

Ольга Казанцева, Ірина Стеценко

ІНФОРМАТИКА 9



Ольга Казанцева
Ірина Стеценко

ІНФОРМАТИКА

ПІДРУЧНИК ДЛЯ 9 КЛАСУ
закладів загальної середньої освіти

Схвалено для використання в освітньому процесі



ТЕРНОПІЛЬ
НАВЧАЛЬНА КНИГА – БОГДАН
2022

УМОВНІ ПОЗНАЧЕННЯ



Інтерактивна
електронна складова

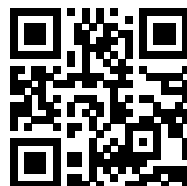


Пройдіть тест



Електронний
кресворд

Завантажуйте безкоштовний інтерактивний додаток,
використовуючи детальну інструкцію, за посиланням:
bohdan-books.com/6746-1



Шановні дев'ятикласники та дев'ятикласниці!

Цей підручник допоможе вам продовжити вивчення інформатики. У розділі 1 ви познайомитеся з класифікацією програмного забезпечення, дізнаєтеся про інформаційну безпеку. Вивчаючи матеріал розділу 2, ви будете здійснювати опрацювання табличних даних засобами табличного процесора MS Excel. Розділ 3 познайомить вас з можливостями створення, редагування та використання баз даних. У розділі 4 продовжите своє знайомство з мовою Python, створюючи програми та Windows-додатки опрацювання масивів. У розділі 5 розпочнете своє знайомство з основами 3D-графіки, використовуючи редактор Blender. Підручник містить практичні вправи для виконання на комп'ютері. Файли-заготовки до деяких вправ містяться в інтерактивній складовій підручника в папці **Files**.

ПУТІВНИК ПІДРУЧНИКОМ

Пропоновані питання вимагають творчого пошуку і поглиблених знань:

* — достатній рівень,
** — високий рівень навчальних досягнень

Ці завдання розраховані на учнів, які захоплюються інформатикою

Практичні завдання дають можливість закріпити вивчений матеріал, працюючи за комп'ютером

Контрольні запитання та завдання

1. Що таке кадр в анімації?
2. Що таке ключовий кадр в анімації?
3. Яка панель використовується для створення анімації в Blender?
4. Як зберегти анімацію у відеофайлі?

Питання для роздумів

- 1*. Як ви думаєте, що відбудеться, якщо після створення декількох ключових кадрів прибрати проміжний кадр?

Завдання для досліджень

- 1*. Дізнайтеся як виконати анімацію руху об'єкту за траєкторією.
2**. Дізнайтеся, що таке рігінг в анімації.

Висновки

3D-модельовання — це технологія побудови об'ємної моделі об'єкту в тривимірному просторі на базі професійних графічних програм.

Створення 3D-зображення відбувається за такими етапами: *побудова 3D-моделі; текстуровання; освітлення; рендеринг.*

Рендеринг — це одержання зображення за побудованою 3D-моделлю з використанням комп'ютерної програми.

Основні дії, які можна виконувати з 3D-об'єктами в Blender — це *переміщення, обертання та масштабування*. В режимі об'єкту ці дії виконуються над суцільним об'єктом, а в режимі редагування над його вершинами, ребрами та гранями.

Екструджування — процес створення нових елементів об'єктів шляхом видавлювання їх із наявних.

Для надання поверхні об'єкту реалістичного виду використовують матеріали та текстури.

Використовуючи технології 3D-друку можна цифрову модель перетворити на матеріальну.

Одним із головних понять створення анімації в Blender є ключовий кадр. У ключових кадрах створюються зміни, що мають відбуватися між ними.

Вміщені запитання і завдання дають змогу перевірити засвоєні знання

Пройдіть тест

У рубриці підсумовано матеріал розділу та виокремлено основні поняття і терміни

Розв'яжіть електронний кросворд


ВПРАВА 46.



Завдання. Завдання. Створіть анімацію, що імітує рух стрілок годинника.

1. Відкрийте файл годинника, створений у вправі 45.
2. Виділіть годинникову стрілку та поверніть її у положення 12 годин (R, а потім Shift+Z).

ПУТІВНИК ІНТЕРАКТИВНОЮ ЕЛЕКТРОННОЮ СКЛАДОВОЮ ПІДРУЧНИКА

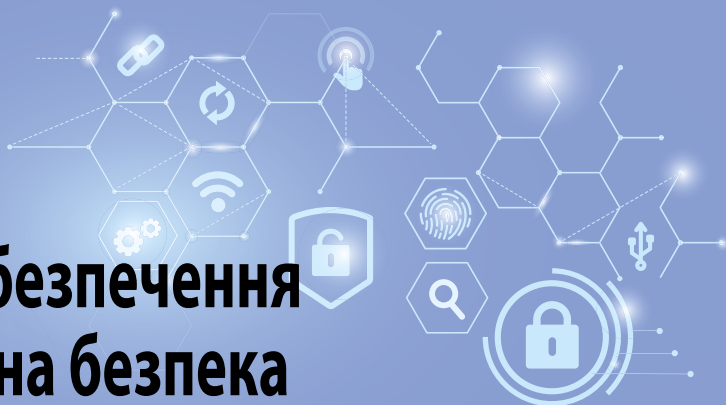
Зручним супроводженням усього матеріалу підручника інформатики є його інтерактивна електронна складова, посилання на використання якої позначено піктограмою .

Головне вікно програми



Розділ 1

Програмне забезпечення та інформаційна безпека



КЛАСИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Зберігання, передавання та обробка інформації

Одна з причин, чому комп'ютерні технології так швидко проникли в усі сфери нашого життя, полягає в тому, що комп'ютер є універсальним пристроєм, тобто він може використовуватися для виконання широкого кола завдань, пов'язаних із обробкою, зберіганням та передаванням інформації. Забезпечується ця універсальність завдяки такій складовій комп'ютерних систем, як програмне забезпечення. Про те, яким буває програмне забезпечення, ми і дізнаємося далі.

Під **програмним забезпеченням (ПЗ)** розуміють сукупність комп'ютерних програм і даних, які містять інструкції для комп'ютера стосовно того, що та як він має робити.

Комп'ютер є програмованим пристроєм. Це означає, що він усе робить у відповідності до програми. Відразу після виготовлення комп'ютер не містить жодних програм і фактично не може працювати.

На такий комп'ютер встановлюють (інстальють) певне програмне забезпечення, і воно, а також програми, з якими комп'ютер може працювати через мережу, визначають, які завдання можна буде виконувати з його допомогою.

Установлені програми зберігаються на зовнішніх пристроях комп'ютера та в його постійній пам'яті. Щоб комп'ютер почав працювати





згідно з тією чи іншою програмою, її потрібно запустити на виконання. Це може бути зроблено в автоматичному режимі (наприклад, після увімкнення комп'ютера) або вручну користувачем, який клацає на ярлик програми. Після цього програма і потрібні їй дані зчитуються з диска та завантажуються в оперативну пам'ять. Центральний процесор послідовно зчитує програму з пам'яті та обробляє інструкції, що в ній містяться.

Зазначимо, що користувачеві може бути доступне програмне забезпечення, розміщене не лише на його комп'ютері, а й у мережі (зокрема, в Інтернеті) — звичайно, за умови, що його комп'ютер підключено до цієї мережі.

За час існування комп'ютерної техніки було розроблено дуже багато різноманітного програмного забезпечення. Залежно від призначення його поділяють на **системне**, **прикладне** та **інструментальне**.

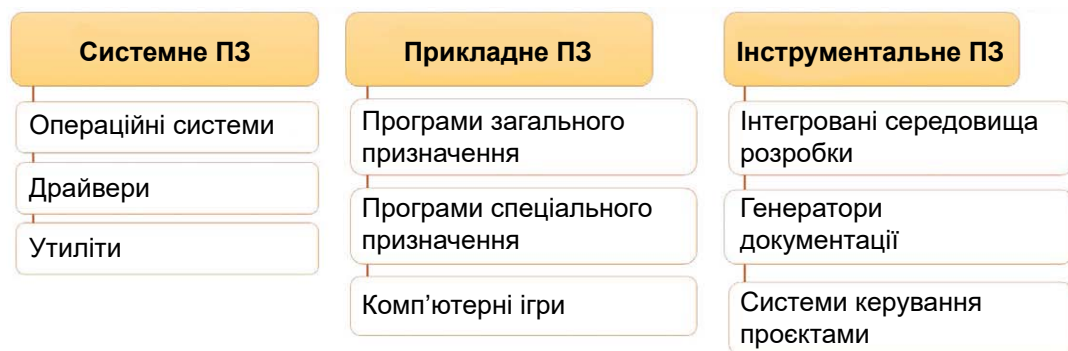


Рис. 1.1. Класифікація програмного забезпечення

Розглянемо ці види програмного забезпечення детальніше.

1.2.

Системне програмне забезпечення

Системне програмне забезпечення керує апаратними засобами комп'ютера, створює платформу для роботи прикладних програм та забезпечує взаємодію з користувачем.

До системного програмного забезпечення належать **операційна система**, **драйвери** та **утиліти**. Більш детально складові системного програмного забезпечення буде розглянуто в наступному параграфі.



1.3.

Прикладне програмне забезпечення

Ще один вид програмного забезпечення — *прикладне програмне забезпечення*.

Прикладне програмне забезпечення призначене для виконання певних завдань користувача та зазвичай розраховане на безпосередню взаємодію з ним.

Сьогодні існує велике розмаїття прикладних програм. Серед них є окремі спеціалізовані програми (медіапрогравачі, браузері, відеоредактори тощо), пакети програм (Microsoft Office — для офісних працівників, Adobe Suite — для дизайнерів), великі системи для керування цілими підприємствами і виробничими процесами (SAP ERP).

Прикладне програмне забезпечення можна умовно розділити на такі категорії, як **програми загального призначення**, **програми спеціального призначення** та **ігрові програми**.

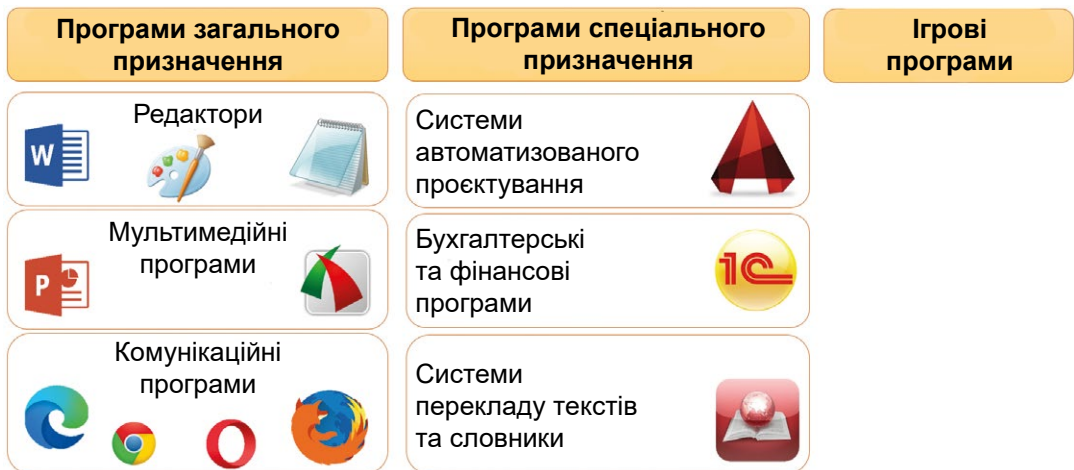


Рис. 1.2. Види прикладного програмного забезпечення

Програми загального призначення

До програм загального призначення можна віднести редактори, що дозволяють обробляти інформацію, яка представлена в текстовій, графічній, звуковій, числовій формах. Наприклад, текстовий процесор MS Word, графічний редактор Paint, текстовий редактор Блокнот.

Сюди також можна віднести мультимедійні програми, що дозволяють працювати з відеофрагментами, звуком, анімаціями, статичними зображеннями та гіпертекстом. Наприклад, програма для



створення презентацій PowerPoint, програма для роботи з відео FastStone Capture.

Комунікаційні програми призначені для підтримки користувацького інтерфейсу при роботі в мережі. Прикладом таких програм є різноманітні браузері.

Програми спеціального призначення

Наступна категорія — це *програми спеціального призначення*. До них можна віднести системи автоматизованого проєктування, що дозволяють виконувати креслення та конструювання з допомогою комп'ютера (наприклад, програма AutoCad); бухгалтерські та фінансові програми, що призначені для ведення бухгалтерського обліку, фінансового звітування та фінансової аналітики діяльності підприємств (наприклад, 1С:Бухгалтерія), а також системи перекладу текстів та словники (наприклад, Abbyy Lingvo).

1.4.

Інструментальне програмне забезпечення

Ще один вид програмного забезпечення — *інструментальне програмне забезпечення*. Його використовують на етапах проєктування, розробки і тестування іншого програмного забезпечення. Для кожного з перелічених етапів воно має свою особливість. Зокрема, розробники (програмісти) зазвичай працюють в інтегрованому середовищі розробки, що має зручний графічний інтерфейс та засоби для виконання всіх завдань. Прикладами є такі середовища програмування, як Python, C++ , Java тощо.

