищеними сайтами.
- 1. Запустіть текстовий процесор і створіть новий документ.
- 2. Визначте, чи увімкнено брандмауер Windows на вашому комп'ютері. Зробіть скриншот вікна налаштування брандмауера й збережіть.
- 3. Запустіть браузер і відкрийте один із захищених сайтів. Зробіть скриншот вікна й збережіть у документі.
- 4. Запишіть у документ пояснення, чому, на вашу думку, знайдений сайт використовує захищений канал зв'язку.
- 5. Відкрийте поштову скриньку й позначте один із листів як спам.
- 6. Відкрийте папку Спам, зробіть скриншот вікна зі вмістом папки та збережіть у документі.
- ⚙ 7. Відновіть переміщений до папки Спам лист як Не спам і перевірте його появу в початковій папці.
- 8. Збережіть документ у файл із назвою Вправа 4.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111796



Надійність сучасних засобів зберігання інформації полягає в надзвичайно ефективних способах передавання й копіювання даних, що робить розповсюджену інформацію практично незнищеною. Сьогодні тривають роботи щодо збільшення довговічності й самих носіїв даних.

Учені Інституту проблем реєстрації інформації Національної академії наук України 2013 року розробили технологію зберігання інформації на оптичних сапфірових дисках, які здатні зберігати дані впродовж десятків тисяч років. На цю роботу їх надихнули таблички шумерів, які зберігають інформацію вже понад 4 тисячі років.



Безпечне зберігання даних на комп'ютері

Серед заходів безпеки перше місце належить особистій організованості людини та її відповідальному ставленню до інформації на ПК.

► Захист конфіденційних даних

Якщо комп'ютером користуються кілька осіб, то для захисту конфіденційних даних необхідно зробити власний обліковий запис, установити пароль і зберігати файли в папках власної бібліотеки. Можна заархівувати її із задаванням пароля. Це також доцільно використовувати й під час листування.

► Захист даних від пошкодження

Для збереження інформації та її захисту від пошкодження варто розділити жорсткий диск на кілька логічних розділів.

Зазвичай жорсткий диск ділять на диски (розділи) C, D і т. д. (рис. 5.1). На диску C установлюють ОС і програмне забезпечення, на решті — зберігають інші дані. Це сприяє збереженню даних у разі пошкодження системного диска, скажімо, вірусом.



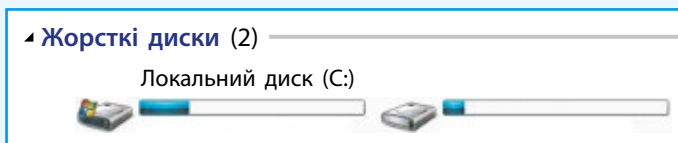


Рис. 5.1

Ось кілька порад для запобігання втраті важливих даних.

- Уникайте зберігання таких даних на системному диску, зокрема на робочому столі, у папках власної Бібліотеки, бо системний диск найчастіше зазнає впливу шкідливих програм.
- Копіюйте дані на зовнішні накопичувачі: зовнішні жорсткі диски, флешносії, хмарні диски, оптичні диски тощо.

! Після збереження даних на флешносії слід дотримуватися безпечного видалення пристрою, щоб не втратити дані, які не встигли скопіюватися на носій із буфера запису.

- Для резервного копіювання файлів і можливості відновлення ОС користуйтеся її утилітами.

З описом реалізації резервного копіювання файлів ознайомтеся за QR-кодом або посиланням rnk.com.ua/114542.



Точка відновлення Windows

Спостереження й запам'ятовування стану ОС і встановлених програм може відбуватися й у автоматичному режимі завдяки створенню так званих «точок відновлення».

Точка відновлення Windows (контрольна точка відновлення) — це збережена інформація про стан операційної системи й установлених програм у певний момент.

Точку відновлення доцільно створювати тоді, коли комп'ютер і програми працюють нормально. ОС створює точки відновлення системного диска автоматично після встановлення програм, зміни налаштувань.

Точку відновлення можна створити в ручному режимі:

Пуск → Панель керування → Система і безпека → Система → Захист системи → Далі → Створити (рис. 5.2) → надати опис точки відновлення → Створити.

Для відновлення видалених файлів можна скористатися функцією контекстного меню диска Відновити попередню версію диска, для якого ввімкнено захист системи. Далі потрібно вибрати: Налаштувати → Відновити параметри системи та попередні версії файлів (рис. 5.3).

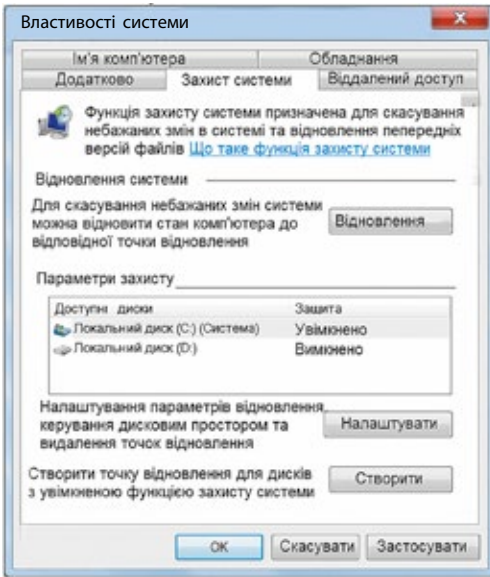


Рис. 5.2

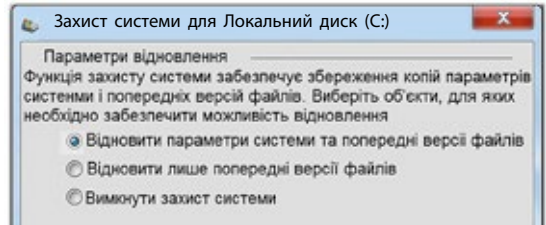


Рис. 5.3



В разі погіршення роботи комп'ютера доцільно повернути його стан до певної точки відновлення. Для цього слід виконати такі дії: Пуск → Всі програми → Стандартні → Службові → Відновлення системи → Далі → вибрати точку відновлення → Далі → Готово.

2023 року американський IT-відеоблогер, інженер Джефф Гірлінг здійснив жартівливий експеримент. Він провів змагання зі швидкості передавання даних між інтернетом і... поштовим голубом. З'ясувалося, що на відстанях до 1000 км голуб, який переносить три SD-карти обсягом по 3 ТБ кожна, забезпечує більшу швидкість передавання, ніж оптичне волокно. Переваги можливі там, де інтернет може бути повільним (нестабільним), наприклад, у віддалених місцевостях.





В ОС Linux Mint для збереження інформації про стан системи та подальшого відновлення стану призначена утиліта Timeshift (рис. 5.4). Послідовність дій для її використання подібна до описаної раніше роботи з точками відновлення ОС Windows.

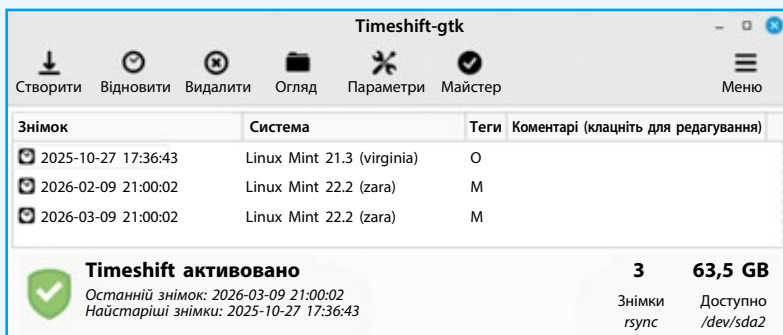


Рис. 5.4

Надійне видалення даних

Іноді конфіденційна інформація може потрапити до сторонніх осіб без зловмисних намірів і зусиль із їхнього боку.



Перед продажем ноутбука з нього видалили папки з особистими документами й фотографіями. Проте під час видалення файлу засобами ОС записані в ньому дані залишаються на диску і їх можна відновити за допомогою спеціальних програм.

Уникнути проблеми, описаної у прикладі 2, допоможе застосування програми для надійного видалення даних. Відповідні засоби має, наприклад, розглянутий раніше архіватор PeaZip (рис. 5.5).

- ✗ Швидке видалення
 - ✗ Видалення з затиранням нулями (Alt + Del)
 - ✗ Надійне видалення файлів (Ctrl + Del)
- Затирання нулями вільного простору
 Надійне очищення вільного простору



Рис. 5.5

Отже, потрібно пам'ятати деякі правила видалення даних.

- Швидке видалення не відрізняється від видалення засобами ОС і не зачіпає записаних у файлі даних.
- Видалення із затиранням нулями передбачає перезапис місця на носії, зайнятого файлом, послідовністю нулів.
- Під час надійного видалення файлу на зайняте ним місце кілька разів поспіль записуються випадкові дані.



Питання для самоперевірки

1. Чим зумовлена надійність сучасних засобів зберігання даних?
2. Які є способи безпечного зберігання даних на комп'ютері.
3. Як виконати резервне копіювання даних системного диска?
4. Як відновити дані з резервних копій?
5. Що таке точка відновлення Windows?
6. Коли варто застосовувати надійне видалення даних із носія?



Вправа 5

■ Виконати архівне копіювання й відновлення даних.

1. Запустіть текстовий процесор і створіть новий документ.
2. Засобами ОС Windows зробіть резервну копію папки за вказівкою вчителя/учительки та збережіть на диску D.
3. Відновіть зі створеної резервної копії файл за вказівкою вчителя/учительки й збережіть його в папці Документи.
4. Запишіть у документ назви логічних дисків, на які розбито жорсткий диск вашого комп'ютера.
5.  Визначте, чи ввімкнено захист системи для дисків. Зробіть скриншот з інформацією про це й збережіть у документі.
6. Збережіть файл із назвою Вправа 5.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111797





ПРАКТИЧНА РОБОТА 2

НАЛАШТУВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ В СЕРЕДОВИЩІ БРАУЗЕРА

Завдання: налаштувати параметри браузера та поштової скриньки, що стосуються безпеки листування.

Обладнання: комп'ютер, з'єднаний з інтернетом, браузер Google Chrome.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть браузер Google Chrome.
2. Налаштуйте конфіденційність користування браузером. Для цього:
 - 1) клацніть на значку у верхньому правому кутку браузера;
 - 2) виберіть команду Налаштування;
 - 3) унизу сторінки виберіть команду Показати розширені налаштування;
 - 4) у розділі Конфіденційність зробіть потрібні налаштування.
3. Перевірте налаштування заборони показу спливних вікон: Меню → Налаштування → Показати розширені налаштування → Конфіденційність → Налаштування вмісту → Спливні вікна (Заборонити).
4. Налаштуйте (якщо не налаштовано) активацію захисту від фішингу та шкідливих програм (див. алгоритм п. 2).
5. Увімкніть, якщо не ввімкнено, у браузері українську мову інтерфейсу (див. попередні алгоритми налаштувань).
6. Самостійно дослідіть налаштування вмісту сторінки браузера.
7. Зайдіть на свою поштову скриньку.
8. Додайте нову папку Шкільне для зберігання листів шкільного призначення.
9. Налаштуйте автоматичний підпис ваших листів, наприклад: З повагою, Ім'я Прізвище.
10. Позначте листи, отримані з невідомих джерел, як спам.
11. З'ясуйте в інтернеті, що таке файли cookies і навіщо їх видаляють.
12. Створіть документ і надішліть відповідь учителю / вчительці.

Зробіть висновок: які налаштування безпеки браузера Google Chrome установлені стандартно, а які потрібно налаштовувати додатково.



ТРИВИМІРНА ГРАФІКА

2

РОЗДІЛ

У цьому розділі ви дізнаєтеся, що таке тривимірна графіка та які програми призначені для роботи з нею. За допомогою тривимірного редактора навчитесь моделювати об'єкти, створювати анімацію тощо.

§6

ОСНОВНІ ПОНЯТТЯ ТРИВИМІРНОЇ ГРАФІКИ

Сучасні комп'ютери дають змогу моделювати просторові об'єкти. Це відкрило нові можливості для наукового та технічного моделювання, створення рекламних відео, спецефектів, комп'ютерних ігор тощо.

Що таке тривимірна графіка

Ви вже створювали векторні й растрові зображення, що є двовимірними, тобто плоскими. А як розробляються об'ємні зображення?



Тривимірна графіка (3D-графіка) — розділ комп'ютерної графіки, в якому розглядають методи створення зображень або відео шляхом моделювання об'ємних об'єктів у тривимірному просторі.

Існує низка програм для роботи з тривимірною графікою, що відрізняються призначенням та функціями. Розгляньмо таблицю:

Програма та її призначення	Опис
Художнє моделювання    3ds Max Blender AUTODESK Maya	Розробка реалістичних та фантастичних зображень і відеоматеріалів; дають змогу довільно змінювати форму об'єктів, добирати матеріали, змінювати їхні властивості (колір, прозорість тощо), керувати рухами об'єктів і їхніх частин
Технічне моделювання   AutoCAD FreeCAD	Розробка деталей і пристроїв із точним дотриманням розмірів, що є принциповим під час підготовки конструкторської документації
Спеціалізовані програми   Sweet Home 3D RenderGold	Sweet Home 3D — моделювання розміщення меблів, дверей, вікон; RenderGold — підготовка декоративних елементів: малюнки для кнопок на сайті, рамки для світлин, візерунки для тла листівки

Щоб закодувати інформацію про форму просторового об'єкта (рис. 6.1), його поверхню потрібно уявно розбити на плоскі грані-многокутники (1) та зберегти дані про ці многокутники: координати вершин (3), ребра (2), колір тощо. Це дає змогу обертати зображення на екрані, розглядати його з різних боків, моделювати освітлення тощо.

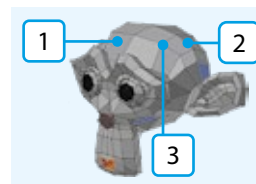


Рис. 6.1

Процес **створення тривимірного зображення** включає такі етапи:

- 1) моделювання, під час якого створюють усі об'єкти, уточнюють їхню форму та взаємне розміщення;
- 2) надання об'єктам кольору та текстур, налаштування освітлення;
- 3) налаштування анімації в разі розробки відеоматеріалу;
- 4) вимальовування (рендеринг) зображення чи відеоматеріалу, тобто формування кінцевого зображення на основі моделі з урахуванням усіх властивостей матеріалів і джерел світла (рис. 6.2).



Рис. 6.2

Застосування тривимірних моделей

Отже, тривимірне моделювання дало змогу змоделювати та змусити рухатися на екрані різні об'єкти. Цю технологію особливо вподобали художники, дизайнери, конструктори, розробники мультфільмів та інші фахівці.



Тривимірне моделювання — процес створення комп'ютерної моделі об'єкта, яка є його візуальним об'ємним образом.

Тривимірна модель — візуальний об'ємний образ певного об'єкта.

Розгляньмо кілька прикладів застосування тривимірних моделей.



Перший український 3D-анімаційний фільм «Микита Кожум'яка» вийшов 2016 року. Фільм режисера Манука Депояна знято за мотивами однойменної казки письменника Антона Сіяніки. Над його виробництвом протягом двох років працювало кількасот людей. Пізніше з'явилися фільми «Викрадена принцеса: Руслан і Людмила» (2018), «Клара та чарівний дракон» (2019), а згодом — український 3D-анімаційний фентезійний фільм «Мавка. Лісова пісня», створений 2023 року за мотивами драми-феєрії Лесі Українки.

Технологія доповненої реальності полягає в доданні штучних об'єктів до зображення на екрані, яке надає камера. Завдяки їй, навівши камеру на сторінку, скажімо, підручника біології, на екрані можна дослідити інтерактивну тривимірну модель кровоносної системи людини.

Тривимірне моделювання та анімація стали потужними помічниками творців кіно. Найчастіше спеціальні відеоефекти застосовують для створення приголомшливих епізодів. Разом із тим вони корисні й для моделювання цілком буденних речей, наприклад, умеблювання.



Над спецефектами для фільму канадського режисера Джеймса Кемерона «Титанік» (1997 рік) працювало 17 студій. У побудованій моделі корабля не було носової частини, тому в епізодах використано її тривимірну модель. У багатьох кадрах ми бачимо змодельовану воду й меблі; під час зіткнення у воду «падають» тривимірні моделі шматків льоду. У 2012 році до 100-річчя загибелі лайнера вийшла у світовий прокат 3D-версія фільму.

Тривимірний друк

Застосування тривимірних моделей не обмежується «віртуальним світом». Протягом кількох десятиліть розвиваються адитивні технології (від англ. *add* — додавати), які полягають у створенні об'єктів поступовим додаванням матеріалу під керуванням комп'ютера.

Пристрій, який за тривимірною моделлю виготовляє з певного матеріалу реальний об'єкт, називають тривимірним принтером (рис. 6.3), а сам процес — тривимірним друком (рис. 6.4).



Рис. 6.3

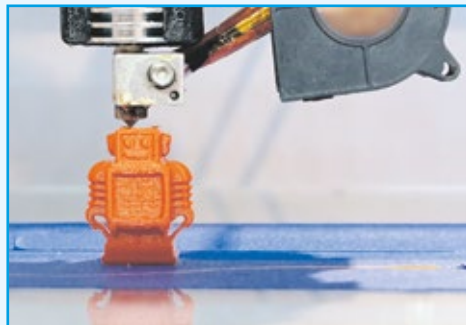


Рис. 6.4

Із поширенням комп'ютерів адитивні технології надзвичайно розвинулися. Нині можливий тривимірний друк із найрізноманітніших матеріалів: різних сортів пластику, металів (зокрема й дорогоцінних), шоколаду та ін. За допомогою будівельних принтерів маємо можливість зі спеціальної суміші «друкувати» цілі будинки.

Одним із перших технологію тривимірного друку описав 1945 року американський винахідник і письменник-фантаст Маррі Лайнстер у своєму оповіданні «Речі минають» (англ. *Things Pass By*). Він сформулював концепцію «конструктора», який нині називають 3D-принтером.

Нині 3D-друк став доступним у багатьох галузях. Так, улітку 2021 року в Амстердамі встановлено перший у світі сталевий міст, виготовлений методом тривимірного друку (рис. 6.5; світлина з вікісховища — [w.wiki/4Lwr](https://www.wiki/4Lwr)). Мосту дали назву Стофбруг (нідерл. *Stoofbrug* — тушкований міст).



Рис. 6.5

Попри спільну назву, для тривимірного друку з різних матеріалів застосовують різні фізичні та хімічні процеси. Так, набули поширення тривимірні принтери, в яких поступово розплавляється тонкий пластиковий шнур і шар за шаром формується предмет. В інших випадках застосовують порошкоподібний або рідкий матеріал, який швидко полімеризується внаслідок опромінення лазером тощо.

Програма Sweet Home 3D

Sweet Home 3D (рис. 6.6) — безкоштовна програма, поширювана за ліцензією GNU GPL, призначена для моделювання інтер'єрів. Її, а також файл українського перекладу, можна завантажити з офіційного сайту. Тут описано версію 6.6. Після запуску програми на екрані з'являється головне вікно (рис. 6.7).



Рис. 6.6

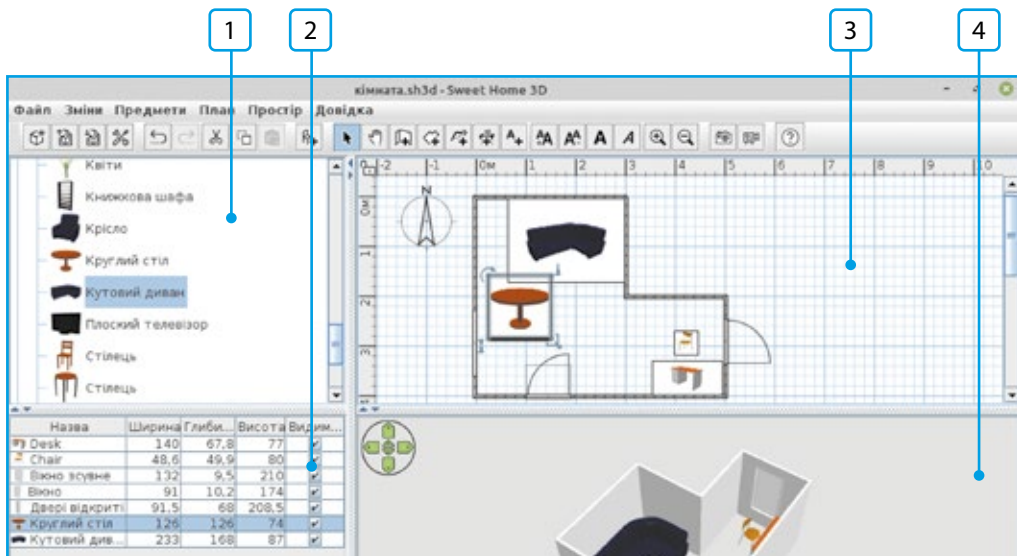


Рис. 6.7

Тут: 1 — каталог предметів; 3 — перелік вибраних предметів;
2 — план; 4 — просторовий вигляд.

Щоб змінити розміри ділянок вікна, слід перетягти межі між ними. Предмети, доступні в каталозі предметів (1), призначені для побудови плану: потрібний предмет слід просто перетягти на план (2). Водночас у переліку вибраних предметів (3) з'явиться опис доданого предмета, а на просторовому вигляді (4) — зображення.

Для навігації в межах плану слугують смуги прокручування та режим перетягування плану — його вмикають кнопкою  на панелі інструментів. Щоб активувати режим вибору об'єктів, слід клацнути кнопку , щоб змінити масштаб — кнопки .

Для здійснення навігації у вікні тривимірного вигляду можна:

- 1) обернути сцену відносно двох осей, пересуваючи мишу, утримуючи натиснутою ліву кнопку;
- 2) змінити масштаб зображення коліщатком миші.



Рис. 6.8

Для цього також використовують навігаційні стрілки (рис. 6.8), розташовані в лівому верхньому куті просторового вигляду.

Властивості предметів, доданих до плану, можна змінити перетягуванням кутів прямокутника, що зображає предмет (рис. 6.9).

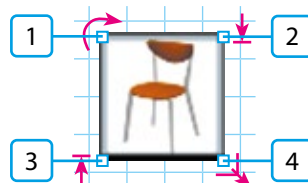


Рис. 6.9

Щоб зберегти розроблену модель, потрібно вибрати команду меню Файл → Зберегти як.... Вікно збереження файлу стандартне. Результатом буде файл із розширенням .sh3d.



Створимо модель житлової кімнати розмірами $3 \times 4,5$ м.

- 1) Вибираємо інструмент Створення стін і будуємо за його допомогою на плані прямокутник, контролюючи розміри за підказками, що з'являються біля вказівника. Побудову останньої сторони прямокутника завершуємо подвійним клацанням.
- 2) У каталозі предметів розгортаємо розділ Двері і вікна. Знаходимо у списку Двері вхідні і перетягуємо цей предмет на зображення на плані тієї стіни, де мають бути двері. Після відпускання кнопки на ділянках плану (рис. 6.10) і просторового вигляду з'являться зображення дверей.
- 3) У каталозі предметів розгортаємо розділ Житлова кімната. Послідовно додаємо, перетягуючи на план, стіл, диван, інші меблі. Одночасно з цим на ділянці просторового вигляду переглядаємо результат (рис. 6.11).

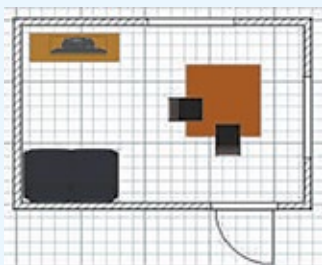


Рис. 6.10



Рис. 6.11



Питання для самоперевірки

1. Порівняйте особливості тривимірної, растрової, векторної графіки.
2. Як кодують інформацію про форму просторового об'єкта?
3. Які бувають програми для тривимірної графіки за призначенням?
4. Що таке «адитивні технології»?
5. Як працює тривимірний принтер?
6. Опишіть послідовність моделювання кімнати в Sweet Home 3D.



Вправа 6

- Дослідити особливості програми для моделювання інтер'єрів, змоделювати інтер'єр навчального кабінету.

1. Програма Sweet Home 3D дозволяє імпортувати нові предмети з файлів у форматах OBJ, DAE/Collada, KMZ або ZDS.

Створіть у текстовому процесорі документ і, скориставшись інтернетом, складіть таблицю з переліком програм, якими можна створювати такі файли.

Збережіть документ у файлі з назвою Вправа 6_1.

2. Користуючись довідковою системою програми Sweet Home 3D (клавіша F1), з'ясуйте:

- як змінити колір предмета;
- як додати на стіну малюнок;
- що відбудеться, якщо перетягти новий предмет на місце, зайняте іншим предметом;
- як моделюють двоповерховий будинок.



3. Змоделюйте кабінет інформатики. Меблі та обладнання доберіть із доступних у програмі.

4. Змініть розміри та колір окремих предметів. Збережіть модель у файлі з назвою Вправа 6_2.

Додайте в документ скриншот вікна програми з моделлю.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111798



«Казка про солом'яного бичка» — перший український анімаційний фільм (режисер, сценарист, художник-аніматор В. Левандовський, 1927 рік). Замість малюнків — плоскі маріонетки. Нині поєднанням плоских маріонеток і 3D-графіки створюються гібридні стилі анімації.

Інтерфейс програми Blender

Для вивчення прийомів роботи з тривимірною графікою розглянемо Blender (рис. 7.1) — потужний редактор художньої тривимірної графіки, який об'єднує засоби для побудови складних композицій, підготовки анімацій та відеоефектів і розробки комп'ютерних ігор.



Рис. 7.1

Blender поширюється за ліцензією GNU GPL. Тут описано версію 2.76b, звантажити яку можна з офіційного сайту програми.

Нідерландський розробник ПЗ Тон Розендал наприкінці 1990-х років розпочав розробку тривимірного редактора Blender. У 2002 році він створив і очолив некомерційну організацію Blender Foundation, яка оголосила в інтернеті збір коштів і викупила Blender у компанії-власниці. Код Blender було опубліковано під вільною ліцензією й спільнота програмістів змогла продовжити розробку програми.

Після запуску програми в центрі головного вікна виводиться **вікно вітання**. У ньому, крім малюнка, є гіперпосилання на сайти з описом Blender та перелік останніх редагованих файлів. Якщо клацнути ім'я файлу, він відкриється для редагування.

Кнопка Відновити попередній сеанс (Recover Last Session) корисна у випадку аварійного завершення попереднього сеансу роботи з програмою. Щоб вікно вітання зникло, слід клацнути за межами гіперпосилань.

Вікно складається з менших вікон (робочих просторів) (рис. 7.2). Найбільшим є вікно 3D-вигляду, в якому видно сцену з кубом, лампою та камерою і панель інструментів. Угорі є вікно інформації (заголовок із головним меню), вікно дерева елементів сцени, вікно властивостей об'єктів і сцени, вікно керування ключами анімації (шкала часу).

Особливість інтерфейсу полягає в тому, що тип, кількість та розміщення вікон можна легко змінити.

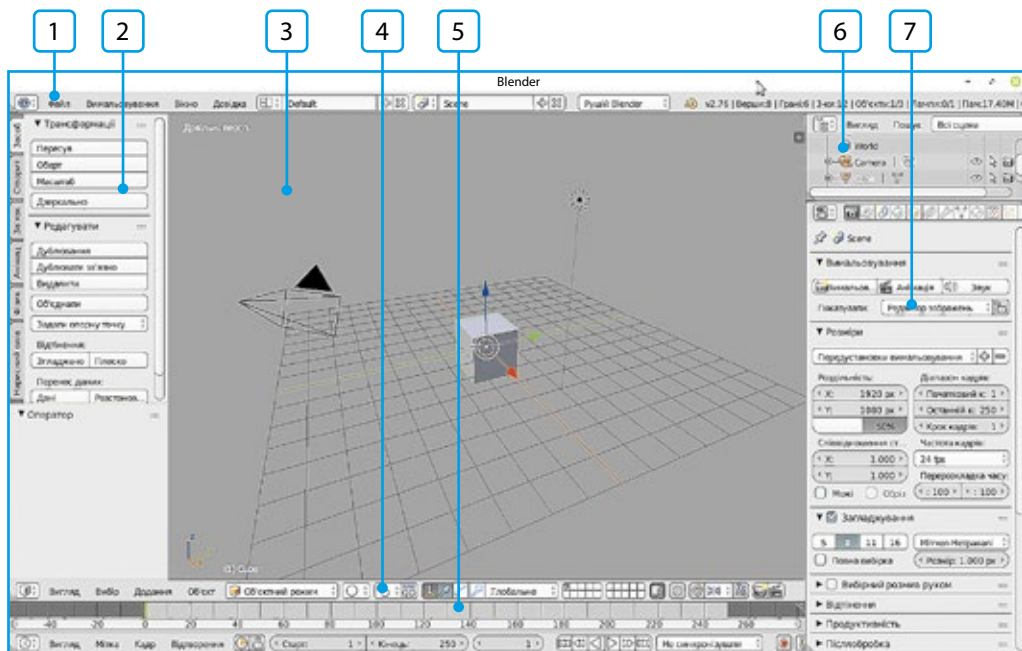


Рис. 7.2

Тут: 1 — вікно інформації; 2 — панель інструментів; 3 — вікно 3D-вигляду; 4 — заголовок вікна 3D-вигляду; 5 — вікно шкали часу; 6 — вікно дерева елементів; 7 — вікно властивостей.

Розгляньмо засоби керування вікнами (рис. 7.3). У лівому нижньому та правому верхньому кутах вікна є позначки у вигляді заштрихованих кутів (1) (наведений на них вказівник набуває вигляду білого хрестика). Як з ними працювати?

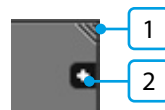


Рис. 7.3

- Якщо одну з позначок перетягти вздовж межі вікна, то воно розділиться на два вікна.
- Якщо позначку перетягти в бік сусіднього вікна такого самого розміру, воно зникне, а розмір поточного вікна відповідно збільшиться.
- Якщо клацнути маркер (2), то відкриється додаткова панель. Щоб сховати цю панель, слід перетягти її ліву межу до краю вікна.

Щоб змінити тип вікна, достатньо клацнути кнопку на початку його заголовка, а потім зі списку, що з'явиться, вибрати тип вікна:

 3D-вигляд,  Властивості,  Огляд файлів,  Шкала часу тощо. Після цього вигляд вікна і піктограма на кнопці зміняться.

Докладніше про типи вікон та їхнє призначення можна дізнатися за QR-кодом або посиланням rpk.com.ua/111814.

Звернімо увагу, що на різних етапах роботи зручно користуватися різними наборами (розкладками) вікон. Blender дає змогу зберігати такі набори, а потім — швидко переходити від одного до іншого. Для цього на інформаційній панелі є список із назвами розкладок (рис. 7.4).

- Тут: 1 — розкрити список;
 2 — поточне значення;
 3 — додати до списку;
 4 — вилучити зі списку;
 5 — поле пошуку.

Достатньо розкрити список, клацнути назву розкладки, і вигляд вікна зміниться. Основною вважатимемо розкладку Default (з англ. *типова*).

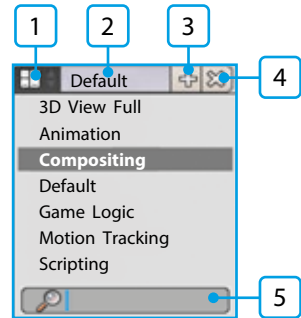


Рис. 7.4

Основні прийоми просторової навігації

Після запуску програми у вікні 3D-вигляду (рис. 7.5) видно координатну сітку, камеру (1), куб (2) і лампу (3).

Центр куба збігається з початком координат. Куб вибрано, тому від його *опорної точки* виходять три кольорові стрілки, які показують напрямки координатних осей.

Масштаб зображення у вікні можна змінити, обертаючи коліщатко миші.

Щоб **вибрати (виділити) об'єкт**, достатньо клацнути його правою кнопкою миші (далі — ПКМ), а щоб **вибрати кілька** — утримувати при цьому клавішу Shift.

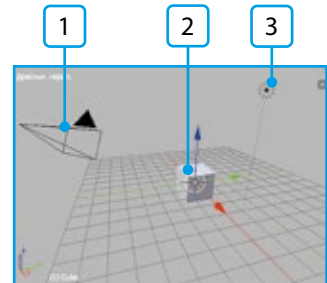


Рис. 7.5

Щоб побачити в центрі вікна вибрані об'єкти, потрібно натиснути клавішу з крапкою на додатковому полі клавіш.

Щоб змінити точку огляду сцени, потрібно пересунути мишу з натиснутим коліщатком (середньою кнопкою миші, далі — СКМ). У лівому нижньому куті вікна 3D-навігації є схематичне зображення координатних осей. При зміні точки огляду воно обертається, вказуючи напрями глобальних осей координат. Якщо ж перед цим натиснути й утримувати клавішу Shift, то замість обертання сцени вона пересунеться без змінення напрямків осей.

Клавіші додаткового поля (позначатимемо їх префіксом NUM) дають змогу змінювати точку огляду сцени й вид проєкції (рис. 7.6).

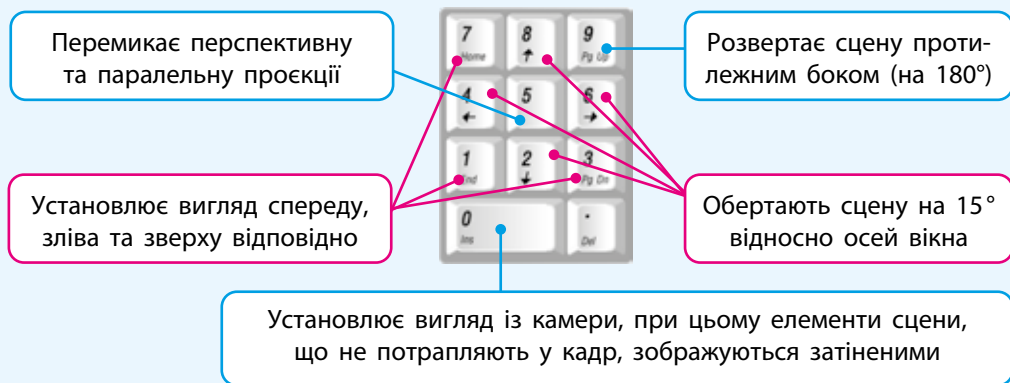


Рис. 7.6

Вам уже відоме поняття перспективи. У *перспективній проєкції* (рис. 7.7) віддаленіші об'єкти виглядають меншими, ніж розміщені ближче. У *паралельній проєкції* (рис. 7.8) спотворення розмірів немає, що дає змогу точніше розміщувати й масштабувати об'єкти (див. далі).

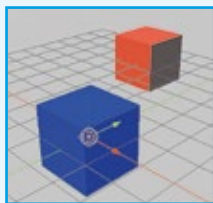


Рис. 7.7

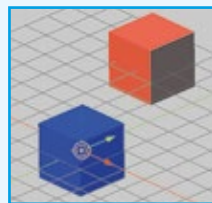


Рис. 7.8

Важливим елементом вікна 3D-вигляду є **просторовий курсор**, або **3D-курсор** (рис. 7.9). Саме там, де він розташований, з'являються додані об'єкти.

Щоб поставити 3D-курсор у певну точку, достатньо її клацнути. Точніше його положення задають у меню, яке викликають сполученням клавіш Shift+S.

Зокрема, у цьому меню наявні команди (рис. 7.10), що встановлюють курсор: у центр вибраного об'єкта чи групи об'єктів (Курсор до вибраного); у початок координат (Курсор до центра); у найближчий вузол тривимірної сітки (Курсор до сітки); у центр останнього з вибраних об'єктів (Курсор до активного).



Рис. 7.9

Прив'язка

Вибране до сітки

Вибране до курсора

Курсор до вибраного

Курсор до центра

Курсор до сітки

Курсор до активного

Рис. 7.10

Додавання об'єктів

Найпростіші композиції будують із 3D-примітивів, що є об'єктами типу **сітка** (іноді кажуть сіть, каркас або меш, від англ. *mesh* — сітчастий). Ребра таких об'єктів справді утворюють сітчастий «каркас».

Щоб додати до сцени об'єкт, на панелі інструментів слід вибрати вкладку Створити, де в списку Каркас натиснути кнопку з назвою примітива (рис. 7.11). Після цього в місці розташування просторового курсора з'явиться новий об'єкт.

Яскрава лінія, якою обведено об'єкт, означає, що його вибрано для подальших операцій. Так, якщо натиснути клавішу G, колір обведення зміниться й активується режим переміщення: досить пересунути мишу й клацнути, щоб змінити розташування об'єкта.

Аналогічно клавіша S вмикає режим змінення розміру (масштабування) об'єкта, а клавіша R — режим обертання (про це далі).

Якщо утримувати Ctrl, то операції виконуватимуться з прилипанням до координатної сітки, що дає змогу точніше змінювати об'єкти.

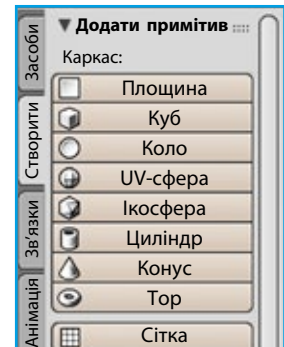


Рис. 7.11

Копіювання та клонування об'єктів

Тривимірні об'єкти можна *копіювати* й *клонувати* подібно до того, як ви це виконуєте в редакторі векторної графіки.

Щоб **створити копію вибраного об'єкта**, достатньо натиснути сполучення клавіш Shift+D (від англ. *Double* — копія) і пересунути вказівник в інше місце та клацнути. З'явиться новий об'єкт, не пов'язаний із початковим. Якщо ж натиснути Alt+D і пересунути вказівник і клацнути, то буде створено клон — копію, пов'язану з початковим об'єктом. Редагування одного з клонів впливатиме відразу на всі.

Робота з файлами

Щоб **записати результат роботи на диск**, слід вибрати команду меню Файл → Зберегти — відкриється Оглядач файлів (рис. 7.12).

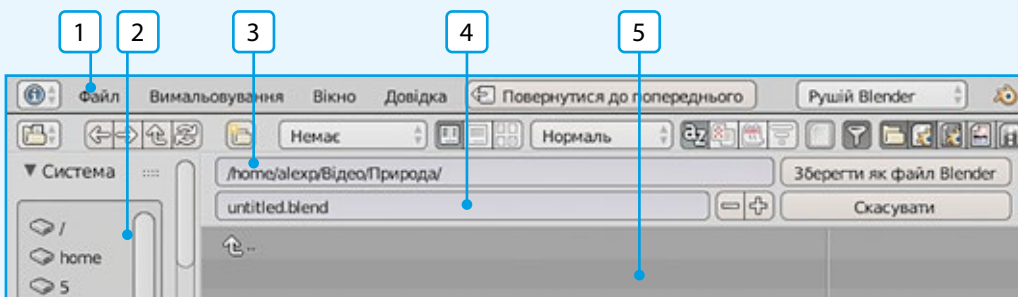


Рис. 7.12

Тут: 1 — меню Файл; 2 — доступні розташування; 3 — активна папка; 4 — ім'я файлу; 5 — вміст активної папки.

У вікні ліворуч можна вибрати диск, у вікні праворуч можна вибрати папку, ввести назву файлу та клацнути кнопку Зберегти файл Blender. Після цього буде створено файл із розширенням .blend.



Питання для самоперевірки

1. Яке призначення програми Blender?
2. Яка різниця між перспективною та паралельною проєкціями?
3. Що таке тривимірні примітиви?
4. Як можна перемістити або повернути об'єкт?
5. Як можна змінити розмір об'єкта?

Вправа 7

- Ознайомитися із середовищем тривимірного редактора Blender.
- 1. Запустити Blender і клацнути у вікні вітання (має зникнути).
Увімкніть українську мову інтерфейсу й підказок (меню File → User Preferences → System → прапорець International Fonts → Language → вибрати мову: Українська → Інтерфейс → Підказки → Зберегти за замовчуванням).
- 2. Ознайомтеся з основними засобами перегляду та керування об'єктами й випробуйте: зміну точки огляду, масштабування, переміщення та повороти куба тощо.
Видаліть куб (вибрати ПКМ → Delete → підтвердити видалення). Клацніть кілька разів у різних місцях 3D-вигляду, спостерігаючи, як переміщується 3D-курсор. Поверніть його в центр сцени, щоб новий об'єкт з'явився саме там (Shift+S).
- ⚙ 3. Додайте куб і увімкніть вигляд спереду (NUM1). Поверніть куб навколо осі Y (зелена вісь) ребром донизу (вибрати куб ПКМ → клавіша R → клавіша Y → повернути куб (для точного повороту утримуйте клавішу Ctrl) → клацнути ЛКМ) (рис. 7.13).
- 4. Перемістіть куб угору (синя стрілка) на три великих квадрати координатної сітки. Для точного переміщення увімкніть паралельну проєкцію (NUM5), збільште масштаб і утримуйте Ctrl.
- ⚙ 5. Копіюванням (Shift+D) додайте і розмістіть три куби (див. рис. 7.13).
- 6. Додайте в центрі сцени об'єкт UV-сферу і збільште (клавіша S → рух мишею → ЛКМ). Розгляньте композицію з різних боків (натискайте NUM2, NUM4, NUM6, NUM8). Збережіть файл із назвою Вправа 7.

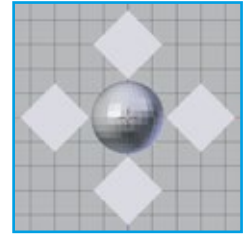


Рис. 7.13



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111799



§8

РОБОТА З ОБ'ЄКТАМИ

Ви вже вмієте швидко змінювати розміри та положення вибраних об'єктів за допомогою клавіш G, S і R та миші. Розгляньмо, як можна виконувати ці операції точніше, з'ясуємо, як стиснути або розтягнути об'єкт уздовж певної осі. А також дізнаймося, як швидко змінити колір об'єкта.

Координатні осі

З уроків математики вам відомо, що положення точки на числовій прямій можна задати одним числом — координатою точки (рис. 8.1). На площині для цього ж знадобиться дві осі, тобто прямокутна система координат і, відповідно, два числа — абсциса й ордината точки (рис. 8.2).

Об'єкти тривимірної сцени розташовані в просторі, тому координати кожної точки задають трьома числами (рис. 8.3).

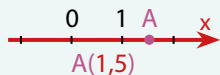


Рис. 8.1

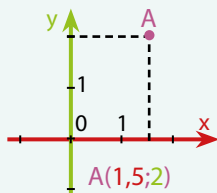


Рис. 8.2

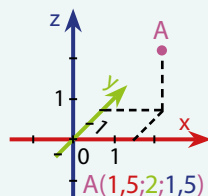


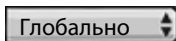
Рис. 8.3



Простір, у якому положення точки задається трьома координатами, називають **тривимірним**, що й дало назву **тривимірній графіці**.

Трансформації об'єктів

Для переміщення об'єктів ви вже користувалися стрілками, які виходять з опорної точки об'єкта. Це один із маніпуляторів трансформацій — маніпулятор переміщення. У заголовку вікна 3D-вигляду є віджети для керування цим та іншими маніпуляторами трансформацій вибраних об'єктів. Ознайоммося з таблицею, розміщеною на с. 49.

Віджет	Дія
	Вмикає/вимикає показ маніпуляторів на вибраних об'єктах (і решти віджетів для їх налаштування)
	Вмикає/вимикає маніпулятор переміщення
	Вмикає/вимикає маніпулятор обертання
	Вмикає/вимикає маніпулятор масштабування
	Список для вибору напрямків осей для маніпуляторів

Розгляньмо докладно, як можна змінити форму або розташування вибраних об'єктів. Ви вже випробували два способи переміщення об'єктів: натискання клавіші G із подальшим рухом мишею та перетягування стрілок маніпулятора переміщення  (рис. 8.4).

Для змінення **розмірів об'єктів** (масштабування) ви натискали клавіші S, а потім рухали мишею. Якщо ж увімкнути маніпулятор масштабування , то замість стрілок, що виходять з опорної точки об'єкта, з'являться кубики (рис. 8.5). Їх перетягування викликає стиснення або розтягнення об'єкта вздовж відповідної осі.

! Щоб увімкнути одночасно кілька маніпуляторів трансформації (рис. 8.6), потрібно утримувати клавішу Shift і по чергово натискати ЛКМ кнопки відповідних інструментів.

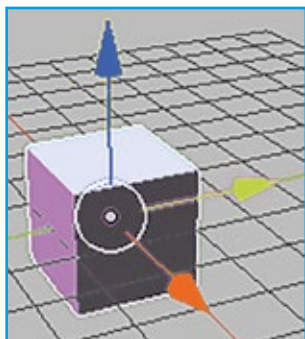


Рис. 8.4

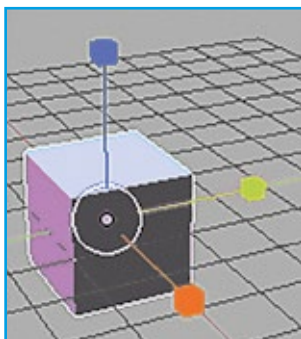


Рис. 8.5

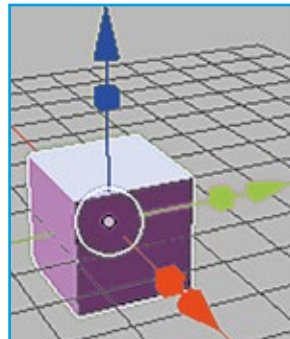


Рис. 8.6

Для обертання об'єкта відносно напрямку огляду ви натискали клавішу R, що активує режим обертання, а потім пересували мишу.

Маніпулятор обертання, який умикається кнопкою , має чотири кола різних кольорів (рис. 8.7). Перетягування будь-якої точки білого кола також дає змогу повернути об'єкт навколо осі, перпендикулярної до площини екрану, а червоного, зеленого і синього — навколо осей x , y і z відповідно.

Усі розглянуті трансформації можна також виконувати, вводючи числові значення у відповідні поля групи Розташування на панелі Трансформації (рис. 8.8).

Панель відкривають маркером . Уведені значення зразу ж впливають на розташування або форму об'єкта. Кнопка із зображенням замка поруч із полем дає змогу блокувати маніпулятори: якщо замок «замкнено», то змінити відповідне значення за допомогою маніпулятора неможливо.

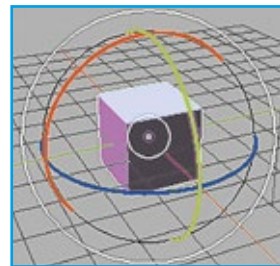


Рис. 8.7

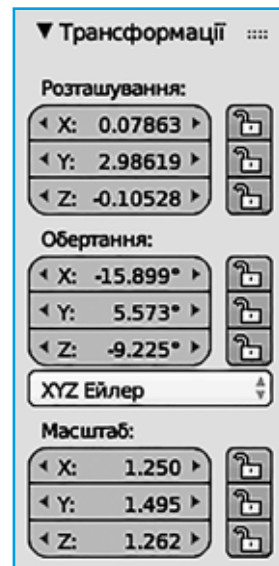


Рис. 8.8

- Деякі гарячі клавіші Blender: A — вибір усіх об'єктів, повторно — скасування вибору; NUM1 — вигляд спереду, NUM3 — справа, NUM7 — згори, R — обертання, S — масштабування, G — переміщення.



Побудуємо з тривимірних примітивів модель сніговика.

- 1) Запустимо Blender і видалимо куб. Якщо тривимірний курсор зміщено, повернемо його в центр сцени. Додамо UV-сферу.
- 2) На вигляді спереду (NUM1) увімкнемо паралельну проєкцію (NUM5) — у лівому верхньому куті вікна 3D-вигляду має з'явитися напис Спереду орто).
- 3) Увімкнемо маніпулятор масштабування  і трохи стиснемо об'єкт уздовж осі Z (синій кубик).
- 4) Увімкнемо маніпулятор переміщення  і перетягнемо UV-сферу вгору так, щоб вона «стояла» на осі X (червона лінія).

- 5) Для подальших дій увімкнемо два маніпулятори (див. рис. 8.6): утримуючи Shift, натиснемо .
- 6) Додамо до сцени ще дві UV-сфери, кожен спочатку трохи зменшимо (клавіша S), трохи стиснемо й перетягнемо вгору, формуючи «тіло» сніговика.
- 7) Додамо конус (майбутню морквину). Зменшимо його розміри, розтягнемо вздовж вертикальної осі. Для припасування конуса на місце зручно ввімкнути прозорість (клавіша Z; рис. 8.9, а).
- 8) Вимкнемо прозорість (клавіша Z) і розглянемо модель сніговика з різних боків (СКМ + рухи мишею; рис. 8.9, б).

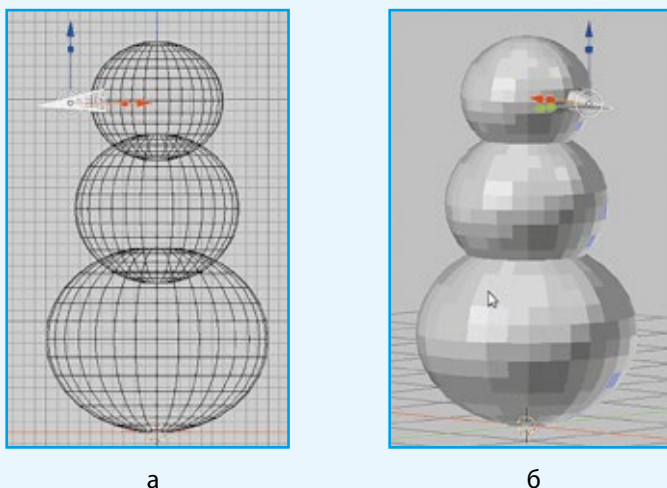


Рис. 8.9

Змінення кольору об'єкта

Blender має розвинені засоби для налаштування матеріалів об'єктів, які дають змогу моделювати *одноколірні матеріали*, додавати до них *процедурні текстури*, накладати на грані растрові *зображення з файлів*, налаштовувати прозорість і блиск матеріалу тощо. Ці та інші властивості матеріалу кодуються в блоці даних матеріалу.