Також до інструментального програмного забезпечення належать генератори документації, системи керування проєктами тощо.

Інтегровані середовища розробки

Генератори документації

Системи керування проєктами



Рис. 1.3. Види інструментального програмного забезпечення

Розглянемо основні етапи роботи з інструментальним програмним забезпеченням.

На **першому етапі** здійснюється планування і проєктування.

**Етап 1. Планування і проєктування**

Визначаються вимоги до системи і коло завдань, які вона має вирішувати. На підставі цього розробляються загальна схема системи, набір її функцій. На цьому етапі також визначаються алгоритми роботи, згідно з якими згодом буде здійснюватися програмування

Документація, що описує будову і функції програмної системи, її інтерфейс, а також витрати часу і коштів на її створення

Другий етап полягає у розробці, тестуванні та створенні документації.

Етап 2. Розробка, тестування та створення документації

Здійснюється написання програм, їхнє тестування, під час якого перевіряється, чи працює система належним чином і чи відповідає вона вимогам; створюється система довідки для користувача та інша документація

Готове програмне забезпечення з документацією

На **третьому етапі** здійснюється впровадження та підтримка програмного забезпечення.

Етап 3. Впровадження і підтримка

Система надходить до користувача, встановлюється у його середовищі, налаштовується і запускається в роботу. У процесі використання системи виконуються завдання з її підтримки (додаткові налаштування, копіювання даних з метою запобігти їхній втраті тощо)

Програмне забезпечення встановлене і працює у користувача

Розглянута класифікація програмного забезпечення за призначенням не є однозначною. Наприклад, браузер Microsoft Edge, який, без сумніву, належить до прикладних програм, є невід'ємною складовою операційної системи Windows, а деякі програми, які загалом вважаються утилітами, мають стільки додаткових функцій, що їх можна віднести до прикладних програм.

ВПРАВА 1

Виконати завдання в середовищі Power Point, вказані у примітках файла **Вправа_1.pptx**.



Контрольні запитання та завдання

1. Що таке програмне забезпечення комп'ютера? Яку роль воно відіграє?
2. Як класифікують програмне забезпечення за призначенням?
3. Перелічіть види прикладного програмного забезпечення. Наведіть приклади.
4. Яке програмне забезпечення вважається системним?
5. Що належить до інструментального програмного забезпечення?
6. Перелічіть етапи роботи з інструментальним програмним забезпеченням.



Питання для роздумів

- 1*. Чому програму MS Excel можна віднести до прикладного програмного забезпечення?
- 2*. Наведіть приклади програм, які не можна однозначно віднести до одного із видів програмного забезпечення: системного, прикладного чи інструментального.



Завдання для досліджень

- 1*. Створіть перелік прикладного програмного забезпечення, необхідного для роботи учня.
- 2*. Створіть власну класифікацію такого виду прикладного програмного забезпечення, як комп'ютерні ігри.
- 3*. Визначте, які програми належать до систем автоматизованого проєктування. Які їхні можливості та де використовуються ці програми?
- 4**. Знайдіть інформацію про різновиди інструментального програмного середовища. Опишіть особливості використання принаймні одного з них.





§2.

СИСТЕМНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

2.1.

Поняття операційної системи

Ви вже знаєте, що програмне забезпечення поділяється на прикладне, системне та програмне. В рамках цього параграфу більш детально розглянемо системне програмне забезпечення. А саме, що таке операційні системи та їхні різновиди.

Операційна система — це набір програм, який забезпечує зв'язок між пристроями і прикладними програмами та керує пристроями, обчислювальними процесами і ресурсами.



Рис. 2.1. Схема взаємодії користувача з програмним та апаратним забезпеченням

На схемі вище видно, що операційна система взаємодіє з апаратним забезпеченням та прикладними програмами, які, у свою чергу, взаємодіють з користувачем. Отже, операційна система є ключовим компонентом комп'ютерної системи, який утворює прошарок між прикладними програмами та апаратним забезпеченням.

Операційна система виконує такі завдання:

- завантаження програм в оперативну пам'ять;
- виконання програм;
- забезпечення доступу до пристроїв введення-виведення;
- керування оперативною пам'яттю;
- керування доступом до даних на жорстких дисках;
- надає інтерфейс користувача.

Проте не всім обчислювальним пристроям потрібна операційна система. Наприклад, без неї працюють вбудовані комп'ютери, які застосовуються в побутових приладах, автомобілях, різноманітному устаткуванні. Ці пристрої постійно виконують лише одну програму, яка запускається на виконання, коли їх вмикають.

Необхідність в операційній системі виникає, коли обчислювальна система має виконувати багато завдань та коли користувачу потрібно надати можливість керувати виконанням окремих програм.



Сьогодні найпоширенішими операційними системами для настільних комп'ютерів і серверів є Windows, Linux та Mac OS різних версій. Для смартфонів та планшетів також розроблено спеціальні операційні системи, їх називають мобільними. Типовими представниками цих систем є Android та iOS.

Найчастіше операційні системи класифікують за такими ознаками:

- за цільовим пристроєм — існують операційні системи для потужних комп'ютерів-серверів, для персональних комп'ютерів, мобільних пристроїв;
- за кількістю програм, що виконуються одночасно, — розрізняють однозадачні та багатозадачні операційні системи;
- за типом інтерфейсу — з текстовим інтерфейсом і графічним;
- за кількістю розрядів даних, що обробляються одночасно — ця класифікація ґрунтується на розрядності процесорів. Найбільш поширеними є 32-розрядні та 64-розрядні. Скористатися перевагами більшої розрядності можна за умови, що операційна система та прикладні програми розроблені спеціально для процесора з такою розрядністю.

2.2.

Складові операційної системи

Операційна система має такі складові:

- ядро, що містить планувальник, який керує розподілом ресурсів;
- драйвери пристроїв, які допомагають керувати обладнанням;
- мережева підсистема;
- файлова система;
- системні бібліотеки;
- оболонка з утилітами (інтерфейс користувача).

Одні з цих складових постійно перебувають в оперативній пам'яті комп'ютера, поки він працює, інші завантажуються і запускаються на виконання лише за потреби.



Дізнайтеся більше про процес завантаження комп'ютера.

Зауважимо, що операційна система — дуже складний програмний продукт, який потребує обережного ставлення. Тому не видаляйте файлів, призначення яких вам невідоме, не порушуйте процес завантаження системи і не змінюйте налаштування, якщо не впевнені в результаті.



2.3.

Драйвери

Важливою складовою системного програмного забезпечення є драйвери.

Драйвери — це спеціалізовані програми, з допомогою яких забезпечується доступ програм до апаратного забезпечення.

Драйвер потрібен для використання будь-якого пристрою комп'ютера (як зовнішнього, так і внутрішнього). Пояснюється це тим, що в операційній системі реалізовано принцип абстрагування від апаратного забезпечення й для керування пристроєм певного типу (наприклад, принтером) вона застосовує стандартний набір команд. Драйвер приймає ці команди, перетворює їх на команди, «зрозумілі» пристрою певної моделі, та надсилає йому. Незавжди здогадатися, що цей підхід значно спрощує підключення до комп'ютера нових пристроїв, адже для цього виробникам достатньо створити відповідний драйвер, а користувачам — установити його у своїй системі.

Зараз завдяки технології Plug and Play («підключай і працюй») багато операційних систем самі розпізнають підключені до комп'ютера пристрої та встановлюють для них драйвери, а отже, користувачам не потрібно цього робити.

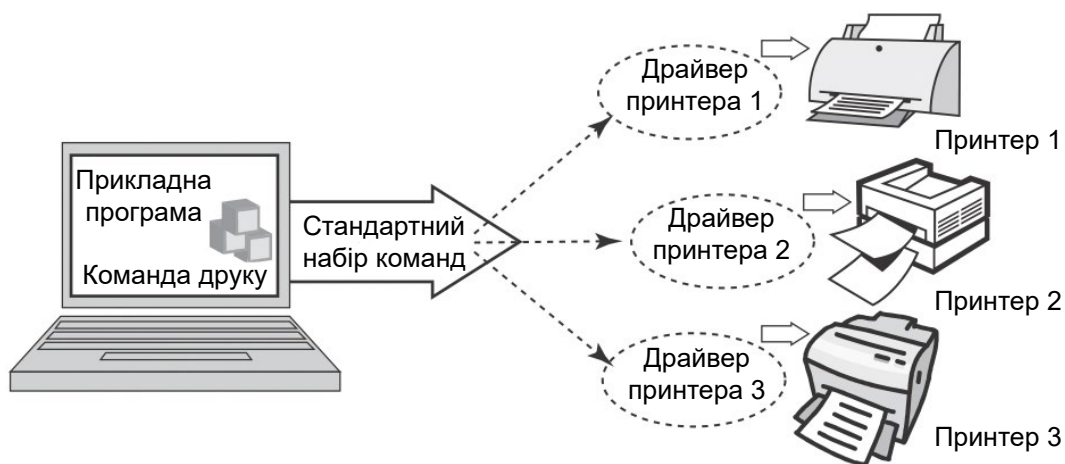


Рис. 2.2. Схема взаємодії пристроїв з використанням драйверів



2.4.

Утиліти

До системного програмного забезпечення відносять також утиліти.

Утиліти — це спеціалізовані програми, призначені для обслуговування та оптимізації роботи операційної системи.

Вони виконують такі завдання, як захист від вірусів, догляд за жорстким диском, моніторинг роботи комп'ютера, точне налаштування його параметрів тощо.

2.5.

Файлова система

Звичайні користувачі безпосередньо не взаємодіють з такими компонентами операційної системи, як ядро або системні бібліотеки, натомість із файловою системою та інтерфейсом користувача стикаються повсякчас.

Ви, звичайно, знаєте, що файлом називають іменовану сукупність даних, яка зберігається на зовнішньому запам'ятовуючому пристрої і пересилається та обробляється як єдине ціле. Файл може містити програму, числові дані, текст, закодоване зображення тощо.

Порядок, що визначає спосіб організації, зберігання та іменування файлів на носіях даних, називають **файловою системою**.

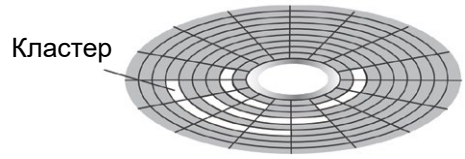
Забезпечують такий порядок відповідні програми, які входять до складу операційної системи. Зазначимо, що поняття файлової системи застосовується не лише до комп'ютерів, а й до інших пристроїв, оснащених носіями даних (цифрових камер, мобільних телефонів, електронних книжок тощо).

Розглянемо функції файлової системи. Файлова система:

- організовує кластери у файли і каталоги;
- відслідковує, які кластери зайняті, вільні або пошкоджені;
- відслідковує призначення імен файлам;
- надає можливість упорядковувати файли за каталогами;
- надає засоби для керування файлами (видалення, копіювання тощо);
- забезпечує шифрування файлів;
- надає можливості керувати доступом до файлів;
- надає програмам засоби для запису даних у файли.



Одне із завдань файлової системи — забезпечити ефективне використання фізичного простору носія. Це досягається завдяки тому, що фізично файли розташовуються на ньому не суцільними великими масивами, а невеликими блоками — *клас-терами*. Такий підхід дозволяє уникнути ситуації, коли внаслідок видалення файла на диску утворюється порожній простір і його не можна заповнити, оскільки неможливо підібрати файл, що має такий самий обсяг. Під час зберігання файла на диску саме драйвери файлової системи забезпечують такий його запис.



Файлова система також визначає довжину імені файла, допустимі в іменах символи, максимально можливий обсяг файла, набір його атрибутів (властивостей).

Для впорядкування файлів на пристроях файлова система надає можливість створювати структури каталогів. У системі Windows їх називають папками (за аналогією з канцелярськими папками). *Ка-талог* — це іменований об'єкт файлової системи, який може містити файли та інші каталоги.

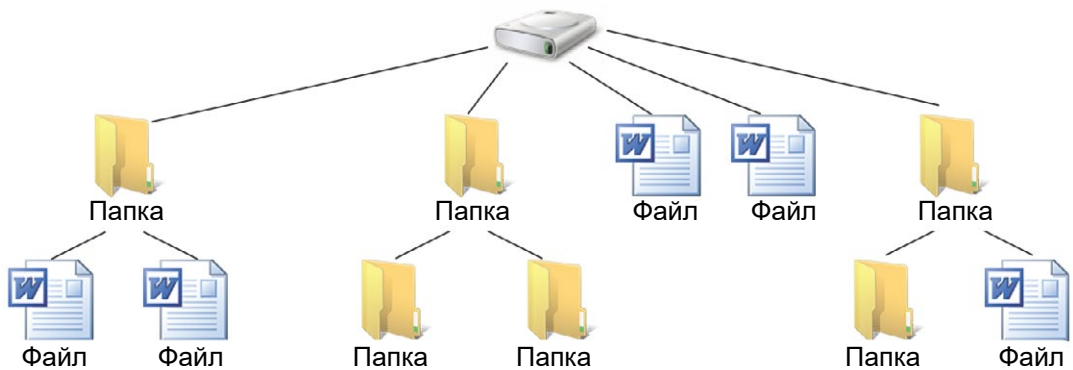


Рис. 2.3. Приклад структури каталогів та файлів у файловій системі

Загалом існує понад п'ятдесят файлових систем, які різняться підходами до іменування файлів, методами організації їх у каталоги та додатковими можливостями, такими, як шифрування файлів чи встановлення обмежень на доступ до них.