Щоб вибраний тривимірний примітив отримав блок даних матеріалу, потрібно перейти на вкладку Матеріал ; клацнути кнопку Новий або вибрати один із матеріалів зі списку  — у вікні властивостей з'являться засоби для налаштування матеріалу.

Тепер, якщо на панелі Розсіювання клацнути прямокутник зі зразком кольору, з'явиться панель, у якій можна вибрати колір матеріалу (рис. 8.10). Вибір відображається у вікні 3D-вигляду. Докладніше про використання матеріалів ви дізнаєтеся з наступних параграфів.



Питання для самоперевірки

1. Що означає термін «тривимірна графіка»?
2. Як кодується положення точки в просторі?
3. Поясніть використання маніпуляторів трансформацій.
4. Як увімкнути одночасно два маніпулятори трансформацій?
5. Як додати до об'єкта новий матеріал?
6. Як змінити колір об'єкта?

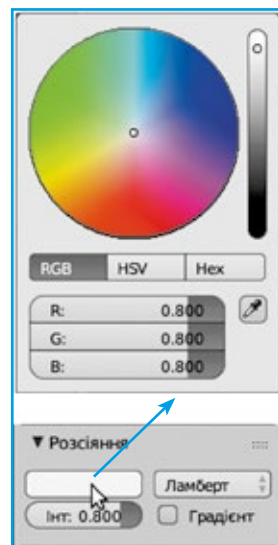


Рис. 8.10



Вправа 8

■ Побудувати за допомогою тривимірного редактора Blender модель сніговика (рис. 8.11).

1. Побудуйте модель сніговика за алгоритмом, наведеним раніше в тексті параграфа.
2. Додайте ще кілька елементів, наприклад очі та рот (із тривимірних примітивів Ікосфера, Циліндр тощо).
3. Розфарбуйте модель, змінивши кольори окремих примітивів.



4. Додайте до сцени інші елементи: поверхню «снігу», «снігові» кулі різних розмірів тощо.
5. Збережіть файл з іменем snowman.blend.



6. Продемонструйте результат однокласникам / однокласницям.

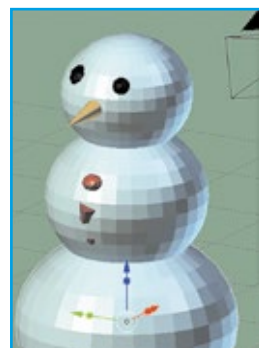


Рис. 8.11



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111800





ПРАКТИЧНА РОБОТА 3

РОБОТА З ТРИВИМІРНИМИ ПРИМІТИВАМИ

Завдання: змодельовати журнальний столик за допомогою тривимірних примітивів.

Обладнання: комп'ютер із тривимірним редактором Blender.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть програму Blender.
2. Додайте куб (або скористайтесь наявним) і, змінюючи його розміри, створіть стільницю. Для перевірки її розташування в кадрі камери натисніть NUM0 і за потреби перемістіть стільницю в потрібний бік.
3. Додайте куб і створіть із нього першу опору. Щоб точно розташувати її відносно стільниці, користуйтеся виглядами спереду (NUM1), збоку (NUM3) і зверху (NUM7), масштабуванням сцени (коліщатко миші) та режимом прозорості (клавіша Z).
4. Клонуванням (сполучення клавіш Alt+D) створіть другу опору. Додайте перемичку між опорами. Розмістіть на поверхні стола червоний циліндр і ще декілька предметів різної форми (тривимірні примітиви різних видів).
5. Задайте кольори частинам столика та предметам (див. рисунок).



6. Додайте площину (підлогу), уточніть її розташування, розміри та колір. Збережіть файл із назвою Практична робота 3.

Зробіть висновок: як змодельовати предмет із тривимірних примітивів; які предмети ви навчилися моделювати.

§9

РЕДАГУВАННЯ ФОРМИ ОБ'ЄКТА. МОДИФІКАТОРИ

В основі тривимірних об'єктів лежить сітка з вершин, сполучених ребрами, яка й визначає їхню форму. Для побудови об'єктів складної форми потрібно навчитися редагувати сітку, тобто виконувати операції з вершинами, ребрами та гранями, утвореними з них.

Режим редагування

Якщо вказівник перебуває у вікні 3D-вигляду, то натискання клавіші Tab переводить вибраний об'єкт у *режим редагування* (рис. 9.1). Для подальшої роботи слід кнопками в заголовку вікна вибрати, з чим саме будуть виконуватися операції: із вершинами (📍), ребрами (📏) чи гранями (📐). В останньому випадку кожна грань позначається квадратиком у центрі.

Якщо утримувати натиснутою клавішу Shift, то можна увімкнути кілька режимів одночасно.

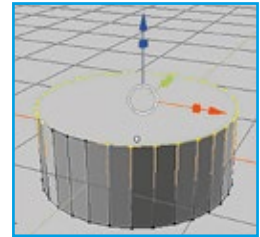


Рис. 9.1

Способи вибирання елементів

Описувані далі операції виконуються над вибраними елементами. Окремі елементи вибирають так само, як і звичайні об'єкти — клацанням ПКМ (із клавішею Shift — кілька), всі елементи — натисканням клавіші A (від англ. *all* — все); повторне натискання знімає вибір зі всіх елементів. Крім того, якщо натиснути:

- клавішу B (від англ. *box* — коробка) — біля вказівника з'являться вертикальна й горизонтальна штрихові лінії: можна вибрати потрібні елементи, охопивши їх у прямокутник;
- клавішу C — увімкнеться режим вибору малюванням: біля вказівника з'явиться коло, яке слід навести на потрібні елементи й клацнути або перетягти — вибрано. (Дію можна повторити кілька разів, змінюючи розмір кола коліщатком миші, й у кінці натиснути Enter.);
- клавішу Shift і вибрати — скасується вибір вибраних елементів. Вибрані елементи можна опрацювати, так само, як і об'єкти.

Якщо увімкнути *режим пропорційного редагування* (клавіша O), то деформація зачепить ще й елементи, сусідні з вибраними, зменшуючись із відстанню (див. приклад далі). Обертанням коліщатка в цьому разі регулюється відстань, на яку поширюється деформація.

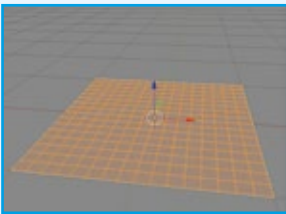
Поділ ребер

Для надання сітці складної форми її ребра й грані слід поділити на менші частини. Після того як вибрано елементи для поділу, на панелі інструментів на вкладці Засоби (група Додавання) слід вибрати команду Поділити. У результаті цих дій посередині вибраних ребер з'являться нові вершини, а ребра, що їх сполучають, утворюють нові грані.

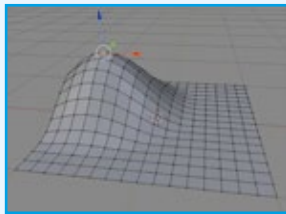


Змоделюємо поверхню території (рис. 9.2). Для цього:

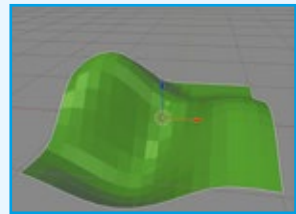
- 1) додамо площину й перейдемо в режим редагування (Tab) — усі вершини буде вибрано; поділимо вибране чотири рази (крок а);
- 2) виберемо одну з вершин (ПКМ), увімкнемо пропорційне редагування (клавіша O) і перемістимо її вгору (крок б);
- 3) повторимо попередню дію кілька разів (крок в).



а



б



в

Рис. 9.2

Моделювання витисканням

Існує ще один спосіб отримання об'єктів складної форми — **витискання** (або екструзія) вибраних елементів.



Змоделюємо кактус, подібний до об'єктів в іграх, де ігровий світ будується з кубиків (рис. 9.3, с. 56). Для цього виконаємо такі дії:

- 1) розпочнемо з куба: перейдемо в режим редагування (Tab) і увімкнемо режим роботи з гранями (📐);

- 2) виберемо верхню грань (крок а) і натиснемо клавішу E (від англ. *extrusion* — витискання) — вершини й ребра вибраної грані продублюються, а сама грань стане рухливою;
- 3) пересунемо нову грань угору та клацнемо ЛКМ — нові ребра й грані буде додано до сітки (крок б);
- 4) повторимо подібні дії з різними гранями, формуючи «кактус» (кроки в, г).

Перегляньте відеоролик із прикладом моделювання кактуса за QR-кодом або посиланням rnk.com.ua/114627.

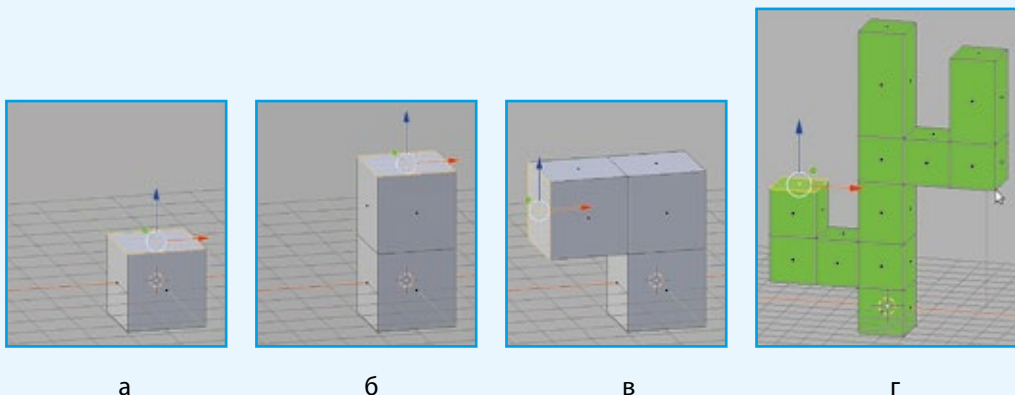


Рис. 9.3

Групування та об'єднання об'єктів

Вибрані тривимірні об'єкти можна групувати або об'єднувати кілька об'єктів у один. Щоб згрупувати **вибрані об'єкти**, слід натиснути сполучення клавіш Ctrl+G (від англ. *Group* — група). Якщо вибрати об'єкт, який входить у групу, він підсвітиться зеленим.

Хоча згруповані об'єкти не «поводяться» як одне ціле, групування дає змогу систематизувати об'єкти в складній сцені. Один об'єкт може належати до кількох груп одночасно.

Щоб **вибрати всі об'єкти групи**, слід вибрати один із них, натиснути сполучення клавіш Ctrl+G і вибрати команду Група. Якщо початковий об'єкт входить у кілька груп, то з'явиться їх список для уточнення, об'єкти якої саме групи потрібно вибрати.

Якщо **вибрати декілька об'єктів** і натиснути Ctrl+J (від англ. *join* — об'єднувати), то їхні сітки буде об'єднано в один об'єкт.

Модифікатори

Багато операцій з об'єктами доступні завдяки модифікаторам.

 **Модифікатор** — операція, яка змінює вигляд або форму об'єкта, не змінюючи його вершин, ребер, граней.

Після видалення модифікатора з об'єкта його дія скасовується. Якщо ж результат дії модифікатора цілком влаштовує, то можна натиснути кнопку Застосувати — сітка об'єкта зміниться відповідно до параметрів модифікатора, а сам модифікатор зникне з об'єкта.

Розгляньмо докладніше модифікатори Поділ поверхні і Дзеркало.

 Додамо до моделі кактуса з прикладу 2 модифікатор Поділ поверхні. Для цього вийдемо з режиму редагування й виконаємо такі дії:

- 1) перейдемо у вікні Властивості в розділ Модифікатори ();
- 2) натиснемо Додати модифікатор;
- 3) виберемо зі списку, що розкрився, модифікатор Поділ поверхні.

У вікні властивостей з'явиться панель налаштування модифікатора (рис. 9.4), а вигляд об'єкта зміниться відповідно до встановлених у ньому значень (рис. 9.5).

Для зменшення обсягу обчислень результат дії модифікатора у вікні 3D-вигляду може показуватися спрощеним. Так, на рис. 9.4 видно, що для вікна 3D-вигляду виконується один поділ поверхні (рис. 9.5). А для кінцевого зображення буде виконано три поділи (рис. 9.6).

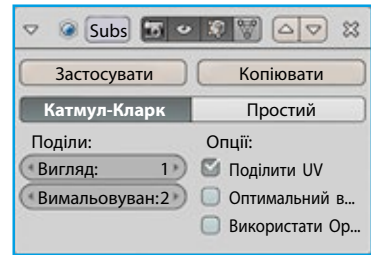


Рис. 9.4



Рис. 9.5

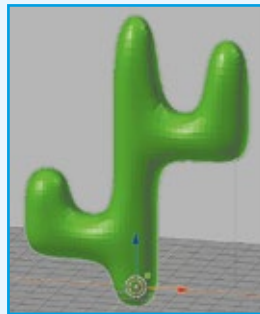


Рис. 9.6

Розгляньмо панель налаштування модифікатора:

Віджет	Опис
	Перемикач використання модифікатора під час рендерингу
	Перемикач показу дії модифікатора у вікні 3D-вигляду
	Переміщення модифікатора вгору або вниз у списку доданих до об'єкта модифікаторів
	Видалення модифікатора з об'єкта

Модифікатор Дзеркало дає змогу прискорити розробку, якщо об'єкт містить симетричні частини. Після додавання модифікатора з'являються дзеркальні копії об'єкта, кількість яких залежить від того, які осі позначено на панелі модифікатора. За таких умов доступним для редагування є лише початковий об'єкт (рис. 9.7).

«Дзеркала» розташовуються перпендикулярно до відповідних осей і проходять через опорну точку початкового об'єкта або іншого, зазначеного в полі Об'єкт-дзеркало на панелі модифікатора (рис. 9.8).

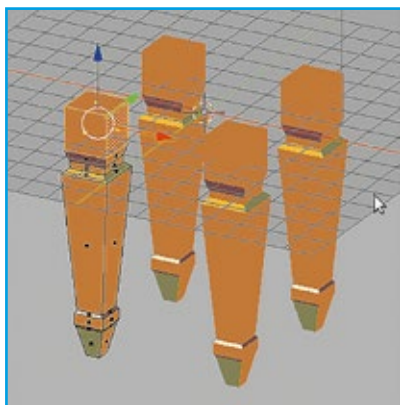


Рис 9.7

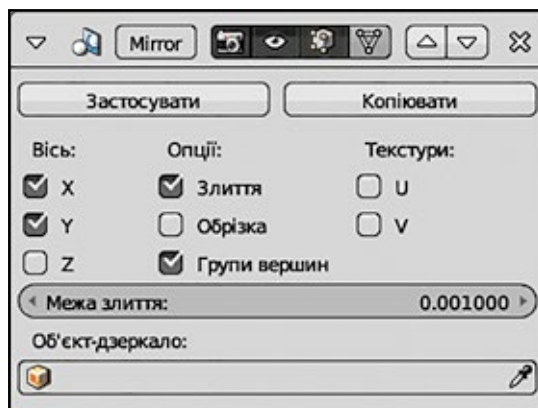


Рис 9.8

4

Щоб змоделювати чотири ніжки для табуретки, нам потрібно:

- 1) змоделювати одну ніжку;
- 2) додати модифікатор Дзеркало, перейти в режим редагування (Tab);

- 3) вибрати всі вершини (A) та перемістити їх від опорної точки, яка залишиться на місці — завдяки цьому з'являться дзеркальні копії.

Після натискання кнопки Застосувати всі дзеркальні копії буде додано до початкового об'єкта, тобто утвориться єдиний об'єкт, а модифікатор з об'єкта зникне.



Питання для самоперевірки

1. Які можливості доступні в режимі редагування сітки?
2. Якими способами можна вибрати елементи сітки?
3. Як і для чого виконують поділ ребер?
4. Поясніть різницю між групуванням та об'єднанням об'єктів.
5. Опишіть, як застосовують витискання.
6. Як додати до об'єкта модифікатор?
7. Яка дія кнопки Застосувати на панелі модифікатора?



Вправа 9

■ Змоделювати тривимірні об'єкти за допомогою пропорційного редагування, витискання й модифікатора Дзеркало.

1. Побудуйте модель поверхні за алгоритмом із прикладу 1. Збережіть її у файлі з іменем `terrain.blend`.
2. На цій самій сцені побудуйте модель кактуса за алгоритмом із прикладу 2. Розмістіть її на створеній раніше поверхні. Збережіть файл.
3. Відкрийте файл з іменем `snowman.blend` (див. вправу 8) і змоделюйте нерівну снігову поверхню.
4. В окремому файлі змоделюйте ніжки для табуретки за алгоритмом із прикладу 4. Додайте до табуретки кругле сидіння. Збережіть файл з іменем `tabouret.blend`.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111801





ПРАКТИЧНА РОБОТА 4

МОДЕЛЮВАННЯ ВИТИСКАННЯМ

Завдання: змоделювати горщик за допомогою витискання вершин.

Обладнання: комп'ютер із тривимірним редактором Blender, файл Практична робота 3.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть програму Blender. Відкрийте файл Практична робота 3.
2. Виберіть циліндр і налаштуйте перегляд так, щоб він був у центрі екрана (клавіша з крапкою на додатковому полі) на вигляді спереду (NUM1). Для точнішого керування процесом вимкніть перспективу (NUM5).
3. Перейдіть у режим редагування (клавіша Tab). Виберіть вершини верхньої основи циліндра.
4. Збільште вибраний фрагмент (клавіша S, перетягування, клацнути ЛКМ), щоб отримати форму зрізаного конуса.
5. Витисніть вибрані вершини вгору (клавіша E, перетягування, клацнути ЛКМ) і збільште фрагмент (див. п. 4).
6. Повторіть операцію декілька разів, щоб отримати форму горщика для кімнатних рослин, подібну до наведеної на рис. 1.
7. Останнє витискання виконайте вниз, щоб отримати порожнисту посудину. Для точнішого контролю товщини стінок перемикайте режим прозорості (клавіша Z; рис. 2).
8. Збережіть файл із назвою Практична робота 4.



Рис. 1

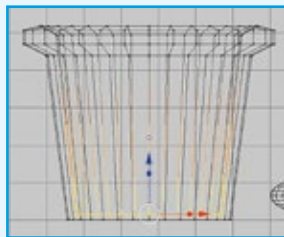


Рис. 2

Зробіть висновок: як виконується витискання, які можливості воно надає.

§ 10

МАТЕРІАЛИ ТА ОСВІТЛЕННЯ. РЕНДЕРИНГ

Так звані процедурні текстури дають змогу імітувати різноманітні матеріали. До матеріалу додається згенероване за певним алгоритмом зображення (у відтінках сірого), яке керує змішуванням власного кольору матеріалу та кольору текстури.

Процедурні текстури

З'ясуємо, як створювати складніші матеріали, на прикладі.



Додамо до матеріалу кубу текстуру деревини. Для цього:

- 1) установимо основний колір матеріалу світло-коричневий;
- 2) відкриємо у вікні властивостей розділ Текстури  і натиснемо кнопку Новий, щоб створити блок текстурних даних;
- 3) клацнемо список Тип і виберемо Деревина (рис. 10.1);
- 4) налаштуємо на панелі Деревина параметри текстури (рис. 10.2);

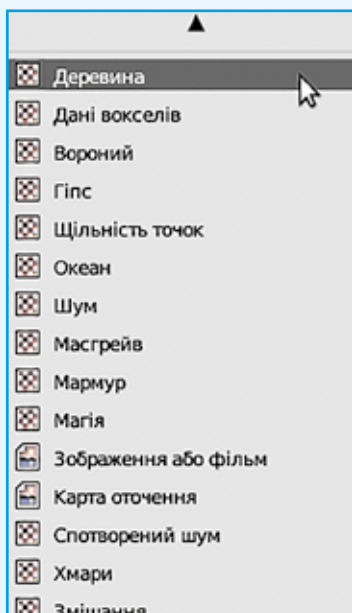


Рис. 10.1

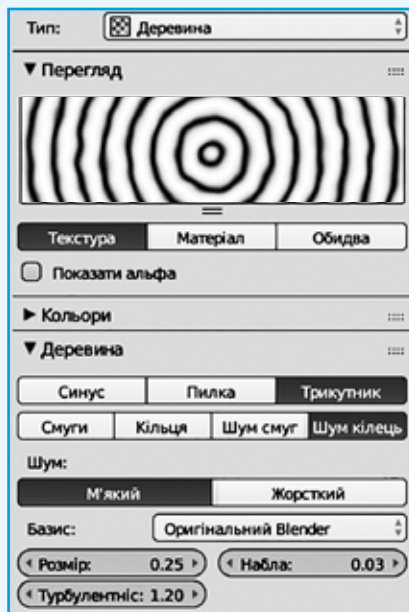


Рис. 10.2

- 5) задамо параметри накладання та розмірів текстури на панелі Розкладка (рис. 10.3);
- 6) виберемо для текстури колір, світліший від кольору матеріалу на панелі Вплив;
- 7) натиснемо кнопку F12 для перегляду кінцевого результату рендерингу (рис. 10.4).

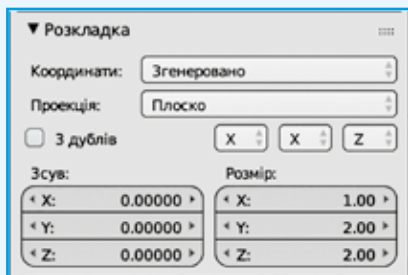


Рис. 10.3

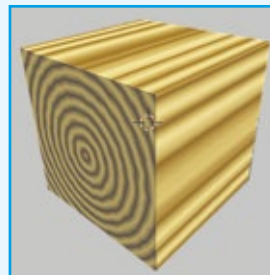


Рис. 10.4

Графічні текстури

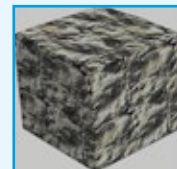
Із файлів форматів PNG, JPG тощо на грані можна накладати графічні текстури (рис. 10.5, а). Для цього слід виконати такі дії:

- 1) за допомогою графічного редактора підготувати растрові зображення для текстур;
- 2) під час додавання текстури зі списку типів текстур (див. рис. 10.1) вибрати Зображення або фільм;
- 3) на панелі Розкладка (див. рис. 10.3) уточнити спосіб накладання текстури.

Так, для об'єкта у формі паралелепіпеда зі списку Проекція слід вибрати варіант Куб (рис. 10.5, б).



а



б

Рис. 10.5

Освітлення

Невдало підібране освітлення може зіпсувати художній задум.

У початковій сцені є одне точкове джерело світла. Щоб **змінити параметри цього джерела**, його потрібно вибрати та на панелі властивостей відкрити розділ Дані (☼). Тут основними параметрами є тип

джерела світла (рис. 10.6): Точка, Сонце, Прожектор, Півсфера, Площина, його енергія та колір, який змішується з власним кольором об'єкта.

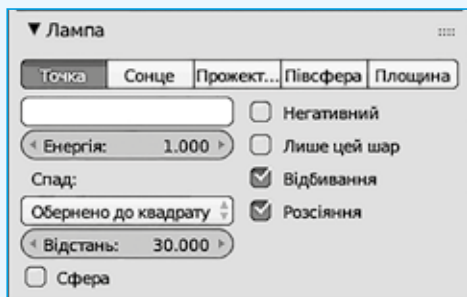
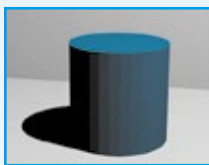


Рис. 10.6



На рис. 10.7 показано результат рендерингу сцени з циліндром, освітленим джерелами світла Точка (а), Сонце (б), Півсфера (в).



а



б



в

Рис. 10.7

Зауважмо, що за потреби до сцени можна додати два й більше джерел світла. Вони можуть бути різних типів, їх розташовують так, щоб досягти бажаного освітлення об'єктів.

Рендеринг

Кінцевою метою процесу тривимірного моделювання у Blender є зображення або відеофрагмент, у якому враховано всі налаштування сцени, матеріалів тощо.



Рендеринг (вимальовування) у комп'ютерній графіці — це процес побудови зображення з тривимірної моделі з урахуванням усіх налаштувань сцени, матеріалів, освітлення тощо.

Як керувати рендерингом? Виберімо розділ Вимальовування у вікні властивостей і розглянемо панель Розміри в цьому розділі (рис. 10.8).

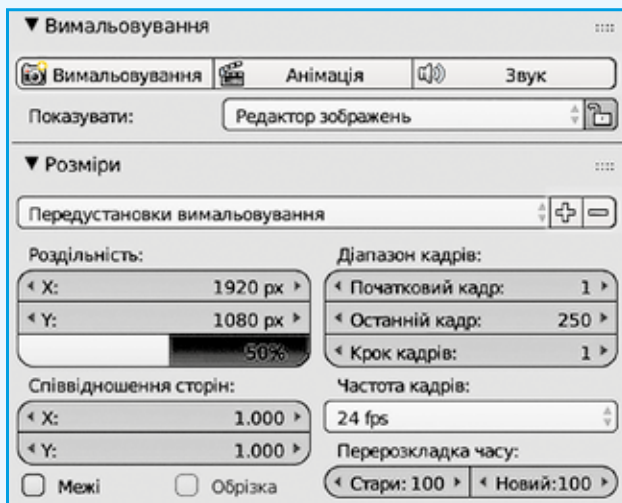


Рис. 10.8

Розміри зображення в пікселях залежать від значень у полях X і Y та розташованого під ними регулятора відсотків. Регулятор зменшує розміри, щоб швидше отримувати пробні зображення, адже сцена з багатьма складними налаштуваннями може вимальовуватися дуже довго. Для кінцевого рендерингу слід установити значення 100 %.

Для того щоб **запустити рендеринг**, слід натиснути кнопку Вимальовування або клавішу F12. Решта параметрів переважно стосуються відео й будуть розглянуті пізніше.

Як ми бачимо, реалістичні текстури та вдале освітлення є важливими елементами сцени, які значно покращують результат рендерингу.



Питання для самоперевірки

1. Що таке рендеринг?
2. Які налаштування слід зробити перед рендерингом сцени?
3. Що таке процедурна текстура?
4. Як створюють і використовують графічні текстури?
5. Як створити матеріал із текстурою деревини?
6. Джерела світла яких видів підтримує Blender?

Вправа 10

■ Дослідити властивості різних джерел світла.

1. Відкрийте файл `snowman.blend`. Запустіть текстовий процесор і створіть новий документ.
2. На вигляді зверху (NUM7) розташуйте джерело світла так, як показано на рис. 10.9. Переконайтеся, що воно має тип Точка. Зробіть скриншот і додайте в текстовий документ.
3. Виконайте рендеринг (F12). Зробіть скриншот, додайте в текстовий документ і зазначте тип джерела світла (Точка).
- ⚙ 4. Змініть тип джерела світла на Сонце. Виконайте рендеринг і додайте в документ скриншот із відповідним підписом.
5. Поверніть джерело світла Сонце в інший бік, виконайте рендеринг і додайте скриншот із підписом.
6. Дослідіть так само джерело світла Півсфера. Збережіть файл із назвою Вправа 10.

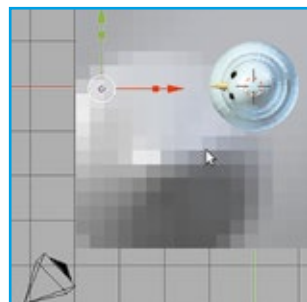


Рис. 10.9



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111802

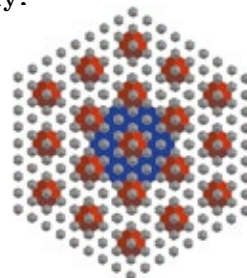


● Blender має вбудовану підтримку мови програмування Python. Це дає змогу скласти програму, яка після запуску побудує у вікні 3D-вигляду об'єкт, який складно створити вручну.

На рисунку — куби різних розмірів, що утворюють просторову фрактальну структуру.

Алгоритм побудови такий: будуємо куб (тут — синій), навколо нього — 26 кубів утричі менших (червоні); для кожного з менших кубів слід виконати попередню дію.

Якщо продовжувати, то кількість кубів дуже швидко зростатиме.



§11

АНІМАЦІЯ КЛЮЧОВИМИ КАДРАМИ. ТЕКСТОВІ ОБ'ЄКТИ

Ви знаєте, що якщо послідовно демонструвати з достатньою частотою нерухомі зображення, виникає ефект анімації. Blender дозволяє розробляти не лише нерухомі (статичні) зображення, а й має потужні засоби для створення анімаційних фільмів.

Розгляньмо основні принципи й налаштування анімації у Blender.

Шкала часу

Для налаштування анімації активно використовується **шкала часу** (рис. 11.1), яка типово розташована внизу вікна Blender (див. рис. 7.2 (позначка 5), с. 42). На ній числами (0, 20, 40...) позначено номери кадрів.



Рис. 11.1

Світло-сірий колір показує діапазон кадрів, які буде опрацьовано під час рендерингу. Спочатку це 250 кадрів, що за частоти 25 кадрів на секунду дасть відео тривалістю 10 секунд.

Щоб **перейти на певний кадр**, потрібно клацнути на шкалі часу поблизу його номера або ввести номер кадру в поле .

Ключові кадри

Одним із методів анімації, доступних у Blender, є метод ключових кадрів. На певних кадрах, які називають **ключовими**, об'єкт налаштовують так, як потрібно, і фіксують це налаштування. Після цього для проміжних кадрів положення, форму тощо буде розраховано автоматично.

Ключові кадри розрізняють за тим, який саме параметр вони фіксують: координати, кут повороту, розмір об'єкта вздовж однієї з осей, колір тощо.





На сцені з розташованими поряд червоним і зеленим кубами налаштуємо анімацію за таким сценарієм: куби нерухомі; червоний куб піднімається; перелітає над зеленим; опускається з іншого боку. На рис. 11.2 показано траєкторію та номери кадрів.

Для цього виконаємо такі дії:

- 1) перейдемо на 20-й кадр, клацнувши на часовій шкалі відповідне місце, і виберемо червоний куб;
- 2) натиснемо клавішу I, коли вказівник перебуває у вікні 3D-вигляду, — з'явиться меню (рис. 11.3);



Рис. 11.2

Вставляння ключового кадру	
Розташування	I
Обертання	I
Розмір	I
РозташОберт	I
РозташМасштаб	I
РозташОбертМасштаб	I
ОбертМасштаб	I

Рис. 11.3

- 3) виберемо тип Розташування, оскільки куб має рухатися поступально (тобто без обертання); це означає, що на перших 20 кадрах куби будуть нерухомими;
- 4) перейдемо на 40-й кадр, пересунемо червоний куб угору, натиснемо I і знову додамо ключовий кадр для розташування;
- 5) повторимо попередній крок ще двічі: на 80-му і 95-му кадрах;
- 6) перейдемо на 1-й кадр і переглянемо анімацію (сполучення клавіш Alt+A) — червоний куб перелетить над зеленим і опуститься біля нього.

Так само вставляють ключові кадри для інших трансформацій (див. рис. 11.3). Скажімо, якщо об'єкт має одночасно рухатись, обертатися й змінювати розміри, то потрібно вибрати тип ключового кадру РозташОбертМасштаб (розташування, обертання, масштабування).

Як уже було сказано, під час анімації можуть змінюватися не лише розташування та розміри об'єкта, а й такі його властивості, як колір, прозорість, енергія джерела світла та ін.



Додамо для рухомого куба (див. приклад 1) анімацію кольору: нехай колір під час руху поступово зміниться на синій. Для цього:

- 1) виберемо червоний куб і перейдемо на 20-й кадр, клацнувши на часовій шкалі відповідне місце;
- 2) відкриємо панель властивостей матеріалу і наведемо вказівник на зразок кольору (рис. 11.4, а);
- 3) натиснемо клавішу з англійською буквою I — навколо зразка кольору з'явиться жовта рамка (це означає, що для цього кольору створено ключовий кадр) (рис. 11.4, б);
- 4) перейдемо на 85-й кадр, змінимо колір на синій, наведемо вказівник на зразок кольору і знову натиснемо I.

Тепер під час перегляду анімації один із кубів не тільки перелітатиме через інший, а й змінюватиме свій колір. Так само можна додавати анімацію для інших параметрів об'єктів.

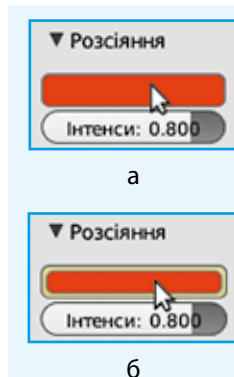


Рис. 11.4

Вирівнювання об'єктів

Як і в редакторі векторної графіки, розташування кількох вибраних об'єктів можна вирівняти. Якщо вибрати команду меню Об'єкт → Трансформації → Вирівняти об'єкти, на лівій панелі з'являється ділянка для уточнення параметрів вирівнювання (рис. 11.5). Тут можна вибрати:

- режим вирівнювання, тобто які саме точки об'єктів буде вирівняно: додатні сторони (крайні точки в додатному напрямі відповідної осі), центри, від'ємні сторони (крайні точки у від'ємному напрямі осі);
- відносно чого буде виконано вирівнювання (активний об'єкт, вибране, 3D-курсор, опорна точка сцени);
- за якими осями слід вирівняти об'єкти (X, Y, Z).

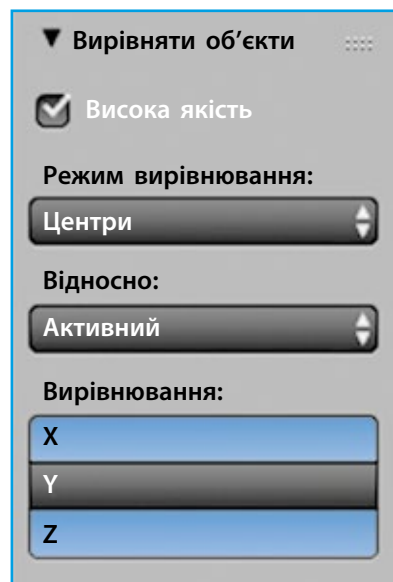


Рис. 11.5

Робота з текстовими об'єктами

Змінюючи форму, розмір, колір тощо тривимірних об'єктів, що містять написи, ми маємо змогу створювати різноманітні ефекти.

Для створення напису слід додати до сцени об'єкт типу Текст (на панелі Додати примітив у групі Інші). Спочатку він містить слово Text. Якщо перейти в режим редагування (клавіша Tab), то можна змінити текст подібно до того, як це робиться в текстовому редакторі.

Іноді при роботі з текстом у вибраному шрифті може не виявитися українських букв. У такому випадку при натисканні клавіш букви не з'являтимуться, проте коди символів у пам'яті збережуться. Букви можуть з'явитися після вибору іншого шрифту.

У вікні Властивості в розділі Дані (F) можна змінити шрифт текстового об'єкта й зробити інші налаштування.

Щоб **вибрати шрифт**, слід на панелі Шрифт клацнути кнопку  і в діалоговому вікні вибрати файл зі шрифтом. Надалі для інших об'єктів цей шрифт буде доступним у списку шрифтів .

 Маємо текстовий об'єкт, який утворений із кривих (це окремий тип об'єктів) і який є плоским (11.6, а). Щоб об'єкт став об'ємним, у вікні властивостей на панелі Геометрія слід задати значення витиснення (11.6, б) і обведення (фаски) (11.6, в). Параметр Роздільність визначає ступінь згладжування фаски (11.6, г).



а



б



в



г

Рис. 11.6

Рендеринг анімації

Ознайоммося з низкою параметрів, які стосуються візуалізації готової сцени. Щоб **створити відеоролик з анімацією**, потрібно:

- 1) увести на панелі Вимальовування (див. рис. 10.8, с. 64) номери початкового й кінцевого кадрів;

- 2) вибрати на панелі Вивід (рис. 11.7) формат відео (наприклад, H.264) і колірний режим (BW — відтінки сірого, RGB — кольоровий);
- 3) клацнути кнопку  і вибрати папку для відеофайлу; на панелі Вимальовування натиснути кнопку Анімація.

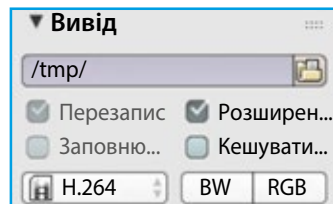


Рис. 11.7

За рендерингом можна спостерігати на екрані, і через певний час у зазначеній папці з'явиться створений відеофайл.

Таким чином, підсумовуємо: щоб отримати відеоролик, потрібно змодельовати об'єкти, налаштувати анімацію та виконати рендеринг.



Питання для самоперевірки

1. Для чого призначена шкала часу?
2. Поясніть принцип анімації ключовими кадрами.
3. Які типи ключових кадрів ви знаєте?
4. Як додати до сцени текстовий об'єкт?
5. Як змінити текст у текстовому об'єкті?
6. Які основні параметри для рендерингу анімації?



Вправа 11

■ Створити анімаційний відеоролик.

1. Запустіть програму Blender. Додайте ще один куб і налаштуйте анімацію за алгоритмом із прикладу 1.
2. Додайте анімацію кольору одного з кубів (див. приклад 2).
3. На панелі Вимальовування задайте розмір кадру камери 720×576 пікселів. Налаштуйте за допомогою ключових кадрів рух камери так, щоб вона поступово наближалася до кубів протягом перших 2 секунд.
4. Додайте площину — поверхню, на якій розміщені куби. Налаштуйте анімацію її кольору на свій розсуд.
5. Додайте до сцени напис «Кубики» жовтого кольору. В кінцевому положенні камери напис має бути в правому нижньому куті.
6.  Перевірте, як виглядають окремі кадри на початку, посередині й у кінці анімації. Для цього клацніть бажане місце на шкалі часу, потім натисніть F12 (вихід із перегляду — клавіша Esc). За потреби змініть розташування джерела світла.

7. Виконайте рендеринг анімаційного ролика (від 1 до 120 кадру) з якістю 25 % у файл із назвою Вправа 11.
8. Перегляньте результат і за потреби змініть налаштування. Виконайте рендеринг з якістю 100 %.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111803



ПРАКТИЧНА РОБОТА 5 СТВОРЕННЯ ТРИВИМІРНОЇ АНІМАЦІЇ

Завдання: змоделювати сцену з графічних примітивів і текстового об'єкта, налаштувати анімацію й створити відеоролик.

Обладнання: комп'ютер із тривимірним редактором Blender.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть програму Blender. Видаліть куб.
2. Додайте до сцени текстові об'єкти (окремий об'єкт для кожної букви), які утворюють слово Україна. За потреби доберіть шрифт з українськими буквами.
3. Виберіть усі текстові об'єкти. На панелі Геометрія змініть величину витиснення й спостерігайте за активним об'єктом (останнім вибраним). Установлену величину витиснення поширте на решту букв (клацнути ПКМ на полі Витиснути → Копіювати до вибраного).
4. Так само встановіть у полі Глибина для всіх вибраних текстових об'єктів однакову глибину обведення.
5. Доберіть для кожної букви інший колір.
6. Установіть камеру, щоб напис був посередині кадру. Для цього:
 - 1) у вікні 3D-вигляду на вигляді зверху (NUM7) перейдіть у таку точку огляду, щоб напис був у центрі вікна;

2) виберіть команду меню Вигляд → Вирівняти перегляд → Вирівн. акт. камеру за виглядом.

Якщо камера виявиться занадто близько або далеко від напису, то на вигляді спереду (NUM1) віддаліть або наблизьте її, щоразу перевіряючи результат на вигляді з камери (NUM0).

7. Позаду букв додайте примітив Площина такого розміру, щоб його краї виходили за межі кадру. Доберіть колір.
 8. Підберіть розташування й тип джерела світла. Результат перевіряйте, виконуючи пробний рендеринг (клавіша F12).
 9. Виберіть першу букву та налаштуйте анімацію, щоб протягом першої секунди (за частоти 25 кадрів/с) вона прилітала на своє місце з-за меж кадру. Для цього виконайте такі дії:
 - 1) додайте на 25-му кадрі ключовий кадр для розташування літери (I → Розташування);
 - 2) перейдіть на 1-й кадр, перемістіть об'єкт за межі кадру й додайте ще один ключовий кадр для розташування.
- Перегляньте результат, натиснувши Alt+A.
10. Налаштуйте анімацію решти букв: нехай деякі з них прилітають обертаючись, змінюють розміри тощо.
 11. Налаштуйте анімацію кольору так, щоб після того, як усі букви займуть свої місця, їхні кольори протягом 1 секунди стали однаковими. Для цього:
 - 1) на кадрі, де починатиметься змінення кольору, для кольору кожної букви додайте ключовий кадр (див. приклад 2 в §11);
 - 2) на кадрі з номером, на 25 більшим, змініть колір кожної літери, ввівши його 16-ковий код (наприклад, 66AAFF), і знову додайте ключовий кадр.
 12. Налаштуйте параметри рендерингу відео:

- початковий кадр — 1;
- кінцевий — після останнього ключового кадру;
- частота кадрів — 25 кадрів/с;
- формат відео — H.264.

Виконайте пробний рендеринг із розміром 25 %. Перегляньте результат, виправте недоліки.

Виконайте кінцевий рендеринг із якістю 100 %. Обговоріть результат з однокласником чи однокласницею.

Зробіть висновок: яка послідовність дій із розробки відеоролика в тривимірному редакторі Blender.



ОПРАЦЮВАННЯ ТАБЛИЧНИХ ДАНИХ

3

РОЗДІЛ

У цьому розділі ви вдосконалисте знання про типи посилань на клітинки та вміння використовувати в розрахунках убудовані функції. Ви опануєте інструменти та методи аналізу даних, а також засоби візуалізації даних.

§12

ТИПИ ПОСИЛАНЬ НА КЛІТИНКИ ЕЛЕКТРОННОЇ ТАБЛИЦІ

При створенні формул ми використовуємо посилання на клітинки — звертаємося до значень, які містяться в інших клітинках. Але посилання «поводяться» по-різному, коли копіюємо або переміщуємо формули.

Відносне посилання



Відносне посилання — посилання на клітинку електронної таблиці, яке містить лише адресу клітинки (наприклад, A1).

Під час копіювання формули в іншу клітинку відносне посилання модифікується так, що назва стовпця й номер рядка змінюються на величину зміщення кінцевої клітинки з формулою відносно початкової.



Формула =A1 у клітинці B1 під час копіювання в клітинку B2 перетворюється на =A2.

Скопіювати формулу можна за допомогою контекстного меню клітинки або сполучень клавіш Ctrl+C, Ctrl+V.

Скопіювати формулу до суміжних клітинок можна перетягуванням маркера автозаповнення клітинки з формулою (рис. 12.1).

Модифікована формула копіюється водночас до всіх клітинок діапазону.



Рис. 12.1

Абсолютне посилання

Модифікація посилань на клітинку дає змогу швидко створювати багато подібних формул. Проте вона потрібна не завжди. Часто деякі посилання не повинні змінюватися при копіюванні формули.



Абсолютне посилання — посилання на клітинку електронної таблиці, яке є незмінним при копіюванні формули в інше місце таблиці.

Для введення абсолютних посилань використовують знак \$. Якщо в будь-яку клітинку скопіювати формулу =\$A\$1, вона не змінить свого

вигляду і вказуватиме на клітинку з адресою A1. Тобто таке посилання завжди звертатиметься до клітинки, вказаної в початковій формулі.

Щоб швидко змінити тип посилання, необхідно встановити курсор на посилання у формулі й натиснути клавішу F4.

Мішане посилання



Мішане посилання — посилання на клітинку електронної таблиці, де одна з частин адреси — назва стовпця або номер рядка — абсолютна (тобто позначена знаком \$), а інша — відносна.

Наприклад: \$A1 або A\$1. Під час копіювання формули з мішаним посиланням модифікуються лише відносні номери рядків і назви стовпців, а абсолютні залишаються незмінними.



Порівняємо результат копіювання формули з клітинки B2 у C3.

Тип посилання	Формула в B2	Результат у C3
Відносне	=A1	=B2
Абсолютне	=\$A\$1	=\$A\$1
Мішане (стовпець)	=\$A1	=\$A2
Мішане (рядок)	=A\$1	=B\$1



При копіюванні адресних клітинок або їх вмісту до інших клітинок посилання на них у формулах не модифікуються.

При переміщенні адресних клітинок (діапазонів) до інших клітинок модифікуються всі посилання на них незалежно від типу посилань та їх розташування.



Розглянемо рис. 12.2. Під час переміщення клітинки A1 на місце клітинки C3 формули в клітинці B2 =A1; =\$A\$1; =A\$1 модифікуються на =C3; =\$C\$3; =C\$3; =\$C3 відповідно.

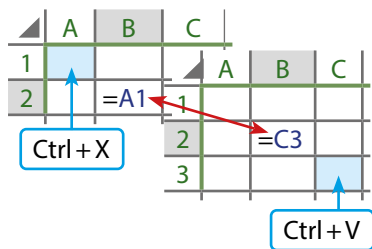


Рис. 12.2

Так само модифікуються всі посилання при переміщенні адресної клітинки після вставляння або видалення стовпців, рядків, клітинок,

діапазонів вище або ліворуч від адресної. Видалення адресних клітинок призводить до помилок у формулах, які посилалися на ці клітинки.

Діапазони клітинок

У тих випадках, коли потрібно опрацювати сукупність чисел, вказують посилання не на одну клітинку, а на певний діапазон.



Діапазон — сукупність клітинок, які можна опрацьовувати як єдине ціле.

Діапазони зручно використовувати у формулах (наприклад, `=SUM(A1:A5)` — обчислити суму 5 чисел), для форматування, копіювання/вставлення одразу кількох клітинок, для побудови діаграм.

Розрізняють два види діапазонів: зв'язні та незв'язні.

Зв'язним називають діапазон, який можна виділити переміщенням указівника з натиснутою ЛКМ. Щоб **виділити зв'язний діапазон**, потрібно перетягнути вказівник від однієї клітинки до іншої або клацнути одну клітинку, а потім із натиснутою клавішею Shift — іншу.

Зв'язний діапазон позначається адресами лівої верхньої й правої нижньої клітинок або двома заголовками рядків (стовпців), відокремленими двокрапкою. Так, B2:C2 — прямокутний діапазон: від клітинки B2 до клітинки C2 (рис. 12.3).

Окрему клітинку також можна вважати зв'язним діапазоном.

Адреса діапазону рядків (стовпців) складається з адрес 1-го й останнього рядків (стовпців), відокремлених двокрапкою: 2:4 — діапазон з усіх клітинок рядків 2, 3 і 4, C:F — стовпців C, D, E і F.

Незв'язним називають діапазон, який складається з кількох зв'язних діапазонів. Щоб **виділити незв'язний діапазон**, слід, утримуючи клавішу Ctrl, виділити кілька клітинок і/або діапазонів. Так, адреса незв'язного діапазону, наведеного на рис. 12.4: A2; C1:D2.

Для скасування виділення діапазону потрібно клацнути будь-яку клітинку. Зв'язний діапазон можна копіювати й переміщувати. Для незв'язних діапазонів такі операції не завжди можливі.

	A	B	C	D
1				
2				
3				

Рис. 12.3

	A	B	C	D
1				
2				
3				

Рис. 12.4



Питання для самоперевірки

1. Що таке відносне посилання?
2. Як записують абсолютні посилання?
3. Як виділити прямокутний діапазон клітинок?
4. Скільки клітинок містить зв'язний діапазон A2:B3?
5. Як виділити незв'язний діапазон?



Вправа 12

- Розрахувати об'єм і вартість засобів, потрібних фермеру для боротьби зі шкідниками, на 4 земельних ділянках. Вхідні дані для розрахунків: площі ділянок, ціни й норми витрат чотирьох засобів.
1. Запустіть табличний процесор. Заповніть таблицю за рис. 12.5.

	A	B	C	D	E	F
1		Засіб 1	Засіб 2	Засіб 3	Засіб 4	
2	Ціна, грн/л	1500	2000	1800	1950	
3	Норма, мл/га	120	90	130	130	
4						Площа, га
5	Ділянка 1					3
6	Ділянка 2					1
7	Ділянка 3					2,8
8	Ділянка 4					4,2
9						
10	Разом					
11	Вартість					

Рис. 12.5

2. Уведіть у клітинку B5 формулу для розрахунку об'єму засобу 1 для ділянки 1. Які частини адрес клітинок мають залишатися незмінними під час копіювання формули вправо й униз? Змініть формулу, додавши, де потрібно, знак \$.
3. Скопіюйте формулу в решту клітинок діапазону B5:E8. Якщо після копіювання виявлено помилки, уточніть формулу в клітинці B5 і повторіть копіювання.
4. У клітинці B10 за допомогою функції SUM створіть формулу для підрахунку загального об'єму засобу 1 і скопіюйте в клітинки C10:E10.

5. У клітинку B11 уведіть формулу для розрахунку вартості засобу 1. Урахуйте, що об'єм розраховано в *мілілітрах*, а ціну наведено за 1 літр. Скопіюйте формулу в клітинки C11:E11.
6. У клітинці F11 розрахуйте загальну вартість усіх засобів. Для клітинок із грошовими сумами встановіть належний формат. Збережіть файл із назвою Вправа 12.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111804



§ 13

ПАРАМЕТРИ СТОРІНКИ. ДРУКУВАННЯ ТАБЛИЦІ

Налаштування структури таблиці й параметрів сторінки в табличному та текстовому процесорах подібні. Налаштування структури включає об'єднання клітинок, додавання та видалення рядків (стовпців) тощо. Параметри сторінки вибирають на вкладці РОЗМІТКА СТОРІНКИ.

Налаштування структури таблиці

► Об'єднання клітинок

Під час оформлення таблиці може виникнути потреба в об'єднанні кількох клітинок, як-от для створення спільного заголовка для кількох стовпців. Найпростіше — це виділити їх і вибрати на вкладці Основне в групі Вирівнювання кнопку Об'єднати й помістити у центрі . У такій клітинці залишаться лише дані з верхньої лівої клітинки.

Щоб скасувати об'єднання клітинок, потрібно ще раз клацнути ту саму кнопку (дані клітинок при цьому не відновлюються).

Розгляньмо деякі операції з рядками й стовпцями таблиці.

► Додавання та видалення рядків і стовпців

Для додавання рядків/стовпців потрібно:

- 1) виділити рядок/стовпець, поруч із яким слід додати новий;
 - Щоб додати новий рядок вище виділеного, виділіть цілий рядок (натиснувши на номер рядка).
 - Щоб додати новий стовпець ліворуч від виділеного, виділіть цілий стовпець (натиснувши на букву стовпця).
- 2) викликати на виділеному рядку/стовпці контекстне меню;
- 3) у контекстному меню вибрати Вставити — буде автоматично встановлено новий порожній рядок або стовпець ліворуч.

Для видалення виділених рядків/стовпців потрібно в контекстному меню вибрати Видалити.

► Змінення розмірів стовпців і рядків

Доречно налаштована ширина та висота клітинок роблять таблицю зручною для аналізу даних. Існує кілька способів налаштування: перетягування меж заголовків, автоматичний підбір, задання значення.

Із 7 класу (§ 21) ви знаєте, що змінити ширину стовпця чи висоту рядка можна вручну, якщо перетягти межу між його заголовком і заголовком наступного рядка або стовпця (рис. 13.1). Якщо двічі клацнути на цій межі, то відповідний розмір стовпця (рядка) автоматично підлаштується під розмір найдовшого (найвищого) вмісту в ньому.



Рис. 13.1

Ширину стовпця (висоту рядка) можна задати числом: виділити один або декілька рядків (стовпців), у контекстному меню вибрати команду Висота рядка... (Ширина стовпця...) і ввести числове значення.

Щоб установити однакову ширину кількох стовпців (висоту рядків), їх слід виділити й перетягти одну з меж або встановити потрібне значення командою контекстного меню Ширина стовпця (Висота рядка).

Перенесення тексту

За наявності великого обсягу текстових даних для того, щоб зберегти компактність електронної таблиці, можна застосувати функцію Перенести текст. Вона розбиває довгий текст у клітинці на декілька рядків, а висота рядка при цьому автоматично збільшується.

Щоб увімкнути перенесення тексту, потрібно виділити клітинку або діапазон клітинок і виконати: ОСНОВНЕ → Вирівнювання → Перенести текст. Можна також примусово перенести текст на новий рядок усередині клітинки, натиснувши Alt+Enter під час редагування клітинки. Це створить ручний розрив рядка; щоб це відображалось коректно, функція Перенести текст повинна бути ввімкнена.

Тепер можна використовувати інші опції вирівнювання у групі Вирівнювання на вкладці ОСНОВНЕ, щоб краще розташувати текст.

Оптимізація таблиці

Якщо таблиця не вміщується на одну сторінку, то вона автоматично розподілиться на кілька сторінок (рис. 13.2). У разі ручної оптимізації таблиці змінюють орієнтацію сторінки, розмір та орієнтацію тексту, розміри рядків і стовпців тощо (рис. 13.3).

Неоптимізований варіант

A	B	C	D	E	F	G	H
Прізвище	Інформатика	Фізика	Хімія	Алгебра	Геометрія	Біологія	Географія
Петренко	12	10	8	9	11	10	12
Іваненко	7	8	9	8	7	9	8
Ткачук	5	6	5	7	6	6	7

Рис. 13.2

Після вирівнювання й орієнтації тексту в клітинках верхнього рядка

Прізвище	Інформатика	Фізика	Хімія	Алгебра	Геометрія	Біологія	Географія
Петренко	12	10	8	9	11	10	12
Іваненко	7	8	9	8	7	9	8
Ткачук	5	6	5	7	6	6	7

Рис. 13.3

Отже, щоб **оптимізувати таблицю**, слід виділити потрібні клітинки верхнього рядка; на вкладці ОСНОВНЕ в групі Вирівнювання задати вирівнювання й орієнтацію тексту; змінити висоту потрібного рядка й ширину стовпців з оцінками.

► Установлення меж клітинок

Таблиця, яку ми бачимо в Excel, розбита на клітинки умовно.

Розгляньмо рис. 13.4. Для **встановлення меж клітинок** потрібно:

- 1) виділити клітинку (діапазон);
- 2) на вкладці ОСНОВНЕ в групі Вирівнювання натиснути кнопку ;
- 3) установити на вкладці Межі вікна Формат клітинок формат межі.

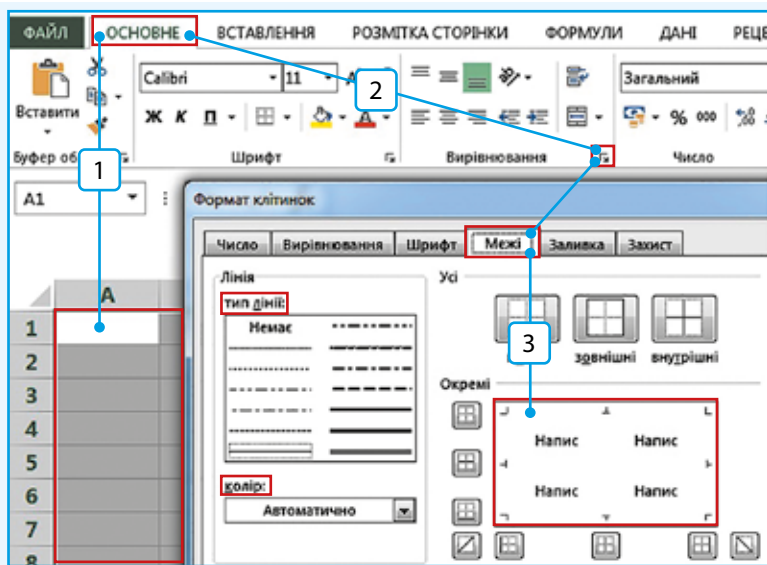


Рис. 13.4

Друк електронної таблиці подібний до друку документа Word. Але роздруковується лише та її частина, де ввімкнено межі чи заливку або є вміст клітинок. Готувати таблицю до друку зручно в режимі ВИГЛЯД → Розмітка сторінки: розміри рядків і стовпців подаються в сантиметрах.

Для **підготовки електронної таблиці до друку** потрібно:

- 1) установити параметри сторінки; оптимізувати таблицю форматкуванням її об'єктів;
- 2) об'єднати діапазони клітинок (за потреби), установити межі.

Надрукувати її можна за допомогою команди ФАЙЛ → Друк (Ctrl+P).



Питання для самоперевірки

1. Як змінити розмір одного стовпця або рядка?
2. Як установити однаковий розмір кількох стовпців (рядків)?
3. Що включає оптимізація таблиці?
4. Як об'єднати декілька клітинок?
5. Як відкрити вікно для налаштування заливки або меж клітинок?
6. Що передбачає підготовка електронної таблиці до друку?



Вправа 13

- Створити таблицю за зразком (рис. 13.5), оформити так, щоб було зручно бачити предмети, прізвища, оцінки за кожен день тижня.

	A	B	C	D	E	F	G
1	Журнал успішності						
2	Предмет	Прізвище	Пн	Вт	Ср	Чт	Пт
3	Математика	Іваненко	10	9	11	12	10
4		Петренко	8	7	9	10	6
5		Сидорчук	11	10	0	10	11
6	Українська мова	Іваненко	8	9	11	12	10
7		Петренко	11	9	9	10	10
8		Сидорчук	11	10	0	10	11

Рис. 13.5

1. Запустіть табличний процесор. Установіть параметри сторінки: всі поля — по 1 см; розмір — A4 (21×29,7 см); орієнтація — книжкова. Установіть режим перегляду: ВИГЛЯД → Розмітка сторінки.
2. У першому рядку об'єднайте клітинки A1:B1, уведіть текст Журнал успішності і вирівняйте його по центру.
3. Уведіть у клітинку C2 текст Пн і автозаповненням створіть список із назвами днів тижня в клітинках C2:G2.
4. Об'єднайте клітинки A3:A5 для предмета Математика, A6:A8 — для предмета Українська мова. Вирівняйте об'єднані клітинки по центру по вертикалі. Установіть межі для всієї таблиці.
5. Виділіть жирним заголовки рядків і стовпців. Додайте заливку клітинок.

6. У стовпці Н уведіть формулу для обчислення середнього бала кожної особи за тиждень. Збережіть файл із назвою Вправа 13.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111805



§ 14

ФУНКЦІЇ. ЛОГІЧНІ ФУНКЦІЇ

Функції, убудовані в електронну таблицю, дають змогу виконувати широкий спектр обчислень і операцій із даними. Завдяки їм вдається швидко будувати формули для складних обчислень.

Класифікація функцій

Функція — заздалегідь визначені формули для виконання обчислень із заданими величинами (аргументами) у зазначеному порядку. Дані, які опрацьовує функція, називають **аргументами**.

Убудовані функції за призначенням поділяють на низку категорій. Аргументами можуть бути числа, вирази, посилання, умови тощо.

Будь-яка функція в Excel характеризується назвою, кількістю аргументів, типами аргументів і результатом.

На рис. 14.1 проілюстровано відому вам функцію: назва — MAX, кількість аргументів — не менше одного; тип аргументу — числовий; результат — найбільше значення зі списку аргументів.

	A	B	C
1	=MAX(B1:C2)	4	1
2		3	4

назва
аргументи
результат

Рис. 14.1

! Аргументи функції записують у дужках і, якщо їх декілька, розділяють символом «;».

Вибрати потрібну функцію в табличному процесорі і ввести її у формулу можна за допомогою діалогового вікна Вставлення функції. Воно відкривається кнопкою виклику функції f_x у рядку формул.

! В україномовній версії Excel назви функцій подано англійською мовою.

Вікно Вставлення функції складається з трьох частин (рис. 14.2).

У першій частині можна ввести опис дії, яку слід виконати, і натиснути кнопку Знайти. Це варто використати, якщо користувач/користувачка не знає чи не пам'ятає, як називається потрібна функція.

Після того як у списку Категорія вибрано категорію функції, у списку Виберіть функцію відображаються лише функції, що входять до неї.

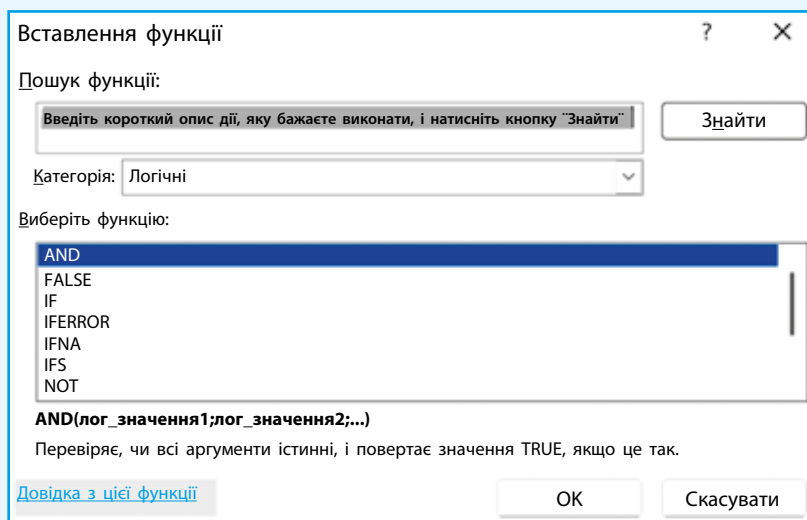


Рис. 14.2

Під списком Виберіть функцію розміщено короткий опис вибраної функції. Розширену довідку з прикладами використання функції можна отримати за посиланням Довідка з цієї функції.

Значення аргументів функції можна вводити в поля для введення аргументів або виділяючи клітинки таблиці, що містять потрібні

аргументи. Після виділення клітинок або діапазонів клітинок їх адреси з'являються в полях для введення аргументів.

Швидко знайти й уставити потрібну функцію можна в групі інструментів Бібліотека функцій на вкладці ФОРМУЛИ (рис. 14.3).

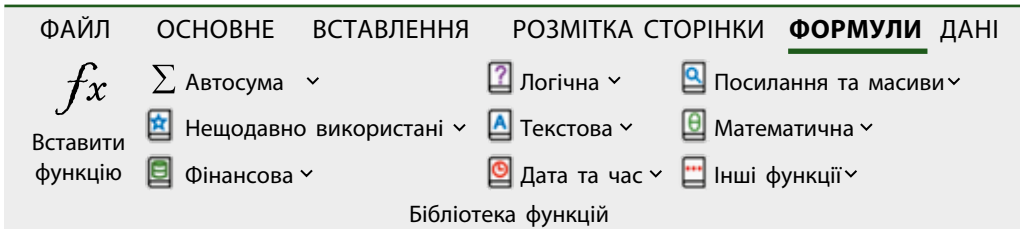


Рис. 14.3

Основні елементи групи Бібліотека функцій і їх призначення такі:

- кнопка fx — відкриває вікно Вставлення функції;
- кнопка Σ — автоматично додає формулу для обчислення суми (функція SUM).

У списку кнопки можна вибрати й швидко вставити функції Середнє (AVERAGE), Кількість (COUNT), Максимум (MAX), Мінімум (MIN);

- інші кнопки — відкривають списки функцій відповідних категорій:
 -  математичні та тригонометричні (SUM, ROUND, INT, PRODUCT, POWER, SIN);
 -  статистичні (AVERAGE, MEDIAN, STDEV, COUNT);
 -  логічні (IF, AND, OR, NOT, IFS).

Логічні функції

Логічні функції, знайомі вам із 7 класу, дають змогу виконувати обчислення, результат яких залежить від певних умов.



Логічна функція — це функція, один або кілька аргументів якої — логічні вирази.

Логічний вираз — вираз, результатом обчислення якого може бути одне з двох значень: ІСТИНА (TRUE) або ХИБНІСТЬ (FALSE).

Із правилами побудови та обчислення значень логічних виразів (як простих, так і складених) ви ознайомились у 7 класі. Пригадаймо їх.

Простий логічний вираз складається з двох виразів, пов'язаних операцією відношення, як от:

- більше >
- менше <
- дорівнює =
- не менше (більше або дорівнює) >=
- не більше (менше або дорівнює) <=
- не дорівнює <>

Наприклад: A1=10 (перевіряє, чи значення в клітинці A1 дорівнює 10); B2>F5 (перевіряє, чи значення в клітинці B2 більше ніж у F5); C3<="01.01.2025" (перевіряє, чи дата в клітинці C3 передує або дорівнює 1 січня 2025 року); D4<>"Кіт" (перевіряє, чи значення в клітинці D4 відрізняється від "Кіт").

Якщо ввести в клітинку формулу, яка містить лише логічний вираз, то в ній з'явиться його значення — TRUE або FALSE.

У функціях Excel на місці логічного виразу може міститися арифметичний вираз. Табличний процесор перевіряє, чи не дорівнює значення арифметичного виразу нулю. Наприклад: 2+2 — трактується як істина (тому що $2+2 \neq 0$), 2-2 — як хибність (тому що $2-2 = 0$).

► Функції AND, OR, NOT

Для побудови складених логічних виразів використовуються логічні функції AND(), OR(), NOT(). Їх аргументами є один (у функції NOT) або більше логічних виразів. Функція AND() повертає TRUE, якщо всі її аргументи істинні. В інших випадках функція повертає FALSE.



=AND(A1>10; B1<20; C1="Так")

Поверне TRUE, тільки якщо A1 більше 10, B1 менше 20 і C1 дорівнює "Так". В іншому випадку — FALSE.

Функція OR() повертає TRUE, якщо істинний хоча б один з її аргументів. Вона повертає FALSE лише тоді, коли всі аргументи хибні.



= OR(A1="Помилка"; B1=0)

Поверне TRUE, якщо A1 дорівнює "Помилка", або B1 дорівнює 0, або істинні обидві умови. Поверне FALSE, тільки якщо A1 не "Помилка" і B1 не 0.

Функція NOT() змінює логічне значення на протилежне.



=NOT(A1=10)

Якщо A1 дорівнює 10, поверне FALSE. Якщо A1 не дорівнює 10, поверне TRUE.



► Функції IF

Обчислення одного з двох виразів, залежно від умови, забезпечує функція IF.

Синтаксис:

IF(Умова; Вираз1; Вираз2)

Вираз1 — обчислюється, якщо Умова істинна.

Вираз2 — обчислюється, якщо Умова хибна.

4

Нехай у відомості успішності потрібно біля оцінки вивести слово «Добре», якщо вона вища від 6 балів, і «Задовільно», якщо інша. Алгоритм розв'язування задачі подано у фрагменті блок-схеми (рис. 14.4). Він реалізується за допомогою функції IF (рис. 14.5): IF(D1>6; "Добре"; "Задовільно")

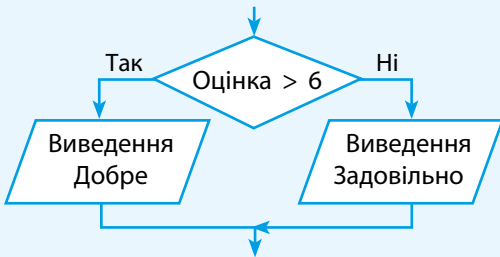


Рис. 14.4

f_x	=IF(D1>6; "Добре"; "Задовільно")		
	C	D	E
	Петренко	10	Добре
	Бунич	8	Добре
	Іваненко	4	Задовільно

Рис. 14.5

5

Іноді доцільно використати вкладені логічні функції IF. Нехай треба підписати успішність за такими «умовними рівнями»: початковий (1–4 бали), середній (5–8 балів), високий (9–12 балів). Маємо (рис. 14.6): =IF(D1<5; "Початковий"; IF(D1<9; "Середній"; "Високий"))

f_x	=IF(D1<5; "Початковий"; IF(D1<9; "Середній"; "Високий"))					
	C	D	E	F	G	H
	Петренко	10	Високий			
	Бунич	8	Середній			
	Іваненко	4	Початковий			

Рис. 14.6

Функція IFS() (доступна в Excel 2016 і новіших версіях) спрощує написання формул із кількома умовами, замінюючи кілька вкладених

функцій IF. Функція IFS послідовно перевіряє кілька умов, і повертає значення, що відповідає першій умові зі значенням TRUE.

Синтаксис:

= IFS(логічний_вираз1; значення_якщо_істина1; логічний_вираз2; значення_якщо_істина2; ...)

Якщо жодна умова не виконується, функція повертає помилку #Н/Д. Щоб цього уникнути, можна додати останню умову TRUE із бажаним значенням.



Потрібно визначити словесну оцінку залежно від значення в клітинці A1:

=IFS(A1>90; "Відмінно"; A1>80; "Добре"; A1>70; "Задовільно"; TRUE; "Незадовільно")

Якщо A1>90, поверне "Відмінно".

Інакше, якщо A1>80, поверне "Добре".

Інакше, якщо A1>70, поверне "Задовільно".

Інакше (якщо всі попередні умови хибні) поверне "Незадовільно".



Розгляньмо приклад аналізу значень температури.



Потрібно надати коментар щодо погодних умов залежно від значення у клітинці B1:

=IFS(B1<0; "Мороз"; B1<15; "Прохолодно"; B1<25; "Тепло"; TRUE; "Спекотно")

Наведемо стислий опис ще кількох логічних функцій.

Назва	Опис	Кількість аргументів
IFERROR	Якщо перший аргумент не є помилкою, то повертає його значення, інакше повертає значення другого аргументу	Два
FALSE	Повертає значення «хибність»	Без аргументів
TRUE	Повертає значення «істина»	Без аргументів

Прототип Excel (першу електронну таблицю VisiCalc) створив у 1979 році студент Гарвардського університету Ден Бріклін за допомоги Боба Френкстона. Тепер Бріклін знаний американський програміст, відомий як «батько табличних процесорів».



Питання для самоперевірки

1. Що таке функція; аргумент функції?
2. Назвіть декілька категорій функцій в Excel.
3. Чи існують в електронних таблицях функції без аргументів?
4. Як позначають операції відношень?
5. Що є першим аргументом логічної функції IF?



Вправа 14

- За даними концентрації PM2.5 (дрібні пилові частинки) у повітрі для кількох міст визначити рівень якості повітря.
1. Запустіть табличний процесор. Створіть таблицю з даними за зразком (рис. 14.7).

	A	B	C
1	Місто	PM2.5 (мкг/м³)	Якість повітря
2	Місто 1	18	
3	Місто 2	55	
4	Місто 3	112	
5	Місто 4	250	
6	Місто 5	35	

Рис. 14.7

2. За допомогою функції IF у стовпці C виведіть оцінку якості повітря за умовними рівнями: «допустимий» ($PM2.5 < 75 \text{ мкг/м}^3$) і «високий» ($PM2.5 \geq 75 \text{ мкг/м}^3$).
- ⚙ 3. У стовпці D за допомогою логічної функції IFS виведіть оцінки концентрації за чотирма рівнями: 1) менше 35,4 → добра; 2) від 35,5 до 55,4 → помірна; 3) від 55,5 до 150,4 → шкідлива; 4) більше 150,5 → дуже шкідлива.
4. Збережіть файл із назвою Вправа 14.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111806



§ 15

МАТЕМАТИЧНІ ТА СТАТИСТИЧНІ ФУНКЦІЇ

Багато функцій в електронних таблицях дають змогу автоматизувати обчислення, швидко опрацьовувати й аналізувати масиви даних. Завдяки їм виконуються складні математичні, логічні операції та автоматично перераховуються результати при зміні вхідних даних.

Математичні функції



Математичні функції в електронній таблиці — це функції, призначені для виконання обчислень над даними клітинок.

Із деякими математичними функціями ви ознайомилися раніше.

Повний перелік таких функцій можна переглянути на вкладці **ФОРМУЛИ** → **Бібліотека функцій** → **Математичні**.

Стисло розглянемо призначення деяких математичних функцій:

Назва	Призначення	Кількість аргументів	Приклад
ROUND	Округлення до вказаної кількості цифр після коми	Два	=ROUND(3.14159; 2) → 3.14
MOD	Остача від ділення	Два	=MOD(15; 4) → 3
PRODUCT	Добуток	Не менше одного	=PRODUCT(D1:D3)
ABS	Модуль числа	Один	=ABS(-7) → 7
SQRT	Квадратний корінь	Один	=SQRT(25) → 5
RANDBETWEEN	Ціле випадкове число в заданому діапазоні	Два	=RANDBETWEEN(1; 100)
FACT	Факторіал числа (1*2*3*...*число)	Один	=FACT(6) → 120



Кількість способів, якими можна розставити на полиці 10 книжок, дорівнює факторіалу числа 10: =FACT(10)

Функція SUMIF() — корисний інструмент, який дає змогу підсумовувати в діапазоні значення, що відповідають певній умові.

Синтаксис:

=SUMIF(діапазон_критеріїв; критерій; [діапазон_сумування])



Обчислимо суму комісії за майно (рис. 15.1).

	A	B	C
1	Вартість майна	Комісія	
2	100 000 грн	7 000 грн	
3	200 000 грн	14 000 грн	
4	300 000 грн	21 000 грн	
5	400 000 грн	28 000 грн	
6	Формула	Опис	Результат
7	=SUMIF(A2:A5;">160000";B2:B5)	Сума комісії за майно вартістю більше 160 000 грн	63 000 грн
8	=SUMIF(A2:A5;">160000")	Сума вартості майна більше 160 000 грн	900 000 грн

Рис. 15.1

Розгляньмо ці формули детальніше.

=SUMIF(A2:A5;">160000";B2:B5) — якщо значення в діапазоні A2:A5 відповідає умові ">160000", до суми додається значення відповідної клітинки з діапазону B2:B5.

=SUMIF(A2:A5;">160000") — аргумент діапазон_сумування опущено, тому до суми додаються значення клітинок із діапазону критеріїв A2:A5, які більші за 160000.

Аргументами одних функцій можуть бути значення, повернуті іншими функціями.

Статистичні функції

Є функції, що допомагають опрацьовувати великі масиви даних.



Статистичні функції в електронних таблицях — це функції, використовувані для аналізу значень діапазонів клітинок.

Статистичні функції зорієнтовані на швидкий розрахунок середніх та інших підсумкових значень, з'ясування взаємозв'язків між змінними.

Проаналізуймо результати дослідження тривалості користування смартфоном на тиждень, яке провели 10 учнів (рис. 15.2). Якщо тривалість користування ввести в діапазон B2:K2, то:

- середній час користування в клітинці B4 знайдемо за формулою `=AVERAGE(B2:K2)`;
- кількість значень, які перевищують середній час — за формулою `=COUNTIF(B2:K2;">"&B4)`.

Зауважмо: у другій формулі в записі критерію рядок ">" і посилання B4 об'єднано оператором &.

	A	B	C	D	E	F	G	H	I	J	K
1	№ учасника	1	2	3	4	5	6	7	8	9	10
2	Час, год/тиждень	5	3	4	5	2	2	4	6	5	3
3											
4	Середній час	<code>=AVERAGE(B2:K2)</code>									
5	К-ть перевищень	<code>=COUNTIF(B2:K2;">"&B4)</code>									

Рис. 15.2

Стисло розгляньмо призначення деяких статистичних функцій:

Назва	Призначення
MIN	Мінімальне значення в діапазоні
MAX	Максимальне значення в діапазоні
AVERAGE	Середнє арифметичне числових аргументів
COUNT	Кількість серед аргументів числових значень
COUNTIF	Кількість клітинок, дані в яких задовольняють певній умові
COUNTIFS	Кількість клітинок, дані в яких задовольняють декільком умовам



Питання для самоперевірки

1. Яке призначення математичних функцій?
2. Яке призначення статистичних функцій?
3. Назвіть кілька статистичних та математичних функцій.
4. Наведіть приклад застосування однієї зі статистичних функцій.
5. Назвіть та опишіть статистичні функції з умовою.

Вправа 15

- Дослідити, якою може бути середня оцінка групи, якщо кількість варіантів відповіді в усіх питаннях однакова, а учасники щоразу вибиратимуть варіант відповіді навмання.
1. Запустіть табличний процесор. Уведіть у таблицю написи, позначені на рисунку синім кольором.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	4	— кількість варіантів відповіді									Номер досліджу:			1
2	3,3	— середній бал												
3														
4	Номер питання	1	2	3	4	5	6	7	8	9	10	11	12	
5	Правильна відповідь	1	2	3	4	1	2	3	4	1	2	3	4	
6														
7	Номер учасника	Відповіді (0 — правильно)												Оцінка
8	1	2	-1	-2	-2	3	2	-1	-2	1	0	-2	-3	1

2. Номери правильних відповідей розрахуйте, як збільшені на 1 остачі від ділення номера питання на кількість варіантів відповіді. Уведіть формулу в клітинку B5, скопіюйте в клітинки C5:M5.
3. У клітинку B8 уведіть формулу: від випадкового числа в межах від 1 до кількості варіантів (A1) відняти номер правильної відповіді на питання 1. Значення 0 означатиме, що відповідь правильна. Скопіюйте формулу в решту клітинок діапазону B8:M11.
4. У клітинках N8:N11 за допомогою функції COUNTIF підрахуйте кількість нулів у відповідних рядках.
5. У клітинці A2 обчисліть середнє арифметичне діапазону N8:N11, округлене до десятих. Збережіть файл із назвою Вправа 15.
6. Після рядка 9 уставте 20 порожніх рядків, виправте номери учасників. Скопіюйте в додані рядки формули з діапазону B9:N9. Змінюючи кількість варіантів відповіді (A1) або номер досліджу (N1), спостерігайте результат досліджу в клітинці A2.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111807



§ 16

УПОРЯДКУВАННЯ ДАНИХ. УМОВНЕ ФОРМАТУВАННЯ

Електронні таблиці мають потужні інструменти для впорядкування та фільтрування великих обсягів взаємозалежної інформації, підбиття підсумків. Тобто надають широкі можливості не тільки для зберігання даних і проведення розрахунків, а й для аналізу даних.

Робота зі списками

Щоб застосувати засоби табличного процесора, призначені для опрацювання й аналізу даних, слід ці дані подати у вигляді списку.



Список в електронній таблиці — це діапазон клітинок зі взаємопов'язаними однотипними даними.

Згадаймо, наприклад, список товарів на складах, розрахункових операцій у магазині, номерів телефонів і адрес абонентів, або гірських вершин України, як на рис. 16.1.

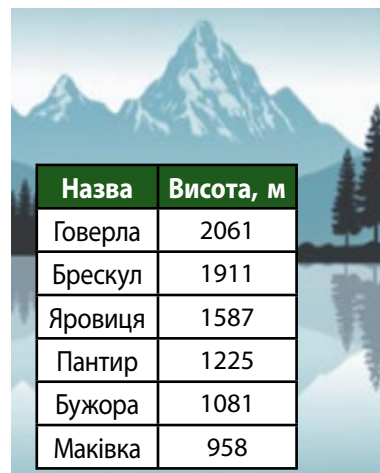
У рядках описуються елементи списку, а в стовпцях — властивості елементів. Зазвичай перший рядок списку містить заголовки стовпців (полів), а кожен наступний рядок — це окремий елемент списку.

При застосуванні команд (сортування, фільтр, підсумки) для опрацювання списків, табличний редактор автоматично розпізнає список і опрацьовує його як базу даних (докладніше про бази даних ви дізнаєтеся в розділі 4).

Список має відповідати **певним вимогам**:

- перший рядок у списку має містити назви стовпців (імена полів);
- кількість стовпців у списку має бути постійною;
- у списку не може бути порожніх рядків і стовпців.

Для зручності опрацювання списку його можна помістити на окремий аркуш. Це дає змогу структурувати дані, застосовувати фільтри, сортування та аналіз.



Назва	Висота, м
Говерла	2061
Брескул	1911
Яровиця	1587
Пантир	1225
Бужора	1081
Маківка	958

Рис. 16.1

Сортування даних

Кожний стовпець списку містить значення певного параметра однотипних об'єктів. Якщо для аналізу даних треба з'ясувати, які об'єкти мають найменші (найбільші) значення певних параметрів, проаналізувати зміни тощо, виконують сортування списку.



Сортування даних в електронній таблиці — це змінення порядку розташування елементів списку за даними певного стовпця (стовпців).

Дані можна сортувати за *зростанням* (текстові — в алфавітному порядку) чи *спаданням* (у зворотному до алфавітного порядку)

Для **негайного сортування даних** за зростанням (спаданням) потрібно клацнути по стовпцю, за даними якого треба відсортувати список, і вибрати тип сортування: ОСНОВНЕ → Редагування → Сортувати й фільтрувати (рис. 16.2) або ДАНІ → Сортування й фільтр.

Якщо перед сортуванням виділити діапазон зі списком, то елементи списку відсортуються за даними 1-го стовпця, порожні клітинки якого розташуються внизу незалежно від типу сортування.

Якщо ж виділити один стовпець, то дані можна буде відсортувати лише в ньому або в усьому діапазоні за даними виділеного стовпця (рис. 16.3, 16.4).

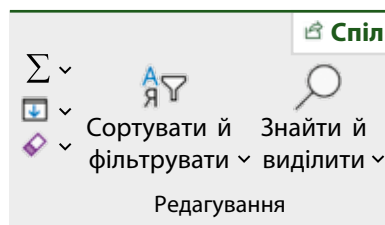


Рис. 16.2

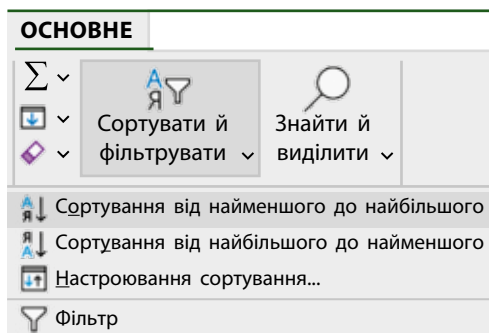


Рис. 16.3



Рис. 16.4



Відсортуємо дані про продаж туристичною фірмою путівок (рис. 16.5, а). Отримаємо таблицю, наведену на рис. 16.5, б.

Країна	Місто	Кількість
Україна	Одеса	3
Україна	Буковель	54
Туреччина	Кемер	20
Єгипет	Хургада	34
Туреччина	Анталія	86
Україна	Львів	20
Єгипет	Шарм Ель Шейх	7

а

Країна	Місто	Кількість
Єгипет	Хургада	34
Єгипет	Шарм Ель Шейх	7
Туреччина	Анталія	86
Туреччина	Кемер	20
Україна	Одеса	3
Україна	Буковель	54
Україна	Львів	20

б

Рис. 16.5

Для **сортування даних за кількома стовпцями** слід виконати дії: ДАНІ → Сортування й фільтрування → Сортування, або ОСНОВНЕ → Редагування → Сортувати й фільтрувати → Налаштування сортування. В обох випадках буде відкрито діалогове вікно Сортування.

У діалоговому вікні Сортування вибирають поля, за якими відбувається сортування. Перше з вибраних полів визначає порядок первинного сортування. Якщо за збігу значень у цьому полі потрібно відсортувати елементи за іншим стовпцем, то слід додати рівень сортування кнопкою **+ Додати рівень** і вибрати цей стовпець зі списку заголовків.

На кожному рівні сортування може виконуватися в порядку зростання чи спадання (рис. 16.6).

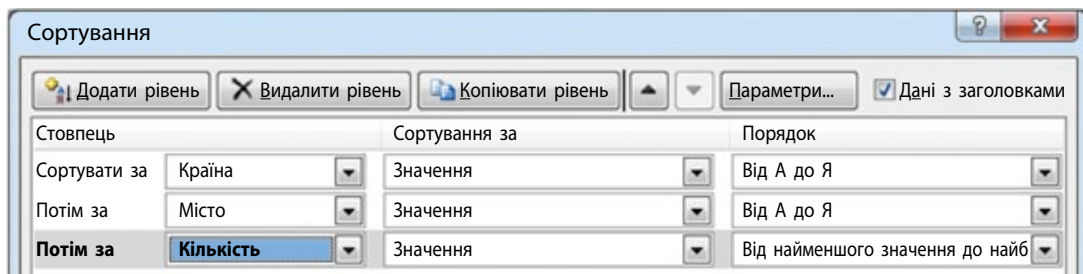


Рис. 16.6

Фільтрування даних

Список може містити дуже багато елементів (у табличному процесорі природною межею слугує максимальна кількість рядків робочого аркуша), тому не завжди треба відображати всі елементи.

 **Фільтрування даних** — це відображення в таблиці тільки тих рядків, дані яких відповідають певним умовам (умовам фільтрування).

При фільтруванні решта рядків таблиці не знищується, а лише приховується й може бути відображена за інших умов.

Вмикають режим фільтрування кнопкою Фільтр на вкладці ДАНІ (див. рис. 16.4). Після цього для кожного поля списку буде автоматично створено набір стандартних фільтрів, доступних у розкривних списках. Кнопки списків відображаються біля заголовка кожного стовпця.

За промовчанням використовується варіант Усі, який вказує, що елементи списку повинні відображатися без фільтрування. Якщо потрібно відобразити елементи, певний параметр яких має конкретне значення, його слід вибрати зі списку значень.

 Таблиця-довідник туристичної фірми містить стовпці з даними про доступні маршрути: номерами записів, назвами країн і назвами міст. Відфільтруємо рядки таблиці, в яких значення поля Країна дорівнює Україна (рис. 16.7).

3	№ ▾	Країна ▾	Місто ▾
4	1	Україна	Одеса
5	2	Україна	Буковель
9	6	Україна	Львів

Рис. 16.7

Як бачимо, у стовпці № відсутні деякі номери, що свідчить про фільтрування даних у таблиці. Також звернемо увагу, як змінилася кнопка розкривного списку в стовпці Країна. Оскільки фільтрування відбулося за цим стовпцем, його позначено значком лійки. Завдяки цьому видно, в яких стовпцях необхідно змінити налаштування фільтрування, щоб повернути початковий вигляд списку.

Скасувати дію фільтра можна повторним натисканням кнопки Фільтр.

Якщо необхідно відобразити елементи, певний параметр яких задовольняє деякій умові, слід вибрати команду Фільтри чисел (для параметрів рядкового типу — Текстові фільтри), після чого в спадному меню вибрати тип умови (рис. 16.8, с. 98).



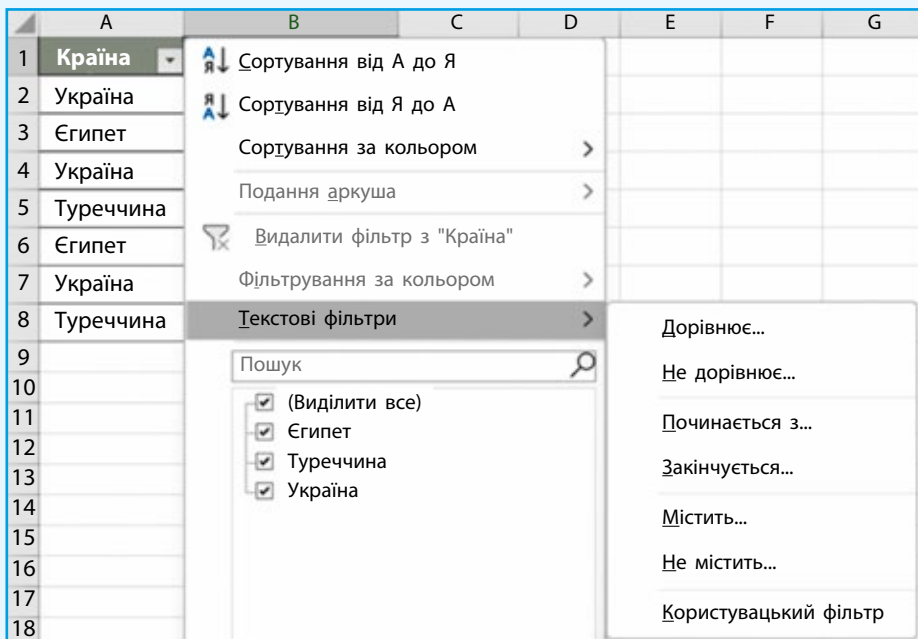


Рис. 16.8

Для стовпців із числовими даними активним буде пункт Фільтри чисел. Можна відмітити варіант Перші 10 і відобразити 10 найбільших (найменших) значень у списку; варіант вище середнього дає змогу відібрати значення, які перевищують середнє значення стовпця.

У вікні Користувачий автофільтр за допомогою логічних операцій можна задати особливі умови фільтрування (рис. 16.9).

Користувачий автофільтр

Показати лише ті рядки, значення яких:
Протяжність в Україні, км

більше

☒ І ☐ АБО

менше

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



Рис. 16.9



3

Дано таблицю протяжності річок на території України. Нам потрібно відобразити дані про річки, протяжність яких в Україні перевищує 500 км. Результат наведено на рис. 16.10.

	A	B	C
1			
2	Назва річки	Куди впадає	Протяжність в Україні, км
5	Псел	Дніпро	717
6	Дністер	Чорне море	705
7	Сіверський Донець	Дон	672
8	Горинь	Прип'ять	659
9	Десна	Дніпро	591
10	Ігулець	Дніпро	549

Рис. 16.10

Умовне форматування

У 7 класі ви детально ознайомилися із засобами умовного форматування, до якого вдаються для унаочнення даних. Пригадаймо, що умовне форматування даних — це автоматичне змінення формату клітинок залежно від значення даних у них.

Щоб виконати умовне форматування, слід виділити діапазон і вибрати потрібне в списку: **ОСНОВНЕ** → **Стилі** → **Умовне форматування**.

Умовне форматування надає можливість оформити клітинки за стандартними правилами або встановити свої. До тих самих клітинок можна застосовувати кілька правил умовного форматування.

Якщо ж правила суперечать одне одному, то «сильнішим» буде правило, встановлене пізніше. Наприклад, при заповненні червоним кольором клітинок зі значеннями «більше 5» та зеленим — клітинок зі значеннями «менше 7» клітинки зі значенням «шість» зафарбуються зеленим кольором (рис. 16.11, с. 100).

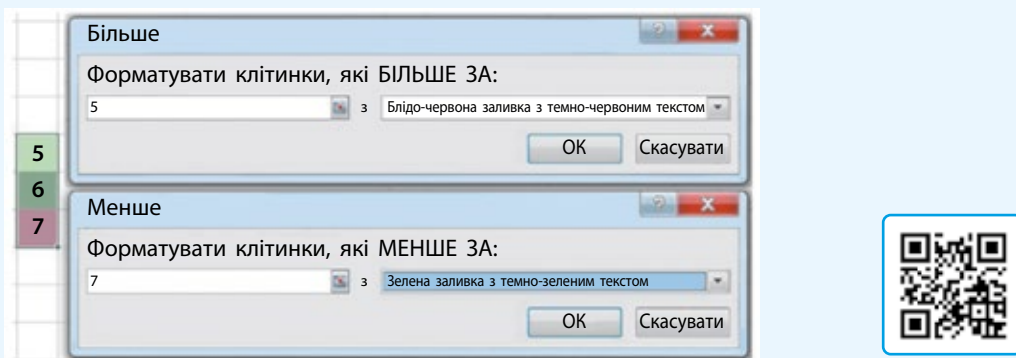


Рис. 16.11

Усі ці інструменти дають змогу перетворити неопрацьовані дані на зрозумілу й упорядковану інформацію. Використовуючи їх разом, можна значно підвищити ефективність аналізу даних, унаочнити їх, щоб зрештою ухвалити обґрунтоване рішення. За QR-кодом (див. вище) або посиланням rnk.com.ua/114800 можна переглянути відеоролік із прикладом роботи з Диспетчером правил умовного форматування.



Питання для самоперевірки

1. У чому полягає сортування даних?
2. Як відсортувати список за даними кількох стовпців?
3. У чому полягає фільтрування даних?
4. Як відбувається фільтрування даних за кількома стовпцями?
5. Що відбувається, коли правила умовного форматування даних суперечать одне одному?
6. Які рядки списку буде відображено, якщо у вікні Користувацький автофільтр умову задано в такий спосіб, як наведено на рис. 16.12.

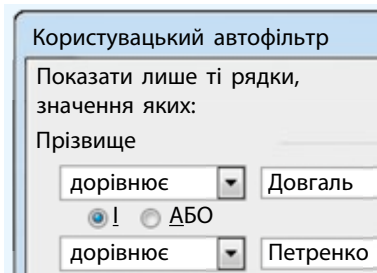


Рис. 16.12

Японець Тацуо Хоріучі (Tatsuo Horiuchi) після виходу на пенсію почав малювати за допомогою... Microsoft Excel! Вже кілька десятиліть він створює картини, пронизані традиціями, культурними мотивами.

Вправа 16

- Здійснити умовне форматування та впорядкування даних.
- Відкрийте табличний процесор і створіть нову робочу книгу. Заповніть таблицю даними, дотримуючись зразка (рис. 16.13). Налаштуйте зовнішній вигляд таблиці. Для цього об'єднайте потрібні клітинки й установіть параметри форматування рядка за головоків стовпців.

	А	В	С
1	Продаж путівок		
2	Країна	Місто	Кількість
3	Україна	Одеса	3
4	Єгипет	Шарм Ель Шейх	7
5	Україна	Львів	20
6	Туреччина	Кемер	20
7	Єгипет	Хургада	34
8	Україна	Буковель	54
9	Туреччина	Анталія	86

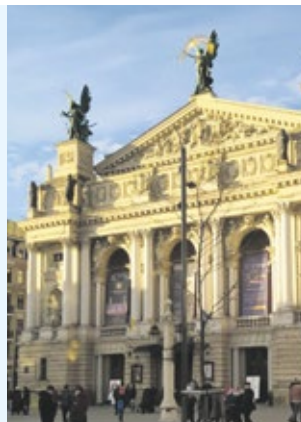


Рис. 16.13

- Упорядкуйте рядки таблиці так, щоб назви країн розташовувались в алфавітному порядку.
- Додайте сортування за стовпцем В в алфавітному порядку.
- Налаштуйте умовне форматування клітинок стовпця С:
 - значення, менші за 10 — фон червоного відтінку;
 - значення від 11 до 40 — фон жовтого відтінку;
 - значення, більші за 40 — фон зеленого відтінку.
- Додайте до таблиці фільтри й відобразіть тільки рядки з даними про путівки до міст України.
- Збережіть файл із назвою Вправа 16.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111808





ПРАКТИЧНА РОБОТА 6

ВИКОРИСТАННЯ ФУНКЦІЙ ТАБЛИЧНОГО ПРОЦЕСОРА. УМОВНЕ ФОРМАТУВАННЯ

Завдання: створити таблицю для обліку продуктів на кухні та планування витрат продуктів для приготування страви за рецептом (див. рисунок). Забезпечити:

- повідомлення в разі нестачі продуктів до нормативу та для приготування страви (умовне форматування);
- розрахунок кількості назв, вартості закупівлі кожного продукту та всіх разом, які потрібно докупити до нормативу;
- підрахунок кількості й вартості продуктів, які потрібно докупити, щоб приготувати страву.

Обладнання: комп'ютер із табличним процесором.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть табличний процесор і створіть нову робочу книгу.
2. **Створення розділу Запаси.** Внесіть у рядки 1:18 дані, дотримуючись зразка (діапазон F4:F18 не заповнюйте).
3. У клітинці B4 налаштуйте умовне форматування, щоб колір тла змінювався, якщо значення в ній менше, ніж у клітинці D4.
4. Застосуйте формат клітинки B4 до клітинок діапазону B5:B16.
5. У клітинку F4 введіть формулу з використанням функції IF: якщо значення в клітинці B4 більше, ніж у клітинці D4, то вивести слово «вистачає», інакше — вивести вартість тієї кількості продукту, яку потрібно докупити.
6. Скопіюйте формулу в клітинки F5:F16.
7. У клітинці F17 розрахуйте загальну вартість потрібних продуктів.
8. У клітинці F18 виведіть кількість чисел у діапазоні F4:F16.
9. **Створення розділу Рецепт.** Внесіть у рядки 19:26 дані, дотримуючись зразка (окрім клітинок C21:D25 і D26).
10. У клітинку C21 введіть формулу
`=SUMIF(A$4:A$16; A21; B$4:B$16)-B21`
 Поміркуйте, що саме в ній розраховується.
 У клітинку D21 введіть формулу
`=IF(C21<0;-C21*SUMIF(A$4:A$16;A21;E$4:E$16);"")`

Поміркуйте, як саме тут відбувається розрахунок.

11. Скопіюйте формули з клітинок C21:D21 у наступні чотири рядки. До клітинок C21:C25 додайте умовне форматування (див. рисунок).
12. У клітинці D26 розрахуйте загальну вартість продуктів, яких не вистачає для приготування страви. Збережіть файл із назвою Практична робота 6. Перевірте, як змінюються результати обчислень, якщо змінювати вхідні дані.

	A	B	C	D	E	F
1	Облік продуктів					
2	Запаси					
3	Продукт	Кількість	Од. вимір.	Норматив	Ціна	Вартість
4	борошно	20	кг	15	25,00 грн	вистачає
5	сіль	1,3	кг	1	22,00 грн	вистачає
6	цукор	2	кг	3	27,00 грн	27,00 грн
7	крупа гречана	0,4	кг	1	47,00 грн	28,20 грн
8	крупа рисова	2,2	кг	2	50,00 грн	вистачає
9	макарони	1,9	кг	1	30,00 грн	вистачає
10	фарш	4	кг	5	270,00 грн	270,00 грн
11	м'ясо куряче	3,2	кг	3	210,00 грн	вистачає
12	яловичина	5	кг	5	350,00 грн	вистачає
13	перець	0,1	кг	0,1	250,00 грн	вистачає
14	сода	0,7	кг	1	80,00 грн	24,00 грн
15	олія	0,6	л	1	80,00 грн	32,00 грн
16	оцет	1,2	л	1	42,00 грн	вистачає
17	Сума закупівлі до норми:					381,20 грн
18	Кількість назв:					5
19	Рецепт					
20	Складник	Потреба	Нестача	Вартість		
21	борошно	25	-5	125,00 грн		
22	фарш	8	-4	1080,00 грн		
23	сіль	0,1	1,2			
24	перець	0,01	0,09			
25	олія	0,3	0,3			
26	Сума закупівлі для рецепта:			1205,00 грн		

Зробіть висновок: функції яких типів використано в роботі; яка з них має найбільше аргументів.

§17

РОЗШИРЕНІ ФІЛЬТРИ. ПРОМІЖНІ ПІДСУМКИ

Автоматичні фільтри придатні для відносно простих випадків аналізу даних. Якщо до даних потрібно застосувати складні логічні вирази (складні умови), використовують розширені фільтри.

Розширені фільтри

Розгляньмо застосування розширених фільтрів на прикладі.



Є потреба відобразити успішність учнів і учениць, які мають більше від 9 балів з алгебри або інформатики. Застосування фільтрів з умовою (>9) до стовпців Алгебра, Інформатика дасть змогу відобразити лише рядок балів Петренка, тоді як заданим критеріям відповідають рядки: Іванова, Петренко, Сидоров (рис. 17.1).

	A	B	C	D	E	F
1	Прізвище	Алгебра	Хімія	Фізика	Інформатика	Середній бал
2	Іванова	7	10	11	12	10
3	Петренко	12	11	8	10	10,3
4	Сидоров	11	9	8	9	9,3
5	Тесленко	9	8	9	7	8,3



Рис. 17.1

Щоб налаштувати розширений фільтр, потрібно:

- 1) скопіювати в окреме місце заголовки стовпців, за якими відбуватиметься фільтрування (наприклад, Алгебра, Інформатика);
- 2) у клітинки під скопійованими заголовками ввести критерії фільтрування (наприклад, >9);
- 3) вибрати ДАНІ → Сортування й фільтрування → Додатково;
- 4) у вікні Розширений фільтр задати прапорці й діапазони (рис. 17.2).

При цьому критерії, об'єднані логічною операцією OR, указуються в різних рядках, а об'єднані логічною операцією AND — в одному. Після натискання ОК у діапазоні для результату з'являться елементи списку, які відповідають умовам фільтрування.