2.6.

Форматування носіїв даних

Файлова система перед першим використанням носія має відповідним чином його підготувати, зокрема визначити, в якій ділянці розміщуватиметься перелік файлів і каталогів та якими будуть класери. З цією метою нові носії завжди піддають форматуванню.

Форматування — це розмітка пристроїв зберігання файловою системою.

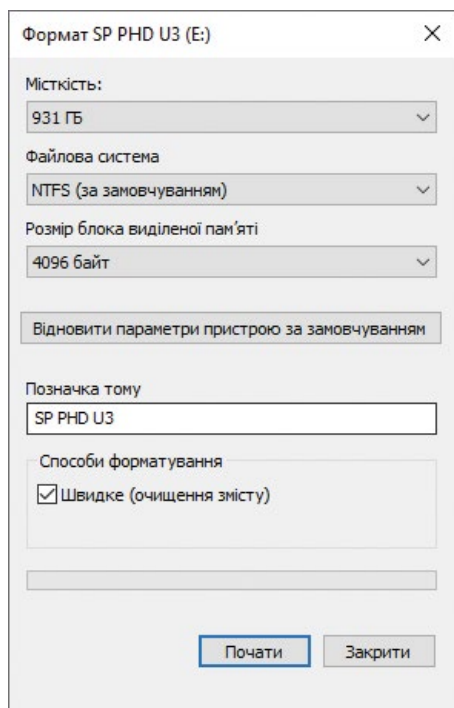


Рис. 2.4. Вікно форматування диска

Розмітка носіїв з лазерним способом запису виконується під час запису даних на них. Форматують також пристрої, на яких уже є дані. При цьому можливі такі варіанти:

- повне форматування — передбачає повторну розмітку диска, а тому всі дані, що містяться на ньому, знищуються;
- швидке форматування — очищуються кореневий каталог і таблиці розміщення файлів, дані в кластерах залишаються, а отже, існує можливість їх відновлення.

Практично кожному користувачеві доводиться формувати свою флешку, а тому розглянемо послідовність дій, які потрібно виконати для цього.

1. Вставити флешку в роз'єм порту USB.
2. Відкрити вікно **Цей комп'ютер**.
3. У контекстному меню флеш-накопичувача вибрати команду **Форматувати...**
4. У вікні **Формат** задати параметри форматування.
5. Натиснути кнопку **Почати**. Потім натиснути кнопку **Закрити**.

ВПРАВА 2



Виконати в середовищі PowerPoint завдання, описані у примітках у файлі **Вправа_2.pptx**.

**Контрольні запитання та завдання**

1. Наведіть приклади системного програмного забезпечення.
2. Які існують види операційних систем? Назвіть приклади.
3. Які складові операційної системи? Яка роль кожної з них?
4. Що таке драйвери?
5. Які програми-утиліти ви знаєте?
6. Які завдання виконує файлова система?
7. Яке призначення має операція форматування носія даних?

**Питання для роздумів**

- 1*. Чому, на вашу думку, спостерігається поступове стирання відмінностей між настільними та мобільними ОС?
- 2**. Чому смартфони та планшетні ПК постачаються з установленою ОС?

**Завдання для досліджень**

1. З'ясуйте, яку ОС встановлено на вашому мобільному пристрої.
- 2*. Знайдіть інформацію про операційну систему Linux. Які переваги та недоліки має ОС Linux порівняно з Windows?





§3.

ЛІЦЕНЗІЇ НА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ,
ЇХНІ ТИПИ

3.1.

Поняття авторського права

Ми вже познайомилися з видами програмного забезпечення. Дізнаємося, як здійснюється його розповсюдження та захист.

Програмне забезпечення є результатом праці багатьох людей, але, на відміну від одягу, меблів чи смартфонів, воно є нематеріальним продуктом. Постачатися програмне забезпечення може у встановленому вигляді разом із пристроєм (комп'ютером, планшетом тощо), існує можливість завантаження файлів програми з Інтернету або одержання на будь-якому носії інформації (диску, флеш-накопичувачі тощо).

Майже все програмне забезпечення захищається законами про авторське право.

Авторське право — це право автора наукових, літературних чи мистецьких творів отримувати блага від результатів своєї інтелектуальної та творчої діяльності.

Для охорони авторського права використовується знак © (копірайт). Знак охорони авторського права використовується з ім'ям фізичної або юридичної особи, якій належать авторські права. Також може бути вказаний об'єкт захисту авторського права. Вказується рік публікації або діапазон дат.



Для охорони авторського права використовується знак © (копірайт)

© С. Петренко, 2020

© ООО «Аїда», 2018–2020



Відсутність знака не означає, що твір не захищений авторськими правами, оскільки, згідно з Бернською конвенцією, авторське право виникає в момент створення твору, і для захисту авторських прав не



вимагається реєстрація твору чи дотримання будь-яких інших формальностей. Знак охорони авторського права не створює додаткових прав. Він, тільки повідомляє, що авторські права належать зазначеній фізичній або юридичній особі.

До речі, щоб надрукувати знак копірайту в одній із офісних програм достатньо перейти на друк в англійській розкладці клавіатури, надрукувати ліву дужку, літеру «с», а потім праву дужку, в результаті введена комбінація символів автоматично зміниться на символ копірайту.

3.2.

Ліцензія на програмне забезпечення

Правила використання програмного забезпечення визначаються ліцензією.

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



Дізнаємося, які ліцензії взагалі бувають і для чого вони потрібні. Взагалі абсолютно усі програми є ліцензованими. LibreOffice або Linux, які є безкоштовними, також є ліцензійними програмами. Напевно, ви звертали увагу, що навіть при встановленні безкоштовної програми все одно необхідно встановити прапорець, що ви приймаєте ліцензійну угоду. Тобто в будь-якому разі там написано, що з цією програмою ви маєте право робити, оскільки ліцензія безпосередньо пов'язана з авторським правом.

3.3.

Види ліцензій

Загалом усе ліцензоване програмне забезпечення можна розділити на дві великі групи: **власницьке ПЗ** і **вільне та відкрите ПЗ**.

Власницьке, або пропрієтарне ПЗ є платним. Видавці (або розробники) можуть продавати його самі або через посередників. Ліцензія посередника надає йому право лише продавати ПЗ. Існують також спеціальні ліцензії для постачальників і виробників обладнання, що надають їм право встановлювати ПЗ на свої вироби.

Приватні особи і підприємства, що купують ПЗ окремо або разом з обладнанням, вважаються кінцевими користувачами. Їхня ліцензія



називається *ліцензією кінцевого користувача* (End User License Agreement — EULA). Вона регламентує, скільки копій ПЗ можна встановити або скільки користувачів можуть мати доступ до встановленого ПЗ. Кінцевий користувач є лише власником своєї копії ПЗ, звичайно він не має авторських прав на нього, а отже, і прав на його розповсюдження (або передавання), навіть безкоштовне.

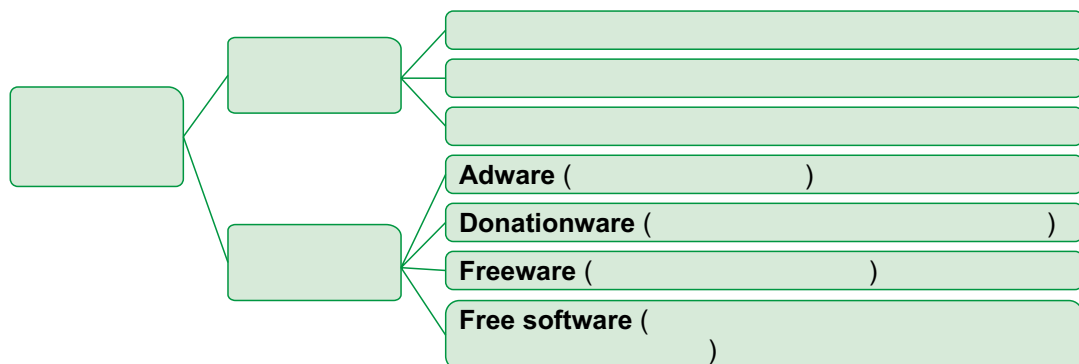


Рис. 3.1. Види ліцензій

Текст ліцензійного договору кінцевого користувача, як правило, вбудований у програмний продукт і відображається під час його встановлення (рис. 3.2). Користувач повинен прочитати і прийняти його, і лише після цього він зможе користуватися програмним продуктом.

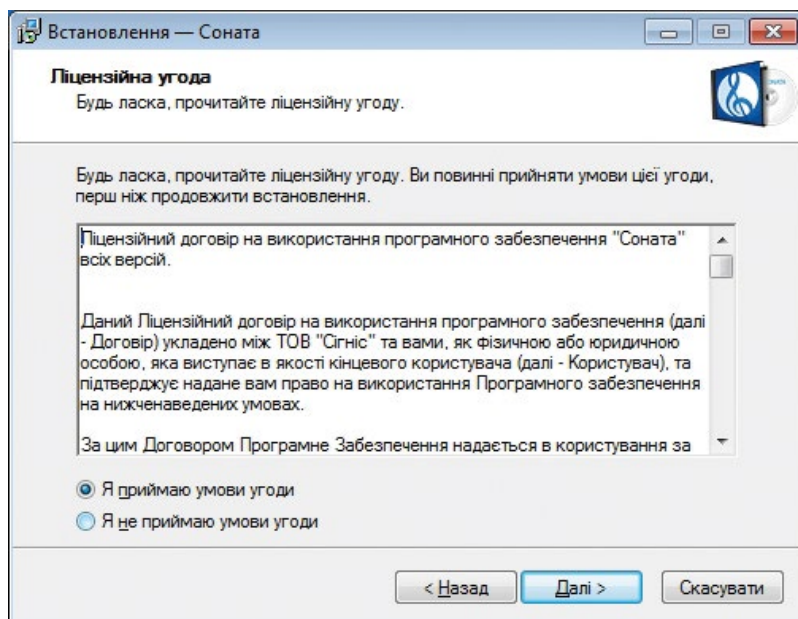


Рис. 3.2. Приклад ліцензійної угоди при встановленні програми



Види ліцензій на власницьке ПЗ

Існують такі види ліцензій на власницьке ПЗ.

Платна ліцензія на довічне використання. Той, хто придбав цю ліцензію, має право на довічне використання програмного забезпечення без додаткових доплат та обмежень.

Платна ліцензія з підпискою. Може використовуватися визначений проміжок часу. Її використання буде продовжено після внесення коштів через даний проміжок часу.

Ліцензія із безкоштовним пробним періодом. Надається безкоштовне використання на вказаний пробний період, після чого для продовження використання програми необхідно внести кошти.

Види ліцензій на вільне та відкрите ПЗ

Вільне та відкрите ПЗ вільно розповсюджується і є переважно безкоштовним (у деяких випадках користувачам пропонують зробити пожертву для розробників програми).

Існують такі види ліцензій на вільне та відкрите ПЗ:

Adware — програмне забезпечення, яке можна використовувати безкоштовно скільки завгодно часу, але при його використанні користувачеві примусово показується реклама.

Часто розробники програмного забезпечення рекламують у такий спосіб свої ж програми, більш функціональні і платні. Крім того, до Adware відносять ті програми, які без відома користувача під час встановлення змінюють домашню сторінку і пошуковий сервіс у браузері, задані за умовчанням, рекламуючи таким чином свої сервіси. Варто, однак, відзначити, що такі програми можуть бути дуже корисними в плані функціональності, а деякі навіть вельми популярні серед користувачів по всьому світу. Базове призначення Adware — це неявна форма оплати за використання ПЗ, що здійснюється за рахунок показу рекламної інформації (відповідно рекламодавці платять за показ їхньої реклами рекламному агентству, рекламне агентство — розробнику програми).

Ліцензія Donationware — це модель ліцензування, при якій користувач одержує повністю функціональне програмне забезпечення з можливістю зробити пожертвування розробнику. Розмір пожертвування може бути фіксованим або встановлюватися користувачем на підставі індивідуального сприйняття цінності програмного забезпечення.

Прикладами таких програм є графічний редактор Gimp, медіапрогравач VLC, текстовий редактор Vim.

Ліцензія Freeware — безкоштовне програмне забезпечення, що не потребує виплат правовласнику і не має обмежень щодо функціональності та часу роботи.