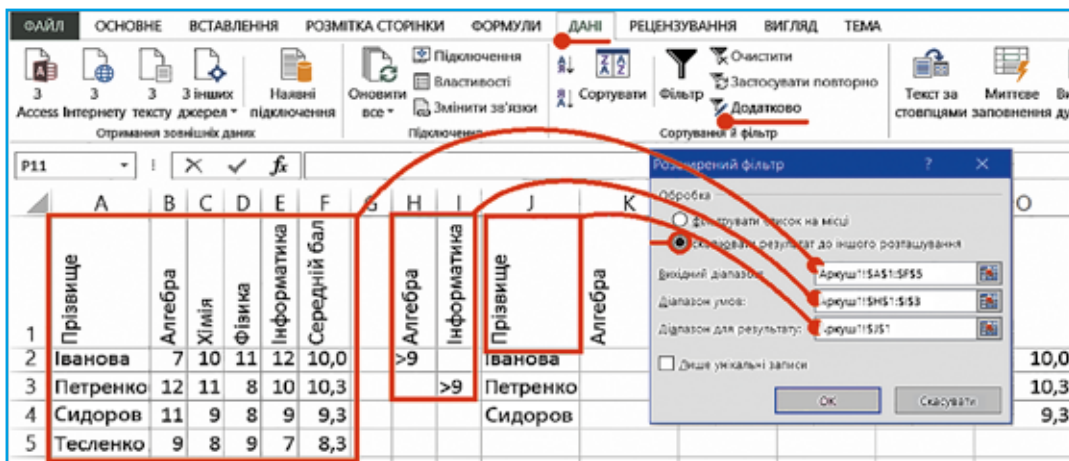


Рис. 17.2

Проміжні підсумки

Підбиття проміжних підсумків — це ефективний інструмент для обробки й аналізу даних списку. Він дає змогу автоматично додавати до списку рядки, що відображають підсумкові значення за певними групами записів. Це спрощує подальший аналіз, візуалізацію даних за допомогою діаграм і підготовку звітів до друку.

Як і раніше, для коректного створення підсумків необхідно, щоб перший рядок у списку містив назви стовпців, а інші рядки — однотипні дані без порожніх рядків.

Для **розрахунку проміжних підсумків** потрібно:

- 1) відсортувати список за тим стовпцем, за яким підбиватимуться підсумки;
- 2) на вкладці ДАНІ в групі Структура вибрати параметр Проміжні підсумки (рис. 17.3, 17.4, с. 106);
- 3) у полі При кожній зміні вибрати стовпець, що містить групи, за якими необхідно підбити підсумки; це повинен бути той стовпець, за яким проводилося сортування списку на кроці 1;

- 4) у полі Використовувати функцію вибрати підсумкову функцію для обчислення проміжних підсумків;
- 5) у полі Додати підсумки до: вибрати стовпець, за даними якого буде розраховано підсумки;
- 6) зняти прапорець Замінити поточні підсумки й натиснути ОК.

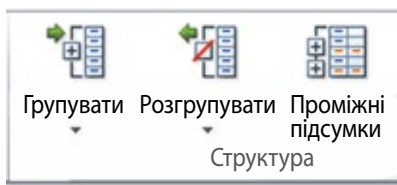


Рис. 17.3

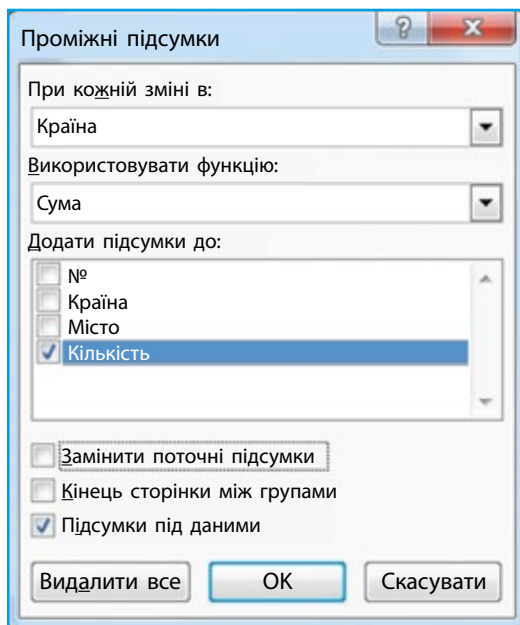
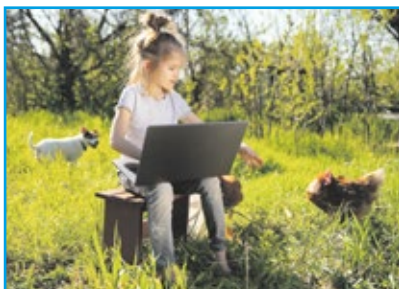


Рис. 17.4



Визначимо кількість путівок, проданих туристичною фірмою до кожної країни (рис. 17.5).

На рисунку подано фрагмент таблиці після підбиття проміжних підсумків за стовпцем Країна. Дані автоматично структуровано, завдяки чому добре видно потрібну інформацію.

Під кожною групою елементів за стовпцем Країна відображено узагальнене значення у стовпці Кількість. Ліворуч від номерів рядків з'явилися позначки структури (кнопки). Структура дає змогу приховати й відобразити рівні деталізації натисканням кнопок із зображенням значка «-» або «+» відповідно.

Для перегляду або приховування детальних даних у структурі слід клацати мишею по символах рівня рядка або стовпця (1 2 3).

1	2	3	3	№	Країна	Місто	Кількість
		+	4	4	Єгипет	Хургада	34
		+	5	7	Єгипет	Шарм Ель Шейх	7
		-	6		Єгипет Підсумок		41
		+	7	3	Туреччина	Анталія	86
		+	8	5	Туреччина	Кемер	20
		-	9		Туреччина Підсумок		106
		+	10	1	Україна	Одеса	3
		+	11	2	Україна	Буковель	54
		+	12	6	Україна	Львів	20
		-	13		Україна Підсумок		77
		-	14		Загальний підсумок		224

Рис. 17.5

Структура може мати до 8 рівнів деталізації, в яких кожен рівень забезпечує детальну інформацію для попереднього рівня. Так на рис. 17.6 рядок, що містить загальний підсумок для всіх рядків, має рівень 1, рядки, що містять підсумки для кожної країни, мають рівень 2, а конкретні дані для міст мають рівень 3.

Після підбиття підсумків виводиться інформація для нижчого рівня структури. Стосовно розглянутого прикладу — це рівень 3; інформація для цього рівня деталізації й відображена на рис. 17.6.

	3	№	Країна	Місто	Кількість
+	6		Єгипет Підсумок		41
+	9		Туреччина Підсумок		106
+	13		Україна Підсумок		77
-	14		Загальний підсумок		224

Рис. 17.6

Для відображення окремо рядків на певному рівні достатньо клацнути кнопку з номером рівня деталізації, який потрібно переглянути. Щоб видалити з таблиці підсумкову інформацію, нам треба виділити всю таблицю й у вікні Проміжні підсумки клацнути кнопку Прибрати все.



Питання для самоперевірки

1. Коли застосувати автоматичний фільтр не можна?
2. Як налаштувати розширений фільтр?
3. Як у розширеному фільтрі записують умови, об'єднані AND?
4. Коли доцільно використовувати проміжні підсумки?
5. Чому їх слід підбивати за відсортованим стовпцем?



Вправа 17

- Створити таблицю із застосуванням розширеного фільтра та проміжних підсумків за описом.
1. Запустіть табличний процесор і заповніть таблицю за зразком (рис. 17.7). Відформатуйте написи та межі таблиці.

	A	B	C
1	Тип ПК	Виробник	Обсяг продажів
2	Ноутбук	Asus	7000
3	Стаціонарний	Sony	5000
4	Ноутбук	Lenovo	30000
5	Стаціонарний	Lenovo	100000
6	Стаціонарний	Acer	7500
7	Планшет	Acer	10000
8	Ноутбук	Acer	150000
9	Ноутбук	Apple	15000



Рис.17.7

2. Застосуванням розширеного фільтра створіть нову таблицю, що міститиме дані про стаціонарні комп'ютери або такі, обсяг продажів яких перевищив 10 000.
3. Обчисліть сумарні обсяги продажів комп'ютерів кожного типу.
4. Приховуйте дані початкової таблиці, залишивши проміжні підсумки продажів комп'ютерів за типами. Збережіть файл Вправа 17.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111809



§ 18

ДІАГРАМИ ТА ГРАФІКИ

Помітною тенденцією останніх років стало застосування інфографіки в різних сферах діяльності, у тому числі в інформатичній галузі. Інфографіка покращує сприйняття, підвищує можливості людини бачити моделі й тенденції.



Інформаційна графіка, або інфографіка, — це графічне візуальне подання інформації, даних або знань, призначене для швидкого й чіткого відображення комплексної інформації.

Інфографіка є одним зі способів викладення інформації за допомогою графіків, діаграм тощо на папері або в електронному вигляді.

Діаграми та їх типи

Одним із засобів табличного процесора, який надає потужні можливості графічної візуалізації табличних даних, є діаграми.

Існує декілька основних типів діаграм. Вибір конкретного типу залежить від того, як саме діаграма відображатиме дані.

Кругова (секторна) діаграма (рис. 18.1) наочно показує співвідношення між частинами.

Для побудови такої діаграми використовують числа, розміщені в одному рядку або стовпці. Приміром, скільки відсотків людей із порушенням зору користуються цифровими послугами.

Гістограму (стовпчикову діаграму) (рис. 18.2) використовують для показу змін одного чи кількох показників за певний проміжок часу.

Гістограми краще ілюструють числові дані, розміщені одночасно в кількох рядках і стовпцях. Приміром, порівняння кількості проданих мобільних пристроїв людям різних вікових категорій по місяцях.

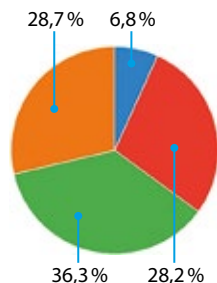


Рис. 18.1

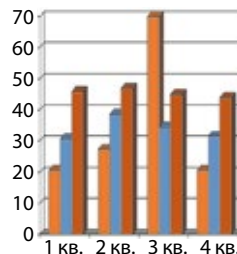


Рис. 18.2

Лінійний графік (рис. 18.3) варто використовувати для відображення змін показників *протягом певного часу*. Приміром, як змінювалася температура протягом дня, або ціна акцій протягом року.

Точкову діаграму використовують для побудови графіків математичних функцій, заданих таблично. Для неї потрібно не менш ніж два ряди даних (один ряд — аргумент, а другий — значення функції).

Якщо після побудови діаграми змінити дані таблиці, відповідні об'єкти діаграми теж зміняться.

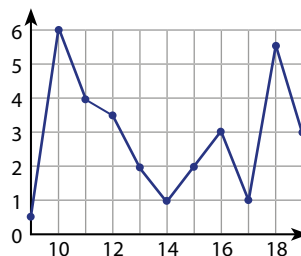


Рис. 18.3

Побудова діаграми

Ознайоммося, які елементи містить діаграма (рис. 18.4).



Рис. 18.4

Тут: 1 — назва діаграми;

2 — область побудови діаграми;

3 — лінії сітки;

4 — легенда;

5 — маркер даних;

6 — назва осі X.

Для **створення діаграми** потрібно:

- 1) заповнити (створити) таблицю даних;
- 2) виділити табличні дані разом із назвами рядків (стовпців);
- 3) перейти на вкладку ВСТАВЛЕННЯ, у групі Діаграми вибрати тип.

Після цього з'являються нові вкладки — Конструктор діаграм і Формат. Вони містять інструменти, що дають змогу змінити заголовок діаграми; додати підписи осей (що показує вісь X, а що вісь Y) та даних (числа біля стовпців чи секторів); змінити кольори, шрифти, стилі; додати легенду (пояснення, що означає кожен колір або символ).

Біля вибраної діаграми з'являються кнопки швидкого налаштування, якими можна змінити тип діаграми, зовнішній вигляд тощо. Ознайомся з їх призначенням.

 Елементи	Вмикання/вимикання назви, легенди, підписів даних
 Стилi	Налаштування стилю та колірної гами
 Фільтри	Налаштування відображення даних та назв

- Щоб **перемістити** діаграму на інше місце аркуша, слід перетягнути фоновий прямокутник.
- Щоб **розмістити** діаграму на окремому робочому аркуші, слід вибрати команду Перемістити діаграму, що є в контекстному меню.
- Щоб **змінити розміри** діаграми (її елементів), треба виділити (клацнути) потрібний елемент і перетягти відповідні маркери зміни розміру. Сектори кругової діаграми, наприклад, можна роз'єднати, перетягуючи їх мишею, за допомогою контекстного меню — змінити колір заливки секторів, їхні контури, розміри, рельєфи та ін.

Графіки

Розгляньмо хід побудови графіка функції на прикладі.



Побудуємо на координатній площині графік функції $y=2\sqrt{|x|}$, заданої таблично на проміжку $[-1; 2]$ із кроком $h=0,2$.

1. Запустимо Excel і створимо таблицю. Упишемо в клітинках A1, B1, C1 назви величин h (крок), x , y (координати). У клітинку A2 введемо крок зміни координати $x(0,2)$.
2. У клітинку B2 введемо перше значення координати $x=-1$.
3. У клітинку B3 введемо формулу, що до поточної координати x додаватиме значення кроку $=B2+\$A\2 .
4. Скопіюємо формулу клітинки B3 у решту клітинок для координати x , доки не отримаємо кінцеве значення діапазону 19.
5. У клітинку C2 введемо формулу розрахунку значення функції $=2*\text{SQRT}(\text{ABS}(B2))$.
6. Скопіюємо формулу для значення функції в решту клітинок.
7. Установимо для клітинок зі значеннями функції числовий формат із двома цифрами після коми.

8. Маємо таблицю аргументів (x) і значень (y) функції $y=2\sqrt{|x|}$ на проміжку $[-1; 2]$ із кроком $h=0,2$. Виділимо всі дані в рядках 2 і 3 (разом із назвами x і y), виберемо ВСТАВЛЕННЯ → Діаграми та зазначимо тип Графік (рис. 18.5).

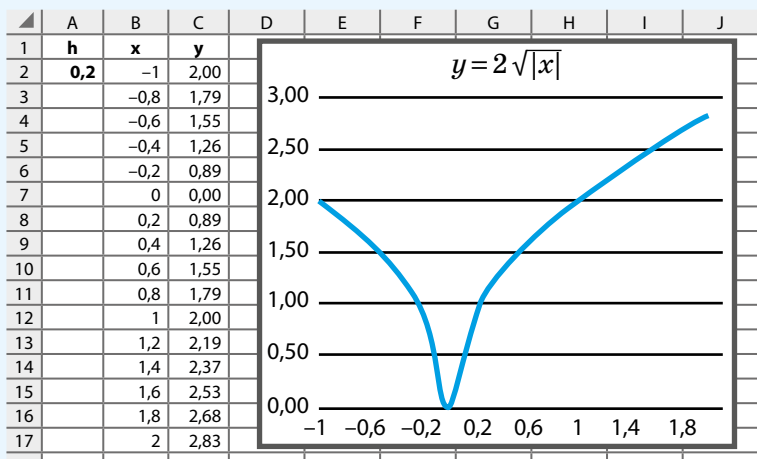


Рис. 18.5

► Мінідіаграми та їх призначення

Якщо кожен із багатьох рядів даних потрібно візуалізувати окремо, то класичні діаграми виявляються громіздкими й незручними. Та вже в Excel 2010 з'явився зручний засіб — мінідіаграми (спарклайни), які будуються в окремих клітинках робочого аркуша.

Мінідіаграма — елемент, що дає змогу візуалізувати ряд чисел, щоб унаочнити їх співвідношення й тенденції.

На рис. 18.6 відображено динаміку продажів оргтехніки за місяцями (мінідіаграми — у правому стовпці).

Обсяг продажу по місяцях					
Назва товару	січень	лютий	березень	квітень	травень
Ноутбук ASUS Eee PC X101CH White	15000 грн	23400 грн	30420 грн	39546 грн	51410 грн
Нетбук Acer Aspire One D270-26Cws) 10.1" White	18000 грн	28080 грн	36504 грн	47455 грн	61692 грн
Моноблок Apple iMac A1311 21.5"	216000 грн	33698 грн	43805 грн	56946 грн	74030 грн
Планшет Apple New iPad (iPad 3) Wi-Fi+4G 64GB	131104 грн	140435 грн	152566 грн	168335 грн	188836 грн
Монітор 17"	37350 грн	58366 грн	63079 грн	82003 грн	106603 грн
Принтер лазерний	53780 грн	69876 грн	30834 грн	118084 грн	153509 грн

Рис. 18.6

Excel підтримує три типи мінідіаграм.

Для **побудови мінідіаграми** потрібно:

- 1) занести дані в таблицю й вибрати клітинку для мінідіаграми;
- 2) на вкладці ВСТАВЛЕННЯ в групі Міні-діаграми вибрати тип (рис. 18.7);
- 3) у вікні Створення міні-діаграм указати діапазон даних, для якого будується діаграма;
- 4) скопіювати діаграму в потрібний діапазон.

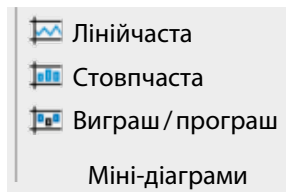


Рис. 18.7

Отже, використання діаграм допоможе нам зробити доповідь про проєкт або дослідження значно цікавішою та переконливішою.



Питання для самоперевірки

1. Які типи діаграм вам відомі? Наведіть приклади.
2. Коли доцільно застосовувати кругову діаграму?
3. Коли доцільно застосовувати стовпчикову діаграму?
4. Які основні кроки побудови діаграм?
5. Як до таблиці додати стовпець мінідіаграм?



Вправа 18

■ Побудувати графік функції $y = \frac{x}{a}$.



1. Запустіть табличний процесор. У клітинку A1 уведіть значення a.
2. У клітинки A2 і B2 уведіть заголовки x та $y = x/a$.
3. Заповніть діапазон A3:A23 значеннями x від -10 до 10 із кроком 1.
4. У клітинку B3 уведіть формулу $=A3/\$A\1 і скопіюйте її вниз до кінця списку значень x.
5. Виділіть діапазон A3:B23. На вкладці ВСТАВЛЕННЯ в групі Діаграми виберіть Точкова з прямими лініями.
Додайте назву діаграми, підписи осей.
6. Змініть число в клітинці A1. Збережіть файл із назвою Вправа 18.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/111810





ПРАКТИЧНА РОБОТА 7

УПОРЯДКУВАННЯ ДАНИХ У ТАБЛИЦЯХ. АВТОМАТИЧНІ ТА РОЗШИРЕНІ ФІЛЬТРИ

Завдання: упорядкувати дані таблиці за різними умовами й правилами.

Обладнання: комп'ютер із табличним процесором.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть табличний процесор і створіть таблицю.
2. Відформатуйте написи за зразком (див. таблицю).

Ціна, \$	Марка автомобіля	Кількість	Ціна, \$	Марка автомобіля	Кількість
12 000	КрАЗ	100	11 000	Мерседес	200
15 000	Форд	125	10 000	Хонда	225
11 000	Форд	150	12 500	Мерседес	250

3. Відформатуйте межі та заливку клітинок за зразком.
4. Зробіть додатково чотири копії аркуша з таблицею (всього п'ять).
5. На Аркуші 1 відсортуйте дані в таблиці за зростанням ціни.
6. На Аркуші 2 відсортуйте рядки, щоб марки автомобілів розташувалися в алфавітному порядку, а кількість — у порядку спадання.
7. На Аркуші 3 увімкніть автоматичні фільтри для всіх стовпців.
8. На Аркуші 3 налаштуйте фільтр відповідного стовпця на відображення рядків тільки марок автомобілів «Форд» і «Мерседес».
9. На Аркуші 4 налаштуйте розширений фільтр для відображення даних про автомобілі, дешевші від \$ 12 300 або кількість яких не перевищує 150.
10. На Аркуші 5 налаштуйте умовне форматування тексту «Форд» червоним кольором.
11. На Аркуші 5 налаштуйте умовним форматуванням сіру заливку клітинок із цінами, меншими за \$ 12 500.
12. Збережіть файл із назвою Практична робота 7.

Зробіть висновок: як упорядкувати дані; які фільтри можна застосувати; для чого використовують умовне форматування.



БАЗИ ДАНИХ. СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ

4

РОЗДІЛ

У цьому розділі ви дізнаєтеся, що таке база даних і система керування базою даних (СКБД), які об'єкти може містити база даних в СКБД Microsoft Access; навчитеся створювати просту базу даних.

§ 19

ЩО ТАКЕ БАЗА ДАНИХ І СИСТЕМА КЕРУВАННЯ БАЗОЮ ДАНИХ

Раніше вважалося, що обсяг даних, який опрацьовує людство, подвоюється кожні 10 років. Нині за оцінками фахівців, обсяг даних подвоюється кожні 2 роки. Як комп'ютер дає змогу їх упорядковувати?

База даних

Кілька тонн безладно вивантажених книжок містять багато інформації, але нею практично неможливо скористатися. Навіть якщо ви впевнені, що в купі є потрібна книжка, то знайти її надзвичайно складно. Пошук суттєво спрощується, коли книжки впорядковано за жанрами та/або авторами й авторками на полицях бібліотеки.



База даних (БД) — упорядкована сукупність взаємопов'язаних даних.

Окрім бібліотеки, базою даних можна вважати журнал успішності, архів, опис майна, матеріалів тощо. Який приклад можете навести ви?

Нині паперова БД поступилася місцем електронній, яка стала загальним позначенням для будь-якої впорядкованої сукупності даних, які зберігаються й опрацьовуються комп'ютером.



Електронна БД — файл або сукупність файлів особливого формату, які містять належним чином структуровані дані, призначені для зберігання, накопичення, опрацювання та використання за допомогою комп'ютера.



Для зберігання даних про телефонні контакти можна створити базу даних у вигляді текстового файлу `phonebook.csv` такої структури:

```
Аліна,0004578111,1:1:2010  
Дмитро,0004654122,1:5:2010  
Аліна,0004789221,8:1:2010
```



Як бачимо, кожен із рядків містить один запис про один телефонний контакт, а саме: ім'я, номер телефону та дату народження. Усі записи зібрано в таблицю, яку й записано в текстовий файл.

Формат файлів CSV (від англ. *Comma-separated values* — значення, розділені комами) призначений для зберігання структурованих текстових даних. Кожен рядок у ньому містить один **запис**, значення **полів** якого відокремлено комами.



Для подальшого використання даних потрібно знати також **назви полів** таблиці. Їх можна зберегти в окремому файлі `phonebook_h.csv`:
Ім'я,Номер телефону,Дата

Розгляньмо на прикладі простої БД, як комп'ютер опрацьовує дані.

Як керують базою даних

Із прикладу 1 зрозуміло, що БД є лише сховищем упорядкованої сукупності даних. Для їх надійного зберігання, редагування та швидкого доступу потрібні додаткові засоби.



Система керування базою даних (СКБД) — комплекс засобів і методів для створення, збереження, оновлення та пошуку інформації в базі даних та керування доступом до неї.

Якщо розглядати бібліотеку як БД, то системою керування такою БД є бібліотечний каталог (рис. 19.1). Він дає змогу стежити за обігом книжок, проводити пошук за назвами чи прізвищами авторів та/або за кодом УДК (універсальний десятиковий код), зазначеним на службовій сторінці книжки. Отже, паперові системи покладаються на фізичні (паперові, друковані) документи, натомість електронні — на програмне забезпечення.



Рис. 19.1

Для роботи з найпростішою електронною БД (див. приклади 1, 2) можна скористатися текстовим процесором. Якщо даних багато, цей спосіб неефективний. Крім того, легко випадково пошкодити дані. Тому для роботи з БД розробляють спеціальні програми.



Електронна СКБД — це комплекс програм для роботи з даними в електронній базі даних.

Отже, робота з БД включає **два напрямки**:

- розробку (проектування структури, конструювання СКБД тощо);
- користування (заповнення даними, пошук і використання даних).

На рис. 19.2 наведено логотипи популярних електронних СКБД.



Рис. 19.2

Для роботи з простою БД із прикладу 1 можна скласти програму.

3

Складемо програму мовою Python, після запуску якої пропонується вибір операції над даними, а після вибору відповідна операція виконується. Основна програма може бути такою:

```
db_file = "phonebook.csv"
op = input("Виберіть операцію:\n1 – показати все\n" +
           "2 – пошук за ім'ям\n3 — додати контакт\n")

if op == '1': show_all()
elif op == '2': show_by_name(input("Уведіть ім'я: "))
elif op == '3':
    n = input("Уведіть ім'я: ")
    t = input("Уведіть номер телефону: ")
    d = input("Уведіть дату народження: ")
    add_contact(n, t, d)
```

Зрозуміло, що для виконання операцій над даними потрібно створити функції, згадані в коді:

- `show_all` — для перегляду всіх записів БД;
- `show_by_name` — для пошуку контактів за ім'ям;
- `add_contact` — для додавання нового контакту до файлу БД.

Функція `show_all` для перегляду всіх записів має відкрити файл і по чергово прочитати й опрацювати кожен його рядок. Завдяки тому, що дані в рядку відокремлено комами, їх легко розділити (пригадайте метод `split`) і вивести в зручному для читання вигляді.

4

Код функції `show_all` для перегляду всіх записів БД.

```
def show_all():
    f = open(db_file, "r")           # Відкриваємо файл для читання
```

```
for k in f:                                # Почергово читаємо рядки з файлу
    record = k.split(',')
    print("Контакт:", record[0]) # Виводимо у зручному вигляді
    print(" тел.:", record[1], " нар.:", record[2])
f.close()                                  # Закриваємо файл
```

Функція `show_by_name` для пошуку контактів за іменем також має послідовно читати рядки із файлу. Для кожного з рядків потрібно перевірити, чи ім'я контакту збігається з шуканим, і якщо так, то вивести номер телефону. У найпростішому варіанті дату народження, яка читається з файлу разом з іншими даними, можна знехтувати.



Код функції `show_by_name` для пошуку в БД записів за іменем контакту.

```
def show_by_name(name):
    f = open(db_file, "r")

    print("Результати пошуку:", name)

    for k in f:                                # Почергово читаємо рядки з файлу
        record = k.split(',')
        if record[0] == name:                  # Якщо ім'я збігається, ...
            print(" ", record[1], end = " ")    # ... то виводимо номер
    f.close()
```

Функція `add_contact` має відкрити файл БД для дозаписування, потім із даних про контакт сформував рядок і записати його у файл. Надалі код функції можна доповнити, наприклад, перевіркою допустимості дати та/або формату номера.



Код функції `add_contact` для додавання в БД даних нового контакту.

```
def add_contact(name, number, date):
    f = open(db_file, "a")    # Відкриваємо файл для дозаписування
    f.write('\n' + name + ';' + number + ';' + date)
    f.close()
    print('Контакт', name, 'додано.')
```

Отже, завдяки СКБД дані захищено від випадкового пошкодження. Далі ви побачите, що СКБД дають змогу працювати з такими обсягами даних, які практично неможливо опрацювати вручну.

З історією розвитку СКБД ви можете ознайомитися за QR-кодом або посиланням rnk.com.ua/114854.



Питання для самоперевірки

1. Чи містить купа безладно вивантажених книжок інформацію (дані) та чи можна назвати її базою даних? Чому?
2. Наведіть приклади паперових баз даних.
3. Яке призначення має система керування базою даних?
4. Які СКБД вам відомі?
5. Поясніть, як виконується програма з прикладу 3.
6. Для чого у функції `show_by_name` виконується перевірка `"if record[0] == name:"`?
7. Як відкриває файл команда `"f = open(db_file, "a")"` із прикладу 6?



Вправа 19

- Випробувати та доповнити просту СКБД Телефонні контакти.
1. Створіть базу даних у файлі `phonebook.csv` (див. приклад 1).
 2. Створіть програму мовою Python у файлі `dbms.py` для виведення меню СКБД (приклад 3). Випробуйте програму.
 3. Доповніть програму функціями для перегляду, пошуку за іменем і додавання контакту в БД (приклади 4–6).
Випробуйте кожну з функцій.
Переконайтеся, що дані нових контактів з'являються у файлі БД.
 4. Створіть файл із заголовками `phonebook_h.csv` (приклад 2).
Доповніть функцію `show_all` так, щоб угорі таблиці виводилися заголовки стовпців.
 5. Створіть функцію `show_by_number` для пошуку в БД записів за номером телефону. Доповніть меню СКБД відповідним пунктом.
 - ⚙ 6. Доповніть функцію `add_contact` так, щоб перед записом у файл виконувалася перевірка, чи там ще немає такого контакту.
Перевірте роботу програми.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112939



Щоб розпочати застосовувати комп'ютер у будь-якій галузі, потрібно налагодити зберігання оцифрованої інформації, яку ми маємо намір ним опрацьовувати. Оскільки інформація може бути дуже різномірною, то існують різні підходи до організації баз даних.

Види баз даних

База даних, із якою ми ознайомилися раніше, придатна лише для найпростіших випадків. Сучасні задачі зберігання та доступу до даних потребують складніших організованих баз даних.

Залежно від того, як організоване зберігання даних, розрізняють ієрархічні, мережеві, реляційні, об'єктно-орієнтовані, гібридні та інші бази даних. Розгляньмо основні з них.

► Ієрархічна база даних

Ієрархічну БД схематично зображено на рис. 20.1. Кожен інформаційний об'єкт, що містить певні дані, в ній називають **вузлом**; між вузлами є **зв'язки**.

Один із вузлів — головний (на рисунку позначений як *0-й рівень*), а інші розташовані на *нижчих рівнях*.

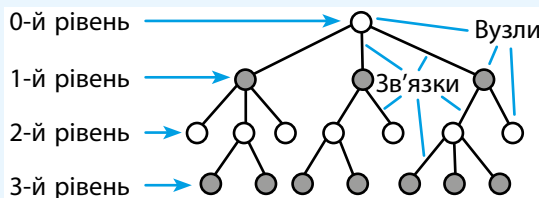


Рис. 20.1

! В ієрархічній БД кожен вузол нижчого рівня має зв'язок тільки з одним вузлом вищого рівня.

Прикладом ієрархічної системи може бути навчальний заклад, відповідне дерево тощо. Який приклад можете навести ви?



Ієрархічну структуру має файлова система комп'ютера (рис. 20.2). Операційна система, яка виконує функції СКБД, дає змогу створювати, змінювати, знаходити файли, папки тощо.



Рис. 20.2

► Мережева база даних

Мережева модель БД — це сукупність об'єктів різного рівня, де кожен об'єкт може бути зв'язаний з іншими.

Складниками мережевої моделі БД, подібно до ієрархічної, є рівень, вузол і зв'язок. Проте в ній можливі довільні зв'язки між елементами сусідніх рівнів, тобто елемент нижчого рівня може мати зв'язок із декількома елементами свого або вищого рівня.



Дані про клієнтів банків можуть зберігатися в БД різних банків і бути пов'язаними між собою (рис. 20.3).

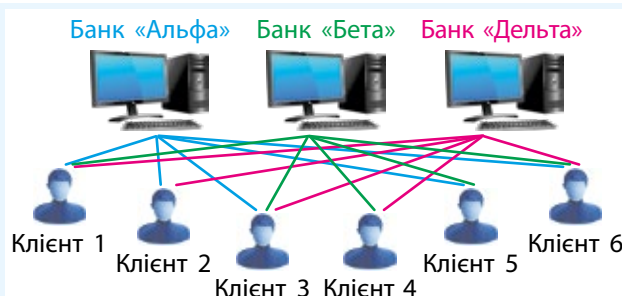


Рис. 20.3

► Реляційна база даних

Основою **реляційної БД** є сукупність таблиць, між якими існують зв'язки. Пошук даних у такій БД зводиться до аналізу зв'язків, автоматичного утворення нових таблиць та пошуку в них даних.



Сторінки журналу успішності можна розглядати як реляційну БД (рис. 20.4), де кожен запис про оцінку пов'язаний із таблицями зі списками учнів/учениць і навчальних предметів.

Інформатика, 1 група		Облік навчальних											
(Назва предмета)													
№ п/п	Прізвище та ім'я учня (учениці)	Місяць і число											
		05/08	12/08	19/08	26/08	03/09	10/09	17/09	24/09	01/10	08/10	15/10	22/10
1	Андрущенко Замис			8			10	9					
2	Беласенко Ярослав Н			8			11	10					
3	Галичків Олег			8			11	10					

Рис. 20.4

Основи роботи саме з реляційними БД докладно розглянемо далі.

► Об'єктно-орієнтована та гібридна бази даних

В **об'єктно-орієнтованих** БД дані зберігаються у вигляді об'єктів, наділених певними властивостями й можливостями (методами) взаємодії з іншими об'єктами. У таких БД зручно встановлювати між об'єктами складні взаємозв'язки (рис. 20.5).

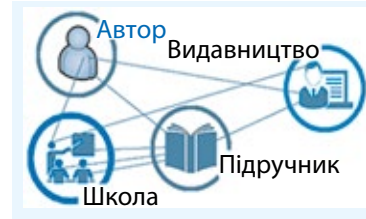


Рис. 20.5

Гібридні БД поєднують можливості реляційних і об'єктних БД, тому їх часто називають об'єктно-реляційними.

Попри різноманіття видів БД, нині найпоширенішими є реляційні. У їх основі лежить так звана модель «сутність — зв'язок».

Модель «сутність — зв'язок»

На початку проектування БД необхідно виконати такі дії:

- 1) проаналізувати предметну галузь (школа, телефонна книга, магазин, автомайстерня та ін.);
- 2) виділити **сутності**, дані про які зберігатимуться;
- 3) з'ясувати, які між ними є **зв'язки**.

Результатом буде модель «сутність — зв'язок» предметної галузі.



Модель «сутність — зв'язок» — це схематичне відображення змістового зв'язку об'єктів майбутньої бази даних.

Наприклад, сутностями можуть бути клас, оцінка, книга, учитель/учителька, співробітник/співробітниця, відділ, підприємство.

Сутність у моделі позначмо прямокутником, у якому, крім назви сутності, можуть бути записані її характеристики — **атрибути**. На рис. 20.6 зображено сутність Учень, яка має атрибути Прізвище та Ім'я.



Зв'язок описує залежність між двома сутностями; його можна охарактеризувати дієсловом.

На малюнку зв'язки позначмо підписаними лініями. Наприклад, на рис. 20.7 виділено зв'язок «Учень має оцінки».

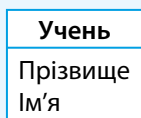


Рис. 20.6

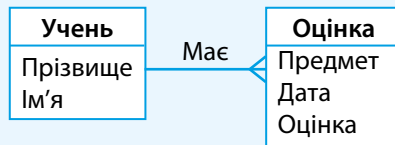


Рис. 20.7

Розгляньмо три основні типи зв'язків між сутностями:

Тип зв'язку	Приклад	Позначення
Один-до-одного	Кожен учень має один щоденник і кожен щоденник належить одному учню	—
Один-до-багатьох	Один учень має багато оцінок, проте кожна оцінка належить одному учню	—>
Багато-до-багатьох	Кожен учень вивчає багато предметів і кожен предмет вивчають багато учнів	>—>

На основі моделі «сутність — зв'язок» будують структуру реляційної електронної БД.

Структура реляційної бази даних

Реляційна БД — це сукупність зв'язаних таблиць, більшість з яких містять інформацію про об'єкти — екземпляри певних сутностей.

4

Таблиця Предмет містить інформацію про навчальні предмети, а таблиця Учень — про учнів і учениць. Таблиця Оцінка пов'язує між собою таблиці Учень і Предмет, тобто описує зв'язок між відповідними двома сутностями.

Рядок таблиці містить дані про один об'єкт (наприклад, про ученицю Бабич Вікторію), а стовпці — значення відповідних атрибутів усіх об'єктів (наприклад, прізвище, ім'я, дату народження).



У таблиці реляційної БД рядки називають **записами**, а стовпці — **полями** (рис. 20.8).

Поля

	№ з/п	Прізвище	Ім'я	День народження
*	1	Іваненко	Андрій	12.12.2009
*	2	Петрина	Максим	01.03.2009
*	3	Сидорук	Сергій	02.05.2009
*	4	Бабич	Вікторія	28.02.2009

Запис: 1

Рис. 20.8

Поле таблиці реляційної БД містить дані одного типу. Запис містить сукупну інформацію про конкретний об'єкт. Навіть порожня БД містить певну інформацію, але не про конкретні об'єкти, а про предметну галузь у цілому (назви таблиць, імена полів, опис типів даних).

Отже, структура реляційної БД визначає зв'язки між таблицями й можливі типи та властивості даних у них.

Проектування реляційної бази даних

Проектування БД починають із визначення її структури:

- 1) визначають перелік даних, які будуть зберігатися, та зв'язки між ними (тобто будують модель «сутність — зв'язок»);
- 2) визначають кількість і призначення таблиць;
- 3) для кожної з таблиць визначають назви полів, їх типи, властивості.

Вдало спроектована БД потребує менше зусиль для подальшого використання та підтримки, оскільки в неї неможливо внести суперечливі дані та зручно отримувати потрібну інформацію.



Не зовсім раціональною «базою даних» є паперовий журнал успішності, у якому на кожній сторінці повторюються прізвище та ім'я учня/учениці. Щоб уникнути повторень, слід «зв'язати» таблицю з оцінками, з таблицею, що містить прізвища. Для цього частину сторінок можна обрізати так, щоби було видно список учнів/учениць на першій сторінці (рис. 20.9). Але цього не варто робити, оскільки клас може ділитися на групи, діти можуть вибувати та ін.

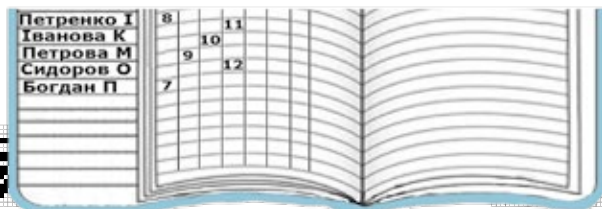


Рис. 20.9

Повторення даних у таблиці знижує надійність електронної БД у цілому: помилившись під час введення, наприклад, назви вулиці в одному із записів, під час пошуку ми ризикуємо отримати неповну інформацію про мешканців цієї вулиці. Проблем можна уникнути, застосовуючи кілька зв'язаних таблиць замість однієї великої.



Проектуючи електронну БД Журнал успішності, виділимо сутності Учень, Предмет і Оцінка, пов'язані, як показано на рис. 20.10. Відповідно, створимо таблиці з такими самими назвами. Завдяки цьому кожна назва шкільного предмета буде записана один раз у таблиці Предмет, особисті дані кожної особи — один раз у таблиці Учень. А інформацію про те, хто і з якого предмета отримав оцінку, буде збережено завдяки зв'язкам (про це — далі).

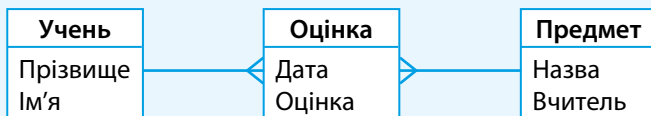


Рис. 20.10



Якщо для кожної сутності виділено окрему таблицю, то це зменшує ймовірність помилки під час введення даних.



Питання для самоперевірки

1. Опишіть особливості реляційної бази даних.
2. Що означає термін «модель “сутність — зв’язок”»?
3. Наведіть приклади сутностей для предметної галузі «Шкільна бібліотека».
4. Які є типи зв’язків між сутностями в реляційній базі даних?
5. Якого типу зв’язок між сутностями Видавництво і Книга; між сутностями Автор і Книга?
6. Опишіть процес проєктування реляційної бази даних.



Вправа 20

- Проаналізувати моделі та об’єкти баз даних.
- 1. Створіть на хмарному диску (наприклад, в обліковому записі Google) документ, у якому зберігатимете відповіді.
Назвіть документ своїм прізвищем та ім’ям, додавши до назви номер вправи (наприклад, Сумлінна Ліна Вправа 20). Надайте доступ учителю/вчительці для редагування документа.
- 2. Запишіть у документ назву, автора (авторів), видавництво, місце й рік видання книжки, наданої вчителем/вчителькою. Знайдіть у книжці код УДК і запишіть у документ.
- 3. Знайдіть в інтернеті таблицю УДК, завантажте та/або відкрийте її. Визначте за знайденою таблицею й запишіть класифікаційну назву, що відповідає кодам УДК книжки.
- 4. Запишіть (зобразіть) по одному прикладу (крім наведених у підручнику) зв’язків сутностей БД різних типів: один-до-одного, один-до-багатьох, багато-до-багатьох.
- 5. У довільному графічному редакторі зобразіть ієрархічну модель структури вашої школи до рівня класів. Додайте малюнок у документ.
- ⚙️ 6. Зобразіть модель «сутність — зв’язок» для предметної галузі Шкільна бібліотека, яка включає не менше трьох сутностей. Додайте малюнок у документ.



Комп’ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112940



Однією із електронних систем керування базами даних (СКБД) є Microsoft Access (далі Access). Файл, який створює Access (із розширенням .accdb), містить таблиці, запити, форми, звіти тощо.

Спроекуємо БД Клас за моделлю «сутність — зв'язок», наведеною на рис. 20.10, с. 126. Структуру таблиць цієї БД показано на рис. 21.1. (У прикладах використано версію Microsoft Access 2016.)

Таблиця Учень		Таблиця Предмет		Таблиця Оцінка	
Назва поля	Тип даних	Назва поля	Тип даних	Назва поля	Тип даних
Номер учня	Число	Код предмета	Число	Номер учня	Число
Прізвище	Текст	Назва	Текст	Код предмета	Число
Ім'я	Текст	Учитель	Текст	Дата	Дата
Дата народження	Дата			Оцінка	Число
Має ООП	Так/Ні			Примітка	Текст
а		б		в	

Рис. 21.1

Як бачимо, структура таблиць відрізняється від моделі «сутність — зв'язок» наявністю полів Номер учня і Код предмета. Пізніше ми їх використаємо, щоб налаштувати зв'язки між таблицями.

Створення порожньої бази даних

Електронну БД Access можна створити на основі шаблону або вручну.



Розглянемо алгоритм створення порожньої БД. Для цього потрібно:

- 1) запустити Access; вибрати Пуста настільна база даних (рис. 21.2);
- 2) у вікні, що з'явиться (рис. 21.3), увести назву БД (у нас: Клас);
- 3) за необхідності кнопкою Знайти розташування для бази даних змінити розташування файлу БД; клацнути значок Створити.

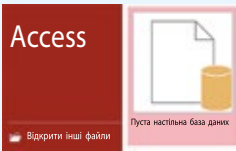


Рис. 21.2

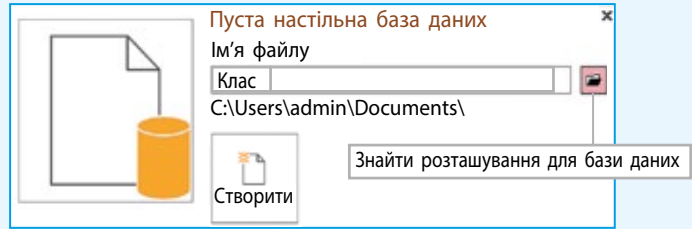


Рис. 21.3

Створена БД міститиме порожню таблицю Таблиця 1, яка матиме єдине поле з назвою Ідентифікатор.

Ознайоммося з інтерфейсом Access, подібним до інтерфейсу знайомих вам офісних застосунків (рис. 21.4).

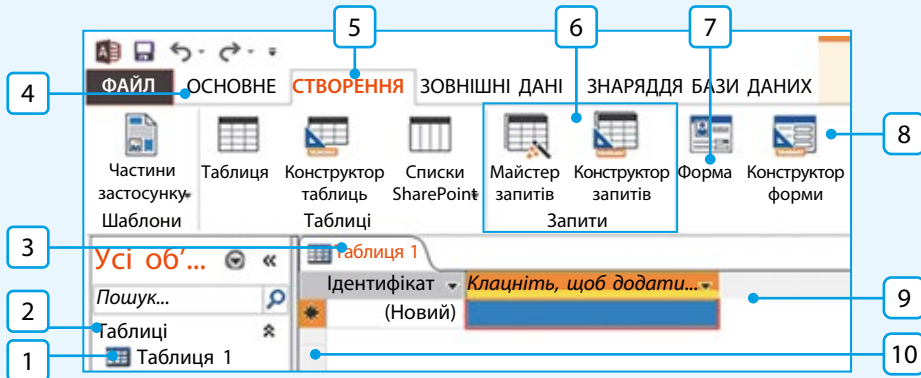


Рис. 21.4

Тут: 1 — значок об'єкта; 2 — область переходів між об'єктами Access; 3 — вкладка об'єкта; 4 — вкладки меню Access; 5 — активна вкладка СТВОРЕННЯ; 6 — група інструментів Запити; 7 — кнопка інструмента; 8 — стрічка інструментів активної вкладки; 9 — робоча область активного об'єкта; 10 — область вибору (запису або поля).

Створення таблиць

Для створення нової таблиці бази даних потрібно перейти на вкладку СТВОРЕННЯ та в групі Таблиці вибрати значок одного з інструментів: Таблиця або Конструктор таблиць.

Кожну таблицю в Access можна опрацьовувати в двох режимах:

- режим конструктора, який призначений для змінювання структури таблиці та вмикається кнопкою або командою Конструктор;
- режим таблиці чи заповнення (вмикається подвійним клацанням на назві таблиці або командою контекстного меню Відкрити).

Змінювати режим таблиці зручно кнопкою Режим, яка на вкладці КОНСТРУКТОР міститься зліва.

► Створення полів у таблиці

Наступним кроком конструювання таблиці є створення її полів.

У режимі конструктора в кожному рядку описують одне поле. У стовпець Ім'я поля записують назву поля.

Назва поля НЕ може:

- бути довшою за 64 символи;
- починатися з пропуску;
- містити крапку (.), знак оклику (!), гравіс ('), квадратні дужки ([]);
- містити керувальні символи (значення ASCII від 0 до 31).

За допомогою контекстного меню поля в режимі конструктора (рис. 21.5) можна додавати поля (рядки), видаляти їх, перейменовувати тощо.

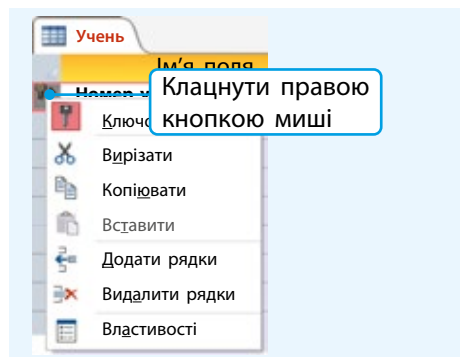


Рис. 21.5

► Типи даних

За промовчанням у режимі конструктора всім полям надається тип Короткий текст. Це незручно, оскільки може виникати необхідність, наприклад, проведення арифметичних операцій над числовими даними тощо. Тому в стовпці Тип даних потрібно вибрати зі списку найоптимальніший тип даних (рис. 21.6). Опис типів даних можна отримати, скориставшись довідкою (клавіша F1 або кнопка «?»).

Учень		
	Ім'я поля	Тип даних
🔑	Номер учня	Автонумерація
	Прізвище	Короткий текст
	Ім'я	Короткий текст
	Дата народження	Дата й час
	Має ООП	Так/Ні



Рис. 21.6

Ознайоммося з деякими типами даних, які стануть у пригоді:

Тип	Призначення
Короткий текст	Зберігання текстів обсягом до 255 символів
Автонумерація	Зберігання натуральних чисел, які створюються автоматично
Число	Зберігання чисел
Дата й час	Зберігання календарних дат і часу
Так/Ні	Зберігання логічних даних

При заповненні таблиці в полі, яке має тип Автонумерація, числа з'являються автоматично, їх неможливо змінити вручну. Залежно від властивостей поля (див. § 22) вони щоразу збільшуються на 1 або генеруються випадково. У разі вилучення записів їхні номери повторно не використовуються, тому не варто застосовувати поле такого типу, як порядковий номер запису.

Вибір оптимальних типів даних дає змогу заощадити ресурси обчислювальної системи, сприяє швидкому опрацюванню даних.

Збереження змін

За допомогою контекстного меню вкладки таблиці (рис. 21.7) можна зберегти зміни, внесені в структуру таблиці.

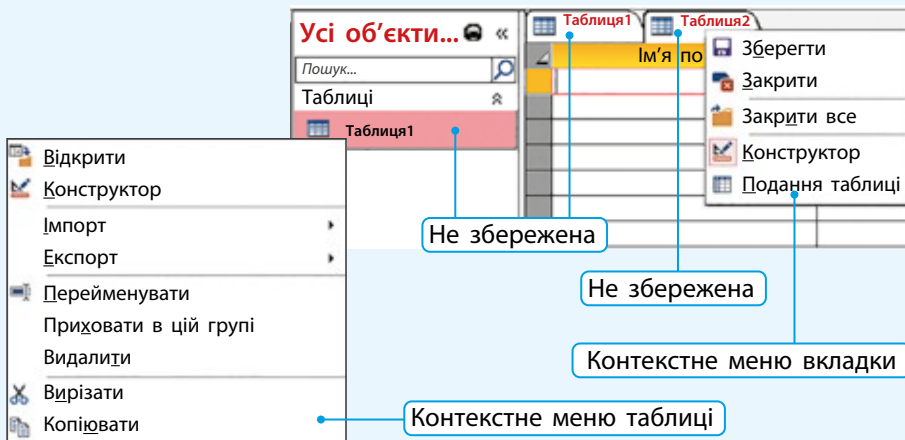


Рис. 21.7

! Таблицю, створену в режимі конструктора, не можна зберегти, доки не буде описано хоча б одне поле (див. Таблиця 2 на рис. 21.7).

Значки та назви збережених таблиць виводяться зліва в Області переходів у полі Усі об'єкти. Таблицю можна перейменувати за допомогою контекстного меню її значка.

2

Створимо в БД Клас (див. приклад 1) таблицю Учень (див. рис. 21.1, а).

- 1) За допомогою контекстного меню таблиці Таблиця 1 відкриємо її в режимі конструктора й уведемо назву Учень.
- 2) Уведемо назви всіх полів таблиці Учень і виберемо для кожного поля тип даних (рис. 21.8). Для поля Номер учня виберемо тип Автонумерація.

Примітка. На початку рядка з описом поля Номер учня автоматично з'явилася позначка ключового поля . Її призначення розглянемо пізніше.

Усі об'єкти... <<		Учень	
Пошук...			
Таблиці			
Учень			
		Ім'я поля	Тип даних
		Номер учня	Автонумерація
		Прізвище	Короткий текст
		Ім'я	Короткий текст
		Дата народження	Дата й час
		Має ООП	Так/Ні

Рис. 21.8

- 3) За допомогою контекстного меню вкладки таблиці Учень збережемо зміни та закриємо вкладку.

Якщо тепер двічі клацнути значок таблиці, то вона відкриється в режимі таблиці. Під час введення даних вона матиме вигляд, як на рис. 21.9.

Учень					
Номер учня ▼	Прізвище ▼	Ім'я ▼	Дата народ ▼	Має ООП ▼	Клацніть, щоб додати ▼
1	Дмитрук	Іванна	01.01.2010	☒	
2	Петренко			☐	
*				☐	

Рис. 21.9



Питання для самоперевірки

1. Яке призначення застосунку Microsoft Office Access?
2. Опишіть порядок створення порожньої бази даних.
3. Як у базі даних створити таблицю?
4. Чому для кожного поля слід вибрати відповідний тип даних?
5. Поясніть, у чому полягає відмінність роботи в режимі таблиці та в режимі конструктора.
6. Опишіть призначення кількох типів даних в СКБД Access.



Вправа 21

- Спроектувати й створити порожню базу даних.
1. Створіть порожню базу даних Клас за алгоритмом із прикладу 1.
 2. Створіть таблицю Учень за алгоритмом із прикладу 2.
 3. У режимі конструктора створіть ще одну порожню таблицю. Керуючись рис. 21.1, б, уведіть назви всіх полів таблиці Предмет і виберіть для кожного поля відповідний тип даних.
 4. Збережіть зміни. У діалоговому вікні зі створення первинного ключа (рис. 21.10) натисніть кнопку Ні.

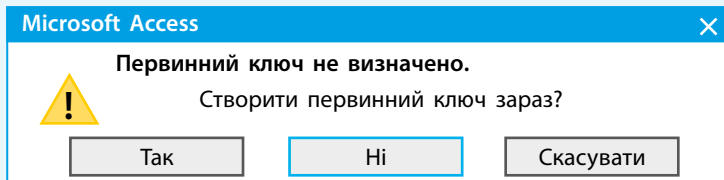


Рис. 21.10

5. Керуючись рис. 21.1, в, створіть таблицю Оцінка. Первинного ключа в ній також не створюйте.
6. Закрийте вікно СКБД Access. При додаткових запитах на збереження чи закриття таблиць відповідайте Так.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112941



§ 22

ЗВ'ЯЗУВАННЯ ТА ЗАПОВНЕННЯ ТАБЛИЦЬ

Таблиці Access, які ми створили, не містять даних. Проте вони й не зовсім порожні, оскільки містять інформацію про властивості даних, які можна в них вносити. Розгляньмо, як налаштувати зв'язки між таблицями та заповнити таблиці даними.

Властивості полів

Кожне поле, залежно від типу даних, має певні властивості, які можна налаштувати після клацання назви типу даних (рис. 22.1). Значення можна вписати в рядок введення власноруч або вибрати з розкривного списку. У вікні справа внизу з'являється опис властивості.

Усі об'єкти...
Пошук...
Таблиці
Оцінка
Предмет
Учень

Ім'я поля	Тип даних
Номер учня	Число
Код предмета	Число
Дата	Дата й час
Оцінка	Число
Коментар	Короткий текст

Властивості поля

Загальні
Підстановка

Розмір поля	Довге ціле число
Формат	Байт
Кількість знаків після коми	Ціле число
Маска вводу	Довге ціле число
Підпис	Одинарне значення
Значення за промовчанням	Подвійне значення
Правило перевірки	Ідентифікатор реплікації
Текст перевірки	Десятькове значення
Обов'язково	Так
Індексовано	Ні
Вирівнювання тексту	Загальне



Рис. 22.1

Ознайоммося з деякими властивостями полів:

Властивість	Опис
Розмір поля	Найбільше значення, яке можна ввести в поле (для текстових типів даних — кількість символів)
Формат	Вигляд даних у таблиці
Маска вводу	Шаблон для полегшення введення даних у таблицю
Значення за промовчанням	З'являється в таблиці автоматично; його можна змінити
Обов'язково	Чи є дане поле обов'язковим для введення в таблицю
Правило перевірки	Обмеження на значення, які вводяться в поле

Маска вводу дає змогу уникнути помилок під час введення і складається зі спеціальних символів. Розгляньмо деякі з них:

Знак (англ.)		Інтерпретація
Заповнення обов'язкове	Заповнення необов'язкове	
0	9	Допускає введення лише десяткових цифр
	#	Допускає введення лише десяткових цифр, пропусків та знаків "+" і "-"
L	?	Допускає введення лише літер
A	a	Допускає введення лише літер і десяткових цифр
&	C	Допускає введення будь-якого символу або пропуску
.: - /		Роздільники (десятковий, тисяч, дат, часу тощо)
<		Вмикає нижній регістр для літерних клавіш
>		Вмикає верхній регістр для літерних клавіш



Якщо в режимі конструктора для короткого формату часу встановити маску вводу 00:00, то під час введення даних у режимі таблиці клітинка виглядатиме так: __:__.



Після введення хоча б однієї цифри потрібно буде обов'язково ввести решту цифр (не плутайте зі властивістю поля Обов'язково).



Налаштувати властивості полів для таблиці *Учень* можна з таких міркувань.

- Навіть подвійні прізвища й імена навряд чи будуть довгими за 30 символів (Розмір поля <30).
- У журналі успішності не може існувати запис із оцінкою без прізвища учня/учениці (Обов'язково).
- Оцінка не може бути більшою від 12 (Розмір поля <13).
- Дату народження вводять відповідно до вибраного формату (Формат; Шаблон).

Первинний ключ

У попередньому параграфі ви вже бачили позначки ключових полів (див. рис. 21.8, с. 132). З'ясуємо, як ключові поля дають змогу створити зв'язки між таблицями.



Первинний ключ таблиці — це одне або декілька полів, значення яких для будь-якого запису є унікальними.

За тим, щоб значення ключа в різних записах відрізнялися, слідкує СКБД.



У таблиці *Учень* поле *Номер учня* автоматично набуває значень, відмінних від інших записів (Автонумерація). Тому саме поле *Номер учня* зручно використати як первинний ключ. Це стосується й поля *Код предмета* в таблиці *Предмети*.

Для створення первинного ключа (ключового поля) слід:

- 1) відкрити таблицю в режимі конструктора;
- 2) вибрати потрібне поле (клацнути мишею будь-де в рядку поля);
- 3) вибрати команду *Ключове поле* в контекстному меню поля (вкладка *КОНСТРУКТОР* у групі *Знаряддя*) (рис. 22.2).

Ліворуч від назви виділеного поля з'явиться позначка, що це поле є ключовим — символ ключа .



Ключове поле

Рис. 22.2



Якщо первинний ключ таблиці складається з декількох полів, то його називають **складеним**.



Прізвища учнів і учениць можуть повторюватися, тож може виникнути ідея для однозначного визначення особи застосувати

складений ключ із полів Прізвище та Ім'я. Але й це не гарантує унікальності даних, адже двоє осіб можуть мати однакові прізвище й ім'я. Гарантовано унікальними, наприклад, є серія та номер паспорта або свідоцтва про народження, номер особової справи учня (учениці) та навіть порядковий номер, який в одному списку не може збігтися для двох осіб.

Для створення складеного ключа слід вибрати кілька полів, утримуючи при цьому клавішу Ctrl, а потім призначити їм властивість Ключове поле.



Якщо учень/учениця з одного шкільного предмета за один день може отримати тільки одну оцінку, то в таблиці Оцінка первинний ключ складатиметься із полів Номер учня, Код предмета, Дата.

Зв'язки між таблицями

Оскільки один учень може отримати декілька оцінок, то кожному запису з таблиці Учень можуть відповідати декілька записів у таблиці Оцінка. Ви вже знаєте, що такий зв'язок називається один-до-багатьох.

Алгоритм установлення зв'язку між таблицями Учень і Оцінка такий:

- 1) Натисніть кнопку Зв'язки на вкладці ЗНАРЯДДЯ БАЗИ ДАНИХ у групі Зв'язки.
- 2) У контекстному меню робочої області вікна Зв'язки виберіть команду Відобразити таблицю або натисніть кнопку Відобразити таблицю на вкладці КОНСТРУКТОР у групі Зв'язки (рис. 22.3).
- 3) У вікні Відображення таблиці (рис. 22.4) послідовно виділяйте назву таблиці й натискайте кнопку Додати.
- 4) Завершіть додавання таблиць кнопкою Закрити.
- 5) Для встановлення зв'язку перетягніть поле Номер учня із таблиці Учень на поле Номер учня в таблицю Оцінка (рис. 22.5, с. 138).



Зв'язки

Рис. 22.3

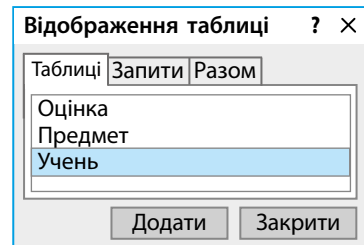


Рис. 22.4



Оскільки поле Номер учня в таблиці Учень є первинним ключем, то цю таблицю називають **головною таблицею** створеного зв'язку. Таблицю ж Оцінка називають **підлеглою таблицею**.

- 6) У вікні Редагування зв'язків установіть прапорці Забезпечення цілісності даних та Каскадне оновлення пов'язаних полів (щоб при зміні значення поля зв'язку в головній таблиці Учень автоматично змінювалося значення у відповідному полі підлеглої таблиці Оцінка) і Каскадне видалення пов'язаних полів (при видаленні запису із головної таблиці видаляються зв'язані дані в підлеглих).
- 7) Натисніть кнопку Створити — між таблицями з'явиться лінія зв'язку двох полів типу один-до-багатьох.

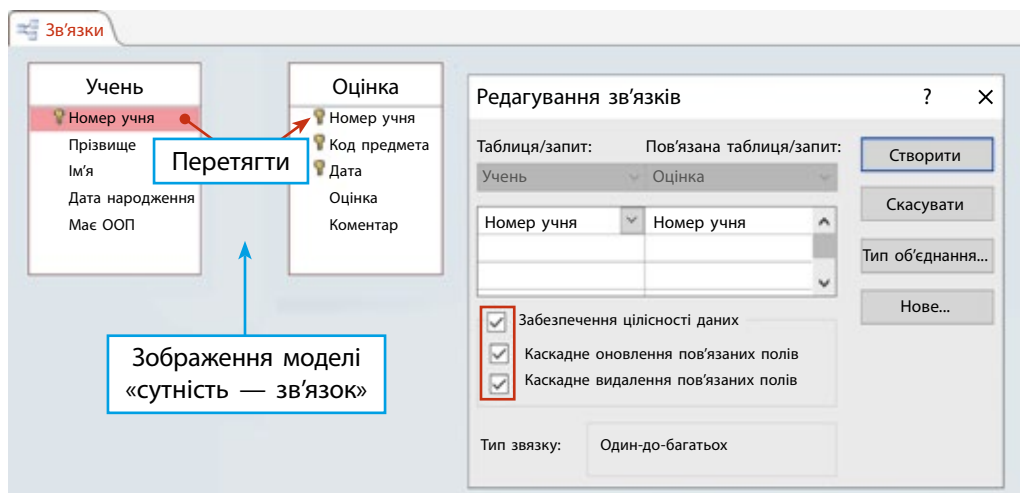


Рис. 22.5

Для збереження схеми зв'язків потрібно в контекстному меню вкладки Зв'язки вибрати команду Зберегти.

Щоб змінити налаштування зв'язку, слід двічі клацнути на лінії зв'язку, а для його видалення — закрити всі таблиці, клацнути по лінії зв'язку та натиснути клавішу Delete.

Уведення даних у режимі таблиці

Найпростіше ввести дані в таблицю в режимі таблиці (рис. 22.6). Для цього потрібно виконати такі дії:

- 1) відкрити таблицю, двічі клацнувши її значок у полі Усі об'єкти;
- 2) увести дані, переходячи до клітинок клавішами зі стрілками, Tab або Enter, або клацаючи мишею клітинку;

- 3) перейти на початок наступного запису, натиснувши Tab або Enter в останньому полі.

Номер учня ▼	Прізвище ▼	Ім'я ▼	Дата народ ▼	Має ООП ▼
1	Дмитрук	Іванна	01.01.2010	☒
2	Петренко			☐
*	(Новий)			☐

Рис. 22.6

! Без заповнення полів, які мають властивість **Обов'язкове**, неможливо ані перейти до заповнення наступного запису, ані видалити поточний запис.

Щоб запис автоматично зберігся, достатньо перейти до іншого запису (зокрема наступного).

Ліворуч від першого поля запису міститься область вибору (позначка 10 на рис. 21.4), де з'являються піктограми, що вказують на стан:

- ▶ — цей запис є поточним (вибраним);
- триває редагування запису.

Підстановки

Якщо поле запису може містити лише значення з обмеженої множини, то ці дані краще не вводити вручну, а вибирати зі списку. Приміром, зручно вибирати зі списку номер групи, клас, колір тощо (рис. 22.7).

У створеній БД Клас заповнювати таблицю Оцінка незручно, оскільки замість прізвищ потрібно набирати їх числові коди. Підстановка дає змогу замість числових кодів бачити в таблиці Оцінка прізвища учнів і учениць зі зв'язаної таблиці Учень.

Розгляньмо алгоритм налаштування підстановки:

- 1) відкрийте таблицю Оцінка в режимі конструктора й для поля Номер учня розгорніть список Тип даних, а в ньому виберіть команду Майстер підстановок (рис. 22.8, с. 140);

Рис. 22.7

Оцінка		
	Ім'я поля	Тип даних
▶	Номер учня	Число
†	Код предмета	Короткий текст
†	Дата	Довгий текст
	Оцінка	Число
	Примітка	Дата й час
		Майстер підстановок

Рис. 22.8

2) у вікні Майстер підстановок клацніть на кнопці Далі (рис. 22.9);

Майстер підстановок

Цей майстер створює поле підстановки, яке відображає список значень, які можна вибрати. Яким чином поле підстановки має отримати значення?

☒ поле підстановки має отримати значення з іншої таблиці або запиту

☐ Я самостійно введу потрібні значення

Скасувати Назад **Далі** Готово

Рис. 22.9

3) у наступному вікні виберіть таблицю Учень і Далі (рис. 22.10);

Майстер підстановок

Яка таблиця або запит має надати значення для поля підстановки?

Таблиця: Оцінки
Таблиця: Предмет
Таблиця: Учень

Подання

☒ Таблиці ☐ Запити ☐ Обидва

Скасувати Назад **Далі** Готово

Рис. 22.10

- 4) у наступному вікні зі списку Доступні поля виберіть Прізвище і клацніть на кнопці «>» та кнопці Далі (рис. 22.11);

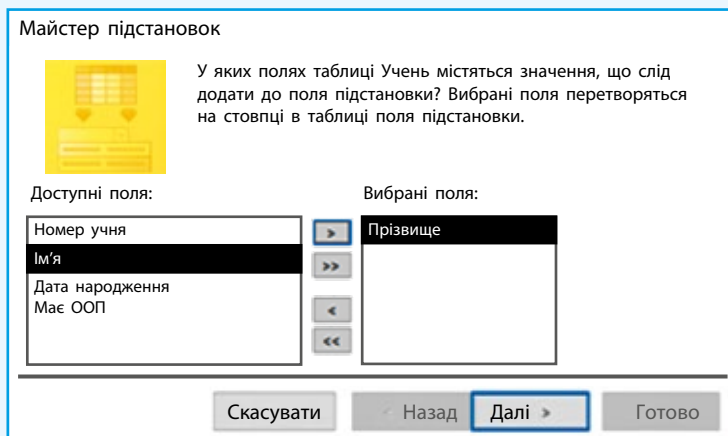


Рис. 22.11

- 5) у наступних вікнах, що з'являтимуться, погодьтеся з усіма налаштуваннями, натискаючи кнопки Далі, Готово, Так.

Тепер у таблиці Оцінка в полі Номер учня замість номерів виводитимуться прізвища із таблиці Учень. Під час заповнення таблиці в цьому полі можна буде вибирати прізвища з розкривного списку.

Під час налаштування підстановки створюється зв'язок між таблицями. Якщо зв'язок за відповідними полями встановлено раніше, то перед налаштуванням підстановки його слід видалити.

Для **редагування наявних підстановок** потрібно:

- 1) відкрити таблицю з підстановкою; відшукати потрібне поле;
- 2) вибрати команду Змінити підстановку контекстного меню поля або натиснути кнопку Змінити підстановку (вкладка ПОЛЯ група Властивості).



Питання для самоперевірки

1. Що визначає властивість Розмір поля?
2. Яке призначення маски вводу: 00:00 для короткого формату часу?
3. Що таке первинний ключ? Коли потрібний складений ключ?
4. Як відкрити таблицю для введення даних?
5. Як розпочати встановлення зв'язків між таблицями?
6. Яке призначення підстановок? Наведіть приклади застосування.



Вправа 22

- Налаштувати властивості полів, зв'язки між таблицями та заповнити таблиці.

1. Запустіть СКБД Access. Відкрийте створену раніше БД Клас.
2. Відкрийте в режимі конструктора таблицю Учень.
3. Налаштуйте властивості полів, спираючись на приклад 2.
4. Зробіть ключовим **№** поле Номер учня. Збережіть таблицю Учень.
- ⚙ 5. Відкрийте в режимі конструктора таблицю Оцінка й налаштуйте для числових типів даних властивості: Розмір поля, Значення за промовчанням, Правило перевірки, Обов'язково.

Підказка. Правилком перевірки для оцінок можна встановити верхню межу: <13.

6. Створіть складений ключ із полів Номер учня, Код предмета й Дата. Збережіть налаштовану таблицю Оцінка.
7. Заповніть таблицю Учень п'ятьма записами та збережіть її.
8. У поле Номер учня (таблиця Оцінка) зробіть підстановку поля Прізвище (з таблиці Учень).
9. У вікні Зв'язки встановіть для зв'язку між таблицями Учень та Оцінка прапорці для забезпечення цілісності даних (див. рис. 22.5).
10. Налаштуйте на свій розсуд таблицю Предмет. Зв'яжіть її з таблицею Оцінка полями Код предмета (головна таблиця — Предмет). Заповніть таблицю п'ятьма записами та збережіть її.
11. У поле Код предмета (таблиця Оцінка) зробіть підстановку поля Назва (з таблиці Предмет). Увімкніть забезпечення цілісності даних.
12. Заповніть таблицю Оцінка десятьма записами та збережіть її. Закрийте (збережіть усі зміни) СКБД Access.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112942



- Microsoft Access, створена компанією Microsoft (США) 1992 року, спочатку призначалася для роботи з великими базами даних. Але вона включає також функції для створення графічного інтерфейсу користувача, тому використовується й для створення застосунків.

§ 23

ЗАПИТИ

Для доступу до інформації, збереженої в БД, призначені особливі об'єкти — запити. Найпростіші запити діють подібно до фільтрів у таблицях — відшукують дані, які відповідають певним критеріям.

Окрім фільтрування даних, запити в БД Access дають змогу автоматично створювати таблиці, додавати, видаляти чи змінювати дані в наявних таблицях тощо (рис. 23.1). На відміну від фільтра в ЕТ, запит може опрацьовувати одночасно декілька об'єктів (таблиць, запитів).



Вибір



Створення
таблиці



Додавання



Оновлення



Перехресний



Видалення

Рис. 23.1



Запит — це засіб відбору та/або опрацювання табличних даних за заданими правилами і критеріями.

Коли ми відкриваємо запит, відбувається пошук інформації в таблицях БД згідно із заданими правилами й критеріями. Результати роботи відкритого запиту виводяться як таблиця. Якщо потрібно, поля та записи впорядковуються й змінюються за вимогами запиту.

Розгляньмо особливості створення запитів деяких типів.

Створення запиту вибору

Створити запит в Access можна за допомогою Майстра запитів або Конструктора запитів. Режим конструктора запитів універсальніший, тож розгляньмо насамперед його.

Пам'ятаймо, що конструювання запиту починається зі створення схеми даних.

Після натискання кнопки Конструктор запитів (рис. 23.2), що на вкладці СТВОРЕННЯ у групі Запити, відкривається вкладка запиту вибору (рис. 23.3, с. 144).

СТВОРЕННЯ



Конструктор
запитів

Рис. 23.2

Вкладка містить поле для схеми даних і бланк запиту. Якщо в ділянці схеми даних немає об'єктів, то відкривається вікно Відображення таблиці, з якого можна додавати об'єкти (таблиці, запити) кнопкою Додати або подвійним клацанням назви об'єкта (див. рис. 23.3).

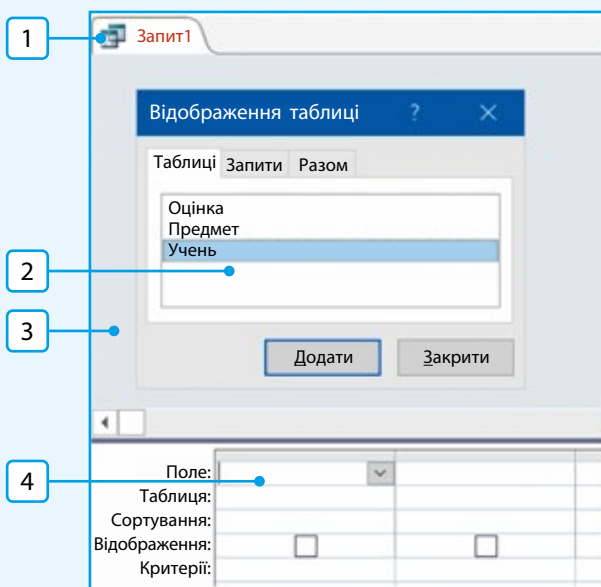


Рис. 23.3

Тут: 1 — вкладка запиту; 2 — вікно додавання таблиць; 3 — ділянка схеми даних; 4 — бланк запиту.

Випадково закрите вікно Відображення таблиці можна відновити відповідною командою контекстного меню ділянки схеми даних.

► Запит вибору зі впорядкуванням (сортуванням) даних

Розгляньмо алгоритм створення запиту вибору даних у БД Клас для виведення таблиці оцінок із зазначенням прізвищ і предметів. Прізвища мають бути впорядковані за алфавітом.

Очевидно, що нам знадобляться всі три зв'язані таблиці, оскільки прізвища зберігаються в одній, оцінки — в другій, а назви предметів — у третій. Отже, виконаємо такі дії.

- 1) Додамо до схеми даних таблиці Учень, Оцінка та Предмети (рис. 23.4). Якщо до схеми даних додано зайву таблицю, клацнемо її й натиснемо клавішу Delete.

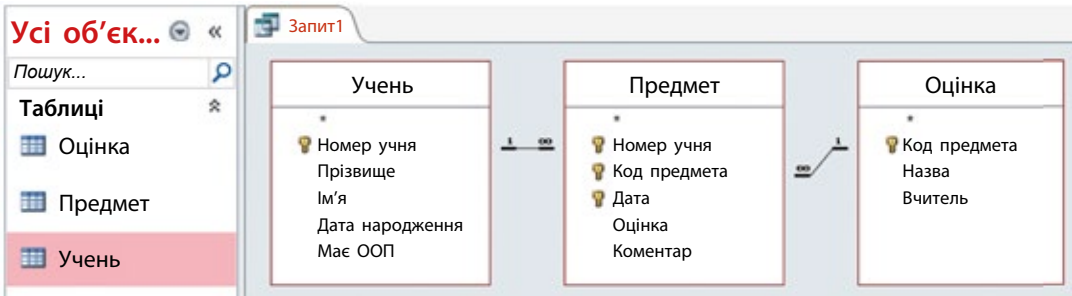


Рис. 23.4

- 2) Якщо з якоїсь причини таблиці не зв'язані, то зв'яжемо їх, перетягнувши ключове поле:
 - Номер учня із таблиці Учень на відповідне поле таблиці Оцінки;
 - Код предмета із таблиці Предмет на відповідне поле таблиці Оцінки.
- 3) У запиті йдеться про прізвища учнів/учениць, оцінки та назви предметів, тож розмістимо назви полів Прізвище, Оцінка, Назва в бланку запиту, двічі клацнувши їх на схемі даних (рис. 23.5). Також назви можна вибрати з розкривних списків у бланку запиту.

Поле:	Прізвище	Оцінка	Назва
Таблиця:	Учень	Оцінка	Предмет
Сортування:	За зростанням		
Відображення:	☒	☒	☒
Критерії:			

Рис. 23.5

! Для видалення з бланку запиту стовпця (поля) його слід виділити, клацнувши над ним при появі стрілки, і натиснути Delete.

- 4) Прізвища в запиті слід упорядковувати за алфавітом, тож у рядку Сортування для поля Прізвище виберемо зі списку значення За зростанням (див. рис. 23.5).
- 5) Залишилося зберегти запит, надавши за необхідності йому нову назву замість стандартної Запит1. Для цього в контекстному меню вкладки запиту виберемо команду Зберегти — у ділянці переходів на панелі Усі об'єкти з'явиться значок із назвою нового запиту.

► Критерії вибору

До запиту можна додати додаткові умови, яким мають відповідати дані. Їх записують для полів, яких вони стосуються, в рядку Критерії бланка запиту.



Щоб отримати з БД Клас список оцінок високого рівня (10, 11, 12) із фізики та хімії, слід до описаного раніше запиту додати критерії, як наведено на рис. 23.6.

Поле:	Прізвище	Оцінка	Назва
Таблиця:	Учень	Оцінка	Предмет
Сортування:	За зростанням		
Відображення:	☒	☒	☒
Критерії:		>9	"фізика"
Або:		>9	"хімія"

Рис. 23.6

Для запису критеріїв використовують оператори порівняння <, >, =, <>, а також логічні оператори and, or, not тощо.

Розгляньмо приклади формування критеріїв відбору текстових даних у запитах вибору:

Критерій	Результат відбору	Критерій	Результат відбору
Петренко	Лише Петренко	K*	Починається з літери K
Not Петренко	Усі, окрім Петренко	*K	Закінчується літерою K
Іванова and Петренко	Лише Іванова і Петренко	*K*	Містить літеру K
Іванова or Петренко	Лише Іванова або Петренко	??K	Містить три символи, останній з яких K
< K	Починається з літер A...Л	Not Like «*K*»	Не містить літеру K
>= K	Починається з літер K...Я	Null	Пусті значення
		Is Not Null	Не пусті значення

► Запуск запиту

Для **запуску запиту**, тобто перегляду результатів у режимі таблиці, потрібно двічі клацнути на його значку (панель Усі об'єкти), або вибрати команду Відкрити в контекстному меню значка, або ж натиснути кнопку Запуск вкладки Конструктор у групі Результати.

Для перемикання між режимами таблиці та конструктора призначена кнопка Вигляд (вкладка Конструктор, група Результати).

Запит із параметрами

Запит із параметрами дає змогу ввести критерії відбору в діалоговому вікні під час виконання запиту.

Для створення запиту з параметрами в режимі конструктора в рядок Критерії замість конкретних значень потрібно вписати у квадратних дужках пояснювальний текст, який з'являтиметься у вікні Введення значення параметра щоразу після запуску запиту.

Наприклад: >[Уведіть оцінку, вище якої здійснювати відбір] (рис. 23.7).

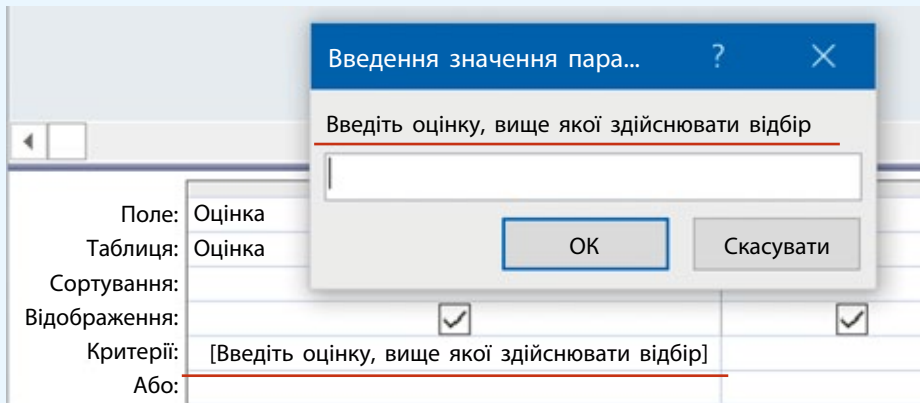


Рис. 23.7

Обчислення в запитах

Запит вибору дає змогу не лише вивести повний список шуканих даних, а й зробити прості підсумкові розрахунки — обчислити середнє арифметичне, кількість, суму тощо для даних певного поля.

Запит, відображений у режимі таблиці, надає таку можливість у вигляді рядка підсумків.

Для додавання рядка підсумків потрібно:

- 1) запустити створений запит (відкрийте в режимі таблиці);
- 2) клацнути кнопку Підсумки (рис. 23.8) на вкладці ОСНОВНЕ у групі Записи;
- 3) розгорнути в потрібному полі список рядка Підсумки й додати бажану дію над даними поля (рис. 23.9).

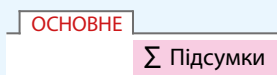


Рис. 23.8



Запит1		
Прізвище ▼	Оцінка ▼	Назва ▼
Дмитрук	11	українська мова
Кобилянська	9	українська мова
Петренко	10	інформатика
Науменко	12	алгебра
Сумлінна	10	фізика
Підсумок ▼		
Немає Сума Середнє Кількість Максимум Мінімум Стандартне відхилення Відхилення		

Рис. 23.9

Щоб приховати рядок підсумків, слід натиснути ту саму кнопку Підсумки на вкладці ОСНОВНЕ в групі Записи.

► Пошук повторюваних значень

Щоб **створити запит** для пошуку повторюваних даних, потрібно:

- 1) натиснути кнопку Майстер запитів (вкладка Створення група Запити);
- 2) у вікні Новий запит вибрати команду Майстер пошуку повторюваних записів і натиснути кнопку ОК;
- 3) у вікнах Майстер пошуку повторюваних записів вибрати потрібні таблиці й поля.

У Вікіпедії є багато сторінок, під час відкривання яких робиться запит до бази даних, яку називають Вікідані (англ. *Wikidata*). Так формується список посилань на іншомовні версії статті, деякі дані для картки на початку статті тощо.



Питання для самоперевірки

1. Що таке запити та для чого їх використовують?
2. Як створити запит вибору даних за певною умовою?
3. Як записати умову відбору даних: містить літеру «а»?
4. Як забезпечити введення параметра під час виконання запиту?
5. Як додати до запиту обчислення підсумків?
6. Як створити запит для пошуку повторюваних даних?



Вправа 23

- Створити запити вибору.

1. Запустіть СКБД Access. Відкрийте створену раніше БД Клас.
2. Створіть Запит1 для виведення списку всіх оцінок із зазначенням прізвищ учнів/учениць і назв предметів. Перевірте його роботу.
3. Створіть Запит2 на відображення прізвищ тих учнів/учениць, в іменах яких є літера «а». Перевірте його роботу.



В україномовному інтерфейсі деяких версій СКБД Access (наприклад, Access 2013) можуть не спрацьовувати запити з критеріями на зразок: *К, К??? тощо. У такому випадку для виконання подібних запитів необхідно увімкнути іншу мову інтерфейсу (Файл → Параметри → Мова → Вибір мов інтерфейсу та довідки...).



- ⚙ 6. Створіть Запит4 на відображення кількості повторюваних прізвищ у таблиці Учень. Перевірте його роботу.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112943





ПРАКТИЧНА РОБОТА 8

РОБОТА З ТАБЛИЦЯМИ І ЗАПИТАМИ

Завдання: створити таблиці бази даних Access за зразком і опрацювати табличні дані за допомогою запитів.

Обладнання: комп'ютер зі встановленою СКБД Access.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть СКБД Access і створіть порожню базу даних з іменем файлу Практична робота 8.accdb.

Відкрийте в режимі конструктора таблицю Таблиця 1, перейменуйте її на Товари й додайте назви трьох полів: Код, Група, Назва.

2. Доберіть необхідні типи даних для кожного з полів відповідно до зразка заповнення (рис. 1). Зробіть поле Код ключовим.



Код ▼	Група ▼	Назва ▼
1	молочна продукція	молоко
2	молочна продукція	йогурт з наповнювачем
3	макаронні вироби	локшина
4	круп	гречка
5	макаронні вироби	спагеті
6	чай	зелений
7	чай	чорний
8	круп	рис
9	молочна продукція	морозиво
10	макаронні вироби	макарони



Рис. 1

3. У вікні Властивості поля → Загальні виконайте такі дії:
 - 1) оптимізуйте (мінімізуйте) розмір кожного поля відповідно до зразка, якщо кількість записів у таблиці Товари не перевищуватиме 250, а довжина рядків у полях Група і Назва не перевищуватиме відповідних значень для товару з кодом 2 (див. рис. 1);
 - 2) решту властивостей полів змініть на свій розсуд;
 - 3) збережіть таблицю.

4. Створіть у конструкторі таблиць нову таблицю. Для цього:
- 1) дайте назву полям таблиці відповідно до зразка (рис. 2);

Код угоди▼	Код товару▼	Кількість тов▼	Ціна тов▼	Дата ▼	Вид розрахун▼	Оплата
1	1	1230	8	22.12.2024	безготівковий	☒
2	6	2301	15,23	05.10.2025	безготівковий	☒
3	4	2365	7	14.10.2025	готівкою	☐
4	7	235	10,24	26.10.2025	безготівковий	☒
5	7	256	12,35	30.10.2025	готівкою	☐

Рис. 2

- 2) установіть типи даних для кожного з полів відповідно до зразка, встановивши, де це можливо, тип Число;
 - 3) установіть ключове поле на свій розсуд;
 - 4) оптимізуйте (мінімізуйте) розмір полів на власний розсуд, керуючись зразком на рис. 2;
 - 5) якщо потрібно, змініть решту властивостей полів;
 - 6) збережіть порожню таблицю з назвою Угоди.
5. Установіть між таблицями Товари й Угоди зв'язок один-до-багатьох за полем Код товару.
- Налаштуйте параметри об'єднання (рис. 3, с. 152).
6. Заповніть таблицю Товари за зразком (див. рис. 1) і збережіть її. Заповніть таблицю Угоди за зразком (див. рис. 2) та збережіть таблицю.
 7. Використовуючи Конструктор запитів, створіть запит для виведення в алфавітному порядку всіх назв макаронних виробів (рис. 4).
 8. Використовуючи Конструктор запитів, створіть запит для виведення в порядку збільшення ціни групи та назви всіх товарів із безготівковим видом розрахунку за них (рис. 5, с. 152).
 9. Створіть запит для виведення групи та назви всіх товарів, на закупівлю яких укладено не більше однієї угоди (рис. 6, с. 152):
 - 1) за допомогою Майстра запитів створіть запит на пошук повторюваних значень у полі Код товару таблиці Угоди.
 - 2) відкрийте цей запит у Конструкторі запитів і додайте таблицю Товари;
 - 3) додайте в бланк запиту поле Назва з таблиці Товари;
 - 4) для кількості повторюваних значень змініть критерій на «<2» і вимкніть відображення всіх полів, крім Назва.

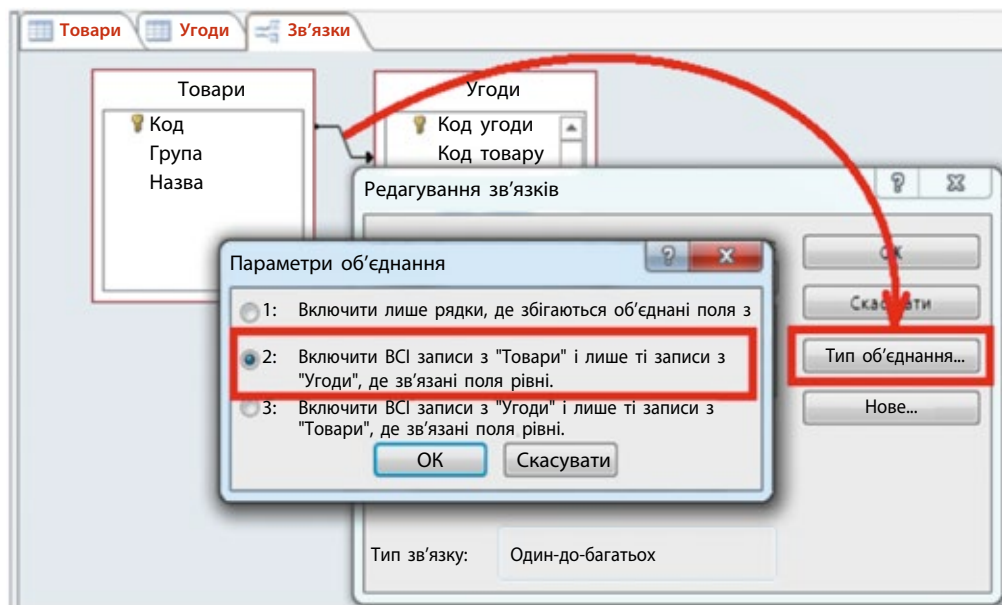


Рис. 3

Запит1	
Назва	
локшина	
макарони	
спагеті	

Рис. 4

Запит2		
Група	Назва	
молочна продукція	молоко	
чай	чорний	
чай	зелений	

Рис. 5

Пошук повторюваних значень для Угоди		
Назва		
йогурт з наповнювачем		
локшина		
макарони		
морозиво		
рис		
спагеті		
молоко		
гречка		
зелений		

Рис. 6

10. Перевірте роботу всіх запитів. Збережіть базу даних.

Зробіть висновок: які особливості створення таблиць і запитів у СКБД Access.

Уведення даних в електронну базу даних у режимі таблиці незручне для повсякденного користування. Щоб робота з даними нагадувала заповнення паперових бланків, застосовують екранні форми.

Створення форми

Для роботи з даними зручно використовувати форми.



Форма — інструмент, що має вигляд бланка для перегляду, заповнення й редагування вмісту таблиць та перегляду результатів запитів.

Після завантаження БД для створення форми можна скористатися одним із засобів у групі інструментів **Форми** на вкладці **СТВОРЕННЯ**.

Хороший результат можна отримати за допомогою майстра форм, тому саме цей режим і розглянемо. Майстер форм ставить докладні питання про записи, поля, макет, формат і на основі відповідей створює форму. Доопрацювати форму можна в режимі конструктора.

Для створення форми за допомогою майстра форм потрібно:

- 1) натиснути кнопку **Майстер форм** (рис. 24.1);
- 2) у вікні **Майстер форм** вибрати таблицю або запит; кнопкою «>» вибрати поля, які міститиме форма;
- 3) у наступному вікні вибрати варіант розміщення об'єктів на формі;
- 4) змінити, якщо потрібно, назву форми та вибрати подальші дії (відкрити форму для роботи чи продовжити розробку в режимі конструктора), після чого натиснути кнопку **Готово**.

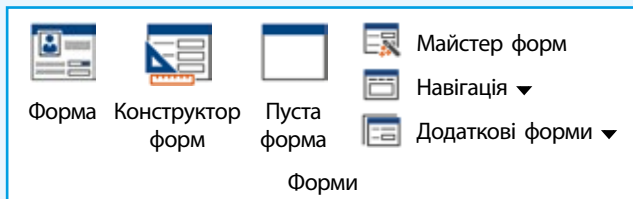


Рис. 24.1

Результатом виконаних дій буде збережена форма і під заголовком **Форми**, що в Області переходів у вікні Усі об'єкти, з'явиться її назва (рис. 24.2).

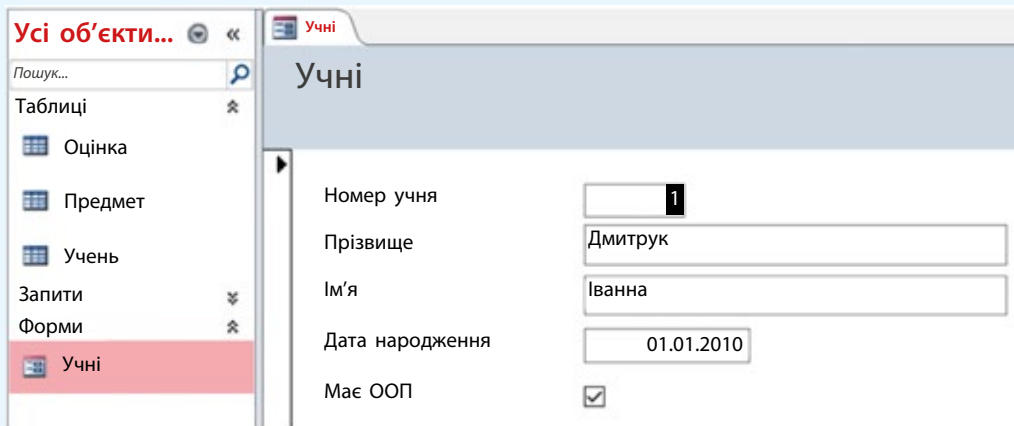


Рис. 24.2

Редагування форми

Розташування, розміри та налаштування об'єктів форми змінюють у Режимі розмітки (макета) або Конструктора, які вмикаються відповідними командами контекстного меню форми або кнопкою **Вигляд**, що в лівому верхньому куті вкладки **Основне** (рис. 24.3).

Форма може містити текстові поля, підписи, кнопки, прапорці тощо, які можна додатково встановити на формі зі вкладки **КОНСТРУКТОР** (рис. 24.4).

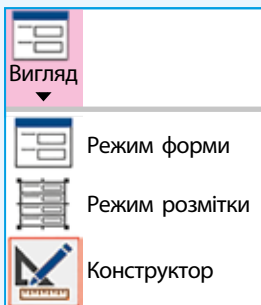


Рис. 24.3



Рис. 24.4

Редагування підписів

Підписи не виводять даних із БД, а лише містять пояснювальний текст. Текст підпису можна довільно змінити. Також інструментами групи Форматування тексту на вкладці ОСНОВНЕ можна встановити потрібний формат (колір, накреслення, вирівнювання, заливку тощо).

Додати підпис можна кнопкою **Aa** (див. рис. 24.4), скасувати хибну дію — кнопкою ↶.

Щоб видалити підпис (як і інші об'єкти форми), потрібно вибрати його й натиснути клавішу Delete.

Редагування чи видалення Підпису не впливає на працездатність форми, на назви полів таблиць і зміст наявних даних.

Редагування текстових полів даних

У режимі форми й режимі розмітки текстове поле виводить дані, а в режимі конструктора — назву поля таблиці чи запиту, звідки ці дані отримано (рис. 24.5).

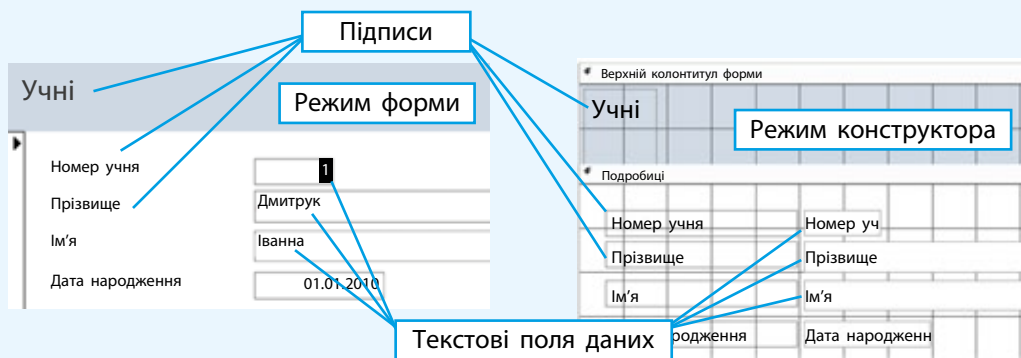


Рис. 24.5

Зміст даних у режимі розмітки чи конструктора змінити неможливо. Але можна, як і для написів, здійснити форматування тексту в них, а також змінити розташування й розміри самих текстових полів.



Видалення текстового поля з даними призводить до видалення пов'язаного з ним підпису та втрати доступу до відповідної інформації під час користування формою.

Оформлення форми

Окрім області даних, форма може мати заголовок та примітку, на яких розміщують як корисну інформацію, так і елементи оформлення. Ці частини форми залишаються видимими, навіть якщо для перегляду області даних необхідна смуга прокрутки. Наприклад, в області заголовка стрічкової форми зручно розмістити назви стовпців.

Щоб змінити зовнішній вигляд форми, можна виконати такі дії:

- 1) відкрити форму в режимі конструктора;
- 2) змінити за необхідності розміри вікна форми, для чого встановити вказівник на межу й при появі двонаправленої стрілки перетягти її до отримання потрібного розміру;
- 3) розширити область заголовка (примітки), перетягнувши її нижню межу;
- 4) додати зображення, для чого скористатися кнопкою Вставити зображення, що на вкладці КОНСТРУКТОР у групі Елементи керування.

Стандартні елементи користування формою

Для перегляду й пошуку записів унизу форми міститься рядок з елементами керування (рис. 24.6). Розгляньмо дію його елементів.



Рис. 24.6

Елемент	Дія
⏮	Перехід до першого запису
⏭	Перехід до останнього запису
⏪	Перехід до наступного запису
⏩	Перехід до попереднього запису
➡📄	Створення нового запису
1 з 5	Параметри поточного запису (тут: перший із п'яти)

Закінчення таблиці

Елемент	Дія
Пошук	Рядок пошуку (автоматично відшукує уведені значення в даних)
☐ Без фільтра ☒ Відфільтровано	Інформує про наявність увімкненого фільтра

Щоб додати новий запис, потрібно:

- 1) натиснути кнопку  і ввести в поля форми відповідні дані;
- 2) перейти до іншого запису — після цього, якщо введені дані відповідають умовам цілісності даних, їх буде внесено у відповідні таблиці.

Якщо ж буде виявлено проблеми (наприклад, не заповнене поле, яке є обов'язковим), то з'явиться повідомлення (рис. 24.7), після закриття якого слід виправити дані й повторити спробу збереження.

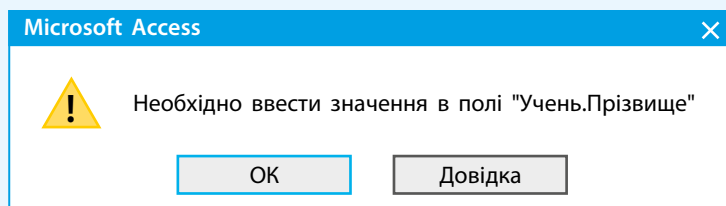


Рис. 24.7



Питання для самоперевірки

1. Для чого використовують форми?
2. Як створити форму за допомогою майстра форм?
3. Як викликати режим конструктора для редагування форми?
4. Як змінити параметри шрифту окремого поля?
5. Як перейти від редагування до користування формою?
6. Опишіть стандартні елементи користування формою.



Вправа 24

■ Створити форми для роботи з базою даних Клас.

1. Запустіть СКБД Access і відкрийте створену раніше базу даних Клас.
2. За допомогою Майстра форм створіть форму Учні, яка містить усі поля таблиці Учень (див. рис. 24.2, с. 154).
3. У режимі конструктора встановіть однакові розміри текстових полів форми. Користуючись формою Учні, введіть у базу даних інформацію ще про п'ятьох учнів / учениць.
4. Створіть форму для роботи з таблицею Предмет та введіть дані ще про 5 навчальних предметів.
5. Створіть запит Всі оцінки, який включає поля: Прізвище, Ім'я з таблиці Учень; Оцінка й Дата з таблиці Оцінка; Назва та Вчитель із таблиці Предмет.

- ⚙ 6. За допомогою Майстра форм створіть на основі запиту Всі оцінки форму Перегляд оцінок із підформою з налаштуваннями, як наведено на рис. 24.8.

Перегляньте за допомогою створеної форми оцінки, користуючись стандартними елементами користування формою.

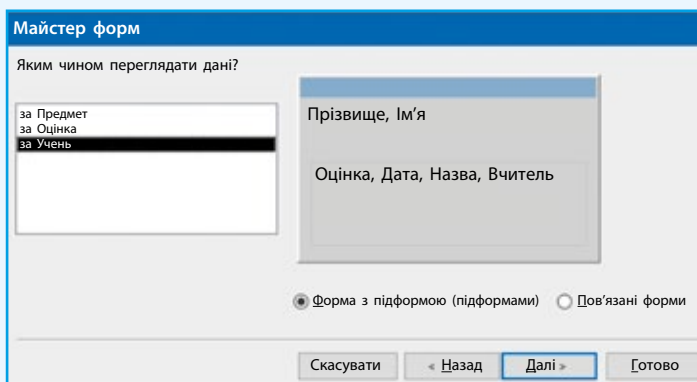


Рис. 24.8



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112944



Відкриті таблиці, запити та форми СКБД Access можна роздрукувати за допомогою команд Файл → Друк. Перед тим варто оформити дані, розмістивши на свій розсуд написи й поля даних, додати заголовки, малюнки, номери сторінок тощо. У пригоді стануть звіти.

Структура звіту

 **Звіт** — це підготовлений до публікації чи друку електронний документ на основі певних даних, отриманих із бази даних.

У режимі конструктора звіту, який відкривається кнопкою Конструктор звітів на вкладці СТВОРЕННЯ в групі Звіти, відображається стандартна (порожня) структура сторінки. Вона включає верхній та нижній колонтитули й ділянку Подобиці для розміщення даних (рис. 25.1).