Прикладами таких програм є відкриті системи для керування вмістом інформації в Інтернеті та для створення невеликих сайтів CMS Joomla та Drupal, програма для видалення із системи файлів CCleaner.

Ліцензія Free software (вільне ПЗ) — це ПЗ, яке відрізняється виключно широкими правами, наданими користувачам. Прихильники вільного ПЗ, як правило, можуть безкоштовно користуватися ним протягом необмеженого періоду часу, а також поширювати його яким завгодно чином, міняти його початковий код та вдосконалювати.

Прикладами такого програмного забезпечення є офісний пакет Open Office, звуковий редактор Audacity, програма візуалізації операційних систем Virtual Box.



Контрольні запитання та завдання

1. Що таке авторське право? З якого моменту автор має права на власну розробку, твір?
2. Що таке ліцензія на програмне забезпечення?
3. На які дві групи можна розділити ліцензоване програмне забезпечення?
4. Які існують види ліцензій на власницьке ПЗ?
5. Опишіть види ліцензій на вільне ПЗ.



Питання для роздумів

1. Чи існує програмне забезпечення, що не вимагає прийняття ліцензійної угоди?
2. Чи може особа, що придбала ліцензію на програму, передавати права на її використання іншій особі?
3. Чи можна розмістити на власному сайті статтю іншого автора, опубліковану в іншому виданні?



Завдання для досліджень

- 1*. Дослідіть прикладне програмне забезпечення, встановлене на вашому комп'ютері. Визначте, який вид ліцензії воно має.
- 2*. Визначте, у чому відмінність між безкоштовним та вільним ПЗ.
- 3*. Дізнайтеся, що таке ліцензія GPL.



ПОНЯТТЯ ІНСТАЛЯЦІЇ ТА ДЕІНСТАЛЯЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. СУМІСНІСТЬ ПРОГРАМ



Поняття інсталяції та деінсталяції

Як зазначалося вище, відразу після виготовлення комп'ютер не містить жодних програм і фактично не може працювати. Спочатку на нього встановлюють (інсталюють) відповідну операційну систему (якщо це передбачено), а потім — певне програмне забезпечення. Більшість настільних комп'ютерів тепер продають з уже встановленою ОС, планшети та смартфони без ОС взагалі не постачають, тому користувачеві немає потреби вміти інсталювати ОС. Але як установлювати прикладні програми, повинен знати кожний користувач. Далі розглянемо поняття інсталяції та деінсталяції програмного забезпечення.

Інсталяція програмного забезпечення — це процес установлення програмного забезпечення на комп'ютер кінцевого користувача.

Зазвичай програми постачаються на дисках або завантажуються з Інтернету у вигляді дистрибутиву — набору, що містить спеціально призначені для інсталяції файли, а також програму-інсталятор.

Як правило, достатньо запустити програму-інсталятор, і вона сама виконає всю роботу з інсталяції — перевірить, чи відповідає система вимогам програми, розпакує і перепише у відповідні папки на жорсткому диску файли програми, зробить відповідні зміни в конфігураційні параметри комп'ютера.

Під час своєї роботи інсталятор виводить діалогові вікна, в яких користувачеві пропонується обрати функціональні компоненти програми, папку для її розміщення тощо. Після завершення роботи інсталятора програму можна запускати на виконання.

Якщо умовами ліцензування передбачена процедура активації ПЗ, після першого запуску програми відобразиться вікно для введення ключа продукту. Правильність ключа перевіряється через Інтернет, тому для успішної активації потрібно, щоб комп'ютер було підключено до Всесвітньої мережі.



Портативні програми

Зазначимо, що процедури інсталяції потребують не всі програми. Є програми, для яких процес установлення обмежується копіюванням файлів програми на жорсткий диск. Його виконує користувач самостійно. Такі програми називають *портативними* або переносними (portable application). Як правило, портативні програми є менш функціональними порівняно з тими, що вимагають установлення. Для вилучення їх з комп'ютера достатньо видалити файли програми з диска.



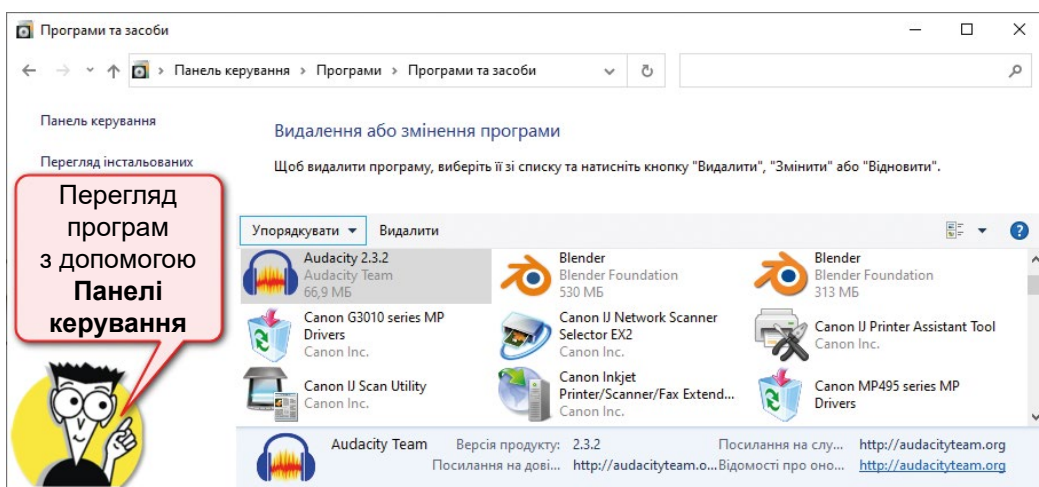
А от програми, що пройшли процедуру інсталяції, у таких спосіб у жодному разі вилучати не можна. Натомість потрібно за допомогою системних засобів або деінсталятора самої програми виконати її деінсталяцію. Під час цієї процедури не лише видаляються файли програми, а й відновлюються змінені нею параметри в операційній системі.

Деінсталяція програмного забезпечення — це процес вилучення програмного забезпечення.

4.2.

Які програми встановлені на комп'ютері

Багато користувачів не знають, які програми встановлені на їхньому комп'ютері. Для більшості програм при встановленні утворюються посилання в меню кнопки **Пуск** за алфавітом. Зауважимо, що видалення посилання з даного меню не означає видалення програми з комп'ютера.





Переглянути встановлені програми можна також з допомогою **Панелі керування** в категорії **Програми та засоби**. З допомогою даної категорії можна відновлювати, змінювати та деінсталювати програмне забезпечення, встановлене на комп'ютері.

Для виконання аналогічних дій існує безкоштовний і зручний у використанні програмний засіб CCleaner.

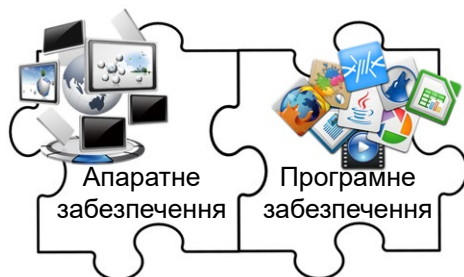
4.3.

Сумісність програмного забезпечення

Для успішного використання програмного забезпечення насамперед слід урахувати його сумісність з операційною системою. Наприклад, прикладні програми, призначені для використання в системі Windows, не працюватимуть на комп'ютері з іншою операційною системою. Проблема сумісності також часто виникає у випадках, коли сучасні програми намагаються встановити на комп'ютерах із застарілими операційними системами або, навпаки, давно розроблені програми — на комп'ютерах під керуванням останніх версій ОС.

Також при встановленні програмного забезпечення слід урахувати технічні характеристики комп'ютера — такі, як розрядність процесора, обсяг оперативної пам'яті, обсяг вільного простору на диску, обсяг відеопам'яті тощо.

Отже, при встановленні програмного забезпечення потрібно враховувати як технічні характеристики комп'ютера, так і програмні засоби, що будуть взаємодіяти із програмою, яка встановлюється.

**Контрольні запитання та завдання**

1. Що таке інсталяція і деінсталяція програмного забезпечення?
2. Яке програмне забезпечення називається портативним?
3. Де можна знайти програми, встановлені на комп'ютері?
4. Як виконати видалення портативних програм; встановлених програм?
- 5*. На що треба звертати увагу при встановленні програмного забезпечення?



Питання для роздумів

- 1*. У яких випадках зручно використовувати портативне програмне забезпечення? Опишіть загальну схему такого використання.
- 2**. Чому програми, призначені для однієї ОС, не працюють під керуванням іншої?



Завдання для досліджень

- 1*. Визначте, які версії MS Office підійдуть для встановлення на вашому комп'ютері.
- 2**. Знайдіть інформацію про придбання, встановлення і видалення прикладних програм для смартфонів під керуванням ОС Android.



§5.

СТИСНЕННЯ ТА АРХІВУВАННЯ ДАНИХ



5.1.

Стиснення даних

Хоча ємність пристроїв зберігання постійно збільшується, питання ефективного використання дискового простору й досі залишається актуальним. Це пов'язано не лише із загальним збільшенням обсягів інформації у світі, а й зі зміною її структури — зростанням частки графічної, відео- та аудіоінформації, що зберігається у файлах великих обсягів. Обсяг файлів зменшують, стискаючи дані.

Стиснення даних — це процедура перекодування даних з метою зменшення їхнього обсягу.

Для цього використовують різні методи, зокрема заміну часто повторюваних послідовностей коротшим значенням (кодом). Такими послідовностями можуть бути масиви пікселів однакового кольору в растровому зображенні, слова в тексті тощо. Наприклад, будь-яка тисяча символів тексту українською мовою містить літер «о» чи «і» на порядок більше, ніж літер «ф». Припустимо, що кожний символ тексту кодується двійковою послідовністю певної довжини. Зрозуміло, що у разі використання коротших кодів для поширених символів загальна довжина закодованого тексту буде меншою.



Буває стиснення **без втрат**, коли вихідні дані можна відновити без викривлень, і з **втратами**, коли дані відновлюються лише з викривленнями, які проте можуть бути непомітними для людини.

ВИДИ СТИСНЕННЯ

без втрат

стиснений файл можна відновити в початковому вигляді, знаючи алгоритм стиснення

$2+2+2+2+2+2+2 \rightarrow 2 \cdot 8$
 КОКОС КО=* $\rightarrow **C$
 ВАРВАР ВАР=* $\rightarrow **$

із втратами

при стисненні частина інформації безповоротно втрачається



Стиснення без втрат застосовують до текстової інформації, а стиснення із втратами — до звукової, графічної та відеоінформації, і саме воно дає змогу значно зменшити обсяги даних. В останньому разі на основі спеціальних досліджень визначають, якою інформацією можна пожертвувати. Відомо, наприклад, що зір людини дуже чутливий до зміни яскравості й значно менше — до кольорового відтінку, тому під час стиснення фотографій інформацію про колір частини пікселів можна видалити. Їхній колір під час розпакування зображення визначатиметься за кольором сусідніх пікселів.

Так, на схемі вище наведено приклад графічного файлу до стиснення та після нього. Якщо стиснутий файл розглядати з великої відстані, різницю важко буде помітити при наближенні в останньому файлі спостерігається погіршення якості.

Стиснення даних використовують для:

- збільшення вільного місця на диску;
- швидкої передачі по мережі;
- зменшення обсягу файла;
- резервного зберігання файлів.

5.2.

Програми-архіватори

Для стиснення даних можна використовувати спеціальні програми — *архіватори*. Файли, створені цими програмами, називають *архівами*.



Архіватор записує вміст файлів в архів і долучає до нього інформацію про імена файлів та обсяги їхніх оригіналів. Архіватор може створити один або кілька архівів з одного чи кількох файлів.

Архіватор — це програма, що стискає та об'єднує файли.

Процедура стиснення потребує певного часу. Чим більше буде стиснуто файл, тим менша швидкість цього процесу, і навпаки.



Програми-архіватори можуть працювати з архівними файлами різних типів. Найпопулярніші типи архівів — ARJ, ZIP, RAR, 7-Z. Найчастіше застосовують ZIP і RAR. Цю практику запроваджено завдяки програмам WinRAR та WinZip, які мають зручний інтерфейс і багато додаткових функцій, а також забезпечують досить високий ступінь стиснення.

Можливість стиснення даних не єдина перевага архіваторів. Щоб не втратити інформацію, її періодично копіюють на інші носії. Ця процедура, яку називають *резервним копіюванням*, є важливою складовою захисту даних. Для створення резервних копій зручно використовувати програми-архіватори, адже вони дають змогу впорядковувати файли. Наприклад, файли можна об'єднувати в архів за датою створення, за належністю до певного завдання чи проєкту тощо.

Отже, основними функціями програм-архіваторів є:

- створення архівів файлів та папок;
- перегляд вмісту архіву;
- додавання файлів в архів;
- знищення файлів в архіві;
- захист від несанкціонованого доступу;
- вилучення файлів з архіву.

5.3.