Колонтитули сторінок, як і в інших офісних документах (Word, Excel та ін.), повторюються на кожній сторінці звіту й можуть містити номер сторінки та інші службові дані.

До звіту ще можна включити колонтитули звіту (ще кажуть, заголовок і примітку), які розташовуються на початку першої й наприкінці останньої його сторінки.

Якщо в контекстному меню макета сторінки вибрати команду Заголовки/примітка звіту, то в конструкторі над верхнім і під нижнім колонтитулом сторінки з'являться відповідні ділянки.

Звіт може містити автоматично відформатовані емблему, назву, точну дату та час, які додаються до заголовка відповідними кнопками на вкладці КОНСТРУКТОР у групі Колонтитули (рис. 25.2, с. 160).

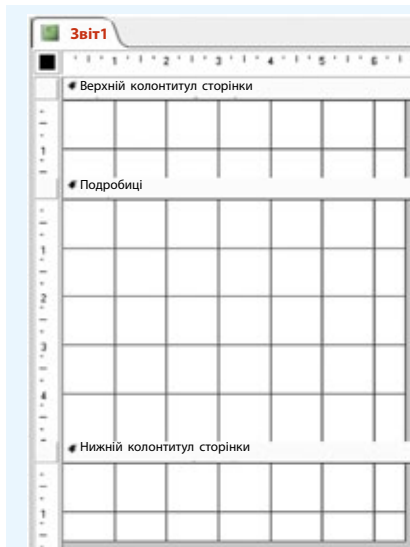


Рис. 25.1

У верхньому колонтитулі сторінки багатосторінкового звіту зручно розмістити заголовки стовпців таблиці з даними.

Перегляд у режимі розмітки (рис. 25.3) дає змогу оцінити, як сторінки звіту розмістяться на аркушах паперу. У цьому режимі ділянки колонтитулів і подробиць при клацанні по них позначаються кольоровими рамками.

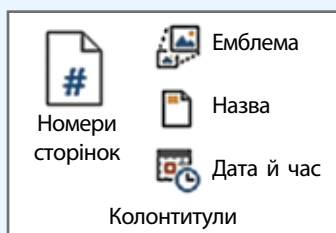


Рис. 25.2

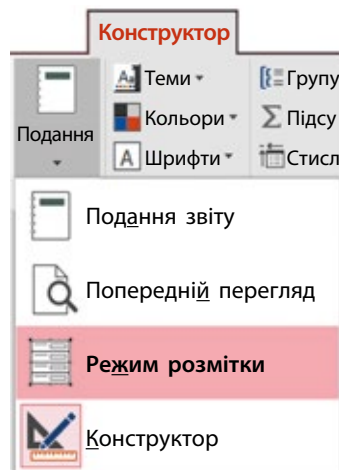


Рис. 25.3

Створення звіту

► Створення звіту на основі об'єкта

Якщо клацнути назву таблиці, запиту чи форми в Області переходів на панелі Усі об'єкти, а потім кнопку Звіт, що на вкладці ОСНОВНЕ в групі Звіти, то з'явиться зображення звіту вибраного об'єкта в Режимі розмітки. Цей звіт міститиме всі поля об'єкта, надалі його можна змінити, відредагувати й відформатувати, щоб надати бажаного вигляду.

► Створення звіту вручну

Звіти використовують для друку свідоцтв, візитівок — документів із однаковим оформленням, які відрізняються лише деякими даними.



Створимо звіт для друку дипломів для переможців шахового турніру між учнями й ученицями класу (рис. 25.4).

Перш ніж розпочати, відкрийте таблицю Оцінка та в полі Коментар зазначте місця, які посіли переможці турніру (наприклад, 1 місце). Конструювання звіту включає створення фонових малюнків, додавання й розташування підписів. Нагороджується, за, у шаховому турнірі та текстових полів із даними про прізвища, імена та місця переможців.

Розгляньмо послідовність дій для створення такого звіту.

- **Налаштування сторінки звіту**

- 1) Натисніть кнопку Пустий звіт на вкладці СТВОРЕННЯ в групі Звіти.
- 2) На вкладці Параметри сторінки налаштуйте розмір сторінки (наприклад, A4) і полів.
- 3) У режимі конструктора установіть нульові колонтитули.

- **Вставлення зображення**

- 4) Підготуйте зображення диплома (рис. 25.5).
- 5) Натисніть кнопку Вставити зображення, що на вкладці КОНСТРУКТОР у групі Елементи керування, виберіть зображення й клацніть у верхньому лівому кутку ділянки Подробиці.
- 6) Командою Розташування контекстного меню зображення помістіть його на задній план.
- 7) У контекстному меню зображення виберіть команду Властивості та на Аркуші властивостей (список Формат) установіть Вирівнювання зображення (Вгорі ліворуч) та Ширину й Висоту згідно з розмірами сторінки (так, для формату A4 — Ширина: 21 см, Висота: 29,7 см).
- 8) У режимі конструктора, за необхідності, перетягуванням меж звіту встановіть його розміри згідно з розмірами зображення.

- **Додавання підписів**

- 9) Додайте три елементи керування Підпис, розмістіть їх (стрілками на клавіатурі або перетягуванням мишею) в потрібних місцях, зробіть написи Нагороджується, за, у шаховому турнірі, відредагуйте і відформатуйте їх (рис. 25.6).



Рис. 25.4



Рис. 25.5



Рис. 25.6

- **Додавання текстових полів із даними**
- 10) Натисніть кнопку Додавання наявних полів, що на вкладці КОНСТРУКТОР у групі Знаряддя (рис. 25.7).
 - 11) На панелі Список полів розгорніть список Учень та подвійним клацанням додайте до звіту поля Прізвище та Ім'я. Зі списку Оцінка додайте поле Коментар.
 - 12) Видаліть клавішею Delete підписи, розташовані ліворуч від доданих текстових полів (рис. 25.8). Потім розташуйте в потрібних місцях та відформатуйте тексти в текстових полях із прізвищем та ім'ям.

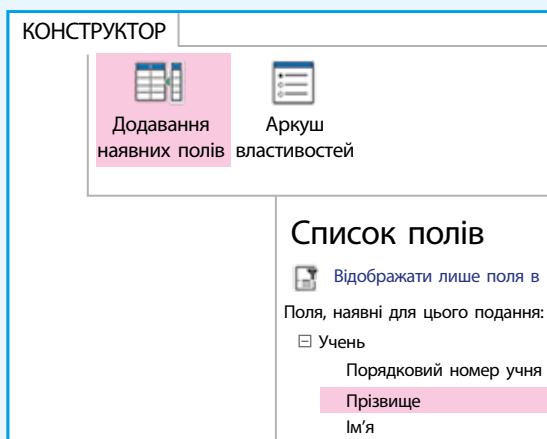


Рис. 25.7

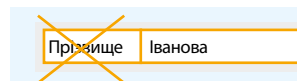


Рис. 25.8



- **Відбір переможців**
- 13) Натисніть кнопку Додавання наявних полів (див. рис. 25.7) та в контекстному меню наявного поля (наприклад, Прізвище) виберіть команду Редагувати джерело записів (рис. 25.9).

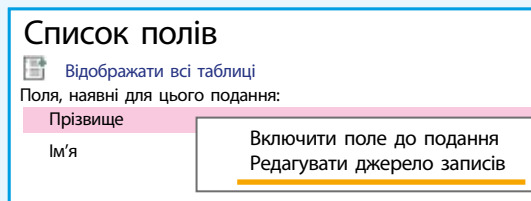


Рис. 25.9

14) Щоб опрацювати тільки ті записи, в яких зазначено місце, в Побудовнику запитів для поля Коментар установіть критерій Not "" (не-порожній рядок).

15) Збережіть побудований запит та створений Звіт.

Отже, наш звіт готовий. Кількість його сторінок дорівнює кількості переможців у таблиці Оцінка. Тепер можна видалити зображення диплома та друкувати звіт на порожніх бланках дипломів або друкувати дипломи разом із зображеннями на чистих аркушах.

► Створення звіту за допомогою майстра звіту

Майстер звітів на відміну від режиму конструктора чи пустого звіту автоматично розташовує вибрані елементи й раціонально форматує. Для створення звіту за допомогою майстра звітів потрібно:

- 1) вибрати команду Майстер звітів на вкладці СТВОРЕННЯ в групі Звіти — відкриється вікно Майстер звітів;
- 2) у полі Таблиці та запити вибрати таблицю або запит — джерело даних для звіту (наприклад, Таблиця: Учень);
- 3) за допомогою кнопки «>» вибрати потрібні поля або скористатися кнопкою «>>» для вибору всіх полів і натиснути кнопку Далі — з'явиться вікно, призначене для задання рівнів групування;
- 4) для прийняття рівня, заданого за промовчанням, натиснути Далі;
- 5) вибрати тип упорядкування та натиснути кнопку Далі;
- 6) вибрати вид макета (наприклад, Табличний) та орієнтацію аркуша (наприклад, Альбомна) і натиснути кнопку Далі;
- 7) у кінцевому вікні ввести назву звіту Учні, натиснути кнопку Готово — у результаті виконаних дій з'явиться звіт Учні.

Звіт можна змінити в режимі конструктора, як описано вище.

Друкування звіту

Access зберігає параметри сторінки разом зі звітом, тому їх потрібно налаштувати лише раз. Згодом їх, за потреби, можна буде змінити.

Щоб змінити параметри сторінки, потрібно:

- 1) відкрити звіт у режимі Попередній перегляд (див. рис. 25.3, с. 160) — внесені зміни негайно відображаються на зображенні на вкладці Попередній перегляд; у групах Макет сторінки та Розмір сторінки вибрати орієнтацію сторінки (Книжкова або Альбомна), натиснути кнопку Розмір, щоб установити розмір паперу, кнопку Поля, щоб налаштувати поля, тощо;

- 2) після внесення змін за допомогою кнопок переходу переглянути кілька сторінок, щоб упевнитись у відсутності проблем із форматкуванням на наступних сторінках.

Для **передавання звіту на принтер** потрібно:

- 1) відкрити звіт у будь-якому поданні або вибрати в області переходів;
- 2) вибрати Файл → Друк → Друк — відкриється діалогове вікно Друк;
- 3) налаштувати запропоновані параметри, зокрема принтер, діапазон друку та кількість копій;
- 4) клацнути кнопку ОК — принтер надрукує звіт.



Питання для самоперевірки

1. Яке призначення звітів?
2. Яку структуру має стандартний звіт?
3. Як швидко створити звіт вибраної таблиці, запиту чи форми?
4. У чому полягає конструювання звіту для заповнення бланків, грамот, атестатів тощо?
5. Як створити звіт за допомогою майстра звітів?
6. Опишіть порядок друкування звіту.



Вправа 25

■ Створити звіти в різний спосіб.

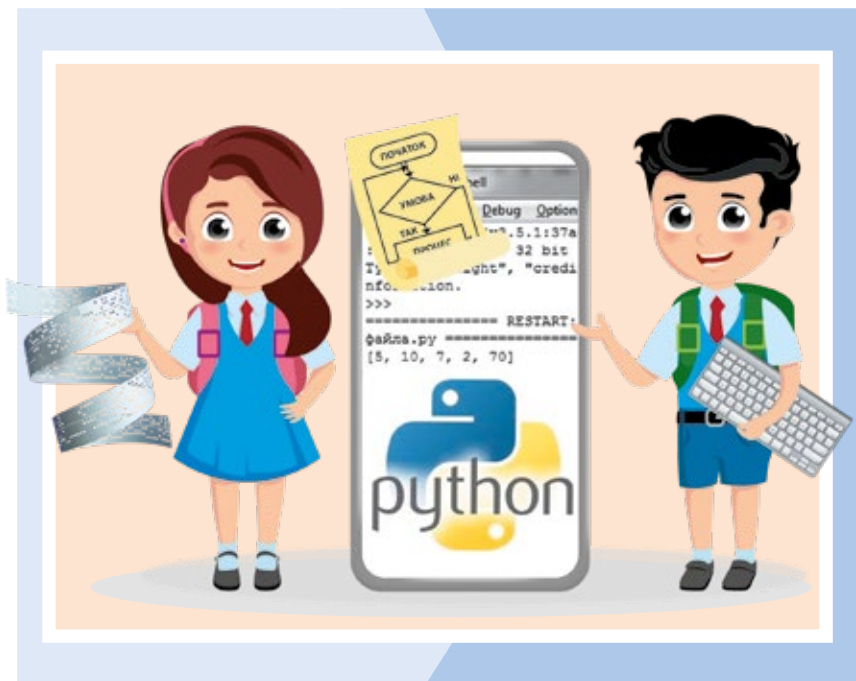
1. Запустіть СКБД Access.
2. Відкрийте створену раніше базу даних Клас.
3. Створіть і збережіть звіт, який складається з усіх полів таблиці Учень.
- 4. Завантажте з інтернету й збережіть файл із зображенням незаповненого диплома (або використайте файл diploma.jpg*).
- ⚙ 5. За алгоритмом, наведеним вище, створіть звіт для заповнення дипломів для нагородження переможців / переможниць змагань.
6. Збережіть звіт разом із малюнком.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112945





АЛГОРИТМИ ТА ПРОГРАМИ

5

РОЗДІЛ

У цьому розділі ви ознайомитеся з поняттям масиву даних, навчитеся застосовувати стандартні алгоритми для розв'язування задач, створювати програми з використанням словників і списків.

Ви вже стикалися з необхідністю зберігання наборів значень, об'єднаних за певною ознакою. Так, у комп'ютерній грі можуть зберігатися координати персонажів на екрані, таблиця рекордів тощо, в обчислювальних програмах — набори чисел.

Основні поняття

Вам уже відомо, що для зберігання сукупностей даних у програмуванні існують структуровані типи даних, або контейнери. Контейнер можна порівняти з картотекою — він зберігає набір значень у певному порядку.

Структурованими типами (контейнерами) називають типи даних, які мають деяку внутрішню структуру та побудовані з простих типів даних. Змінна або константа структурованого типу має кілька компонент, об'єднаних спільною назвою.

Розгляньмо таблицю зі значеннями температури повітря. У ній кожному значенню відповідає порядковий номер дня тижня.

Масив t значень середньодобової температури протягом тижня							
Номер елемента i	0	1	2	3	4	5	6
Значення температури $t[i]$, °C	-5	-2	0	0	2	5	4



Одновимірний масив — пронумерована послідовність даних одного типу, що мають спільну назву.

Спільна назва t послідовності значень температури означає, що ці дані належать до одного масиву, а розрізнити їх можна за номером (індексом) (рис. 26.1).

У мові Python для зберігання масивів даних найчастіше використовуються списки (list).



Рис. 26.1



Список (list) — упорядкований набір об'єктів, можливо, різних типів (чисел, рядків, списків тощо), який можна змінювати.

Хоча Python дає змогу зберігати в списку елементи різних типів, наша мета — навчитися опрацьовувати **масиви даних**, тобто сукупності однотипних значень, об'єднаних за певною ознакою.

Створення списків

Створити порожній список можна двома способами:

- за допомогою функції `list()`: `a = list()`;
- використавши квадратні дужки: `a = []`.

У програмі значення списку записують у квадратних дужках `[]` і відокремлюють одне від одного за допомогою коми.



Створимо змінні типу список:

```
a = [] # Порожній список
b = [3.14, 9.8, 2.4, 1.0] # Список із дробових чисел (тип float)
animal = ["Кіт", "Собака", "Миша", "Хом'як"] # Список із рядків (тип str), рис. 26.2
```



Список `animal` →

0	1	2	3
"Кіт"	"Собака"	"Миша"	"Хом'як"

Рис. 26.2



Перевірити, чи є в списку певний елемент, допоможе ключове слово `in`. Визначимо, чи є в списку `letters` елемент `'a'`:

```
if 'a' in letters: print ('так')
else: print ("символу 'a' в списку немає")
```

Додати елемент до списку можна двома способами:

За допомогою методу <code>append()</code>	За допомогою операції об'єднання «+»
Додати до списку <code>a = [2, 6, 4]</code> елемент 10: <pre>>>> a.append(10) >>> print(a) [2, 6, 4, 10]</pre>	Заповнити список числами від 0 до 4: <pre>>>> a = [] >>> for i in range(5): a = a+[i] >>> a [0, 1, 2, 3, 4]</pre>

Додати елемент можна лише у *створений* список (у т. ч. порожній).

3

Створимо список із 5 цілих чисел, що вводяться з клавіатури (рис. 26.3).

```
a = []
for i in range(5):
    n = int(input(str(i)+'?'))
    a.append(n)
print(a)
```



```
0? 3
1? 5
2? 2
3? 15
4? 90
[3, 5, 2, 15, 90]
>>>
```

Рис. 26.3

4

Заповнимо список 5 випадковими числами.

```
from random import*
a = []
for i in range(5):
    a = a+[randint(1, 10)]
print(a)
```

Результат роботи програми може бути таким: [9, 1, 6, 3, 8].

Генератори списків

Ще один спосіб, за допомогою якого в мові Python можна створити список, — використання генераторів списків.



Генератор списків — це синтаксична конструкція, яка дає змогу створити список та заповнити його значеннями за певними правилами.

Синтаксис генератора списків:

```
list_variable = [<вираз> for x in iterable],
```

де x — значення, які вибираються з послідовності `iterable`, `<вираз>` — вираз, значення якого обчислюється для кожного значення x і використовується як елемент списку `list_variable`.

Конструкція береться у квадратні дужки; це означає, що результатом роботи генератора є список.

5

Створимо список із перших 10 натуральних чисел.

```
>>> a = [i for i in range(1, 11)]
>>> a
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

6

Створимо список `a` з квадратів елементів списку `b`.

```
>>> b = [2, -2, 4, -4, 7, 5]
```

```
>>> a = [i**2 for i in b]
```

```
>>> b
```

```
[4, 4, 16, 16, 49, 25]
```

7

Створимо список `a` з перших літер елементів списку `animal`.

```
>>> animal = ["Кіт", "Собака", "Миша", "Хом'як"]
```

```
>>> a = [x[0] for x in animal]
```

```
>>> a
```

```
['K', 'C', 'M', 'X']
```



Доступ до елементів списку

Доступ до елемента списку можна отримати за його індексом (номером у списку). Нумерація елементів починається з нуля.

Завдяки нумерації можна визначати довжину списку, тобто кількість елементів у ньому, отримувати зрізи.

8

Дано список `a = [2, 15, 3, -5, 10]`. Проаналізуємо результати виконання в консолі дій зі списком.

№	Дія	Команда	Результат виконання
1	Визначити довжину списку	<code>len(a)</code>	5
2	Вивести елемент списку з індексом 3	<code>a[3]</code>	-5
3	Вивести перші 3 елементи списку	<code>a[0: 3]</code>	[2, 15, 3]
4	Вивести елементи списку, починаючи з елемента з номером 3 і до кінця	<code>a[3:]</code>	[-5, 10]
5	Вивести зріз списку довжиною 1 елемент	<code>a[3: 4]</code>	[-5]

9

Замінімо нулями перші 3 елементи списку `a`:

```
a = [2, 15, 3, -5, 10]
```

```
a[0: 3] = [0]*3
```

```
print(a) # Буде виведено [0, 0, 0, -5, 10]
```

Отже, списки є потужним інструментом для збереження масивів — сукупностей однотипних даних, розташованих у певному порядку.

**Питання для самоперевірки**

- Обговоріть спільні ознаки та відмінності між рядками й списками.
- У змінній `letters` міститься список `['a', 'b', 'c', 'd']`. Обчисліть значення виразів:

а) <code>letters[int('2'*2)//11];</code>	в) <code>letters[-1];</code>	д) <code>letters[1:3];</code>
б) <code>letters[3];</code>	г) <code>letters[::2];</code>	е) <code>letters[:2].</code>
- Створено список `a = [2, 15, 3, -5, 10]`. Визначте вміст списку `a` після виконання команд:

а) <code>a = a+[11];</code>	б) <code>a[3] = 0.</code>
-----------------------------	---------------------------
- Збережіть імена кількох друзів і подруг у списку з назвою `names`. Виведіть ім'я кожної особи, звернувшись до кожного елемента списку.
- За допомогою генератора списків заповніть список квадратами перших 10 натуральних чисел.
- За допомогою генератора списків заповніть список `a` непарними числами в діапазоні від 1 до 20.

**Вправа 26**

- Створити програму опрацювання списку рядкових даних.
- Завантажте IDLE Python. Поміркуйте над тим, яку інформацію можна було б зберігати в списку.
Створіть список `nazva` з 6 рядкових величин: назв професій, видів спорту, назв країн, видів рослин тощо.
 - Змінній `a` надайте значення третього елемента списку `nazva`. Виведіть значення `a` на екран.
 - Виведіть на екран перші три елементи списку `nazva`.
 - Програмно додайте до списку `nazva` ще два значення. Виведіть список на екран.
 - ⚙ За допомогою генератора списків створіть список `letters` із перших літер елементів списку `nazva`. Виведіть список `letters` на екран.
 - Збережіть файл з іменем `Vprava26.py`.

**Комп'ютерне тестування**

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112946



§27

ФУНКЦІЇ ТА МЕТОДИ СПИСКІВ

Для опрацювання сукупностей однотипних даних Python має вбудовані функції та методи списків.

Методи списків

Виклик методів має такий синтаксис:

Назва_списку.Назва_методу(<параметри>)

Зауважмо, що методи списків змінюють сам список, тому результат виконання не потрібно зберігати в іншу змінну.

Розгляньмо на прикладі списку `a = [2, 15, 3, -5, 3]` деякі методи:

Метод	Призначення	Приклад	Результат
<code>list.insert(i, x)</code>	Вставляє на <i>i</i> -те місце елемент зі значенням <i>x</i>	<code>a.insert(2, 78)</code>	<code>a = [2, 15, 78, 3, -5, 3]</code>
<code>list.remove(x)</code>	Видаляє зі списку перший елемент зі значенням <i>x</i>	<code>a.remove(3)</code>	<code>a = [2, 15, -5, 3]</code>
<code>list.index(x)</code>	Повертає індекс першого елемента зі значенням <i>x</i>	<code>k = a.index(2)</code>	<code>k = 0</code>
<code>list.count(x)</code>	Повертає кількість елементів зі значенням <i>x</i>	<code>k = a.count(3)</code>	<code>k = 2</code>
<code>list.sort()</code>	Упорядковує список за зростанням	<code>a.sort()</code>	<code>a = [-5, 2, 3, 3, 15]</code>
<code>list.reverse()</code>	Переставляє елементи списку у зворотному порядку	<code>a.reverse()</code>	<code>a = [3, -5, 3, 15, 2]</code>



Список `results = [2, 3, 7, 15, 1, 9]` містить суми балів, набрані учасниками змагань. Визначимо три найкращі результати (див. рисунок):

```
results.sort() → [1, 2, 3, 7, 9, 15]
results.reverse() → [15, 9, 7, 3, 2, 1]
print(results[:3]) → Виведено: [15, 9, 7]
```

Функції для роботи зі списками

Функції для роботи зі списками використовують список як аргумент: Назва_функції(назва_списку). Більшість функцій, на відміну від методів, не змінюють сам список, а повертають певне значення.

Розгляньмо на прикладі списку `a = [1, 5, 7, 5, 31, -5]` деякі функції:

Метод	Призначення	Приклад	Результат
<code>len(list)</code>	Повертає довжину списку	<code>k = len(a)</code>	<code>k = 6</code>
<code>max(list)</code>	Повертає значення найбільшого елемента	<code>m = max(a)</code>	<code>m = 31</code>
<code>min(list)</code>	Повертає значення найменшого елемента	<code>m = min(a)</code>	<code>m = -5</code>
<code>sum(list)</code>	Повертає значення суми елементів	<code>s = sum(a)</code>	<code>s = 44</code>
<code>del(list[n])</code>	Видаляє елемент з індексом <code>n</code>	<code>del(a[2])</code>	<code>a = [1, 5, 5, 31, -5]</code>



Видалимо зі списку `a = [1, 5, 7, 31, -5]` найбільший елемент:

```
>>> m = max(a)           # m = 31
>>> n = a.index(m)       # n = 3
>>> del(a[n])            # a = [1, 5, 7, -5]
```

Також є низки засобів для роботи зі списками й рядками.

Метод	Призначення	Приклад	Результат
<code>s.split(символ)</code>	Розбиття рядка <code>s</code> на список за заданим символом-роздільником	<code>s = "2 56 12 5"</code> <code>a = s.split()</code> # За промовчуванням роздільник пропуск	<code>a = ['2', '56', '12', '5']</code>
<code>list(рядок)</code>	Розбиття рядка на список символів	<code>s = "2 56 12 5"</code> <code>a = list(s)</code>	<code>a = ['2', ' ', '5', '6', ' ', '1', '2', ' ', '5']</code>
<code>c.join(список)</code>	Об'єднання списку в рядок із роздільником <code>c</code> між символами	<code>a = ['2', '56', '12', '5']</code> <code>s = '+'.join(a)</code>	<code>s = "2+56+12+5"</code>



Упорядкуємо за неспаданням числа, які утворюють рядок `s`:

```
s = "2 56 12 5 12"
a = s.split()
a.sort()
s = ' '.join(a)
print(s)          # Буде надруковано "2 5 12 12 56"
```

Отже, розглянуті функції дають змогу будувати складні алгоритми опрацювання списків.



Питання для самоперевірки

- У змінній `a` міститься список `[2, 8, 1, 4, 3, -5, 3]`. Обчисліть:
а) `a.index(4)`; б) `a.count(3)`; в) `len(a)`; г) `max(a)`.
- У змінній `a` міститься список `[2, 8, 1, 4, 3, -5, 3]`. Якого значення набуває змінна `a` після виклику методів?
а) `a.insert(3, 10)`; б) `a.sort()`; в) `del(a[0])`; г) `a.remove(4)`.
- Збережіть у списку назви кількох країн. Застосуйте до списку функції `sort()` і `reverse()`. Виведіть результат у консоль.
- Заповніть список десятьма випадковими цілими числами в діапазоні `[0, 10]`. Визначте кількість чисел `5` у створеному списку.
- У текстовому файлі збережено номери премійованих осіб. Напишіть програму для перевірки, чи не потрапив хтось у список двічі.



Вправа 27

- Створити програму опрацювання списку рядкових даних.
- 1. Завантажте IDLE Python. Відкрийте файл із назвою `Vprava26`.
- 2. Змінній `a` надайте значення найбільшого елемента списку `pazva`. Виведіть значення `a` на екран.
- 3. Замініть найбільший елемент списку рядком `***`.
- 4. Упорядкуйте список `pazva` за зростанням і виведіть на екран.
- ⚙ 5. Поміняйте місцями найменший та найбільший елементи списку `pazva`. Виведіть список на екран.
- 6. Видаліть зі списку останній елемент. Виведіть список на екран.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112947



Розв'язування багатьох задач, що потребують виконання дій над елементами масиву, значення або індекс яких відповідає заданій умові, базується на типових алгоритмах опрацювання масивів.

Перебирання елементів списку

Якщо потрібно виконати дії лише над елементами списку, що мають задану ознаку, слід послідовно перебрати елементи списку й перевірити кожний елемент на відповідність певній умові. Перебирати зручно в циклі, де параметр послідовно набуває значень елементів списку або їх індексів. Розгляньмо це на прикладі виведення списку:

Послідовне звернення до елементів за їх індексами	<code>a = [2, 5, 14, 62, -5]</code> <code>for i in range(len(a)):</code> <code>print(a[i])</code>
Перебір усіх наявних у списку елементів. Застосовується, коли не потрібно змінювати значення елементів списку	<code>for item in a:</code> <code>print(item)</code>

Типові алгоритми опрацювання елементів списку

► Заміна значень усіх елементів списку

Алгоритм розв'язування задачі: послідовно перебрати всі елементи списку та значення чергового елемента змінити за певним правилом (наприклад, змінити знак на протилежний, надати нульове значення, збільшити або зменшити на певну величину).



Подвоїмо значення елементів списку `a = [2, 5, 14, 62, -5]`.

`a = [2, 5, 14, 62, -5]`

`for i in range(5):`

`a[i] = a[i]*2     # Список набуває значення a = [4, 10, 28, 124, -10]`

► **Заміна значень залежно від умови**

Алгоритм розв'язування задачі: послідовно перебрати всі елементи списку; якщо черговий елемент відповідає деякій умові, його значення змінити за певним правилом. Умова може включати перевірку кратності значення елемента списку певному числу, додатності, рівності нулю тощо. Може перевірятися значення індексу елемента списку (наприклад, для змінення елементів, що стоять на парних місцях).

2

Поділимо на 2 парні елементи списку $a = [2, 5, 14, 62, -5]$.

```
if a[i]%2 == 0:
```

```
    a[i] = a[i]//2    # Список набуває значення a = [1, 5, 7, 31, -5]
```

3

Елементи списку $a = [2, 5, 14, 62, -5]$, що мають парні індекси, потрібно замінити нулем.

```
a = [2, 5, 14, 62, -5]
```

```
for i in range(5):
```

```
    if i%2 == 0:
```

```
        a[i] = 0
```

```
print(a)    # Буде надруковано [0, 5, 0, 62, 0]
```

```
[0, 5, 0, 62, 0]
```

► **Визначення кількості елементів із заданою властивістю**

4

Визначимо кількість додатних елементів списку $a = [2, 5, 14, 62, -5]$.

```
k = 0    # Лічильник елементів із заданою властивістю
```

```
for item in a:
```

```
    if item>0:
```

```
        k = k+1
```

```
print ('k = ', k)    # Буде надруковано k = 4
```



► **Визначення найбільшого (найменшого) елемента списку**

Хоча в Python убудовано функції `max(list)`, `min(list)`, слід розуміти сутність алгоритму пошуку найбільшого (найменшого) елемента списку.

Алгоритм розв'язування задачі: припустити, що найбільшим елементом списку є перший (з індексом 0), після чого послідовно перебрати елементи, починаючи з другого; якщо черговий елемент більший за знайдений раніше, запам'ятати його значення.

```
mx = a[0]    # Змінна для збереження найбільшого значення елемента
```

```
k = 0    # Змінна для збереження індексу найбільшого елемента
```

```
for i in range(1, len(a)):    # Перебираємо елементи, починаючи з першого
```

```

if a[i] > mx:          # Якщо i-й елемент більший за mx,
    mx = a[i]         # значення mx замінюється на значення a[i];
    k = i             # у змінній k запам'ятовується значення індексу i

```

► Обчислення суми (добутку) елементів із заданою властивістю

5

Знайдемо суму додатних елементів списку $a = [2, -3, 1, 6, -5]$.
`suma = 0` # Змінна для збереження значення суми
`for item in a:`
 `if item > 0: suma = suma + item`
`print('suma = ', suma)` # Буде надруковано `suma = 9`

6

Знайдемо добуток ненульових елементів списку $a = [2, 0, 2, 6, 0]$.
`dobutok = 1` # Змінна для збереження значення суми
`for i in range(len(a)):`
 `if a[i] != 0: dobutok = dobutok * a[i]`
`print('dobutok = ', dobutok)` # Буде надруковано `dobutok = 24`

Отже, тепер ви зможете створювати ефективні алгоритми розв'язування більш складних задач опрацювання масивів даних. З алгоритмами опрацювання списків ви можете детальніше ознайомитися, переглянувши відеоролік за QR-кодом або посиланням rnk.com.ua/115068.



Питання для самоперевірки

- Дано список $a = [3, 8, 5, 7, 6]$. Виконайте дії з елементами списку.
 - `for i in range(5):`
 `a[i] = -a[i]`
 - `for i in range(5):`
 `if a[i] % 3 == 0:`
 `a[i] = a[i] // 3`
 - Дано список $a = []$. Запишіть команди для реалізації таких дій:
 - ввести з клавіатури значення елементів списку;
 - знайти добуток елементів списку;
 - знайти найменший елемент списку;
 - знайти кількість додатних елементів списку.
- Створіть список `nazva`, який містить назви 5 країн світу. Сформуйте з елементів списку й виведіть на екран речення за зразком:
 Україна, Польща, Німеччина — це назви країн.

4. Складіть програму для виконання завдання: список `Numbers[]` заповнено 50 випадковими числами в межах 100. Визначити, скільки елементів списку відрізняються від останнього елемента.

Вправа 28

- Створити програму Календар спостережень за погодою. Проаналізувати список значень середньодобових температур за березень.

- 1. У IDLE виберіть команду `File → New File`.
Завантажте модуль `random`. Створіть порожній список `t` і додайте до списку 31 випадкове число в діапазоні типових для вашої місцевості березневих температур.

Виведіть список на екран.

```
from random import *
```

```
t = []
```

```
for i in range (31):
```

```
    t.append(randint(-10, 10))
```

```
print(t)
```



2. Запишіть код для підрахунку кількості днів, коли температура була вищою від нуля. Виведіть знайдене значення з поясненням:
`print(k, ' днів температура була вищою від 0')`

3. Визначте число стрибків температури (сусідства двох чисел з різними знаками):

```
k = 0
```

```
for i in range (30):
```

```
    if t[i]*t[i+1]<0: k += 1
```

```
print ('Стрибків температури було:', k)
```

4. Визначте дату найнижчої температури, виведіть значення.

5. Визначте різницю між найменшим і найбільшим значеннями температури.

- ⚙️ 6. Знайдіть середнє значення температури повітря за березень.

Виведіть дати, коли температура відрізнялася від середньомісячної не більше ніж на 1°C .

Збережіть файл з іменем `Vprava28.py`. Перевірте роботу програми.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112948



Іноколи виникає потреба в упорядкуванні даних у таблиці за деякою ознакою. Числові дані можна відсортувати за величиною (наприклад, розташувати значення маси деталей від найменшого до найбільшого), рядкові — за алфавітом (упорядкувати список).

Методи сортування масивів



Сортування елементів масиву — впорядкування їх за деякою ознакою.

Як ви вже знаєте (див. с. 218 підручника для 7 класу), клас `list` у Python має метод `sort()`, який упорядковує список за неспаданням. Щоб упорядкувати список за незростанням, атрибуту `reverse` слід надати значення `True`.



Упорядкуємо елементи списку `arr` за незростанням.

```
arr = [23, 12, 3, 45, 6, 7, 12, 4, 21, 81]
```

```
arr.sort(reverse = True)
```

```
# a = [81, 45, 23, 21, 12, 12, 7, 6, 4, 3]
```

Щоб зрозуміти сутність процесу сортування, розгляньмо два найпростіші алгоритми сортування масиву. Нехай потрібно впорядкувати елементи списку `arr` із 10 елементів за неспаданням:

```
arr[0] ≤ arr[1] ≤ ... ≤ arr[9].
```

Сортування вибором максимального елемента

Алгоритм сортування вибором полягає в пошуку на неопрацьованому зрізі списку максимального значення та подальшому обміні цього значення з останнім елементом неопрацьованого зрізу. На наступному кроці неопрацьований зріз зменшується на один елемент.

Щоб здійснити сортування вибором максимального елемента, слід:

- 1) відшукати максимальний елемент у зрізі `arr[0: 10]`;
- 2) поміняти місцями знайдений елемент і `arr[9]`;
- 3) відшукати максимальний елемент у зрізі `arr[0: 9]`;

4) поміняти місцями знайдений елемент і `arr[8]`;

<...>

5) максимальний елемент зрізу `arr[0: 2]` поміняти місцями з `arr[1]`.

У таблиці наведено вигляд масиву `arr` на кожному кроці сортування (кольоровим тлом виділено відсортовану частину).

0	1	2	3	4	5	6	7	8	9
21	27	15	20	1	29	12	18	26	2
21	15	20	1	27	12	18	26	2	29
15	20	1	21	12	18	26	2	27	29
15	1	20	12	18	21	2	26	27	29
1	15	12	18	20	2	21	26	27	29
1	12	15	18	2	20	21	26	27	29
1	12	15	2	18	20	21	26	27	29
1	12	2	15	18	20	21	26	27	29
1	2	12	15	18	20	21	26	27	29



Реалізуємо алгоритм у вигляді функції. Список `arr` заповнимо 10 випадковими числами.

```
from random import randint
```

```
def sort_select():
```

```
    for k in range(len(arr), 0, -1):
```

```
        mk = max(arr[0 : k])
```

```
        m = arr[0 : k].index(mk)
```

```
        arr[k-1], arr[m] = arr[m], arr[k-1]
```

Переглядається зріз [0: k]

Знайшли максимальний елемент

у зрізі `arr[0: k]`

Визначили його індекс

поміняли місцями з останнім у зрізі

```
arr = []
```

```
for i in range(10):
```

```
    arr.append(randint(1, 30))
```

```
print(arr)
```

```
sort_select()
```

```
print(arr)
```

Заповнення списку випадковими числами

Виклик функції сортування

Приклад виконання:

```
[21, 27, 15, 20, 1, 29, 12, 18, 26, 2]
```

```
[1, 2, 12, 15, 18, 20, 21, 26, 27, 29]
```

Окрім сортування вибором, існують і інші алгоритми сортування: наприклад, сортування обміном (метод бульбашки), сортування вставками, швидке сортування, сортування злиттям. Вони відрізняються способом упорядкування елементів, швидкістю роботи.

Так, за сортування обміном (або бульбашкою) послідовно порівнюють сусідні елементи й міняють місцями, якщо вони розташовані неправильно.

Детальніше про цей метод сортування можна дізнатися за QR-кодом або посиланням rnk.com.ua/114606.



Питання для самоперевірки

- Де може міститися найменший елемент списку, якщо список упорядковано:
 - а) за зростанням;
 - б) за спаданням;
 - в) не впорядковано?
- Початкове значення змінної s дорівнює 0. Для кожної пари сусідніх елементів списку a виконується перевірка: якщо $a[i] \geq a[i+1]$, то значення s збільшується на 1. Визначте, якого значення набуде змінна s , якщо список:
 - а) впорядкований за зростанням;
 - б) упорядкований за спаданням;
 - в) не впорядкований.
- За тиждень зафіксовано такі температури: $[3, -1, 5, 2, 0, 4, -2]$. Упорядкуйте значення за спаданням. Поясніть, як це зробить алгоритм сортування вибором (опишіть перші два проходи).
- Є список прізвищ: ["Іваненко", "Бондар", "Сидоренко", "Андрущенко"]. Відсортуйте його за алфавітом. Поясніть, чим відрізняється сортування чисел від сортування рядків.
- Дано список результатів забігу на 100 м восьми спортсменок. Складіть програму для визначення трьох найкращих результатів.



У галереї програмістів у Музеї комп'ютерної історії в Кремнієвій долині серед портретів піонерів програмування всього світу є і портрет Катерини Логвинівни Ющенко, видатної української фахівчині з програмного забезпечення комп'ютерів, яка стояла біля витоків вітчизняної школи теоретичного програмування.



Вправа 29

- Скласти програму для розв'язування задачі. У фігурному катанні загальна оцінка якості виконання елемента обчислюється як усічене середнє оцінок, даних 9 суддями. Для цього відкидаються найвища та найнижча оцінки, а з решти обчислюється середнє арифметичне. Складіть програму для визначення оцінки за цими правилами.

1. В IDLE виберіть команду File → New File.
Завантажте модуль random: `import random`.
2. Створіть порожній список `grades`.
3. Додайте до списку 9 випадкових чисел з інтервалу (4, 6). Щоб згенерувати випадкове дійсне число, викличте метод `random.uniform(4, 6)` і округліть згенероване число до 1 десяткового знаку:

```
for i in range(9):
    x = random.uniform(4, 6)
    grades = grades+[round(x, 1)]
```

Виведіть список на екран.
4. Упорядкуйте список. Виведіть упорядкований список на екран.
- ⚙ 5. Знайдіть середнє арифметичне елементів, що утворюють зріз списку згідно з умовою задачі.
6. Виведіть результат з одним десятковим знаком.
Збережіть файл з іменем `Vprava29.py`. Перевірте роботу програми.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112949



- Тім Пітерс, розробник і автор «Дзену Python», у 2002 році для стандартного методу сортування списків розробив гібридний алгоритм Timsort. Цей алгоритм виявився настільки ефективним, що згодом його запозичили для використання в Java, Android і навіть у рушії V8 для JavaScript, зробивши архітектурне рішення команди Python стандартом для всієї ІТ-індустрії.

Вибирати елементи списку для виконання потрібних дій зручніше, коли вони наочно показані та доступні в графічному вікні. Таку можливість надають віджети класу `ListBox`.

Об'єкт класу `ListBox` дає змогу виводити список елементів, з якого користувач/користувачка може вибрати один або декілька пунктів.

Створення об'єктів класу `ListBox`

Вам уже відомий **синтаксис створення віджета**:

```
[змінна = ]ListBox([батьківський_віджет, ][атрибут = значення, ...])
```

Для налаштування віджета класу `ListBox` у виклику конструктора можна задати значення атрибутів, як от: `bg`, `fg` (кольори фону й тексту); `font` (тип і кегль шрифту); `width`, `height` (висота й ширина елемента; за промовчанням — висота 10 рядків, ширина 20 символів).

! Важливо задати значення атрибута `selectmode`, яке визначає, скільки елементів списку можна вибрати.

За промовчанням у віджеті натисканням миші можна вибрати тільки один елемент. Якщо встановити `selectmode = EXTENDED`, то можна вибрати декілька елементів, утримуючи натиснутою клавішу `Ctrl`.



Створимо віджет класу `ListBox`:

```
lbox_mova_progr = ListBox(width = 40, selectmode = EXTENDED)
```

```
lbox_mova_progr.pack() # Віджет розміщується у вікні
```

Після створення віджета класу `ListBox` необхідно заповнити список об'єкта елементами.

Щоб додати елемент до списку, слід викликати метод `insert()`:

```
<назва об'єкта>.insert(<індекс>, <значення>)
```

Першим аргументом методу `insert` передається індекс вставлення елемента, другим — його значення.

Щоб додати елемент у кінець списку, замість індексу можна вказати константу `END`.

2

Відобразимо у створеному в прикладі 1 віджеті класу *Listbox* перелік мов програмування зі списку `mova_progr` (рис. 30.1). Для цього в циклі `for` переберемо елементи `mova_progr` і кожен із них додамо до віджета:

```
mova_progr = ["Python", "JavaScript", "C#", "Java",
              "C/C++", "PHP", "Visual Basic.NET", "Ruby"]
# Заповнення за допомогою методу insert()
for mova in mova_progr:
    lbox_mova_progr.insert(END, mova)
```

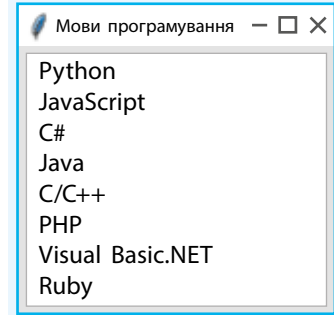


Рис. 30.1

Основні методи *Listbox*

Окрім методу `insert()`, клас *Listbox* має інші корисні методи для керування поведінкою об'єктів та їхнім вмістом. Розгляньмо деякі з цих методів.

- Метод `<назва об'єкта>.curselection()` повертає набір індексів вибраних елементів.

3

Викличемо метод `curselection()` для віджета `lbox_mova_progr`. Метод викликається в обробнику події для кнопки Застосувати метод `curselection()` (рис. 30.2):

```
def btn_click():
    print (lbox_mova_progr.curselection())           # Буде виведено (1, 3, 6)
```

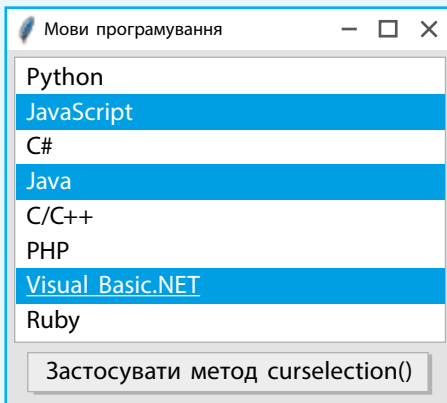


Рис. 30.2



- Метод <назва об'єкта>.**delete**(n) видаляє зі списку Listbox елемент з індексом n.

4

Видалимо елемент з індексом 6.

`lbox.delete(6)`

Вигляд списку після виконання методу подано на рис. 30.3.

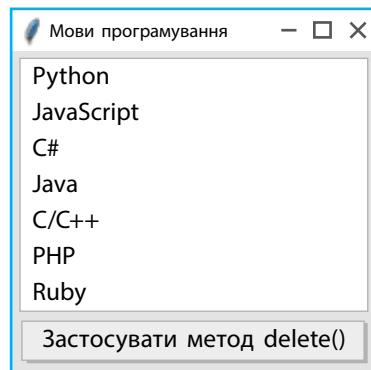


Рис. 30.3

- Метод <назва об'єкта>.**get**(n) повертає текст елемента з індексом n.

5

Вивести в консоль елемент списку `lbox_mova_progr` з індексом 3:
`print(lbox_mova_progr.get(3))` # Буде виведено Java

6

Створити список `elem` з елементів списку `lbox_mova_progr`:
`elem = list(lbox1.get(0, END))`

- Метод <назва об'єкта>.**size**() повертає кількість елементів у списку Listbox.

7

Визначити кількість елементів у списку `lbox_mova_progr`:
`k = lbox_mova_progr.size()`
`print(k)` # Буде виведено 8

Застосування методів класу Listbox

Додавати рядки до списку Listbox можна під час виконання програми, вводячи значення з клавіатури.

8

Запрограмуємо додавання рядків до списку віджета `lbox` класу `Listbox`, уводячи по черзі значення в поле віджета `entry_add` класу `Entry` і натискаючи кнопку `badd` (рис. 30.4).

```
from tkinter import*
def badd_cl():
```

```

lbox.insert(END, entry_add.get()) # Додавання рядка з поля
                                # entry_add до списку lbox
entry_add.delete(0, END)         # Очищення поля entry_add
k = lbox.size()                  # Визначення кількості рядків у списку lbox
lab2.config(text = 'кількість елементів '+str(k)) # Виведення значення k
root = Tk()
lbox = Listbox(root, selectmode = EXTENDED, width = 10, height = 8)
lbox.grid(row = 0, column = 0, rowspan = 5)
                                # Віджет lbox розміщується у вікні
Label(root, text = 'Уведіть число:').grid(row = 0, column = 1)
entry_add = Entry(root) # Створення
                        # текстового поля
entry_add.grid(row = 1, column = 1)
badd = Button(text = "Додати число ",
              command = badd_cl)
badd.grid(row = 2, column = 1)
lab2 = Label(root, text = '****')
lab2.grid(row = 3, column = 1)
root.mainloop()

```

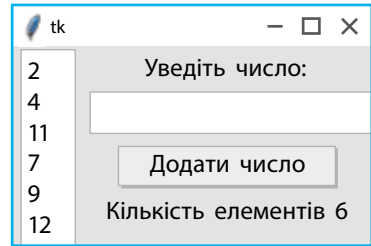


Рис. 30.4

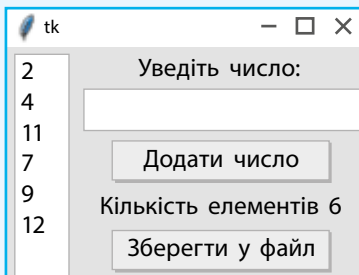


Додамо до коду з прикладу 8 можливість зберігати список віджета *lbox* у текстовому файлі (рис. 30.5). Для цього слід створити кнопку Зберегти у файл і описати для неї обробник події:

```

def saveList():
    f = open('list.txt', 'w')
    f.writelines("\n".join(lbox.get(0, END)))
    f.close()
bsave = Button(root, text = "Зберегти у файл", command = saveList)
bsave.grid(row = 4, column = 1)

```



Метод `join()` перетворює рядки, які повернув метод `get()`, в один рядок із додаванням роздільника `'\n'`, щоб елементи списку записалися у файл в окремих рядках

Рис. 30.5



Обчислимо суму чисел, які містяться в списку `lbox`, створеному в прикладі 8 (рис. 30.6).

У вікно програми слід додати кнопку Знайти суму та описати в коді обробник події для неї:

```
def bsum_cl():
    elem = list(map(int, lbox.get(0, END)))
    s = sum(elem)
    lab2.config(text = 'Сума елементів '+str(s))
    <...>
bsum = Button(text = "Знайти суму",
               command = bsum_cl)
bsum.grid(row = 4, column = 1)
```

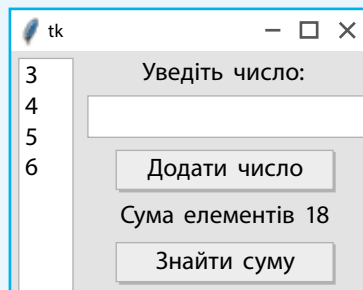


Рис. 30.6



Обчислимо суму чисел, які містяться у вибраних рядках списку `lbox`, створеному в прикладі 8 (рис. 30.7).

Функцію `bsum_cl()` слід змінити так:

```
def bsum_cl():
    select = list(lbox.curselection()) # Список індексів вибраних елементів
    s = 0
    for i in select:
        s = s+int(lbox.get(i))         # Додавання елементів зі списку lbox,
                                     # індекси яких містяться в списку select
    lab2.config(text = 'Сума вибраних чисел '+str(s))
```

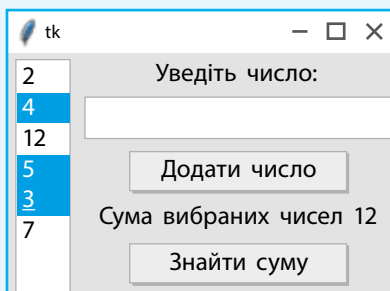


Рис. 30.7



Отже, віджет `Listbox` дає змогу унаочнити роботу зі списками, полегшити опрацювання значень у списку, вибір елементів тощо.

- IT-послуги є третьою за величиною статтею експорту України (у 2025 році він сягнув понад 6,66 млрд доларів США). Крім того, в Україні народилися такі технологічні гіганти, як Grammarly (оцінка понад 13 млрд доларів США), GitLab і Preply; активно впроваджується штучний інтелект у безпілотні системи, розвивається робота з BigData (із величезними масивами даних).



Питання для самоперевірки

1. Поясніть алгоритм створення об'єкта класу *Listbox*.
2. Створіть об'єкт *lb* класу *Listbox*. Заповніть список 5 випадковими цілими числами.
3. Виведіть у консоль число елементів у списку *lb*.
4. Виведіть у консоль індекс вибраного елемента списку *lb*.
5. Виведіть у консоль текст 3-го елемента списку *lb*.
6. Видаліть 4-й елемент списку *lb*.



Вправа 30

- Написати програму Покупки, що складається з двох списків *Listbox*. Перший список містить перелік товарів, другий — перелік покупок. При натисканні на одну кнопку товар має переходити з одного списку в інший. При натисканні на другу кнопку — повертатися (людина передумала купувати). Забезпечити можливість множинного вибору елементів списку та їх переміщення.
1. В IDLE виберіть команду *File* → *New File*. Завантажте модуль *tkinter*. Створіть вікно програми із заголовком Покупки. Завершіть програму командою `root.mainloop()`.
 2. Додайте у вікно програми віджет *box1* класу *Listbox*, занесіть до списку перелік товарів:


```
box1 = Listbox(selectmode = EXTENDED)
box1.grid(row = 0, column = 0, rowspan = 6)
products = ["ЯБЛУКА", "БАНАНИ", "МОРКВА", "ХЛІБ", "МОЛОКО", "КАВА",
            "ЧАЙ", "ЦУКОР", "ПЕЧИВО", "ЦУКЕРКИ"]
for p in products:
    box1.insert(END, p)
```

 Додайте у вікно віджети *btn1*, *btn2* класу *Button* (рис. 30.8, с. 188).

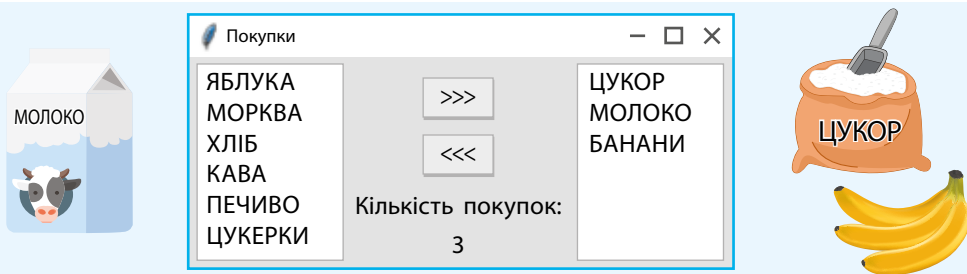


Рис. 30.8

3. Створіть об'єкт box2 класу Listbox, залиште його порожнім.
4. Опишіть обробник події для кнопки btn1 (перенесення елементів зі списку box1 до списку box2):

```
def to_box2():
    select = list(box1.curselection())
    select.reverse()
    for i in select:
        box2.insert(END, box1.get(i))
        box1.delete(i)
```

Підказка. У функції to_box2 () із вибраних елементів формується список, порядок елементів якого змінюється на обернений. Це робиться, щоб видалення елементів починалося з кінця списку.

Опишіть обробник події для кнопки btn2 (перенесення елементів зі списку box2 до списку box1).

5. Додайте віджет lab1 класу Label із заголовком Кількість покупок і віджет lab2 класу Label для виведення кількості елементів у списку box2.

Додайте до описів функцій команди для змінювання значення атрибута text віджета lab2 на значення box_2.size():

```
k = box2.size()
lab2.config(text = str(k))
```

6. Збережіть файл з іменем Vprava30.py. Перевірте роботу програми.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112950



Уявіть собі, що ви готуетесь до доповіді. Чи виглядатиме вона наочніше, якщо числові дані подати на кольоровій діаграмі? Як ви знаєте, діаграми будуються на основі числових даних, що містяться в таблицях, і допомагають аналізувати й порівнювати дані.

Побудова діаграм із використанням графічних методів

Діаграма — зображення, на якому числові дані подано геометричними фігурами. Розгляньмо способи побудови діаграм із використанням методів модуля `tkinter` і засобів бібліотеки `matplotlib`.

Діаграми можна створювати, програмуючи креслення графічних примітивів (відрізок, прямокутник тощо) методами полотна `Canvas`, знайомими вам із курсу 8 класу.

► Програмування побудови гістограми

Гістограма (стовпчикова діаграма) використовується для порівняння елементів табличної величини.

Побудуймо гістограму, яка ілюструє показник успішності учнів 9-А класу з декількох предметів, значення якого занесено в список `ndu` (рис. 31.1). Областю побудови діаграми буде полотно (об'єкт класу `Canvas`) розмірами `sw`×`sh` пікселів.

Щоб запрограмувати побудову діаграми, з'ясуймо, як пов'язані координати й розміри її частин із розмірами полотна. Слід пам'ятати, що в `tkinter` початок координат полотна лежить у його лівому верхньому куті, вісь `x` напрямлена праворуч, вісь `y` — униз (рис. 31.2, с. 190).

Нехай діаграма містить `k` стовпців. Тоді для ширини `ab` стовпця разом із проміжком можна взяти величину `sw/k`. Оскільки після останнього стовпця проміжку немає, це дасть змогу залишити пообіч діаграми вільне місце шириною `f = b/2`.

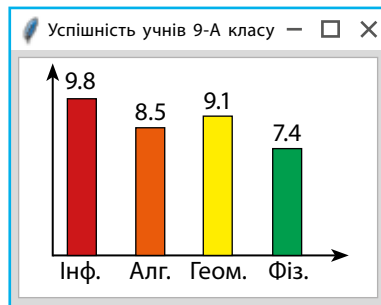


Рис. 31.1

Для правильного розташування кожного стовпця потрібно розрахувати координати x_r і y_r його верхнього лівого кута. Координати x_r сусідніх стовпців відрізняються на ab , тобто $x_r = f + i * ab$ (тут i — номер стовпця, починаючи з 0).

Нижні краї стовпців розмістимо на одному рівні: $y_l = ch / 1.1$. Для розрахунку висоти, а отже й y_r , знадобиться масштабний множник. Його розрахуємо за найбільшим значенням у списку ndu : $step = ly / \max(ndu) / 1.1$.

У двох попередніх формулах ділення на 1,1 забезпечує місце для написів над і під діаграмою. Додамо їх викликом методу `create_text(x, y, text)`. Координати написів розрахуємо на основі x_r , y_r і y_l (див. програмний код).

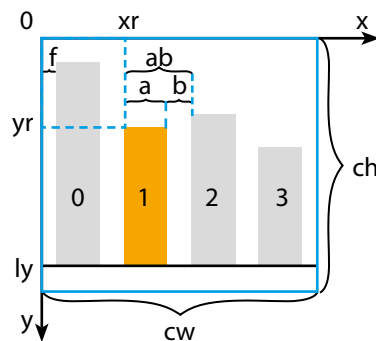


Рис. 31.2



Програмний код, який реалізує побудову діаграми, може бути, наприклад, таким:

```
from tkinter import *
root = Tk()
root.title("Успішність учнів 9-А класу")
colors = ['red', 'orange', 'yellow', 'green']
subject = ['Інф.', 'Алг.', 'Геом.', 'Фіз.']
ndu = [9.8, 8.5, 9.1, 7.4]
k = len(ndu)
cw, ch = 300, 220
ab = cw / k
a, b = ab * 0.6, ab * 0.4
f = b / 2
ly = ch / 1.1
step = ly / max(ndu) / 1.1
canvas = Canvas(root, width = cw, height = ch)
canvas.pack()
canvas.create_line(0, ly, cw, ly)
for i in range(k):
    x_r = f + i * ab
```



```

yr = ly - ndu[i]*step
canvas.create_rectangle(xr, yr, xr+a, ly, fill = colors[i])
canvas.create_text(xr+a/2, ly + 10, text = subject[i])
canvas.create_text(xr+a/2, yr-10, text = str(ndu[i]), font = "Arial 13")
root.mainloop()

```

Завдяки розрахунку параметрів діаграми на основі даних таблиці, програма має гнучкість: змінюючи списки `colors`, `subject`, `ndu` та значення `sw`, `ch`, можна будувати діаграми з різною кількістю стовпців та різними розмірами.

► Програмуємо побудову графіка

Графік дає змогу проілюструвати, наприклад, змінення значення величини протягом певного проміжку часу.

Координатні осі зобразимо відрізками. Для малювання стрілок до переліку параметрів методу `create_line()` додамо параметри `arrow` та `arrowshape`. Атрибуту `arrow` надамо значення `LAST` (задає розміщення стрілки в кінці лінії), значення атрибута `arrowshape` задає розміри стрілки.



Побудуємо графік середньодобової температури повітря за тиждень. Значення температури зберігаються в списку `a` (рис. 31.3).



Рис. 31.3

```

from tkinter import*
root = Tk()
canvas = Canvas(root, width = 300, height = 220, bg = 'white')
canvas.pack()
a = [14, 12, 15, 12, 9, 10, 7]

```

```
# Побудова координатних осей
canvas.create_line(10, 200, 10, 10, arrow = LAST, arrowshape = "10 20 10")
canvas.create_line(10, 200, 280, 200, arrow = LAST, arrowshape = "10 20 10")
day = ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Нд']
x1 = 30
y1 = 220-a[0]*10
canvas.create_text(x1, 210, text = day[0], anchor = W)
canvas.create_text(x1, y1-10, text = str(a[0]), anchor = W, font = "Arial 12")
for i in range(1, 7):
    x = x1+30
    y = 220-a[i]*10
    canvas.create_line(x1, y1, x, y, width = 3, fill = "lightblue")
    canvas.create_text(x, 210, text = day[i], anchor = W)
    canvas.create_text(x, y-10, text = str(a[i]), anchor = W, font = "Arial 12")
    x1 = x
    y1 = y
root.mainloop()
```

Побудова діаграм засобами бібліотеки matplotlib

Matplotlib — це багатоплатформна бібліотека візуалізації даних і побудови діаграм для Python.

Перш ніж використовувати засоби бібліотеки matplotlib, її потрібно встановити. Після цього можна в Python-файлі підключати модуль pyplot із цієї бібліотеки, який містить добірку функцій для створення діаграм і налаштування їхнього вигляду.

Імпортуємо модуль pyplot і для зручності позначимо його як plt.

```
import matplotlib.pyplot as plt
```

Для виведення зображення на екран використовується функція plt.show().



Побудуємо стовпчикову діаграму для відображення елементів списку data:

```
import matplotlib.pyplot as plt
data = [5., 25., 50., 20.]
plt.bar(range(len(data)), data)
plt.show()
```



Діаграму побудовано в окремому графічному вікні, яке має заголовок Figure 1 (рис. 31.4).

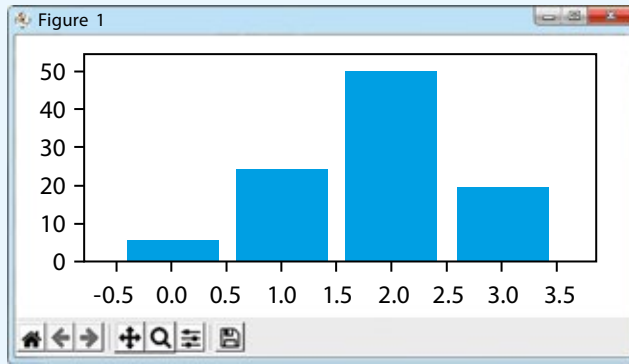


Рис. 31.4

У таблиці пояснено призначення кнопок панелі інструментів, розташованої внизу вікна діаграми:

	Повертає графік до первісного стану (як при першому запуску)
	Переміщує назад/уперед історію зміни графіка
	Перетягує (ліва кнопка миші) або встановлює масштаб графіка (права кнопка миші)
	Збільшує зображення
	Відкриває додаткові налаштування вікна (наприклад, відступи від країв)
	Викликає діалог збереження графіка у файл

Для підвищення наочності діаграми можна налаштувати властивості title (заголовок), xlabel, ylabel (підписи осей), grid (координатна сітка). У прикладі 3 функції plt.bar() передано тільки один список чисел, і matplotlib пов'язує цей список із послідовністю чисел x: 0, 1, 2, 3, Якщо ж точку графіка задають парою значень (x, y), то слід передати два масиви: у першому — значення для осі x, а в другому — для y.

4

Побудуємо діаграму з прикладу 1 (рис. 31.5).

```
import matplotlib.pyplot as plt
plt.title('Успішність учнів 9-А класу')
ndu =[9.8, 8.5, 9.1, 7.4]
subject = ['Інф.', 'Алг.', 'Геом.', 'Фіз.']
plt.xlabel('Предмети', color = 'gray')
plt.ylabel('Середній бал', color = 'gray')
plt.grid(True)
plt.bar(subject, ndu)
plt.show()
```



Рис. 31.5

5

Для побудови графіків призначена функція plot(). Побудуємо графік із прикладу 2, задавши синій колір лінії (параметр 'b') (рис. 31.6).

```
import matplotlib.pyplot as plt
plt.title('Середньодобова температура за тиждень')
a = [14, 12, 15, 12, 9, 10, 7]
day = ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Нд']
plt.xlabel('Дні тижня', color = 'gray')
plt.ylabel('Середньодобова температура', color = 'gray')
plt.grid(True)
plt.plot(day, a, 'b')
plt.show()
```



Рис. 31.6

6

Побудуємо кругову діаграму прибутку підприємства за 4 квартали року за наведеною таблицею. Результат наведено на рис. 31.7.

	Прибуток, тис. грн
I квартал	100
II квартал	40
III квартал	80
IV квартал	70

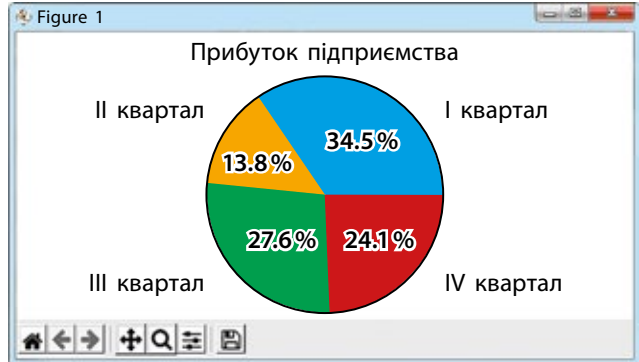


Рис. 31.7

```
import matplotlib.pyplot as plt
plt.title('Прибуток підприємства')
a = [100, 40, 80, 70]
kv = ['I квартал ', 'II квартал ', 'III квартал ', 'IV квартал ']
plt.pie(a, labels = kv, autopct = "%.1f%%")
plt.show()
```

Атрибут autopct дає змогу відображати частку кожного значення.



Питання для самоперевірки

1. Обговоріть переваги й недоліки двох способів побудови діаграм.
2. Опишіть алгоритм побудови графіка певної величини з використанням графічних методів модуля tkinter.
3. Назвіть методи класу `matplotlib.pyplot`, призначені для візуалізації табличних величин.
4. Поясніть призначення інструментів інтерактивної навігації графічного вікна.
5. Додайте до проєкту Календар спостережень за погодою (див. вправа 28) процедуру побудови графіка кількості опадів за місяць.



Вправа 31

- У таблиці наведено дані про обсяг продажів книжкового магазину за місяцями за період із січня по червень.

	Січень	Лютий	Березень	Квітень	Травень	Червень
Обсяг продажів, тис. грн	11,0	17,0	18,0	32,5	22,5	15,5

- 1) Визначити сумарний обсяг продажів за півріччя.
- 2) Побудувати: а) гістограму обсягу продажів за місяцями; б) графік обсягу продажів за місяцями.
1. В IDLE виберіть команду `File → New File`. Завантажте модулі `tinker`, `matplotlib.pyplot`. Створіть вікно програми із заголовком Обсяг продажів (рис. 31.8). Завершіть програму командою `root.mainloop()`.
2. Збережіть масиви назв місяців і значень обсягу продажів у списках `month`, `sales`.
`month = ['Січень', 'Лютий', 'Березень', 'Квітень', 'Травень', 'Червень']`
`sales = [11.0, 17.0, 18.0, 32.5, 22.5, 15.5]`
 Задайте заголовок вікна діаграми:
`plt.title('Обсяг продажів')`
3. Додайте у вікно програми віджет `box1` класу `Listbox`, занесіть до списку значення списку `sales`:
`box1 = Listbox(selectmode = EXTENDED)`
`box1.grid(row = 0, column = 0, rowspan = 6)`
 for p in sales:
 `box1.insert(END, p)`

4. Додайте у вікно віджет btn1 класу Button:

```
btn1 = Button(text = "Стовпчикова діаграма", command = btn1_cl)
btn1.grid(row = 2, column = 1)
```

Опишіть функцію btn1_cl(), яка містить команди для побудови стовпчикової діаграми за значеннями елементів масиву sales:

```
plt.xlabel('Місяці', color = 'gray')
plt.ylabel('тис. грн', color = 'gray')
plt.bar(month, sales)
plt.show()
```

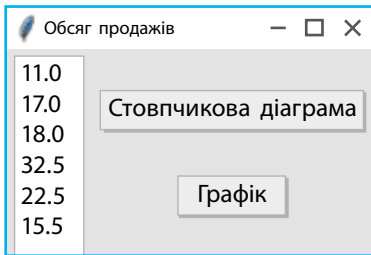


Рис. 31.8



Рис. 31.9

5. Додайте у вікно віджет btn2 класу Button:

```
btn2 = Button(text = "Графік", command = btn2_cl)
btn2.grid(row = 4, column = 1)
```

6. Опишіть функцію btn2_cl(), призначену для побудови графіка за значеннями елементів масиву sales.

Збережіть файл з іменем Vprava31.py.

Випробуйте роботу програми.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112951





ПРАКТИЧНА РОБОТА 9

СКЛАДАННЯ І ВИКОНАННЯ АЛГОРИТМІВ ЗНАХОДЖЕННЯ СУМ І КІЛЬКОСТЕЙ ЗНАЧЕНЬ ЕЛЕМЕНТІВ ТАБЛИЧНИХ ВЕЛИЧИН ЗА ЗАДАНИМИ УМОВАМИ

Завдання: створити програму для розв'язання задачі.

Дано таблицю кількості опадів протягом року за місяцями:

Номер місяця	1	2	3	4	5	6	7	8	9	10	11	12
Кількість опадів, мм	54	12	30	15	40	32	10	30	46	60	56	38

Визначити:



- кількість опадів за рік;
- у якому місяці кількість опадів була найменшою;
- у якому місяці кількість опадів була найбільшою;
- у які місяці кількість опадів була меншою від 40, кількість таких місяців.

Хід виконання роботи — за QR-кодом або посиланням rnk.com.ua/115069.

Обладнання: комп'ютер із середовищем програмування IDLE Python.

Хід роботи

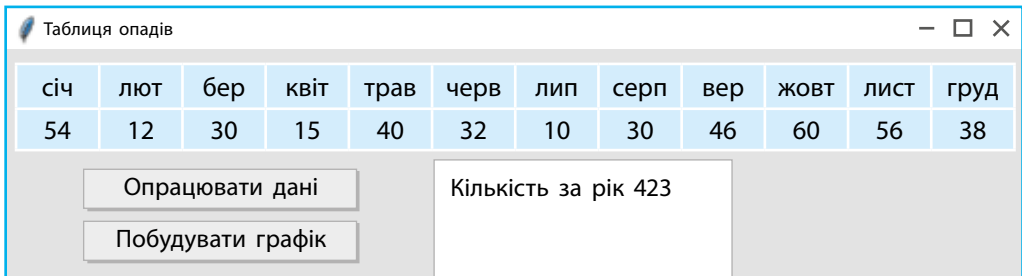
Під час роботи за комп'ютером дотримуйтеся правил безпеки.

- В IDLE виберіть команду File → New File. Завантажте модулі tkinter, matplotlib.pyplot.
- Створіть вікно програми із заголовком Таблиця опадів. Завершіть програму командою root.mainloop().
- Створіть списки для збереження масивів назв місяців і кількості опадів:

```
month = ['січ', 'лют', 'бер', 'квіт', 'трав', 'черв', 'лип', 'серп', 'вер', 'жовт', 'лист', 'груд']
mas = [54, 12, 30, 15, 40, 32, 10, 30, 46, 60, 56, 38]
```
- Створіть масив об'єктів класу Label, які будуть відображати назви місяців:

```
list_1 = []
for i in range (12):
    list_1.append(Label(text = month[i], bg = 'lightblue', width = 4))
    list_1[i].grid(row = 0, column = i, padx = 2, pady = 2)
```

5. Створіть масив list_2 об'єктів класу Label, який буде відображати кількість опадів (значення зі списку mas).
6. Створіть командні кнопки btn1 Опрацювати дані, btn2 Побудувати графік (див. рисунок).



січ	лют	бер	квіт	трав	черв	лип	серп	вер	жовт	лист	груд
54	12	30	15	40	32	10	30	46	60	56	38

Опрацювати дані

Побудувати графік

Кількість за рік 423

7. Створіть віджет box1 класу Listbox:


```
box1 = Listbox()
box1.grid(row = 3, column = 4, rowspan = 5, columnspan = 5)
```
8. Опишіть функцію btn1_cl() — обробник події для кнопки btn1. Обчисліть суму елементів списку mas, додайте результат до списку box1:


```
def btn1_cl():
    s = sum(mas)
    box1.insert(END, "Кількість за рік "+str(s))
```
9. Доповніть функцію btn1_cl(). Додайте до списку box1 назву місяця, у якому кількість опадів була найменшою.
10. Доповніть функцію btn1_cl(). Додайте до списку box1 назву місяця, у якому кількість опадів була найбільшою.
11. Доповніть функцію btn1_cl(). За алгоритмом пошуку елементів із заданою властивістю знайдіть кількість місяців, у яких опадів було менше за 40 мм, і виведіть до списку box1 назви цих місяців.
12. Опишіть функцію btn2_cl() — обробник події для кнопки btn2, призначену для побудови графіка за значеннями списку mas. Збережіть файл з іменем Pract9.py.

Зробіть висновок: як застосовувати методи списків і типові алгоритми опрацювання табличних величин.

Списки зручно застосовувати для опрацювання таблиць, які містять один рядок даних, наприклад, таблиця щомісячних опадів за один рік. А якщо потрібно проаналізувати кількість опадів за місяцями за кілька років? Тут знадобляться таблиці з кількох рядків і стовпців.

Поняття двовимірного масиву

Двовимірний масив являє собою таблицю з n рядків і m стовпців.



Двовимірний масив — це масив, у якому для задання розташування елемента необхідно вказати значення двох індексів.

Перший індекс двовимірного масиву вказує номер рядка, а другий — номер стовпця.



Збережемо дані таблиці (рис. 32.1), у змінній a :
 $a = [[1, 2, 3, 12], [23, 5, 13, 4], [8, 6, 7, 11]]$

Кожний елемент списку a також є списком, що містить дані з одного рядка таблиці (тобто маємо список списків). Довжина списку $\text{len}(a) = 3$. Звернутися до вкладеного списку (рядка таблиці) можна за його індексом: $a[0] = [1, 2, 3, 12]$.

1	2	3	12
23	5	13	4
8	6	7	11

$a[1] = [23, 5, 13, 4]$

Рис. 32.1

Для перебору рядків списку використовується цикл `for`.

Нехай $b = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]$. Розгляньмо два варіанти:

Можна перебрати індекси вкладених списків — елементів списку	Можна перебрати наявні в списку елементи — рядки таблиці
for i in range ($\text{len}(b)$): $\text{print}(b[i])$	for row in b : $\text{print}(\text{row})$

При використанні обох варіантів циклу в консоль буде виведено результат, який наведено на рис. 32.2.

[1, 2, 3, 4]
[5, 6]
[7, 8, 9]

Рис. 32.2

Отже, щоб звернутися до елемента у вкладеному списку, слід вказати два індекси. Елемент, розташований на перетині i -го рядка та j -го стовпця масиву A , позначають $A[i][j]$.

Елементи списку a з прикладу 1 мають такі значення:

$a[0][0] == 1$, $a[0][1] == 2$, $a[1][0] == 23$, $a[1][1] == 5$, $a[1][2] == 13$.

Послідовно отримати доступ до кожного елемента двовимірного масиву можна за допомогою вкладених циклів:

- зовнішній цикл перебирає рядки таблиці;
- внутрішній цикл перебирає елементи рядка.



Виведемо значення списку a на екран порядково, додаючи пропуск між елементами. Зовнішній цикл `for i` перебирає індекси елементів списку a — рядків двовимірного масиву. Внутрішній цикл `for j` перебирає індекси елементів списку $a[i]$ (i -го рядка масиву).

Згадаємо, що значення `end = ''` методу `print()` потрібне, щоб після виведення курсор залишився в тому самому рядку. Команда `print()` із порожнім списком виведення переводить курсор на наступний рядок.

```
a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]
for i in range(len(a)):
    for j in range(len(a[i])):
        print(a[i][j], end = ' ')
    print()
```



Щоб вивести в консоль значення двовимірного масиву b порядково, можна в зовнішньому циклі перебирати рядки, а у внутрішньому циклі — наявні в рядку елементи:

```
b = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]
for row in b:
    for elem in row:
        print(elem, end = ' ')
    print()
```

```
1 2 3 4
5 6
7 8 9
>>>
```

Результат наведено на рис. 32.3.

Рис. 32.3



Двовимірний масив, у якому рядки мають різну довжину, називають зубчастим масивом.

Надалі переважно розглядатимемо двовимірні масиви з однаковою довжиною рядків.

Створення двовимірного масиву

Нехай потрібно створити двовимірний масив для збереження даних прямокутної таблиці, у якій n рядків і m стовпців, і заповнити його нулями. Це можна зробити в такий спосіб: створити порожній список, а потім додати до нього n елементів-списків — рядків масиву.

```
n = 3
m = 4
a = []
for i in range(n):
    a.append([0]*m)
```

Також це можна зробити з використанням генератора. Для заповнення чергового елемента списку a щоразу генеруємо список $[0]*m$:

```
n = 3
m = 4
a = [[0]*m for i in range(n)]
```

Введення елементів

Щоб створити двовимірний масив із n рядків, уводячи значення елементів із клавіатури, потрібно в циклі, що виконується n разів, зчитати рядок чисел, розділених пропусками. Далі слід звести введені дані за допомогою функції `map` до числового типу, і сформувати функцією `list` список, який потім додати до основного (спочатку порожнього), створюючи таким чином двовимірну структуру.

```
n = int(input())          # Кількість рядків масиву
a = []                   # Створюється порожній список
for i in range(n):
    row = list(map(int, input().split())) # Зчитується та формується один
                                           # рядок масиву як список цілих чисел
    a.append(row)         # Список row додається в кінець основного списку a
```



Заповнимо двовимірний масив із 3 рядків і 4 стовпців випадковими числами:

```
from random import*
a = [[], [], []]          # Порожній двовимірний масив із трьох рядків
for i in range(len(a)):
    for j in range(4):
        a[i] = a[i]+[randint(1,10)] # Додавання елемента до i-го рядка
print(a[i])
```

5

Якщо значення елемента залежить від його індексів, то обчислити елементи двовимірного масиву можна за певною формулою. Нехай кожний елемент масиву 3×4 дорівнює більшому з його індексів:

```
a = [], [], []
```

```
for i in range(len(a)):
```

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

```
        a[i] = a[i]+[max(i, j)]
```

```
print(a[i])
```

Результат наведено на рис. 32.4.

0	1	2	3
1	1	2	3
2	2	2	3

Рис. 32.4

6

Елементи, для яких $i == j$, утворюють головну діагональ масиву. Заповнимо двовимірний масив 5×5 у такий спосіб: елементам головної діагоналі присвоїти значення 1, елементам, що розташовані над нею — значення 2, елементам під головною діагоналлю — значення 0 (рис. 32.5):

```
a = [], [], [], [], []
```

```
for i in range(5):
```

```
    for j in range(5):
```

```
        if i == j: a[i].append(1)
```

```
        elif i < j: a[i].append(2)
```

```
        else: a[i].append(0)
```

```
print(a[i])
```



1	2	2	2	2
0	1	2	2	2
0	0	1	2	2
0	0	0	1	2
0	0	0	0	1

Рис. 32.5



Питання для самоперевірки

- Назвіть кількість рядків і кількість стовпців у двовимірному масиві, для збереження якого створено такий список:

а) $a = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1]]$;

б) $c = []$ for i in $\text{range}(10)$; в) $b = [], [], []$.

- Збережіть дані (рис. 32.6) у змінній `mas`.

- Запишіть фрагмент коду для введення значень двовимірного масиву з 5 рядків та 5 стовпців із клавіатури та збереження значень його елементів у списку списків.

1	2	3	4
8	7	6	5
9	10	11	12

Рис. 32.6

- Створіть список списків для збереження значень двовимірного масиву з 4 рядків і 6 стовпців; елементи — випадкові числа.

- Запишіть команди виведення значень двовимірного масиву A .

- Визначте, який вигляд має таблиця $A[4, 5]$, якщо:

а) $A[i, j] = i$; б) $A[i, j] = \max(i, j)$; в) $A[i, j] = |i-j|$; г) $A[i, j] = 1$.

Вправа 32

- Територіальна громада ухвалила рішення здавати в оренду частину території, що має форму квадрата розміром 5×5 км. Вартість оренди кожної квадратної ділянки 1×1 км визначено з урахуванням локальних умов і занесено в таблицю а. Із метою організації відкритого тендеру на оренду на вебсайті громади оприлюднено карту території.



Напишіть програму, що дає змогу розрахувати вартість оренди для довільної прямокутної групи суміжних ділянок, якщо протилежні кути групи ділянок розташовані в елементах таблиці з індексами $(i1, j1)$, $(i2, j2)$.

1. В IDLE виберіть команду File → New File.

Завантажте модулі tkinter, random. Створіть вікно програми tk із заголовком Вартість оренди. Завершіть програму командою tk.mainloop().

2. Створіть двовимірний масив об'єктів класу Label, який буде відображати карту території.

```
list_b = [[], [], [], [], []]
```

```
for i in range(5):
```

```
    for j in range(5):
```

```
        lab = Label(tk, bg = 'lightblue', width = 4)
```

```
        list_b[i].append(lab)
```

```
        list_b[i][j].grid(row = i, column = j, padx = 2, pady = 2)
```

3. Опишіть функцію output(), призначену для виведення значень елементів масиву а до елементів масиву list_b.

```
def output():
```

```
    for i in range(5):
```

```
        for j in range(5):
```

```
            list_b[i][j].config(text = str(a[i][j]))
```

4. Заповніть масив а розмірами 5×5 випадковими числами в діапазоні (1, 10) і запишіть команду виклику функції output().



5. Додайте до вікна напис Координати верхнього лівого кута і створіть два об'єкти класу Entry для введення індексів ділянки, що розміщена в лівому верхньому куті групи ділянок.

```
Label(tk, text = 'Координати верхнього лівого кута:').grid(row = 5,  
    column = 0, columnspan = 5)
```

```

enx1 = Entry(tk, width = 3)
enx1.grid(row = 6, column = 1)
eny1 = Entry(tk, width = 3)
eny1.grid(row = 6, column = 3)

```

Додайте до вікна напис Координати нижнього правого кута і створіть два об'єкти класу Entry для введення індексів ділянки, що лежать у правому нижньому куті групи ділянок.

Додайте об'єкт envart класу Entry для виведення загальної вартості оренди визначеної групи ділянок:

```

envart = Entry(tk, width = 6)
envart.grid(row = 10, column = 2)

```

6. Створіть командну кнопку — об'єкт класу Button:
 Button(text = 'Обчислити', command = but_click,
 width = 14).grid(row = 9, column = 0, columnspan = 5)
 Опишіть функцію but_click() — обробник події натискання на кнопку.

```

def but_click():
    x1 = int(enx1.get())
    y1 = int(eny1.get())
    x2 = int(enx2.get())
    y2 = int(eny2.get())
    var = 0
    for i in range(x1, x2+1):
        for j in range(y1, y2+1):
            var = var+a[i][j]
    envart.delete(0, END)
    envart.insert(0, str(var))

```

Випробуйте роботу програми (рис. 32.7).
 Збережіть файл з іменем Vprava32.py.



Рис. 32.7



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112952





ПРАКТИЧНА РОБОТА 10

РОБОТА З ДВОВИМІРНИМИ МАСИВАМИ У ПРОГРАМІ З ГРАФІЧНИМ ІНТЕРФЕЙСОМ

Завдання: запрограмувати введення з клавіатури й виведення значень двовимірного масиву за допомогою графічних компонентів.

Двовимірний масив a складається з 4 рядків і 4 стовпців.

Організувати заповнення цього масиву:

1) за формулою $a[i][j] = (i+1)*(j+1)$ — завдання № 1;

2) випадковими числами — завдання № 2;

3) за формулою $a[i][j] = \begin{cases} 0, & \text{якщо } i \leq j, \\ 1, & \text{якщо } i > j \end{cases}$ — завдання № 3.

Хід виконання роботи — за QR-кодом або посиланням rnk.com.ua/115070.

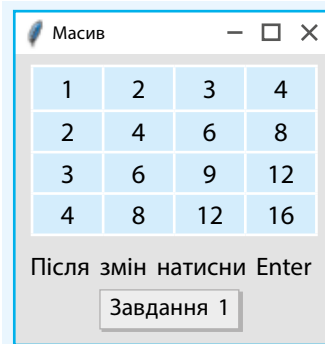
Обладнання: комп'ютер із середовищем програмування IDLE.



Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

- В IDLE виберіть команду File → New File.
Створіть вікно програми tk із заголовком Масив (див. рисунок). Завершіть програму командою `tk.mainloop()`.
- Створіть двовимірний масив a розмірами 4×4 , заповнений нулями:
`a = [[0]*4 for i in range(4)]`
- Створіть двовимірний масив об'єктів класу Entry:
`list_en = [], [], [], []`
`for i in range(len(a)):`
 `for j in range(len(a[i])):`
 `en = Entry(tk, bg = 'lightblue', width = 6)` # Створюємо об'єкт Entry
 `en.grid(row = i, column = j, padx = 2, pady = 2)`
 `list_en[i].append(en)` # Додаємо об'єкт en до i -го рядка
 # масиву `list_en`
- Додайте до вікна програми віджет класу Label для виведення тексту "Після змін натисни Enter".
`Label(tk, text = "Після змін натисни Enter").grid(row = 5, column = 0, columnspan = 3)`



5. Запишіть команду виклику для об'єкта tk методу bind() для обробки натискання клавіші Enter.
tk.bind('<Return>', read)
6. Опишіть функцію read(), призначену для зчитування значень із текстових полів масиву list_en і занесення їх до масиву a:
def read(event):
 for i in range(len(a)):
 for j in range(len(a[i])):
 if list_en[i][j].get().isdigit():
 x = int(list_en[i][j].get())
 a[i][j] = x
 else: print('Не введено: ', i, j)
7. Опишіть функцію output(), призначену для виведення значень елементів масиву a до полів елементів масиву list_en.
def output():
 for i in range(len(a)):
 for j in range(len(a[i])):
 list_en[i][j].delete(0, END)
 list_en[i][j].insert(0, str(a[i][j]))
8. Створіть командну кнопку із написом 'Завдання 1' (див. рисунок).
Button(text = 'Завдання 1', command = b1_click,
 width = 14).grid(row = 6, column = 0, columnspan = 5)
9. Опишіть функцію — обробник події b1_click() для завдання №1.
def b1_click():
 for i in range(len(a)):
 for j in range(len(a[i])):
 a[i][j] = (i+1)*(j+1)
 output() # Виклик функції для виведення значень елементів
 # масиву a до полів елементів масиву list_en
10. Створіть командну кнопку із написом 'Завдання 2', атрибуту command надайте значення b2_click.
11. Опишіть функцію — обробник події b2_click() для завдання №2.
12. Створіть командну кнопку із написом 'Завдання 3'.
Опишіть функцію b3_click() — обробник події для цієї кнопки (завдання №3).
Збережіть файл з іменем Pract10.py.



Зробіть висновок: як організувати введення та виведення елементів двовимірному масиву.

До типових задач опрацювання двовимірних масивів належать завдання на обчислення суми, добутку, кількості, середнього арифметичного, максимуму, мінімуму елементів кожного рядка, кожного стовпця або деякої частини масиву.

Уявіть, що ви керуєте фірмою й вам потрібно проаналізувати динаміку продажів товарів (див. таблицю).

Продаж телефонів, шт.				
№	Марка	Січень	Лютий	Березень
1	Nokia	80	110	120
2	Samsung	80	90	80
3	Alcatel	60	70	130



Якщо вас цікавить сумарний обсяг продажів за січень, то потрібно виконати додавання елементів першого стовпця таблиці числових даних про кількість проданих телефонів.

Щоб знайти загальну кількість проданих телефонів марки Samsung, треба знайти суму елементів другого рядка.

Щоб знайти загальну кількість проданих телефонів усіх марок за квартал, то потрібно обчислити суму всіх елементів масиву.

Додавання значень елементів двовимірного масиву

Рядки двовимірного масиву зберігаються як списки, тому при опрацюванні масиву за рядками зручно застосовувати методи роботи зі списками.



Щоб знайти суму елементів 2-го рядка масиву $a(n \times m)$, можна викликати метод `sum()` для списку, в якому зберігаються елементи 2-го рядка: `s = sum(a[2])`

2

Щоб знайти суму елементів двовимірного масиву `a`, потрібно перебрати всі елементи списку `a` (рядки двовимірного масиву), і застосувати функцію `sum` до кожного рядка.

```
s = 0
```

```
for i in range(len(a)):          # До значення суми додається сума
    s = s+sum(a[i])              # елементів i-го рядка
print(s)
```

Опрацювати двовимірний масив за стовпцями можна за допомогою вкладених циклів: зовнішній цикл перебирає індекси стовпців, а внутрішній — індекси рядків.

3

Знайдемо суму 2-го стовпця масиву (рис. 33.1).

```
a = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1]]
```

```
s = 0
```

```
for i in range(len(a)):
    s = s+a[i][2]
print(s)    # Буде надруковано 13
```

$j = 2$

1	2	3	4
5	6	1	1
7	8	9	1

Рис. 33.1

4

Знайдемо суми стовпців масиву `a` (рис. 33.2).

```
for j in range(4):    # Зовнішній цикл по стовпцях
```

```
    s = 0
```

```
    for i in range(len(a)):

```

```
        s = s+a[i][j] # Додавання елементів j-го стовпця

```

```
    print(s)          # Друк суми елементів j-го стовпця
```

1	2	3	4
5	6	1	1
7	8	9	1

Рис. 33.2

Пошук елементів двовимірного масиву, що задовольняють певну умову

Для організації пошуку у двовимірному масиві елементів із заданими властивостями слід за допомогою вкладених циклів перевірити кожний елемент на відповідність деякій умові.

5

Знайдемо кількість парних елементів у кожному стовпці масиву `tabl` (3×4).

```
tabl = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1]]
```

```
for j in range(4):          # Зовнішній цикл за стовпцями
```

```
    k = 0    # Перед початком перегляду кожного стовпця k обнуляється
```

```
    for i in range(3):      # Внутрішній цикл за рядками

```

```
        if tabl[i][j]%2 == 0:
```



```

k = k+1
print('У ', j, '-му стовпці k = ', k)

```

Результат наведено на рис. 33.3.

```

У 0-му стовпці k = 0
У 1-му стовпці k = 3
У 2-му стовпці k = 0
У 3-му стовпці k = 1

```

Рис. 33.3

6

Знайдемо в кожному рядку масиву `tabl` (3×4) кількість елементів, які перевищують середнє арифметичне елементів масиву.

```

tabl = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1]]

```

```

s = 0

```

```

for i in range(len(tabl)):      # Обчислення суми елементів масиву
    s = s+sum(tabl[i])

```

```

sr = s/12      # Обчислення середнього арифметичного елементів масиву
print('sr = ', sr)

```

```

for i in range(len(tabl)):      # Зовнішній цикл за рядками
    k = 0

```

```

    for j in range(len(tabl[i])):

```

```

        if tabl[i][j]>sr: k = k+1

```

```

        print('У', i, '-му рядку таких елементів ', k)

```

Результат наведено на рис. 33.4.

```

sr = 4.0

```

```

У 0-му рядку таких елементів 0

```

```

У 1-му рядку таких елементів 2

```

```

У 2-му рядку таких елементів 3

```



Рис. 33.4

Фільтрування двовимірного масиву в одновимірний список

Для розв'язування багатьох задач користувачу/користувачці потрібно проаналізувати підсумкові значення, отримані опрацюванням таблиці за рядками або за стовпцями.

Наприклад, як за даними таблиці Продаж телефонів (с. 208) визначити, телефонів якої марки продано найбільше? Нам потрібно визначити номер рядка, елементи якого утворюють найбільшу суму. Для цього зручно сформулювати одновимірний масив, значеннями якого є суми елементів рядків двовимірного масиву.

7

Визначимо номер рядка двовимірного масиву `tabl`, сума елементів якого найбільша (рис. 33.5).

```
tabl = [[80, 110, 120], [80, 90, 80], [60, 70, 130]]
```

```
mas_sum = []
```

```
for i in range(len(tabl)):
```

```
    mas_sum = mas_sum+[sum(tabl[i])]
```

```
        # Сума елементів i-го рядка додається до списку mas_sum
```

```
m = max(mas_sum) # Визначається максимальний елемент списку mas_sum
```

```
k = mas_sum.index(m) # Визначається індекс елемента списку
```

```
print('m = ', m) # m = 310 # mas_sum, який дорівнює m
```

```
print('k = ', k) # k = 0
```

tabl			mas_sum
80	110	120	310
80	90	80	250
60	70	130	260

Рис. 33.5

8

Заповнимо список `b` значеннями максимальних елементів стовпців масиву `tabl` (3×4).

```
tabl = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1]]
```

```
b = []
```

```
for j in range(4): # Зовнішній цикл за стовпцями
```

```
    m = tabl[0][j] # Перед початком перегляду кожного стовпця m
```

```
    # набуває значення елемента, розміщеного
```

```
    # в цьому стовпці у рядку з індексом 0
```

```
    for i in range(1, 3): # Внутрішній цикл за рядками
```

```
        if tabl[i][j]>m: m = tabl[i][j]
```

```
    b.append(m) # Максимальний елемент j-го стовпця додається
```

```
    # до списку b
```

```
print('b = ', b) # b = [7, 8, 9, 4]
```

Таким чином, опрацювання даних двовимірного масиву здійснюється через застосування типових алгоритмів опрацювання одновимірних масивів до його рядків або стовпців.



Питання для самоперевірки

1. Запишіть команди для обчислення суми елементів двовимірного масиву `tabl` (4×4).
2. Знайдіть суму елементів головної діагоналі масиву `tabl` (4×4).
3. Як знайти максимальний елемент стовпця з індексом 2 масиву `tabl` (5×4)?
4. Як знайти середнє арифметичне елементів рядка з індексом 1 масиву `tabl` (6×6)?

5. Заповніть список а середніми арифметичними значеннями елементів кожного рядка масиву `tabl` (6×6).
6. У 5 міст відправили вагони з десятьма різними видами товарів. Складіть програму для обчислення загальної маси продукції, відправленої в кожне місто, і загальної маси продукції кожного виду.



Вправа 33

- Торговельна мережа має 4 магазини. Інформація про щомісячний прибуток кожного магазину за квартал (тис. грн) зберігається в рядках двовимірного масиву (див. таблицю).
Скласти консольну програму для опрацювання масиву даних.



№ магазину	Січень	Лютий	Березень
1	300	150	200
2	250	230	170
3	120	180	190
4	220	135	210



1. В IDE виберіть команду `File → New File`. Створіть двовимірний масив `tabl` і заповніть його даними таблиці.
2. Для опрацювання даних таблиці виконайте такі завдання:
 - 1) за введеним номером магазину вивести його середній прибуток за три місяці;
 - 2) визначити сумарний дохід мережі магазинів;
 - 3) визначити, який магазин за квартал має найбільший прибуток;
 - 4) визначити, у якому місяці торговельна мережа отримала найбільший прибуток.
3. Збережіть файл з іменем `Vprava33.py`. Перевірте роботу програми.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112953





ПРАКТИЧНА РОБОТА 11

СКЛАДАННЯ АЛГОРИТМУ ЗНАХОДЖЕННЯ ПІДСУМОВИХ ВЕЛИЧИН ДЛЯ ДВОВИМІРНОГО МАСИВУ

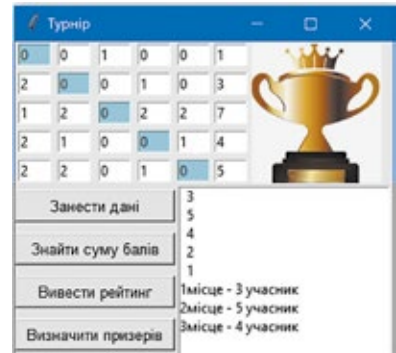
Завдання: розробити програму для визначення переможця та призерів математичного турніру (5 команд).

Обладнання: комп'ютер із середовищем програмування IDE.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. В IDE виберіть команду File → New File. Створіть вікно програми tk із заголовком Турнір, завершивши програму командою `tk.mainloop()`.
2. Створіть двовимірний масив `a` розміром 5×6 , заповнений нулями:
`a = [[0]*6 for i in range(5)]`
3. Створіть таблицю з об'єктів Entry (5×6) для введення результатів. Головну діагональ виділіть кольором.
4. Додайте кнопки:
 - Занести дані (`b1_click`) — зчитування даних у масив `a`;
 - Знайти суму балів (`b2_click`) — обчислення сум балів;
 - Визначити рейтинг (`b3_click`); призерів (`b4_click`).
5. Додайте віджет Listbox для відображення рейтингу та призерів.
6. Реалізуйте функції:
 - `b1_click()` — зчитування даних із таблиці в масив;
 - `b2_click()` — обчислення сум елементів кожного рядка та запис їх у 6-й стовпець;
 - `b3_click()` — формування рейтингу за спаданням суми балів;
 - `b4_click()` — виведення трьох перших місць.
7. Перевірте роботу програми, заповнивши стовпці 0–4 турнірної таблиці (виграш — 2 бали, нічия — 1, програш — 0).
8. Додайте віджет Canvas і розмістіть зображення призу для переможця. Збережіть файл під назвою `Pract11.py`.



Зробіть висновок: як застосовувати алгоритми опрацювання двовимірною масиву для розв'язування задач.

Списки дають змогу пов'язувати сукупність даних у єдину структуру. Якщо ж потрібно пов'язати кожний елемент списку з назвою ознаки, яку він характеризує, застосовують структуру даних «словник».

Створення словників

Можна створити словник, який описує об'єкт людина, і зберегти в ньому назви властивостей об'єкта (ім'я, вік, стать, професія тощо) та їх значення. Об'єднавши такі словники в список, можна ефективно програмувати опрацювання великих обсягів персональних даних.



Словник (об'єкт типу `dict`) — це неупорядкована сукупність пар ключ: значення.

На рис. 34.1 видно, що в словнику кожен елемент даних (Семешук, Антоніна, 2010, велоспорт) позначено ключовим словом (прізвище, ім'я, рік, захоплення), тобто поєднано два набори взаємопов'язаних даних.

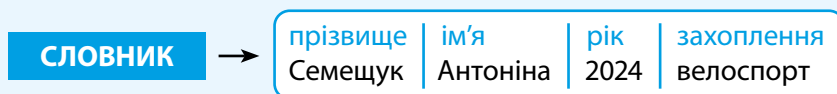


Рис. 34.1

Кожне значення в словнику має унікальний **ключ**, який є величиною одного з незмінюваних типів: рядком, логічним значенням, цілим або дійсним числом.

Щоб **створити словник**, потрібно помістити у фігурні дужки `{}` пари ключ: значення, розділені комами:

```
dictionary = {ключ1: значення1, ключ2: значення2, ...}
```



Створимо порожній словник `zakupy` та словник `magazyn` для збереження пар товар: ціна.