Архівування файлів і папок у Windows

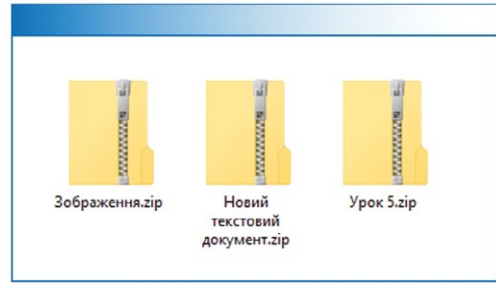
В операційну систему Windows вбудовано засоби для створення ZIP-архівів та керування ними. У програмі **Провідник** архіви подаються у вигляді папок із зображенням змійки, що мають розширення .zip.

Щоб створити архів, потрібно клацнути правою кнопкою миші на значок папки (або файла), яку необхідно стиснути, і в контекстному меню вибрати команду **Надіслати ▸ Стиснута папка**. В результаті у поточній папці з'явиться стиснута папка з таким самим ім'ям, як у архівованого об'єкта.



Щоб відкрити стиснуту папку, достатньо двічі клацнути на її значок. Після цього у вікні буде відображено вміст цієї папки. Для відкриття файла, збереженого в стиснутій папці, потрібно двічі клацнути значок файла.

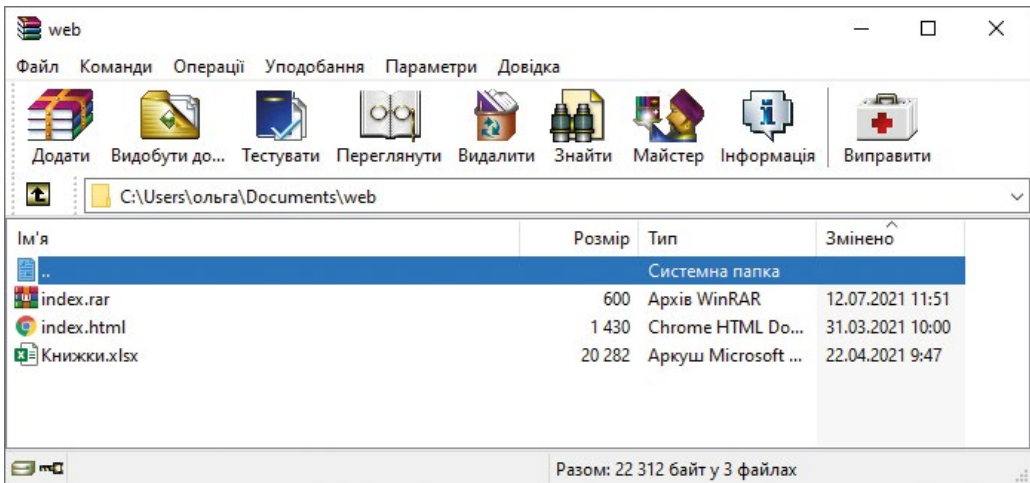
Можливості засобів архівування в операційній системі Windows обмежені, тому для створення архівів та керування ними зазвичай застосовують архіватор.



5.4.

Програма-архіватор WINRAR

Архіватор WinRAR має простий і зручний інтерфейс та забезпечує високий ступінь стиснення. Це умовно-безкоштовна програма, тобто її можна безкоштовно використовувати впродовж сорока днів.



Якщо програма WinRAR встановлена на комп'ютері, запустити її можна з допомогою меню кнопки **Пуск**, і, вибравши у папці **WinRAR** програму **WinRAR**, в результаті на екрані відкриється вікно програми.

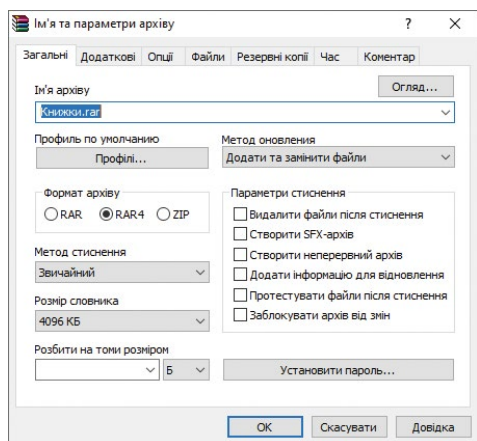
Як і більшість Windos-програм, другий рядок програми містить меню, команди якого дозволяють виконувати різноманітні дії в програмі. Для зручного використання програми передбачено панель інструментів. Призначення інструментів є інтуїтивно зрозумілим за їхньою назвою.



5.5.

Архівування файлів

Розглянемо процес архівування файлів на прикладі програми WinRAR 6.00. Після запуску програми WinRAR у її вікні буде відображено вміст тієї папки, з RAR-файлами якої працювали востаннє.



1. Для створення нового архіву, перейти до папки, де містяться файли, які потрібно заархівувати.
2. Виділити файли і папки, які слід заархівувати, та натиснути інструмент **Додати**.
3. У діалоговому вікні **Ім'я та параметри архіву** ввести ім'я архіву або підтвердити ім'я, запропоноване програмою за умовчанням.
4. У групі параметрів **Формат архіву** встановити формат нового архіву (RAR, RAR4 або ZIP).
5. У полі **Метод стиснення** обрати метод стиснення.

У разі застосування методу без стиснення файли поміщаються до архіву в початковому вигляді. Найменший обсяг архів матиме в разі вибору максимального стиснення. Цей метод доцільно вибирати під час створення архівів, призначених для довгострокового зберігання або для передавання через Інтернет.

6. Натиснути кнопку **ОК**. В результаті має запуститися процес архівації.

Якщо на комп'ютері встановлено програму WinRAR, файли можна пакувати в RAR-архів безпосередньо в програмі **Провідник**. Для цього слід виділити файл (або папку), клацнувши його правою кнопкою миші, та в контекстному меню вибрати команду **Додати до архіву**. В результаті відкриється вікно **Ім'я та параметри архіву**. Налаштування параметрів у цьому вікні вже було розглянуто вище.



Дізнайтеся про багатотомні та саморозпаковувані архіви.

Видобування файлів з архіву

Перший спосіб. У середовищі програми архіватора обрати файл архіву та натиснути кнопку **Видобути до**, після чого у вікні, що відкрилося обрати місце розташування файлів та інші параметри.




Другий спосіб. У контекстному меню файла-архіву обрати одну із команд: **видобути файли**, **видобути у поточну папку** тощо.

5.6.

Ступінь стиснення

Одним із основних параметрів файлів-архівів є ступінь стиснення файла, що дорівнює відношенню обсягу стисненого файла до обсягу вихідного файла, виражене у відсотках. Отже, чим меншою є ступінь стиснення, тим більше стиснуто файл.

Побачити ступінь стиснення можна з допомогою інструменту . В результаті на вкладці **Інформація** буде відображено загальний та стиснутий розміри файла, а також ступінь його стиснення (рис. 5.1).

На схемі нижче представлено файли, які стискаються краще, а які — гірше. Найбільше підлягають стисненню файли, що мають текст, нестиснені зображення, звук та відео. Найгірше стискаються випадкові дані, файли з розширенням .exe, а також графічні, звукові та відеофайли у стисненому форматі.

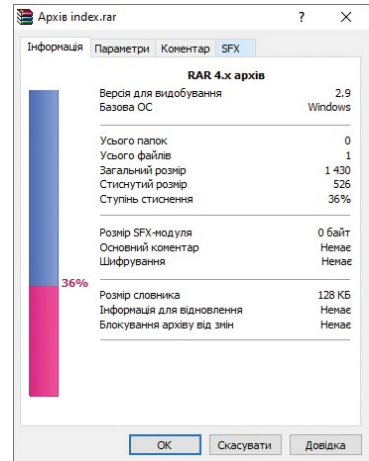


Рис. 5.1. Відображення ступеня стиснення файла

Добре стискаються

(*.txt)

(*.doc, *.xls)

(*.bmp)

(*.wav)

(*.avi)

Погано стискаються

(*.exe)

(*.gif, *.jpg, *.png)

(*.mp3, *.wav)

(*.mpg, *.wmv)



Контрольні запитання та завдання

1. Що таке стиснення даних?
2. Яким буває стиснення даних?
3. Для чого застосовують програми-архіватори?
4. Складіть і опишіть порядок резервного копіювання файлів із використанням програми-архіватора.





Питання для роздумів

- 1*. Одним із методів стиснення зображень є такий: у файлі окремо зберігається палітра кольорів, а при записі даних самого зображення замість кольору пікселя записується його номер у палітрі. Чи буде ефективним цей метод для стиснення чорно-білих зображень? Поясніть свою відповідь.



Завдання для досліджень

- 1*. Дізнайтеся про створення багатотомних архівів.
2*. Дізнайтеся, що таке саморозпаковувані архіви.
3**. Знайдіть інформацію про алгоритм стиснення Лемпеля – Зіва – Велча (англ. Lempel – Ziv – Welch, LZW) і дізнайтеся, як він працює.



ПРАКТИЧНА РОБОТА 1. «АРХІВУВАННЯ ТА РОЗАРХІВУВАННЯ ДАНИХ»



Створити архіви різних видів з допомогою програми-архіватора та видобути файли з архівів.

1. Створіть звичайний архів, до якого помістіть кілька файлів. Для цього в програмі **Провідник** перейдіть до папки з файлами, які потрібно заархівувати (оберіть довільну папку на вашому комп'ютері).
 - а) Виберіть ці файли, клацніть їх правою кнопкою миші та в контекстному меню виберіть команду **WinRAR ▶ Додати до архіву**. Відобразиться вікно **Ім'я та параметри архіву**. (Якщо вибрати команду **WinRAR ▶ Додати до <ім'я_файла>**, то архів буде створено в тій самій папці, де містяться вихідні файли.)
 - б) Оберіть шлях до папки для створення архіву та задайте ім'я архіву.
 - в) Виберіть формат **ZIP**, залиште нормальний метод стиснення. Клацніть кнопку **ОК**. В результаті архів буде створено.
2. Розархівуйте створений щойно архів.
 - а) Перейдіть до папки, в якій було створено архів. Двічі клацніть ім'я файла цього архіву. В результаті відкриється вікно програми-архіватора.
 - б) Із допомогою миші виберіть усі файли архіву та оберіть інструмент **Видобути до**, щоб розпакувати архів.



3. Створіть саморозпаковуваний архів. Для цього виконайте такі дії.
 - а) Мишею оберіть файли, клацніть їх правою кнопкою миші та в контекстному меню виберіть команду **WinRar** ▶ **Додати до архіву**, щоб відкрити вікно **Ім'я та параметри архіву**.
 - б) У вікні задайте ім'я архіву, виберіть формат **RAR**, залиште нормальний метод стиснення та встановіть прапорець **Створити SFX-архів**. Клацніть кнопку **ОК**. Архів буде створено.
4. Перемістіть створений щойно архів до іншої папки. Двічі клацніть файл архіву. Його вміст буде розпаковано в поточну папку.
5. Перемістіть до нової папки файл фотографії з розширенням **.jpg**, аудіофайл з розширенням **.mp3** та будь-який текстовий файл у форматі **.txt**. Створіть для кожного з них архіви у різних форматах (**rar**, **rar4**, **zip**). Порівняйте обсяги початкових файлів та архівів. Визначте ступінь стиснення, обравши інструмент **Інформація** у вікні архіватора. Результати запишіть у таблицю, наведену нижче. Зробіть висновки щодо ступеня стиснення різних типів файлів у різних форматах архівів.

Файл	Формат архіву	Обсяг вихідного файла	Обсяг архіву	Ступінь стиснення
.jpg	rar			
	rar4			
	zip			
.mp3	rar			
	rar4			
	zip			
.txt	rar			
	rar4			
	zip			



§7.

ІНФОРМАЦІЙНА БЕЗПЕКА

7.1.

Поняття інформаційної безпеки



З розвитком інформаційної індустрії з'являється все більше можливостей обробки та використання різноманітної інформації. Наявність цих можливостей спричиняє бажання зловмисниками скористатися цією інформацією, пошкодити її або зробити недоступною для користувача. Розглянемо, які існують загрози щодо втрати та пошкодження інформації, та визначимо шляхи уникнення таких загроз. Саме ці питання розглядає такий розділ інформатики, як *інформаційна безпека*.

Поняття інформаційної безпеки, залежно від його використання, розглядається у декількох ракурсах. В загальному випадку **інформаційна безпека** — це стан захищеності інформаційного середовища суспільства, який забезпечує його формування, використання і розвиток в інтересах громадян, організацій, держави.

Що ж стосується інформаційної безпеки особистості, то вона характеризується як стан захищеності особистості, різноманітних соціальних груп та об'єднань людей від впливів, здатних проти їхньої волі та бажання змінювати психічні стани і психологічні характеристики людини, модифікувати її поведінку та обмежувати свободу вибору.

7.2.

Комп'ютерні віруси

Користувачі проводять значну частину свого професійного і особистого життя в мережі. Але не всі усвідомлюють, що кіберзагрози, якими насичена мережа Інтернет, здатні завдати непоправної шкоди не тільки у фінансовому, а і в особистому плані. Більшість загроз несуть в собі шкідливі програми. До них належать віруси, трояни, мережеві хробаки, шпигунські програми, програми-вимагачі, рекламні програми тощо. З їхньою допомогою хакери можуть заволодіти вашими обліковими даними, паролями або даними банківських карт, заблокувати доступ до інформації, що зберігається на комп'ютері,



вимагаючи викуп за розблокування, видалити всю вашу особисту інформацію, знизити швидкість роботи комп'ютера і багато іншого.

Однією із перших шкідливих програм стала програма, що з'явилася на початку 70-х років в американській військовій комп'ютерній мережі ARPAnet. Програма дістала назву Creeper. Вона самостійно переміщалася по мережі та передавала свою копію на віддалений комп'ютер. При активізації цієї програми на комп'ютерах виводилося повідомлення I'm the creeper. Catch me if you can. (Я рептилія. Спіймайте мене, якщо зможете).

А власне термін «комп'ютерний вірус», вважається, уперше вжив співробітник Лежайського університету (США) Фред Коен. Сталося це 1984 року на конференції з безпеки інформації.

Відтоді минуло чимало часу, чіткого визначення комп'ютерного вірусу так і не з'явилося, проте багато різних програм мають властиві вірусам риси. Наведемо одне з поширених його визначень.

Комп'ютерний вірус — це програма, що активізується під час виконання іншої, зараженої нею програми. Після активізації вірус самовідтворюється, вражаючи програми на жорсткому диску та інших носіях.

Основна властивість вірусу — здатність самовідтворюватися, тобто створювати копії свого програмного коду без участі людини. Вірус може руйнувати дані або виконувати менш шкідливі дії (виводити на екран повідомлення, розсилати електронну пошту тощо).



7.3.

Середовище розповсюдження вірусу

Комп'ютерні віруси, як і звичайні віруси, є паразитами, яким для розмноження потрібен «хазяїн-носіє» — вони ховають у ньому свій код. Залежно від того, де розміщується цей «хазяїн-носіє», розрізняють такі типи вірусів.

Завантажувальні віруси. Розміщуються в тій самій ділянці диска, що й файли операційної системи, та активізуються в момент її завантаження.

Файлові віруси. Зазвичай містяться у виконуваних файлах (такі файли мають розширення імені .exe, .com, .sys або .dll).

Мережеві віруси. «Оселяються» в комп'ютерах мережі та розповсюджуються нею.



Макровіруси. Розміщуються у файлах документів із макросами. **Макроси** — це невеликі програми, написані мовою програмування, яку підтримують деякі прикладні програми. Наприклад, програми Microsoft Office підтримують мову Visual Basic, тож написані нею макровіруси можуть міститися в текстових документах Word і електронних таблицях Excel.

7.4.

Деструктивні можливості вірусів

Віруси по-різному впливають на роботу комп'ютера. Одні лише зменшують обсяг вільної пам'яті на диску, інші ж створюють різні ефекти: графічні, звукові тощо. Є віруси, що можуть призвести до серйозних перебоїв у роботі системи і навіть спричинити втрату даних.

Ознаки того, що комп'ютер уражено вірусом, можуть бути такими:

- сповільнення та збої в роботі комп'ютера, його часті «зависання»;
- неможливість завантажити операційну систему;
- припинення роботи чи неправильна робота програм, що раніше функціонували нормально;
- відправлення даних у мережу за відсутності дій з боку користувача, які можуть спричинити таке відправлення;
- збільшення кількості файлів на диску, змінення їхніх обсягів, дати й часу створення;
- періодична поява на екрані монітора недоречних повідомлень;
- руйнування файлової структури.

7.5.

Шкідливі програми



Дізнайтеся більше
про випадки ураження
шкідливими програмами

Окрім комп'ютерних вірусів, є й інші шкідливі програми, які дають змогу зловмисникам збирати інформацію, модифікувати та пошкоджувати її, а також порушувати роботу комп'ютера чи використовувати його ресурси зі зловмисною метою.

Троянські програми

Найбільшого поширення набули шкідливі програми, які називають *троянськими програмами*, троянськими конями або просто троянами.



Трояни — це шкідливі програми, що збирають інформацію, модифікують та пошкоджують її, порушують роботу комп'ютера чи використовують його ресурси у зловмисних цілях.

Троян не є вірусом, оскільки не здатен самовідтворюватися. Він запускається користувачем або автоматично. Щоб це сталося, трояна маскують під іншу програму (називають її іменем та призначають йому значок відомої програми) або надають неправдиву інформацію щодо призначення трояна (наприклад, описують його як безкоштовну екранну заставку).

Як же розпізнати троянську програму?

Якщо комп'ютер заражений, він буде працювати дуже повільно через навантаження, що надається трояном на обчислювальні ресурси пристрою.



Мережеві хробаки

Із розвитком мереж з'явилася ще одна категорія шкідливих програм — *мережеві хробаки*.

Мережеві хробаки — шкідливі програми, які потрапляють до комп'ютера через мережу. Мережевий хробак може спричинити втрату програм і даних, а також викрадення персональних даних користувача.

Вони здатні до самовідтворення, але, на відміну від комп'ютерних вірусів, не «оселяються» в інших програмах.

Як же розпізнати мережевого хробака?

Як правило, хробаки споживають велику кількість оперативної пам'яті й пропускної здатності мережі, тому сервери і робочі станції часто просто перестають відповідати і реагувати на сигнали.



Шпигунські програми

Ще один тип шкідливих програм — *шпигунські програми*.

Присутність їх в системі практично непомітна для користувача і часто не піддається виявленню. Така програма збирає інформацію про історію пошуку, звички та вподобання користувача — або ж його конфіденційні дані (наприклад, дані банківської картки) і без відома





користувача пересилає зібрану інформацію на віддалені сервери, що належать зловмисникам.

Як же розпізнати шпигунську програму? Присутність шпигунського ПЗ часто супроводжується появою нових або невідомих значків в панелі завдань унизу екрана, а також виконанням пошукових запитів іншим пошуковим провайдером, аніж налаштований за замовчуванням. Час від часу можуть з'являтися повідомлення про помилки при виконанні операцій, які раніше виконувалися без труднощів.

Програми-вимагачі



Рис. 7.1. Вікно програми-вимагача

До шкідливих програм також відносять програми-вимагачі. Ці програми обмежують доступ користувача до системи комп'ютера і вимагають викуп за повернення доступу до пристрою. Найнебезпечніші атаки були проведені програмами-вимагачами WannaCry, Petya, Cerber, Cryptolocker та Locky.

Програми-вимагачі створюються високопрофесійними програмістами. На пристрій користувача програма може проникнути через файл-вкладення в повідомленні електронної пошти

або через браузер у разі відвідування сайту, інфікованого даним типом шкідливого програмного забезпечення. Вона також може проникнути на пристрій користувача з локальної мережі.

Інфікування програмою-вимагачем видно неозброєним оком, оскільки в більшості випадків пристрій повністю заблоковано, і ви просто не зможете ним скористатися.

Рекламні програми



Менш шкідливою, але доволі розповсюдженою є рекламна програма, що використовується недобросовісними рекламними агентствами для відображення спливаючих повідомлень рекламного характеру за допомогою діалогових вікон і модулів в основній системі або в браузері. У своїй більшості рекламні програми дратують, але не створюють реальної загрози безпеці. Однак деякі використовуються для збору особистих даних, відстеження історії відвідувань і навіть для перехоплення введеного користувачем тексту.



7.6.

Зловмисні дії в інформаційному просторі

Поява на комп'ютері небажаного шкідливого програмного забезпечення виникає внаслідок різноманітних зловмисних дій. Зловмисні дії, з якими ви вже знайомі ще з сьомого класу, — це *спам-атаки* та *фішинг*. Які ще бувають зловмисні дії, розглянемо далі.

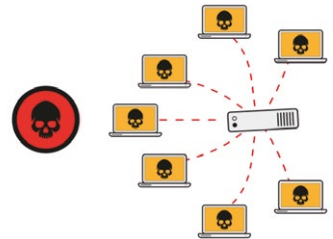
Дії, що завдають шкоду комп'ютерам, виконують особи, яких у суспільстві називають **хакерами**. По суті, звичайний хакер — це талановитий програміст, який добре володіє керуванням або модифікацією комп'ютерних систем та мережі. Проте існують і хакери-зловмисники, що використовують свої знання і навички з метою отримання несанкціонованого доступу до банківських або персональних даних з метою отримання вигоди.

Під хакерською атакою розуміють несанкціоноване проникнення в систему комп'ютера з метою подальшої маніпуляції його роботи і пов'язаних з ним систем. Це досягається використанням спеціальних програм, які маніпулюють обробкою прийнятих через підключення до мережі даних, щоб отримати доступ до конфіденційної інформації.

DDoS-атаки

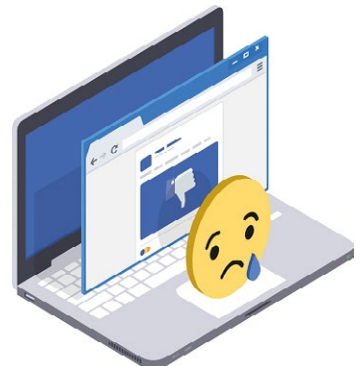
Якщо ви коли-небудь стикалися з тим, що при спробі відкрити веб-сайт завантаження було незвично повільним, з'єднання переривалося або ви так і не змогли дочекатися повного завантаження сторінки, то цілком можливо, що він перебував під так званою *DDoS-атакою*.

Атаки DDoS намагаються вивести з ладу сайти або цілі мережі, перевантажуючи їх запитами, що відправляються з тисяч інфікованих комп'ютерів, більш відомих як мережа під назвою *ботнет*.

**Кібербулінг**

На сьогоднішній день одним з найбільш згубних типів онлайн-атак є *кібербулінг*. Під **кібербулінгом** розуміють цькування, образи чи погрози, висловлені жертві за допомогою засобів електронної комунікації.

Кібербулінг зазвичай проявляється у відправці жертві повідомлень з погрозами або публікацій, що принижують гідність жертви, фотографій і відео в соціальних мережах або навіть у створенні підблених вебсайтів. На жаль, дії кібербулінгу часто





мають трагічні наслідки, що призводять до пошкодження здоров'я або навіть загрожують життю особи.

7.7.

Правила особистого захисту

Існує ряд правил, які дозволяють захистити особисті дані від втручання зловмисників.

Не повідомляйте свою електронну адресу на сумнівних сайтах і форумах.

Не переходьте за посиланнями, надісланими в підозрілих або незрозумілих повідомленнях.

Не завантажуйте вкладені файли з повідомлень електронної пошти, які ви не очікували.

Не повідомляйте нікому свої персональні дані по телефону, особисто або в повідомленні електронної пошти.

Уважно проаналізуйте адреси сайту (URL), на який ви були переадресовані.

Підтримуйте свій браузер оновленим і своєчасно встановлюйте оновлення безпеки.

7.8.

Профілактика ураженню шкідливими програмами

Також існують правила, яких слід дотримуватися, аби зменшити ймовірність піддатися кібератакам, ураженню комп'ютерними вірусами або нанесенню шкоди шкідливим програмним забезпеченням:

- не запускайте на виконання незнайомі програми;
- не користуйтеся піратськими копіями програмних продуктів;
- не відкривайте одержані через Інтернет файли без їхньої перевірки на наявність вірусів;
- аби не втратити важливі дані, періодично створюйте резервні копії важливих файлів на зовнішніх носіях інформації;
- використовуйте якісні антивірусні засоби та програми захисту від усіляких можливих кібератак.

7.9.

Антивірусні програми

Для боротьби з комп'ютерними вірусами та шкідливим програмним забезпеченням розробляють антивірусні програми, або антивіруси. Одним із поширених антивірусів є avast, роботу з цією програмою ми вивчимо докладніше. Потужні, постійно оновлювані засоби захисту від вірусів вбудовано також в операційну систему Windows.



Програми цієї категорії зазвичай виконують такі дії:

- сканування пам'яті комп'ютера, а також файлів, що записуються та читаються;
- сканування пам'яті та вмісту дисків за розкладом;
- сканування стиснених файлів;
- розпізнавання поведінки, властивої комп'ютерним вірусам;
- автоматичне оновлення антивірусних баз через Інтернет;
- перевірка даних, які надходять з Інтернету;
- ведення журналів подій, що стосуються антивірусного захисту.



До речі, відомий програміст Пітер Нортон у 1988 році зробив цікаву заяву. Він уважав, що вірусів узагалі не існує, і порівняв комп'ютерні шкідливі програми з міфами про крокодилів, що живуть в нью-йоркських каналізаціях. Однак це не завадило Пітеру розпочати власний антивірусний проєкт під назвою Norton AntiVirus.



Відповідно комп'ютерні віруси, крім руйнівної дії, дали життя новій галузі економіки — щорічно антивірусні компанії заробляють до 2 мільярдів доларів на виробництві антивірусного програмного забезпечення.