```
zakupy = {}
```

```
magazyn = {'яблука': 15, 'груші': 25, 'огірки': 9, 'помідори': 15, 'морква': 9}
```

Словник визначає взаємозв'язок пар ключ: значення. Оскільки ключ не повторюється, за ним відбувається доступ до значення.



Отримаємо зі словника `magazyn` елемент із ключем "яблука":

```
>>> magazyn['яблука']
15
```

Якщо ключа в словнику немає, буде згенеровано помилку:

```
>>> magazyn['цибуля']
```

KeyError: 'цибуля'

Щоб запобігти такій помилці, потрібно перевірити наявність ключа в словнику за допомогою умови `<ключ> in <словник>`:

```
if 'цибуля' in magazyn:
```

```
    print(magazyn['цибуля'])
```

```
else: print('Товар відсутній')      # Буде надруковано Товар відсутній
```

Для додавання елемента в словник слід звернутися до елемента за його ключем і присвоїти йому значення. Якщо ключ у словнику вже існує, поточне значення буде замінено новим.



Установимо для словника `magazyn` значення елемента з ключем 'морква':

```
>>> magazyn['морква'] = 5
```

Для видалення елемента за ключем застосовується команда `del`:

```
del magazyn['яблука']
```

Методи словників

Розгляньмо деякі методи об'єктів типу `dict` на прикладі:

```
books = {'І. Котляревський': 'Енеїда', 'М. Гоголь': 'Тарас Бульба',
         'Т. Шевченко': 'Гайдамаки', 'О. Гончар': 'Собор'}
```

- Метод **keys()** повертає `dict_keys()` — придатне для перебору (ітеративно) подання ключів словника, об'єкт особливого типу, який зберігає зв'язок зі словником:

```
>>> books.keys()
```

```
dict_keys(['І. Котляревський', 'М. Гоголь', 'Т. Шевченко', 'О. Гончар'])
```

Щоб за об'єктом `dict_keys()` отримати список ключів, слід викликати функцію `list()`:

```
>>> list(books.keys())
```

```
['І. Котляревський', 'М. Гоголь', 'Т. Шевченко', 'О. Гончар']
```

- Метод `values()` повертає `dict_values()` — придатне для перебору (ітеративно) подання значень словника. Щоб отримати список значень, викликають функцію `list()`:

```
>>> list(books.values())
['Енеїда', 'Тарас Бульба', 'Гайдамаки', 'Собор']
```

Перебір елементів словника

Послідовно перебрати всі пари "ключ: значення" словника можна за допомогою циклу. Параметр циклу послідовно отримує ключі, за якими в тілі циклу можна отримати й опрацювати відповідні їм значення.

4

Переберемо всі ключі в словнику `books` і виведемо на екран ключі з відповідними значеннями (рис. 34.2):

```
for key in books:
    print(key, ' - ', books[key])
```

```
I. Котляревський – Енеїда
М. Гоголь – Тарас Бульба
Т. Шевченко – Гайдамаки
О. Гончар – Собор
>>>
```

Рис. 34.2

Для перебору лише значень (без огляду на ключі) для словника можна викликати метод `values()`.

5

Виведемо на екран усі значення зі словника `books` (рис. 34.3):

```
for value in books.values():
    print(value)
```

```
Енеїда
Тарас Бульба
Гайдамаки
Собор
>>>
```

Рис. 34.3

Об'єкти класу `Combobox`

Віджет `tkinter Combobox` (поле зі списком) — це розкривний список для вибору одного зі значень. Якщо натиснути на кнопку, то можна побачити список (рис. 34.4), а при виборі елемента списку він стане поточним. Клас `Combobox` описано в модулі `tkinter ttk`, тому для використання віджета слід імпортувати цей модуль.

6

Створимо віджет `Combobox` і присвоїмо значення, які міститиме список (рис. 34.4):

```
from tkinter import ttk
combo1 = ttk.Combobox(root, values = ['Енеїда', 'Тарас Бульба', 'Гайдамаки',
    'Собор'])
combo1.pack()
```

Елемент списку, що виводиться в полі, можна програмно замінити на інший елемент, використовуючи метод `current(index)`.

Зробимо поточним елемент списку з індексом 3 (рис. 34.5), додавши команду:

```
combo1.current(3)
```

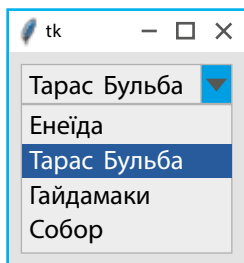


Рис. 34.4

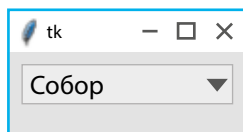


Рис. 34.5



Метод `current()` можна також використовувати для отримання індексу поточного елемента. Отримати значення поточного елемента можна за допомогою методу `get()`. Збережемо в змінних `index` та `item` індекс і значення вибраного елемента списку:

```
index = combo1.current()
```

```
item = combo1.get()
```

```
print('індекс вибраного елемента – ', index, ', значення – ', item)
```

У результаті виконання коду буде виведено рядок:

```
індекс вибраного елемента – 3, значення – Собор
```



Питання для самоперевірки

- Дано словник: `magazyn = {'яблука': 15, 'груші': 25, 'огірки': 9}`. Яким буде результат виконання в консолі команд?
а) `magazyn['груші'];` б) `magazyn['морква']`.
- Як додати до словника (див. п. 1) товар капуста вартістю 5 грн?
- Опишіть способи послідовного перебору значень у словнику.
- Як створити віджет `Combobox` із бажаним списком?
- Як зі списку `Combobox` отримати значення вибраного елемента?
- Напишіть програму, яка моделює діалог в овочевому магазині: покупець уводить назву товару й кількість кілограмів, яку хоче придбати; програма обчислює загальну вартість покупки. Для збереження даних використовуйте словник із п. 1.

Вправа 34

- Створити програму Банкноти України, яка виводить імена українських діячів, портрети яких зображено на грошових знаках.

1. В IDLE виберіть команду File → New File. Завантажте модулі tkinter, ttk, messagebox.

Створіть вікно розміром 400×250 із заголовком Банкноти. Збережіть файл із назвою Vprava34.

2. Додайте до вікна віджети lab_nom класу Label, ent_nom класу Entry для уведення номіналу банкноти.

Додайте до вікна віджети lab_name класу Label, ent_name класу Entry для виведення імені особи, що зображена на банкноті.

3. Створіть об'єкт btn класу Button:

```
btn = Button(text = 'Визначити ім'я', command = btn_click)
```

```
btn.place(x = 100, y = 40)
```

➤ 4. Створіть словник bank для збереження пар <номінал>: <особа>:

```
bank = {1000: 'Володимир Вернадський', 500: 'Григорій Сковорода',  
        200: 'Леся Українка'}
```

Доповніть його елементами, що відповідають іншим банкнотам.

5. Опишіть функцію btn_click:

```
def btn_click():  
    nom = int(ent_nom.get())  
    if nom in bank:  
        name = bank[nom]  
        ent_name.delete(0, END)  
        ent_name.insert(0, name)
```

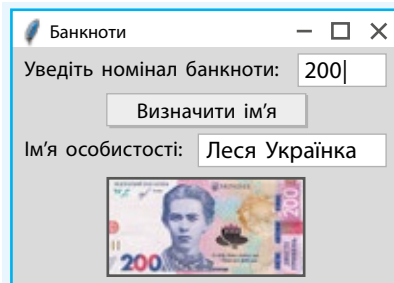
```
    else: messagebox.showerror('Помилка', 'Такої банкноти в обігу немає')
```

⚙ 6. Додайте до вікна полотно canvas для виведення зображення. Скопіюйте в папку з файлом програми графічні файли із зображеннями (наприклад, із назвами файлів 1000.png, 500.png тощо). Створіть словник bank_image для збереження пар <номінал>: <ім'я файлу>:

```
bank_image = {1000: '1000.png', 500: '500.png', 200: '200.png'}
```

У тілі функції btn_click() до команд, що виконуються у випадку, коли умова nom in bank істинна, додайте команди для зміни зображення в canvas.

Перевірте роботу програми.





Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей за QR-кодом або посиланням rnk.com.ua/112954



ПРАКТИЧНА РОБОТА 12

СТВОРЕННЯ ПРОГРАМИ З ВИКОРИСТАННЯМ СЛОВНИКІВ І СПИСКІВ

Завдання: скласти програму для побудови речень із трьох наборів слів і перекладу їх англійською мовою.

Обладнання: комп'ютер із системою програмування.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. В IDLE виберіть команду File → New File. Підключіть модулі tkinter, ttk, random. Збережіть файл із назвою Pract12.
2. Створіть вікно програми root із заголовком Перекладач.
3. Підготуйте словники, в яких будуть зберігатися слова для списків віджетів Combobox.
`c1 = {'Собака': 'A dog', 'Кінь': 'A horse', 'Пташка': 'A bird', 'Сонце': 'The Sun'}`
`c2 = {'голосно': 'loudly', 'швидко': 'fast', 'високо': 'high', 'яскраво': 'brightly'}`
`c3 = {'гавкає': 'barks', 'бігає': 'runs', 'літає': 'flies', 'сяє': 'shines'}`
4. На основі словника c1 сформуєте список ключів: `key1 = list(c1.keys())`.
5. Так само створіть списки `key2`, `key3`.
6. Слова зі списків помістіть у три розкривних списки (віджети Combobox), розміщені на формі методом grid() (див. рисунок, с. 220):
`combo1 = ttk.Combobox(root, width = 10, values = key1)`
`combo1.current(0)`
7. Створіть об'єкт класу Entry для виведення сформованого речення:
`fraza = Entry(text = "", width = 35)`
8. Створіть командну кнопку Скласти фразу:
`btn1 = Button(text = 'Скласти фразу', width = 20, font = 'Arial 12',`
`command = btn1_click)`

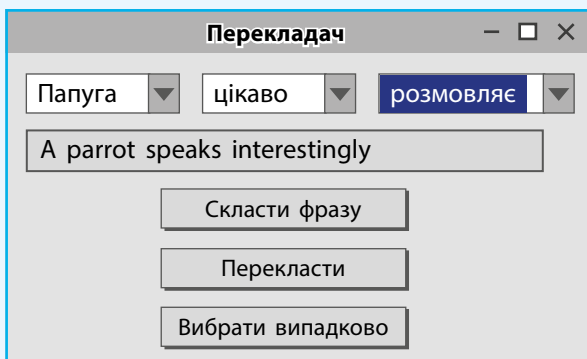
9. Створіть кнопки Перекласти, Вибрати випадково.
10. Запрограмуйте процедуру обробки події для командної кнопки btn1. Рядок s створюється як результат конкатенації рядків, вибраних у віджетах combo1...combo3, між якими додаються пропуски.
Отримане значення виводиться в текстове поле fraza;

```
def btn1_click():
    s = combo1.get()+' '+combo2.get()+' '+combo3.get()
    fraza.delete(0, END)
    fraza.insert(0, s)
```
11. Запрограмуйте процедуру обробки події для командної кнопки btn2. Рядок s створюється як результат конкатенації значень, що відповідають ключам, отриманим із віджетів combo1...combo3. Отримане значення виводиться в текстове поле fraza.

```
s = c1[combo1.get()]+ 'c3[combo3.get()]+ 'c2[combo2.get()]
```
12. Запрограмуйте можливість випадкового формування речення. Створіть процедуру обробки події для командної кнопки btn3. Задайте індекс вибраного елемента випадковим чином:

```
combo1.current(randint(0, 4))
```

 Випробуйте програму. Перевірте роботу кнопок.



Зробіть висновок: які особливості побудови програми з використанням словників і списків.



ПРАКТИКУМ З ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

6

РОЗДІЛ

У цьому розділі ви створите різні інформаційні продукти, користуючись інструментами програмних середовищ, із якими ознайомились у 9 класі.



ПРАКТИЧНА РОБОТА 13

СТВОРЕННЯ ОБ'ЄКТІВ ДЛЯ ПРОГРАМИ SWEET HOME 3D

Завдання: створити за допомогою тривимірного редактора Blender об'єкт для використання в програмі для моделювання інтер'єрів Sweet Home 3D.

Обладнання: комп'ютер із застосунками Blender і Sweet Home 3D.

► Теоретичні відомості

Як ви знаєте (див. §6, вправа 6), застосунок для моделювання інтер'єрів Sweet Home 3D дає можливість не обмежуватися типовим набором предметів, а імпортувати нові предмети з файлів у форматі OBJ. Створювати такі файли можна за допомогою вже знайомого вам тривимірного редактора Blender.

Щоб якомога точніше відтворити в Blender реальний об'єкт, потрібно виміряти його розміри й вибрати масштаб.

Наприклад, якщо вибрати масштаб 1 одиниця Blender = 10 см, то розміри табуретки, що має кругле сидіння (рис. 1), можуть бути такими, як наведено в таблиці:



Рис. 1

Параметр	Реальний розмір, см	Розмір у Blender
Висота	44	4,4
Діаметр сидіння	40	4
Товщина сидіння	1,8	0,18
Товщина ніжки	4,5	0,45
Ширина планки	4	0,4
Товщина планки	1,5	0,15



Для дотримання точності, перш ніж додати об'єкт, поставте курсор у початок координат (Shift+S → Курсор до центра).

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть Blender. Візьміть за основу куб, що є в початковій сцені, та змодельуйте розташовану вертикально ніжку табуретки (тут і далі орієнтуйтеся на розміри, наведені в таблиці).
2. Додайте до «ніжки» модифікатор Дзеркало. Налаштуйте віддзеркалення за осями X та Y, щоб вийшло 4 ніжки (див. §9, с. 58).
3. Додайте планки. Для цього:
 - 1) змодельуйте одну й продублюйте її модифікатором Дзеркало;
 - 2) в об'єктному режимі зробить копію планки (Shift+D, потім ЛКМ) і на вигляді зверху поверніть її на 90°.
4. Додайте сидіння (циліндр). Експортуйте модель у файл із назвою tabouret формату OBJ (меню Файл → Експортувати → Wavefront (.obj)).

! Використовуйте для назви файлу саме англійські букви. Це дасть змогу уникнути проблем на наступних кроках.

5. Запустіть Sweet Home 3D. Імпортуйте створений предмет із файлу. Для цього:
 - 1) виберіть команду меню Предмети → Імпортувати предмети). У вікні Помічник імпорту меблів натисніть Вибрати модель, відшукайте файл tabouret.obj і натисніть ОК, а потім — Продовжити; Малюнок предмета у вікні помічника можна розглянути краще, перетягуючи й обертаючи коліщатко.
 - 2) перевірте, чи предмет правильно зорієнтований, за потреби виправте його розташування, керуючись інструкцією у вікні; натисніть Продовжити;
 - 3) уточніть параметри предмета (рис. 2, с. 224):
 - назви предмета та категорії, в яку він має потрапити (причому, Мої предмети);
 - один із розмірів (за встановленого прапорця Зберегти пропорції решта розмірів зміняться автоматично);
 - колір тощо;
 потім натисніть Продовжити;
 - 4) розмістіть предмет так, як він має виглядати на піктограмі в колекції й натисніть Закінчити.
6. Переконайтеся, що новий предмет з'явився в каталозі предметів (див. §6, с. 38, рис. 6.7, позначка 1).

Додайте цей предмет на план разом з іншими предметами, щоб упевнитися, що розміри задано правильно.

7. Змодельуйте в Blender та імпортуйте у Sweet Home 3D ще один предмет (тумбочку, книжкову полицю, парту, стілець тощо).
8. Побудуйте модель кімнати з використанням декількох примірників щойно імпортованих предметів. Збережіть її у файл з іменем Практична робота 13.sh3d.
9. З'ясуйте за допомогою довідкової системи Sweet Home 3D (розділ редагування налаштувань, плану та просторового перегляду), як змінити колір предмета, доданого на план.
Змініть колір одного з нових предметів. Збережіть файл.

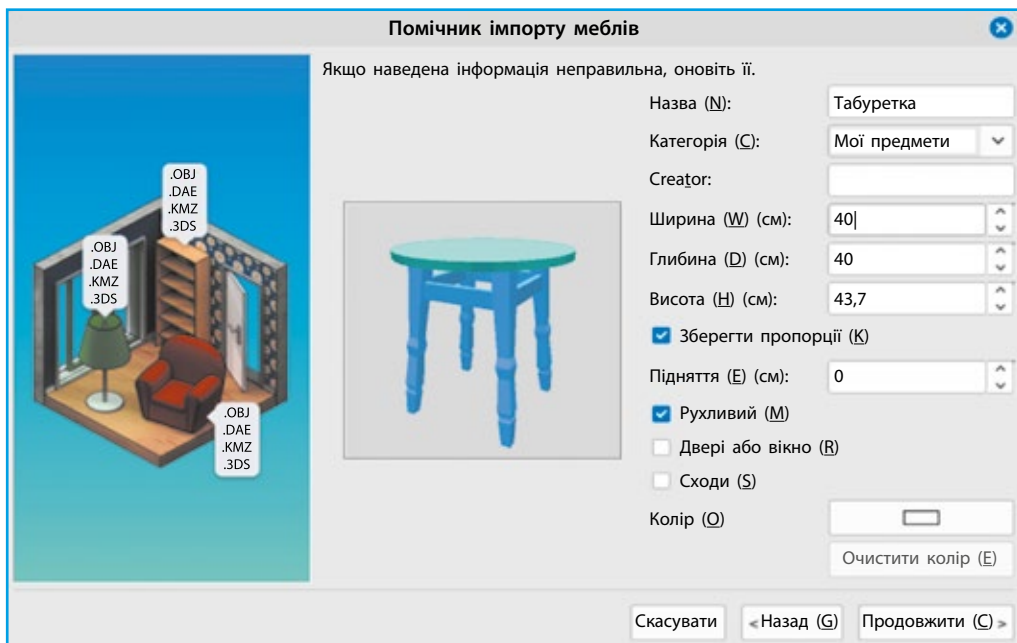


Рис. 2

Зробіть висновок: які можливості відкриває функція імпортування об'єктів у Sweet Home 3D; як змінити кольори предмета, доданого на план; що слід урахувати при підготовці в Blender моделі для Sweet Home 3D.



ПРАКТИЧНА РОБОТА 14

СТВОРЕННЯ ОБ'ЄКТІВ У BLENDER ЗА ДОПОМОГОЮ СКРИПТІВ МОВОЮ PYTHON

Завдання: створити й випробувати скрипти для програмного створення тривимірних об'єктів.

Обладнання: комп'ютер із тривимірним редактором Blender.

► Теоретичні відомості

Blender має вбудований інтерпретатор мови Python і надає доступ до модуля bpy, який дає змогу змінювати будь-які дані (сцени, тривимірні об'єкти тощо), створювати нові інструменти, генерувати за допомогою програм тривимірні об'єкти та багато іншого.

Для створення тривимірного об'єкта нам знадобляться такі засоби з модуля bpy:

- `m = bpy.data.meshes.new(name)` — метод для створення порожнього блоку даних тривимірної сітки (`name` — назва блоку даних);
- `m.from_pydata(verts, edges, faces)` — метод для додавання до блоку даних тривимірної сітки `m` вершин зі списку `verts`, ребер зі списку `edges` і граней зі списку `faces` (див. приклад далі);
- `obj = bpy.data.objects.new(name, m)` — метод для створення тривимірного об'єкта на основі тривимірної сітки `m`;
- `bpy.context.scene.objects.link(obj)` — метод для поміщення об'єкта `obj` у тривимірну сцену.

Розгляньмо приклад простого скрипту для створення в сцені об'єкта:

```
import bpy
```

```
# Підготовка даних
```

```
name = 'Трикутник'
```

```
verts = [(0, 0, 0), (0, 0.5, 0), (1, 0, 0)]
```

```
# Вершини
```

```
faces = [(0, 1, 2)]
```

```
# Грані
```

```
# Створення блоку даних сітки
```

```
mesh = bpy.data.meshes.new(name)
```

```
mesh.from_pydata(verts, [], faces)
```

```
# Створення об'єкта й розміщення його на сцені
```

```
obj = bpy.data.objects.new(name, mesh)
```

```
bpy.context.scene.objects.link(obj)
```



Список `verts` містить тривимірні координати (див. рис. 8.3, с. 48) трьох вершин. У списку `faces` зазначено, що об'єкт матиме лише одну грань, вершини якої мають у списку `verts` номери 0, 1 і 2. Список ребер порожній, оскільки потрібні ребра буде створено разом із гранню. Якщо запустити цей скрипт, то у вікні тривимірного вигляду з'явиться об'єкт із однією трикутною гранню (рис. 1). Далі його можна використовувати, як і інші тривимірні об'єкти.

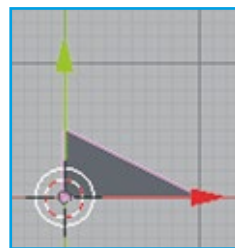


Рис. 1

Списки вершин, ребер і граней можна не лише задавати явно, а й формувати програмно. Наприклад, щоб створити пилкоподібний об'єкт із заданою кількістю «зубів», наведений на рис. 2, можна дані для побудови сформувати таким фрагментом коду:

```
name = 'Пилка'
n = 3                                     # Кількість зубів
faces = []
verts = [(0, 0, 0), (0, 1, 0)]           # Створюємо дві початкові вершини

for i in range(1, 2*n+1):
    verts += [(i, 0, i%2), (i, 1, i%2)]   # Створюємо дві нові вершини
    faces.append((2*i-2, 2*i-1, 2*i+1, 2*i)) # Створюємо нову грань
```

Числами показано порядок додавання вершин у список `verts`:

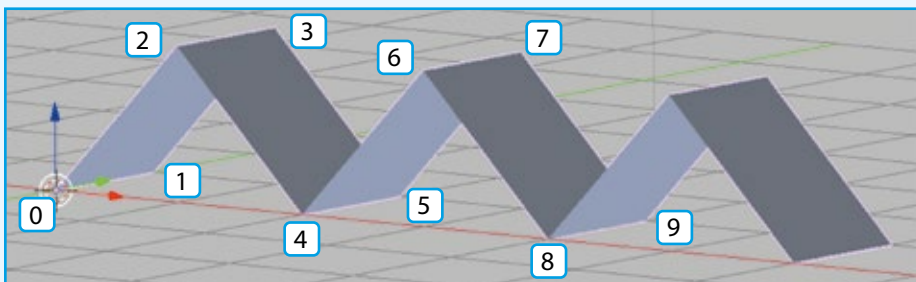


Рис. 2

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть Blender. Видаліть куб. Поділіть вікно 3D-вигляду по вертикалі на дві приблизно однакові ділянки (див. рис. 7.3, с. 42).

Тип правої ділянки змініть на Текстовий редактор (див. с. 43).

- У ділянці текстового редактора натисніть  **Новий** і введіть наведений на с. 225 код простого скрипту.
- Із контекстного меню ділянки текстового редактора виберіть команду Запуск скрипту (рис. 3). Переконайтеся, що в ділянці 3D-вигляду з'явився новий об'єкт трикутної форми. Перемістіть створений об'єкт від центра сцени.

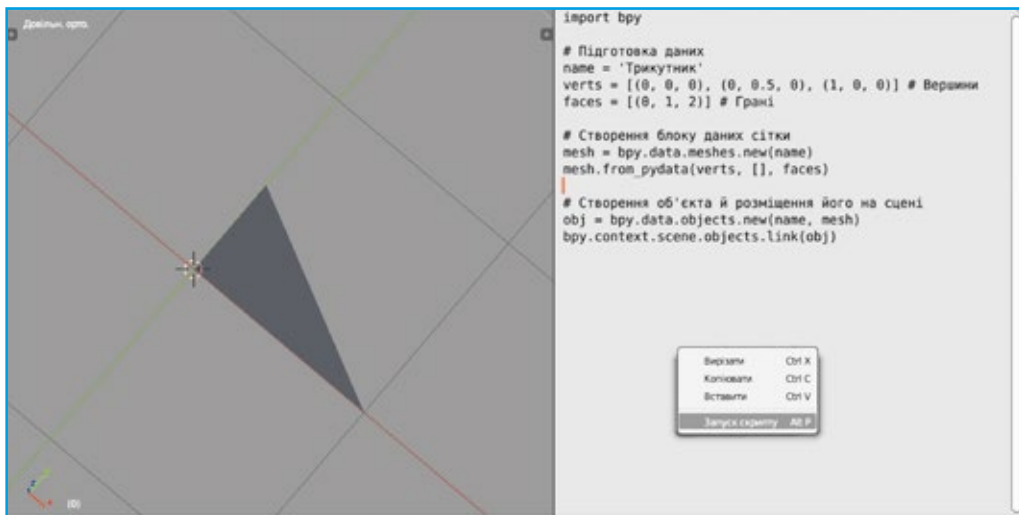


Рис. 3

- Змініть скрипт. Для цього:
 - додайте до списку `verts` вершину з координатами `(0, 0, 1)`:
`verts = [(0, 0, 0), (0, 0.5, 0), (1, 0, 0), (0, 0, 1)];`
 - додайте до списку `faces` грань із номерами вершин 1, 2 і 3:
`faces = [(0, 1, 2), (1, 2, 3)].`

Запустіть скрипт і переконайтеся, що створено об'єкт із двох граней (рис. 4). Перемістіть створений об'єкт.

- Додайте до списку `faces` ще дві грані: з номерами вершин 0, 1, 3 і 0, 2, 3.

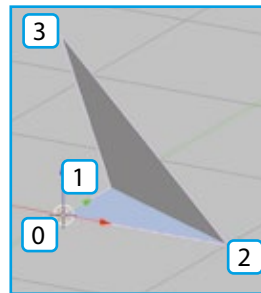


Рис. 4

Запустіть скрипт і переконайтеся, що створено чотиригранник. Перемістіть його в інше місце.

6. Створіть новий скрипт для побудови об'єкта, наведеного на рис. 5. Перевірте роботу скрипту.

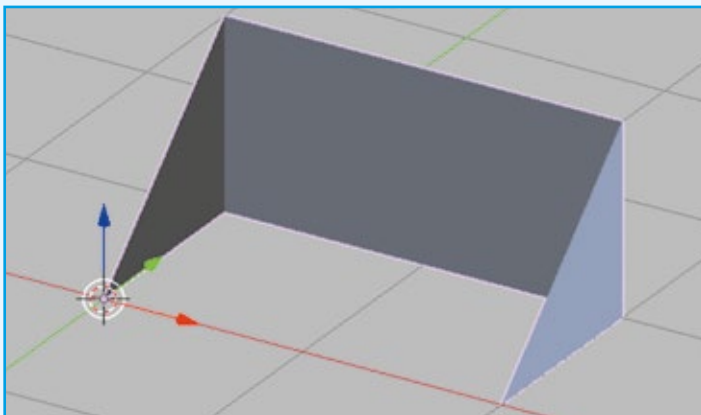


Рис. 5

7. Створіть скрипт для побудови «пилки», дані для якої створюються програмно (див. с. 226). Перевірте його роботу.
8. Поекспериментуйте з програмним кодом. Наприклад, змініть форму зубців (рис. 6) і/або доповніть зубці бічними гранями. Збережіть файл Практична робота 14.



Рис. 6

Зробіть висновок: яка послідовність створення об'єкта за допомогою скрипту; які є способи створення даних для побудови сітки об'єкта; які можливості відкриває використання мови Python у Blender.



ПРАКТИЧНА РОБОТА 15

СТВОРЕННЯ ПРОГРАМИ «МАГІЧНИЙ КВАДРАТ»

Завдання: скласти ігрову програму, в якій гравець/гравчиня має заповнити порожні клітинки квадрата 3×3 числами від 1 до 9 так, щоб утворити магічний квадрат.

Магічний квадрат — квадратна таблиця ($n = m$), у якій суми чисел кожного ряду (рядка, стовпця й діагоналі) однакові.

Обладнання: комп'ютер із середовищем програмування IDLE.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

У Python IDLE виконайте команду File → New File.

1. Завантажте модулі tkinter, random. Створіть вікно програми із заголовком Магічний квадрат (рис. 1). Завершіть програму командою tk.mainloop().

2. Створіть двовимірний список:

```
a = [[1]*3 for i in range(3)]
```

3. Створіть двовимірний масив об'єктів класу Entry:

```
list_en = [], [], []
```

```
for i in range(len(a)):
```

```
    for j in range(len(a[i])):
```

```
        # Створюємо об'єкт класу Entry
```

```
        en = Entry(tk, bg = 'lightblue', width = 3, font = 'Arial 18')
```

```
        en.grid(row = i, column = j, padx = 2, pady = 2)
```

```
        # Додаємо об'єкт en до i-го рядка масиву list_en
```

```
        list_en[i].append(en)
```

4. Створіть віджет b1 класу Button із заголовком Новий квадрат:

```
b1 = Button(tk, text = 'Новий квадрат', command = b1_click, width = 12,
            font = 'Arial 14')
```

```
b1.grid(row = 5, column = 0, columnspan = 5)
```

5. Створіть віджет b2 класу Button із заголовком Перевірити.

6. Додайте у вікно програми віджет lb класу Label:

```
lb = Label(tk, text = "...", font = 'Arial 14')
```

```
lb.grid(row = 7, column = 0, columnspan = 3)
```

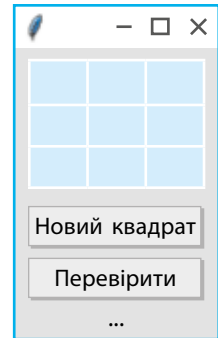


Рис. 1

7. Створіть обробник події для кнопки Новий квадрат.

У цій функції потрібно очистити вміст полів Entry і текст напису lb. У поле елемента list_en[1][1] виводиться випадкове число в діапазоні від 1 до 9.

```
def b1_click():
    for i in range(len(a)):
        for j in range(len(a[i])):
            list_en[i][j].delete(0, END)
    k = randint(1, 9)
    list_en[1][1].insert(0, str(k))
    lb.config(text = "...")
```

8. Створіть обробник події для кнопки Перевірити. Запрограмуйте зчитування значень із текстових полів масиву list_en і занесення їх до списку a:

```
for i in range(len(a)):
    for j in range(len(a[i])):
        a[i][j] = int(list_en[i][j].get())
```

9. Запрограмуйте перевірку «магічності» квадрата.

Якщо числа розташовані правильно, то сума чисел кожного ряду (діагоналі, рядка, стовпця) дорівнює 15. Якщо чергова сума не дорівнює 15, прапорцева змінна Prap набуває значення False.

Перевіримо суму чисел головної діагоналі:

```
prap = True
magic = 15
suma = 0
for i in range(len(a)): suma = suma+a[i][i]
if suma != magic: prap = False
```

10. Запрограмуйте перевірку сум чисел у рядках і стовпцях:

```
for i in range(len(a)):
    if sum(a[i]) != magic: prap = False
for j in range(len(a[i])):
    s = 0
    for i in range(len(a)): s = s+a[i][j]
    if s != magic: prap = False
```

11. Запрограмуйте перевірку суми чисел допоміжної діагоналі:

```
suma = 0
for i in range(len(a)):
    suma = suma+a[i][2-i]
if suma != magic: prap = False
```

12. Якщо після обчислення всіх 6 сум змінна `Prap` не змінила значення, то квадрат заповнено правильно (рис. 2):

```
if prap: lb.config(text = "Перемога!")
```

```
else: lb.config(text = "Спробуй ще!")
```

Збережіть програму у файлі `Pract15`. Перевірте роботу програми. Чи за будь-якого початкового числа вдається заповнити магічний квадрат?

Зробіть висновок: як створити обробники подій для елементів керування в програмі з графічним інтерфейсом.

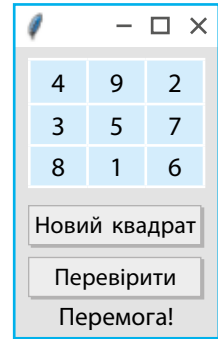


Рис. 2



ПРАКТИЧНА РОБОТА 16

СТВОРЕННЯ ПРОГРАМИ «КВАДРАТ ПОЛІБІЯ»

Завдання: створити програму, яка шифрує текст за допомогою квадрата Полібія. Користувач/користувачка вводять повідомлення, а програма виводить зашифрований текст у вигляді пар чисел.

Обладнання: комп'ютер із середовищем програмування IDE.

► Теоретична частина

Полібій (близько 200 р. до н. е. — близько 118 р. до н. е.) — давньогрецький історик, державний і військовий діяч, відомий як автор «Історії» в 40 томах, що описує історію Середземномор'я.

Квадрат Полібія — це таблиця, у якій розміщено букви алфавіту. Кожна буква позначається номерами рядка та стовпця (на рисунку — приклад для українського алфавіту).

Порядок розташування символів у квадраті Полібія є таємним ключем, який дає змогу зашифрувати й розшифрувати повідомлення.

	0	1	2	3	4	5
0	А	Б	В	Г	Ґ	Д
1	Е	Є	Ж	З	И	І
2	Ї	Й	К	Л	М	Н
3	О	П	Р	С	Т	У
4	Ф	Х	Ц	Ч	Ш	Щ
5	Ь	Ю	Я	'	,	.

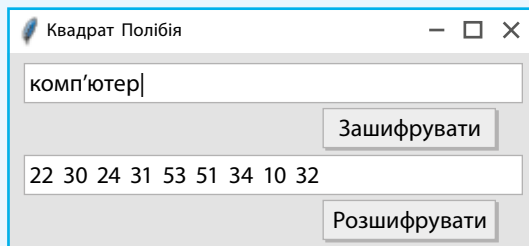
Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

У Python IDLE виконайте команду File → New File.

1. Завантажте модуль tkinter. Створіть вікно програми із заголовком Квадрат Полібія (див. рисунок).

Завершіть програму командою `root.mainloop()`.



2. Розмістіть у вікні віджет Entry для задання повідомлення, яке треба зашифрувати:
`text1 = Entry(width = 80, font = "Arial 14")`
`text1.grid(row = 0, column = 0)`
3. Додайте до вікна об'єкт `b_shifr` класу Button:
`b_shifr = Button(text = "Зашифрувати", command = shifrText)`
`b_shifr.grid(row = 1, column = 0)`
4. Розмістіть у вікні віджет `text2` класу Entry для виведення зашифрованого повідомлення та кнопку `b_rozsh` із підписом Розшифрувати, як наведено на рисунку.
5. Сформуйте двовимірний список `buk`, що міститиме алфавіт і символи «'», «,», «.»
`S = "АБВГГДЕЄЖЗИЙЙКЛМНОПРСТУФХЦЧШЩЬЮЯ,."`
`buk = [[], [], [], [], [], []]`
`for i in range(6):`
`for j in range(6):`
`buk[i].append(S[i*6+j])`
6. Створіть обробник події для кнопки Зашифрувати. Ця функція:
 - зчитує текст із поля `text1`;
 - для кожної букви шукає її позицію `(i, j)` у квадраті `buk`;
 - формує новий рядок із парами чисел;
 - виводить результат у поле `text2`.

```
def shifrText():
    sh = "" # Рядок для зберігання зашифрованого тексту
    vp = text1.get() # Початкове повідомлення, яке потрібно
    for k in range(len(vp)): # зашифрувати
        for i in range(6):
            for j in range(6):
# Якщо k-та буква з тексту збігається з буквою в таблиці buk[i][j],
# до рядка sh додаються координати букви: str(i) — номер рядка,
# str(j) — номер стовпця, ' ' — пропуск для відокремлення наступної пари.
                if vp[k] == buk[i][j]:
                    sh = sh+str(i)+str(j)+' '
    text2.delete(0, END)
    text2.insert(0, sh)
```

Наприклад: якщо буква "С" міститься в клітинці buk[1][2], то до рядка sh додається "12".

7. Створіть обробник події для кнопки Розшифрувати. Ця функція:
 - зчитує зашифрований рядок sh із поля text2;
 - команда split() розбиває рядок sh на список k за пропусками; кожен елемент — пара цифр, які вказують позицію букви у квадраті;
 - цикл for n in k: перебирає елементи списку k. i = int(n[0]) — номер рядка, j = int(n[1]) — номер стовпця;
 - buk[i][j] додає до результату vp;
 - виводить результат у поле text1.

```
def rozshText():
    sh = text2.get()
    vp = ""
    k = sh.split()
    for n in k:
        i = int(n[0])
        j = int(n[1])
        vp = vp+buk[i][j]
    text1.delete(0, END)
    text1.insert(0, vp)
```



8. Збережіть програму у файлі Pract15. Запустіть програму. Зашифруйте повідомлення (наприклад, своє ім'я або слово з 5–7 літер).
9. Уведіть у поле рядок «22 30 24 31 53 51 34 10 32». Розшифруйте.

Зробіть висновок: як створити обробники подій для реалізації такого способу шифрування.

Аргументи — дані, які опрацьовує функція.

Архівування — процес стиснення даних, що зберігаються у файлах.

База даних — упорядкована сукупність взаємопов'язаних даних.

Брандмауер — технічний пристрій або програмний засіб для контролю даних, що надходять до комп'ютера через мережу.

Генератор списків — синтаксична конструкція, призначена для створення списків і заповнення їх значеннями за певними правилами.

Двовимірний масив — масив, в якому для задання розташування елемента необхідно вказати значення двох індексів.

Діаграма — графічне зображення, у якому числові дані подано геометричними фігурами.

Діапазон — сукупність комірок, які можна опрацьовувати як єдине ціле.

Драйвер — спеціальна програма, що забезпечує взаємодію комп'ютера із зовнішніми пристроями.

Електронна база даних — файл або сукупність файлів особливого формату, які містять структуровані дані, призначені для зберігання, накопичення, опрацювання за допомогою комп'ютера.

Електронна система керування базою даних — комплекс програм для роботи з даними в електронній БД.

Запит — засіб відбору та/або опрацювання табличних даних за заданими правилами та критеріями.

Захищений сайт — сайт, який використовує для обміну даними протоколи захищеного зв'язку.

Комп'ютерний вірус — програма, яка копіює й поширює свій програмний код і може негативно впливати на виконання програм і роботу комп'ютера в цілому.

Ліцензія на програмне забезпечення — угода, яка надає право на його встановлення й використання.

Логічна функція — функція, один або декілька аргументів якої — логічні вирази.

Логічний вираз — вираз, який набуває одного із двох значень: True — істина або False — хибність.

Математичні функції в електронній таблиці — функції, призначені для виконання обчислень над даними клітинок.

Мінідіаграма — елемент, що дає змогу візуалізувати ряд чисел, щоб унаочнити їх співвідношення й тенденції.

Модель «сутність — зв'язок» — схематичне відображення змістового зв'язку об'єктів бази даних.

Одновимірний масив — структура даних — пронумерована послідовність даних одного типу, що мають спільну назву.

Програмне забезпечення — сукупність програм, що створюють умови для роботи комп'ютера.

Рендеринг (вимальовування) — процес побудови зображення з урахуванням усіх налаштувань матеріалів, освітлення тощо.

Розархівування — процес повернення даних, записаних в архіві, до початкового стану.

Система керування базою даних — комплекс засобів і методів для створення, збереження, оновлення та пошуку інформації в базі даних із контролем доступу до них.

Словник — структура даних — невпорядкована сукупність пар ключ: значення.

Сортування даних в електронній таблиці — встановлення порядку розташування рядків за даними певного стовпця (стовпців).

Сортування елементів масиву — упорядкування за деякою ознакою.

Сортування обміном (метод бульбашки) — метод упорядкування масиву шляхом послідовного порівняння та обміну сусідніх елементів, якщо попередній елемент виявляється більшим від наступного.

Список — упорядкований набір об'єктів, можливо, різних типів (чисел, рядків, списків тощо), який можна змінювати.

Статистичні функції в електронній таблиці — функції для аналізу значень діапазонів клітинок.

Точка відновлення Windows (контрольна) — збережена інформація про стан операційної системи та встановлених програм у певний час.

Тривимірна графіка (3D-графіка) — розділ комп'ютерної графіки, у якому розглядають методи створення зображень або відео шляхом моделювання об'ємних об'єктів у тривимірному просторі.

Тривимірна модель — візуальний об'ємний образ певного об'єкта.

Умове форматування даних — автоматична зміна формату комірок залежно від значення даних у них.

Фільтрування даних — відображення в таблиці тільки тих рядків, дані яких відповідають певним умовам.

АЛФАВІТНИЙ ПОКАЖЧИК

А

Архів 12
— із саморозархівуванням 14
Архіватор 14
Архівування даних 12

Б

База даних 116
Брандмауер 24

Г

Генератор списків 168

Д

Двовимірний масив 200
Деінсталяція програми 10
Діаграма 110
Діапазон 76
— зв'язний 76
— незв'язний 76
Драйвер 7

З

Запит 143
Захищений сайт 25
Звіт 159
Зв'язок між таблицями 138

І

Інсталяція програми 9
Інформаційна безпека 22

Л

Ліцензія 8
Логічна функція 85
Логічний вираз 85

М

Математична функція 90

Мінідіаграма 112
Модель сутність — зв'язок 124

О

Одновимірний масив 166

П

Первинний ключ таблиці 136
Переносна програма 10
Посилання 74
Програмне забезпечення 6

Р

Рендеринг 63
Розархівування даних 112

С

Системи керування базами
даних 117
Словник 214
Сортування даних 95
Сортування елементів
масиву 178
Список 94
Статистична функція 91

Т

Точка відновлення Windows 28
Тривимірна графіка 34
Тривимірне моделювання 35

У

Умовне форматування даних 99

Ф

Фільтрування даних 97
Форма 153
Форматування носія даних 8
Функція 83

ЗМІСТ

Передмова.....	3
----------------	---

Розділ 1 Програмне забезпечення та інформаційна безпека

§ 1. Програмне забезпечення комп'ютера.....	6
§ 2. Архівування даних. Архіватори	12
Практична робота 1. Архівування та розархівування даних	16
§ 3. Шкідливі програми та захист комп'ютера	17
§ 4. Захист комп'ютерних систем. Безпека в інтернеті	21
§ 5. Зберігання даних	27
Практична робота 2. Налаштування параметрів безпеки в середовищі браузера	32

Розділ 2 Тривимірна графіка

§ 6. Основні поняття тривимірної графіки	34
§ 7. Тривимірний редактор Blender	41
§ 8. Робота з об'єктами	48
Практична робота 3. Робота з тривимірними примітивами	53
§ 9. Редагування форми об'єкта. Модифікатори	54
Практична робота 4. Моделювання витисканням	60
§ 10. Матеріали та освітлення. Рендеринг	61
§ 11. Анімація ключовими кадрами. Текстові об'єкти	66
Практична робота 5. Створення тривимірної анімації	71

Розділ 3 Опрацювання табличних даних

§ 12. Типи посилань на клітинки електронної таблиці	74
§ 13. Параметри сторінки. Друкування таблиці	78
§ 14. Функції. Логічні функції	83
§ 15. Математичні та статистичні функції	90
§ 16. Упорядкування даних. Умове форматування	94

Практична робота 6. Використання функцій табличного процесора. Умовне форматування.....	102
§ 17. Розширені фільтри. Проміжні підсумки.....	104
§ 18. Діаграми та графіки.....	109
Практична робота 7. Упорядкування даних у таблицях. Автоматичні та розширені фільтри	114

Розділ 4 Бази даних. Системи керування базами даних

§ 19. Що таке база даних і система керування базою даних	116
§ 20. Реляційна база даних	121
§ 21. СКБД Microsoft Access	128
§ 22. Зв'язування та заповнення таблиць	134
§ 23. Запити	143
Практична робота 8. Робота з таблицями і запитамі	150
§ 24. Форми.....	153
§ 25. Звіти.....	159

Розділ 5 Алгоритми та програми

§ 26. Одновимірні масиви. Списки.....	166
§ 27. Функції та методи списків.....	171
§ 28. Опрацювання елементів списку	174
§ 29. Алгоритми впорядкування масиву.....	178
§ 30. Об'єкти класу Listbox	182
§ 31. Візуалізація елементів табличної величини	189
Практична робота 9. Складання і виконання алгоритмів знаходження сум і кількостей значень елементів табличних величин за заданими умовами	198
§ 32. Двовимірний масив. Вкладені списки	200
Практична робота 10. Робота з двовимірними масивами у програмі з графічним інтерфейсом	206
§ 33. Опрацювання двовимірних масивів даних	208
Практична робота 11. Складання алгоритму знаходження підсумкових величин для двовимірного масиву.....	213

§ 34. Словники	214
Практична робота 12. Створення програми з використанням словників і списків	219

Розділ 6 Практикум з використання інформаційних технологій

Практична робота 13. Створення об'єктів для програми Sweet Home 3D.....	222
Практична робота 14. Створення об'єктів у Blender за допомогою скриптів мовою Python	225
Практична робота 15. Створення програми «Магічний квадрат»	229
Практична робота 16. Створення програми «Квадрат Полібія»	231
Комп'ютерний словник	234
Алфавітний покажчик	236



Відомості про користування підручником

№ з/п	Прізвище та ім'я учня / учениці	Навчальний рік	Стан підручника	
			на початку року	у кінці року
1				
2				
3				
4				
5				

Навчальне видання

БОНДАРЕНКО Олена Олександрівна
ЛАСТОВЕЦЬКИЙ Василь Васильович
ПИЛИПЧУК Олександр Павлович
ШЕСТОПАЛОВ Євген Анатолійович

«ІНФОРМАТИКА»

Підручник для 9 класу закладів загальної середньої освіти

Рекомендовано Міністерством освіти і науки України

Видано за рахунок державних коштів.

Продаж заборонено

Підручник відповідає Державним санітарним нормам і правилам
«Гігієнічні вимоги до друкованої продукції для дітей»

Редактор *Л. А. Каюда*. Технічний редактор *А. В. Плisko*.

Художнє оформлення *В. І. Труфена, Т. В. Задорожної*.

Комп'ютерна верстка *О. Л. Піковець*. Коректор *Н. В. Красна*.

Підписано до друку 27.04.2026. Формат 70×90/16.

Папір офсетний. Гарнітура Шкільна. Друк офсетний.

Ум. друк. арк. 17,79. Обл.-вид. арк. 22,28.

Наклад 54281 прим. Зам. № 0348.

ТОВ Видавництво «Ранок», пр. Незалежності, 5, Харків, 61022;
вул. Деревлянська, 13, Київ, 04119. E-mail: office@ranok.com.ua
Свідоцтво суб'єкта видавничої справи ДК № 8348 від 02.06.2025.

Надруковано у друкарні ТОВ «ФАКТОР-ДРУК».

61030, Україна, м. Харків, вул. Саратовська, 51.

Свідоцтво суб'єкта видавничої справи ДК № 5496 від 23.08.2017.

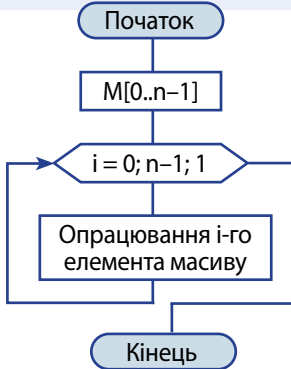
Тел. +38(057)717-53-55. E-mail: office@druk.factor.ua



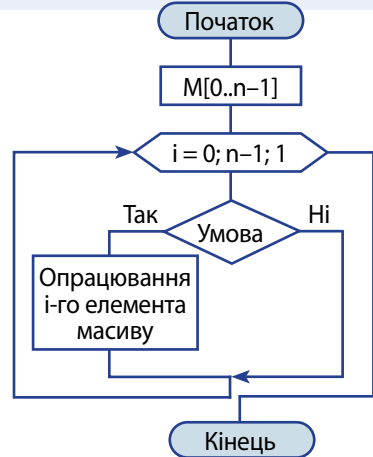
Служба підтримки:
rnk.com.ua/107946

ОПРАЦЮВАННЯ

Опрацювання елементів одновимірного масиву $M[0..n-1]$



Опрацювання елементів одновимірного масиву $M[0..n-1]$, що задовольняють певній умові



Знаходження кількості елементів одновимірного масиву $M[0..n-1]$, що задовольняють певній умові

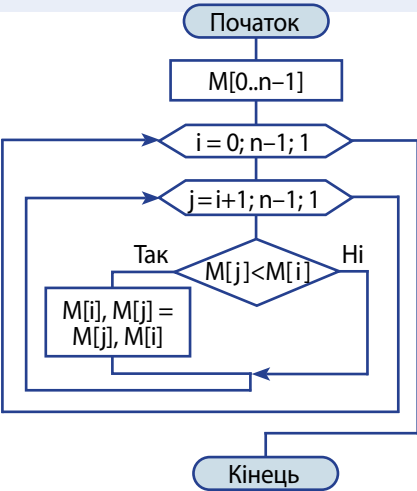


Знаходження суми S елементів одновимірного числового масиву $M[0..n-1]$, що задовольняють певній умові

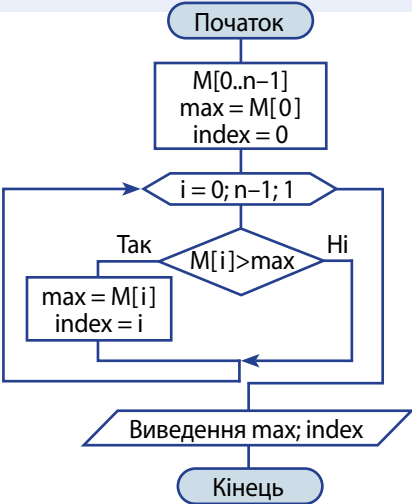


ЕЛЕМЕНТІВ МАСИВУ

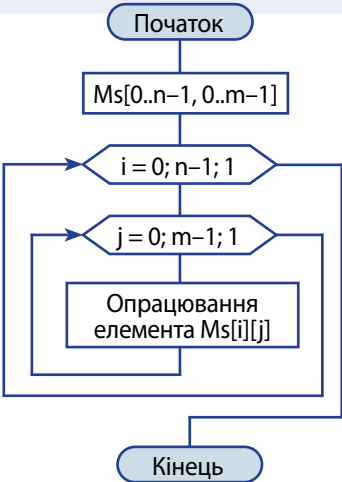
Упорядкування за зростанням
(неспаданням) елементів
одновимірного масиву $M[0..n-1]$



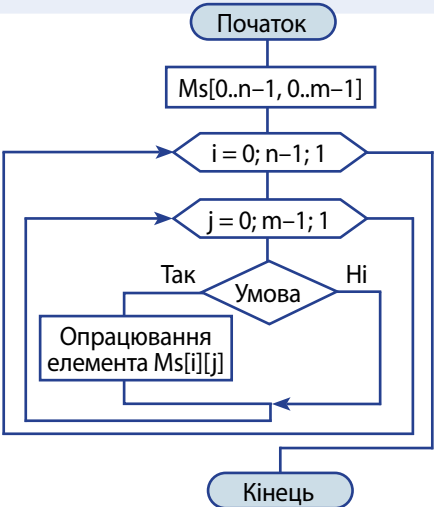
Знаходження значення і першого
індексу найбільшого елемента в одно-
вимірному числовому масиві $M[0..n-1]$



Опрацювання елементів
двовимірного масиву
 $Ms[0..n-1, 0..m-1]$



Опрацювання елементів двовимірного
масиву $Ms[0..n-1, 0..m-1]$,
що задовольняють певній умові



Підручник для 9 класу
закладів загальної середньої освіти

ІНФОРМАТИКА

Особливості підручника:

- › зорієнтованість матеріалу на власний досвід учнівства, підготовку до життя в інформаційному суспільстві
- › покроковий опис виконання практичних робіт
- › приклади з різних галузей людської діяльності

Інтерактивний електронний додаток:

- › інтерактивні тести за всіма темами курсу
- › підсумкові роботи до кожного розділу
- › додаткові матеріали до вправ і практичних робіт
- › навчальні відео, інтерактивні вправи, 3D-моделі

Навчально-методичний комплект:

- | | |
|-------------------------------------|-------------------------|
| › підручник | › робочий зошит |
| › інтерактивний електронний додаток | › розробки уроків |
| | › презентації до уроків |



ranok.com.ua

**УСІ
КНИГИ
ТУТ!**

Навчально-
методична
література

(057) 727-70-90

pochta@ranok.com.ua



ua.izzi.digital



Сучасна
інтерактивна
освітня
платформа