Класифікація антивірусних програм

За ознакою розміщення в оперативній пам'яті антивірусні програми поділяють на резидентні та нерезидентні:

- **резидентні** (починають свою роботу при запуску операційної системи, постійно знаходяться в пам'яті комп'ютера і здійснюють автоматичну перевірку файлів);
- **нерезидентні** (запускаються на вимогу користувача або відповідно до заданого для них розкладом).

Також антивірусні програми розділяють **за способом захисту** від вірусів.

- **Програми-детектори**, або сканери. Ці антивіруси знаходять віруси в оперативній пам'яті, на внутрішніх або зовнішніх носіях та виводять повідомлення при виявленні вірусу.
- **Програми-лікарі** знаходять заражені файли та «лікують» їх.



- **Програми-вакцини**, або імунізатори. Виконують імунізацію системи (файлів, каталогів), блокуючи дію вірусів.
- **Програми-ревізори** є найбільш надійними в плані захисту від вірусів. Ревізори запам'ятовують початковий стан програм, каталогів, системних ділянок диска до моменту інфікування комп'ютера, потім порівнюють поточний стан з початковим, виводячи знайдені зміни на дисплей.
- **Програми-монітори** починають свою роботу при запуску операційної системи, постійно знаходяться в пам'яті комп'ютера і здійснюють автоматичну перевірку файлів. Отже, діють за принципом «тут і зараз».
- **Програми-фільтри**, або сторожі — виявляють вірус на ранній стадії, поки він не почав розмножуватися. Програми-фільтри — це невеликі резидентні програми, метою яких є виявлення дій, характерних для вірусів.



7.11. Можливості антивірусної програми avast

Роботу з антивірусними програмами розглянемо на прикладі умовно безкоштовної програми avast.

Щоб захистити свій комп'ютер із використанням антивірусної програми, насамперед потрібно встановити її на комп'ютері. Якщо йдеться про безкоштовний засіб на зразок avast, то завантажити інсталяційний файл можна з офіційного сайту **avast.ua**.

Безкоштовна версія антивірусу Free Antivirus надає такі можливості:

- блокування вірусів та іншого шкідливого ПЗ;
- пошук слабких місць у захисті мережі wi-fi;
- захист паролів;
- додатковий рівень захисту від програм-вимагачів.

Даного набору можливостей для домашнього використання комп'ютера більш ніж достатньо.

Доступ до розширених можливостей цієї програми є платним. До них належать: безпечний запуск підозрілих програм, захист від спаму та фішингу, захист від стеження через вебкамеру тощо. Проте, який саме рівень захисту обрати для власного комп'ютера, обирати власне вам.

Установити антивірус дуже просто — для цього достатньо запустити на виконання завантажений файл. Після встановлення антивірусу він запропонує виконати перше сканування комп'ютера.



Як правило, антивірус постійно працює у фоновому режимі та захищає комп'ютер. Це можна побачити у прихованому меню панелі завдань.

Крім того, антивірус повинен надавати можливість проводити сканування на наявність вірусів вручну. Тому практично всі антивіруси мають вкладки або команди для запуску цієї процедури. Не є виключенням і avast.

Для відкриття антивірусу можна обрати відповідне посилання у меню кнопки **Пуск** або у прихованому меню панелі завдань. Після запуску антивірус насамперед сповіщає про поточний стан захисту системи.

Загалом інтерфейси антивірусних програм подібні, адже набір їхніх команд та інших елементів визначається призначенням цих програм. Більше команд можна побачити безпосередньо в меню програми.

Сканування комп'ютера

1. Обрати кнопку **Захист** ► **Перевірка на віруси**.
2. Обрати один із запропонованих видів сканувань:

Інтелектуальне сканування. Виконує сканування за встановленими за умовчанням параметрами програми.

Повне антивірусне сканування. Виконує сканування усіх файлів та папок комп'ютера.

Вибіркове сканування. Цей спосіб дозволить обрати необхідні папки та диски, які необхідно перевірити.

Сканування під час завантаження системи. Цей вид сканування здійснюватиме сканування на віруси при наступному завантаженні операційної системи.

Якщо жоден із способів не влаштовує, слід обрати опцію **Особливе сканування**. Завдяки цій опції створюється нове сканування, в якому можна врахувати бажані опції сканування.

Будь-яка антивірусна програма у своїй роботі спирається на власну базу даних з описами вірусів, а отже, для надійного захисту комп'ютера дуже важливо, щоб база даних періодично оновлювалася, адже нові віруси з'являються мало не щодня. Тому зазвичай автоматичне оновлення встановлюється в антивірусних програмах за умовчанням. Але керувати цим процесом можна і вручну.

Так, в avast для цього необхідно відкрити меню та обрати команду **Налаштування**, після чого команду **Оновлення**.

Крім оновлення, вікно налаштувань надає можливості керування й іншими параметрами. Так, наприклад, обравши команду **Винятки** можна додати імена програм та адреси вебсторінок, яким дозволено, виконання на комп'ютері.



Підозрілі файли, виявлені екранами в режимі реального часу, автоматично переміщуються в спеціальну область — карантин, де вони ізолюються. При ручному скануванні у разі виявлення підозрілих та заражених файлів програма відобразить повідомлення про підозрілі файли і доступні варіанти дій з ними — видалити або перемістити в карантин. Файли, розміщені в карантині, можуть бути проаналізовані та в майбутньому дозволені для використання або навпаки розпізнані як шкідливі програми та видалені.

Щоб переглянути файли, розміщені в карантині, необхідно обрати команду **Захист**, а після цього **Сховище вірусів**.

7.12.

Захист від шкідливих програм у Windows 10

На комп'ютері під керуванням операційної системи Windows 10 антивірусну програму можна і не встановлювати. Адже до цієї системи вже вбудовано засіб **Захист від вірусів і загроз**. За умовчанням цей засіб у режимі реального часу слідкує за безпекою вашого комп'ютера і у разі виникнення загрози повідомить про це.

Відкрити цей засіб можна, обравши в меню **Пуск** команду **Налаштування**. У вікні, що відкриється, потрібно обрати категорію **Оновлення та захист**. У наступному вікні слід обрати команду **Безпека Windows**. Після чого обрати засіб **Захист від вірусів та загроз**.

В цьому вікні ви побачите рівень захисту вашого комп'ютера, а також інформацію про поточні загрози (звичайно, якщо вони існують). Для боротьби з загрозами запропоновано такі варіанти дій, як *видалити*, *помістити в карантин* або *дозволити виконання на комп'ютері*.

Як і в будь-якому антивірусі, в даному вікні можна здійснити налаштування захисту від вірусів та загроз.

Крім засобу **Захист від вірусів та загроз** у Windows вбудовано ще одну програму захисту вашого комп'ютера, що має назву **Брандмауер**. Цей засіб виконує функцію перевірки і фільтрації даних, що надходять з Інтернету. Залежно від налаштувань брандмауер може пропустити їх або заблокувати. Брандмауер захищає комп'ютер від проникнення в нього хакерів або зловмисних програм (наприклад, хробаків) через мережу або Інтернет. Крім того, брандмауер запобігатиме надсиланню зловмисних програм із вашого комп'ютера на інші.

**ВПРАВА 7**

Виконайте швидку перевірку на віруси вашого комп'ютера та однієї із папок.

План перевірки з використанням вбудованого засобу **Захист від вірусів та загроз** в ОС Windows10. У разі виконання вправи на комп'ютері з іншою операційною системою виконайте аналогічні дії з антивірусною програмою, встановленою на вашому комп'ютері.

1. Виконайте швидку перевірку на віруси комп'ютера.
 - a. Натисніть **Пуск** ► **Налаштування**.
 - b. Оберіть **Оновлення та захист**.
 - c. Оберіть категорію **Безпека у Windows**.
 - d. Оберіть додаток **Захист від вірусів та загроз**.
 - e. В області **Поточні загрози** натисніть кнопку **Швидка перевірка**.
2. Якщо буде знайдено заражений або підозрілий на зараження файл, програма виведе відповідне повідомлення та для боротьби із загрозою запропонує такі варіанти дій: *видалити*; *помістити в карантин*; *дозволити виконання на комп'ютері*.
3. Після завершення перевірки ознайомтеся зі звітом, в якому має бути вказано, коли була здійснена перевірка, кількість виявлених загроз, скільки тривала перевірка та скільки було перевірено файлів.
4. Виконайте перевірку на віруси папки **Документи**. Для цього в контекстному меню папки оберіть команду **Сканування за допомогою Windows Defender**. В результаті має відкритися вікно **Безпека у Windows** і розпочнеться сканування обраної папки.

**Контрольні запитання та завдання**

1. Що таке інформаційна безпека?
2. Які шкідливі програми ви знаєте?
3. Що таке комп'ютерний вірус? Які ознаки ураження вірусом?
4. Які існують види вірусів за середовищем їхнього розповсюдження?
5. Які зловмисні дії ви знаєте?
6. Як уникнути дій шкідливих програм?
7. Які існують види антивірусних програм за способом захисту?

**Питання для роздумів**

- 1*. Що спільного та які відмінності між вірусами, троянами та хробаками?
- 2*. Чи доцільно на комп'ютер встановлювати декілька антивірусних програм?



Завдання для досліджень



- 1*. Дізнайтеся, що таке кейлогери, як вони потрапляють до комп'ютерів користувачів.
- 2*. Дізнайтеся, що таке ботнет і які його функції під час виконання DDoS-атак.
- 3*. Дослідіть можливості однієї з антивірусних програм.



Висновки

Залежно від призначення програмне забезпечення поділяють на **прикладне, системне та інструментальне**.

Системне програмне забезпечення керує апаратними засобами комп'ютера, створює платформу для роботи прикладних програм та забезпечує взаємодію з користувачем.

Прикладне програмне забезпечення призначене для виконання певних завдань користувача та зазвичай розраховане на безпосередню взаємодію з ним.

Програмне забезпечення, яке використовується на етапах проєктування, розробки і тестування іншого програмного забезпечення називають **інструментальним**.

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

Інсталяція ПЗ — це процес установлення програмного забезпечення на комп'ютер кінцевого користувача.

Стиснення даних — це процедура перекодування даних з метою зменшення їхнього обсягу. Для стиснення даних на комп'ютері використовують **архіватори**.

Стабільній роботі комп'ютера можуть загрожувати такі шкідливі програми: комп'ютерні віруси, трояни, мережеві хробаки, шпигунські програми, програми-вимагачі, рекламні програми тощо.

Поява на комп'ютері небажаного шкідливого програмного забезпечення виникає внаслідок різноманітних зловмисних дій, таких як **хакерські атаки, спам, фішинг, DDoS-атаки** тощо.

Для боротьби зі шкідливими програмами розробляють **антивірусні програми**.



Розділ 2

Опрацювання табличних даних



АБСОЛЮТНІ, ВІДНОСНІ ТА МІШАНІ ПОСИЛАННЯ



Посилання на адреси клітинок

У даному розділі ми продовжимо ознайомлення з табличним процесором. Як ви пам'ятаєте з 7-го класу, в табличному процесорі зручно можна представити табличні дані, виконувати обчислення та працювати зі списками. В межах даного параграфа розглянемо види посилань табличного процесора, завдяки яким обчислення можна виконувати значно зручніше та швидше.

Нагадаємо, що **табличний процесор** — це прикладна програма для опрацювання даних, поданих у таблицях. Прикладом такої програми є Microsoft Excel. В даному параграфі завдання буде розглянуто на прикладі MS Excel 2019. Таблиці, які опрацьовує електронний процесор, називають **електронними таблицями**. Робоче поле табличного процесора MS Excel, представлене у вигляді таблиці, називають **робочим аркушем**. Робочий аркуш поділений на рядки та стовпці. Стовпці мають назви, що позначаються латинськими літерами. Рядки таблиці позначаються натуральними числами. **Клітинка таблиці** — це елемент для збереження даних у таблиці. Кожна клітинка має адресу, що складається з назви стовпця і номера рядка, на перетині яких вона знаходиться. Клітинка може містити **число**, **текст** або **формулу**.

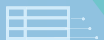
Якщо в таблиці потрібно створити список або застосувати ту саму формулу декілька разів в одному стовпці чи рядку, зручно використовувати **автозаповнення**,



Пригадайте основні поняття, пов'язані з електронним процесором.



Дізнайтеся про різні версії Excel.



	A	B	C	D	E
1			=D3		
2			=D4		
3	=B5	=C5	=D5	=E5	=F5
4			=D6		
5			=D7		

при якому утворюється послідовність чисел або копіюються формули з урахуванням нових даних.

Як вам уже відомо, під час копіювання формули, адреса клітинки, що входить до неї, змінюється залежно від того, куди буде здійснено копіювання.

Але це не завжди зручно. Іноді в завданні використовують дані, які є сталими і не потребують змін. А отже, у формулі вони також мають бути сталими. Розглянемо такий випадок на прикладі наступного завдання.

ПРАКТИЧНЕ ЗАВДАННЯ 1



Відомо ціни товарів у гривнях. Потрібно перевести ціну товару з гривень у деяку грошову одиницю, наприклад, в холінги.

	A	B	C
1	Курс гривні/холінги	10	
2			
3	Найменування	Ціна, грн.	Ціна, хол.
4	Пенал	100	10,00
5	Рюкзак	850	
6	Ручка	20	

	A	B	C
1	Курс гривні/холінги	10	
2			
3	Найменування	Ціна, грн.	Ціна, хол.
4	Пенал	100	10,00
5	Рюкзак	850	#ДІЛЕННЯ/0!
6	Ручка	20	#ЗНАЧЕННЯ!

Зрозуміло, що для виконання поставленого завдання, в клітинку **C4** треба записати формулу **=B4/B1**.

Для автоматизації обчислень в наступних клітинках у 7-му класі ми використовували автозаповнення. Використаємо автозаповнення і зараз.

Але, що ми бачимо? У клітинках **C5** та **C6** виникли помилки. Давайте розбиратися, чому вони з'явилися.

Застосуємо до аркуша режим перегляду формул та проаналізуємо їх. Для цього на вкладці **Формули** оберемо інструмент **Показати формули**.

Отже, ми бачимо, що в клітинках **C5** і **C6** ділення відбувається не на значення клітинки **B1**, у якій розташовано курс валют, а на клітинки, розташовані нижче (**B2** та **B3**). Звичайно, якщо змінити у цих формулах посилання з **B2** та **B3** на **B1**, ділення буде виконуватися правильно.

	A	B	C
1	Курс гривні/холінги	10	
2			
3	Найменування	Ціна, грн.	Ціна, хол.
4	Пенал	100	=B4/B1
5	Рюкзак	850	=B5/B2
6	Ручка	20	=B6/B3

Виправлення формул у такий спосіб для двох рядків, звичайно, не є проблемою. А якщо таких формул потрібно ввести не три, а двадцять або навіть сто і більше? Що тоді робити? Як не припуститися помилки і скоротити час на введення формул?



Якщо уважно поглянути на останні, вже правильно введені формули, то всі вони містять посилання на ту саму клітинку **B1**.

Щоб посилання на клітинку **B1** в результаті копіювання формули не змінювалося, адресу клітинки **B1** слід «зафіксувати». У програмі Excel це роблять за допомогою символу **\$**. Його можна розташовувати перед назвою стовпця і перед номером рядка залежно від того, яку частину формули потрібно зафіксувати.

	A	B	C
1	Курс гривні/холінги	10	
2			
3	Найменування	Ціна, грн.	Ціна, хол.
4	Пенал	100	=B4/B\$1
5	Рюкзак	850	=B5/B\$1
6	Ручка	20	=B6/B\$1

Для нашого випадку знак долара потрібно поставити перед номером рядка, оскільки формулу потрібно копіювати по рядках у межах одного стовпця.

Залежно від того, яку частину адреси зафіксовано та чи зафіксовано її взагалі, розрізняють відносні, абсолютні та мішані посилання на адреси клітинок. Їхні види представлено в таблиці нижче на прикладі адреси клітинки **A1**.

Таблиця 8.1. Види посилань

Адреса	Тип посилання	Змінення адреси після її копіювання
A1	Відносне	Змінюються назва стовпця та номер рядка
\$A\$1	Абсолютне	Адреса не змінюється
A\$1	Мішане	Змінюється тільки назва стовпця
\$A1	Мішане	Змінюється тільки номер рядка

Зазначимо, що під час введення та редагування формул ви можете швидко змінювати тип посилання на адреси, натискаючи клавішу **F4**. Нижче показано, як це відбувається на прикладі адреси клітинки **A1**.

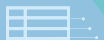
F4
A1 → **\$A\$1** → **A\$1** → **\$A1** → **A1**

ВПРАВА 8.1



Завдання. Створити таблицю за зразком та ввести формули у зафарбовані клітинки для виконання необхідних розрахунків.

	A	B	C	D	E	F	G
	Країна	Площа території, тис. км ²	у %	Чисельність, тис. ос.	у %	Густота, тис. ос. км ²	у %
1							
2	Ірландія	70,3		4609		65,3	
3	Іспанія	504,6		47500		95,15	
4	Італія	301,3		60588		201,2	
5	Всього						



РОЗДІЛ 2. Опрацювання табличних даних

1. Створіть таблицю та виконайте форматування відповідно до зразка.
Примітка. Для розташування тексту в клітинці у два рядки скористайтеся інструментом **Переносити текст** на вкладці **Основне**. Для створення верхнього індексу виділіть число, в області **Шрифт** відкрийте діалогове вікно та встановіть опцію **Надрядковий**.
2. Зробіть активною клітинку **B5**.
3. Оберіть інструмент **Автосума** та натисніть **Enter**. В результаті суму площ країн буде знайдено.
4. До клітинки **C2** введіть формулу обчислення площі Ірландії. Для цього необхідно значення площі Ірландії розділити на загальну площу.
5. Виділіть клітинку **C2** та введіть формулу **=B2/B\$5**. (Знак **\$** встановлено перед номером рядка, оскільки **B5** є сталим для кожної країни)
6. Скопіюйте формулу вниз, використовуючи автозаповнення.
7. Самостійно заповніть формулами й інші порожні клітинки таблиці.
8. Встановіть у діапазонах клітинок **C2:C4**, **E2:E4**, **G2:G4** відсотковий формат.

ВПРАВА 8.2



Завдання. Створити таблицю відповідно до зразка та виконати в ній необхідні обчислення.

	A	B	C	D	E	F	G	H	I
1	№	Найменування	Ціна в грн	Кількість	Ціна в \$	Ціна в €	Сума в грн	Сума в \$	Сума в €
2	1	Пенал	100	1					
3	2	Портфель	400	1					
4	3	Ручка	7	5					
5	4	Олівець	6	12					
6	5	Обкладинка	43	20					
7	6	Щоденник	55	1					
8	7	Зошит	12	40					
9	8	Папка для зошитів	29	1					
10	9	Ластик	5	2					
11	10	Лінійка	15	2					
12									
13		Курси валют:	\$						
14			€						

1. Створіть таблицю за зразком, увівши необхідні дані та виконавши форматування.
2. До клітинок **D13** та **D14** уведіть поточні курси валют.
3. Заповніть формулами клітинки, виділені зеленим кольором. Для формул використовуйте посилання на клітинки **D13** та **D14**, що містять курси валют.



4. Скопіюйте формули із зелених клітинок вниз, застосувавши автозаповнення.
5. Установіть у клітинках **E2:I11** грошовий формат відповідно до зазначених валют.



Виконайте вправу 8.3.



Контрольні запитання та завдання

1. Яке призначення табличного процесора?
2. Що таке адреса клітинки?
3. Які дані може містити клітинка?
4. Які існують види посилань на адреси клітинок?
5. Як виконати перехід від одного виду посилання до іншого при введенні формули?



Питання для роздумів

- 1*. У яких випадках можна використовувати відносні посилання, а коли — абсолютні чи змішані? Наведіть приклади.
- 2*. Чи існують випадки, коли замість мішаного посилання можна скористатися абсолютним посиланням. Наприклад, **\$A\$1** замість **A\$1**. Наведіть приклади.



Завдання для досліджень

- 1*. Дослідіть, для чого використовують у MS Excel стиль посилань **R1C1**.

§9.

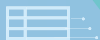
ІМЕНА КЛІТИНОК ТА ЇХНІХ ДІАПАЗОНІВ

9.1.

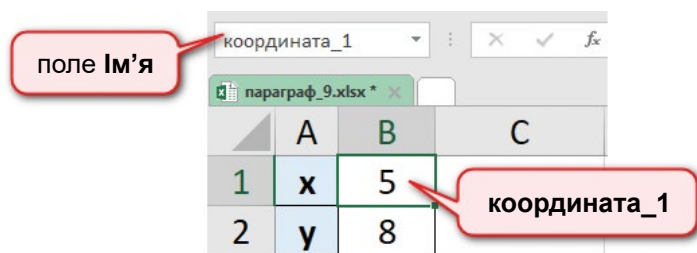
Створення імен клітинок та діапазонів

Якщо клітинка або діапазон клітинок часто використовуються, у формулах їх зручно називати. Розглянемо, що таке імена клітинок і як їх використовувати.

Ви знаєте, що кожна клітинка має свою адресу і за умовчанням її ім'я складається з назви стовпця та номера рядка на перетині яких вона знаходиться. Але клітинці можна надати й інше ім'я на власний розсуд користувача.



Так, наприклад, клітинці **B2** можна надати ім'я **координата_1**. Як правило, імена надають відповідно до тих даних, що містить клітинка. Також можна надавати імена і діапазонам клітинок.



Створення імені клітинки або діапазону

1 спосіб

1. Виділити клітинку.
2. Перейти до поля **Ім'я** (ліворуч від рядка формул).
3. Увести ім'я клітинки, наприклад, **координата_1**.
4. Натиснути **Enter**. В результаті ім'я буде створено.

2 спосіб

1. Виділити клітинку.
2. На вкладці **Формули** обрати інструмент **Визначити ім'я**.
3. Перейти до поля **Ім'я**.
4. Ввести ім'я клітинки, наприклад, **координата_1**.
5. Натиснути **ОК**. В результаті ім'я буде створено.

Аналогічно можна надати ім'я і діапазону клітинок.

Для переходу до клітинки, що має ім'я, необхідно у поля **Ім'я** обрати бажане ім'я клітинки. В результаті табличний курсор переміститься саме до тієї клітинки, що має це ім'я.

Для створення імен клітинок та діапазонів існують певні правила.

1. Ім'я може містити букви, цифри, символи: «?», «.», «_», «\».
2. Ім'я не може містити пробіли — їх, як правило, замінюють символом підкреслення.
3. Ім'я може починатися лише з букви, символів «_» або «\».
4. Не можна створювати ім'я, що розпізнається як адреса клітинки (**A5**, **BX3** тощо).
5. Ім'я не може містити більше, ніж 255 символів.
6. Не можна називати клітинки літерами **R** і **C**.

Слід також зауважити, що при використанні імен клітинок та діапазонів необхідно враховувати, що за умовчанням вони є абсолютними посиланнями.



Звичайно, створеними іменами клітинок можна керувати. А саме: переглядати, видаляти, змінювати саме ім'я, посилання на клітинки або діапазони тощо. Для цього необхідно обрати інструмент **Диспетчер імен** на вкладці **Формула**.

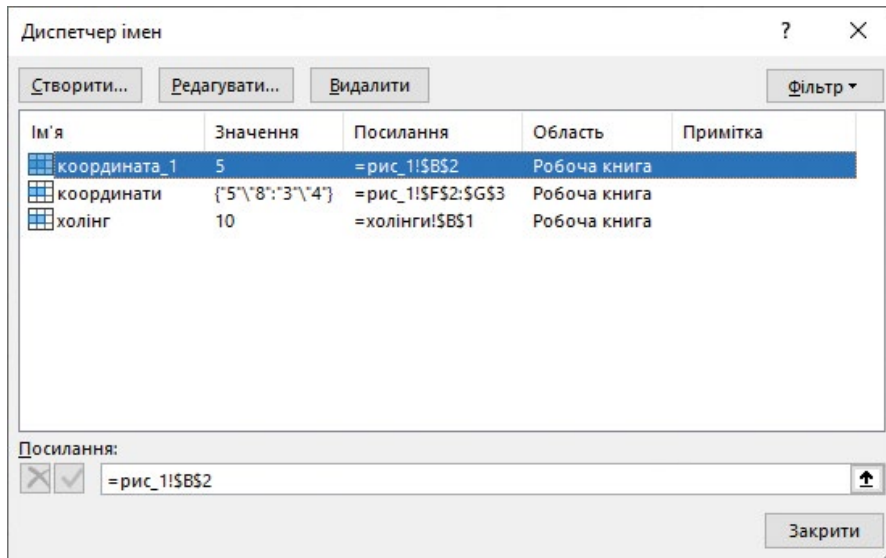


Рис. 9.1. Вікно, що містить створені імена

У даному вікні відображено перелік імен з їхніми значеннями, посиланнями, областю застосування та примітками. З використанням відповідних кнопок можна створити нове ім'я, а також редагувати та видаляти вже наявне.

9.2.

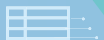
Використання імен. Практичне завдання 1



Розглянемо використання імен клітинок на практиці. Згадаємо завдання з попереднього параграфу, де необхідно було знайти ціну товару в холінгах. Тут доцільно надати ім'я клітинці **B1**, що містить курс холінгу, можна назвати клітинку, наприклад, **холінг**.

1. Для надання імені виділіть клітинку **B1**.
2. Оберіть поле **Ім'я** та введіть ім'я **холінг**.
3. Натисніть **Enter**.
4. У клітинці обчислення ціни товару в холінгах до клітинки **C4** введіть формулу: **=B4/B1**. (При введенні формули клацайте на відповідні клітинки, а не

	A	B	C
1	Курс гривні/холінги	10	холінг
2			
3	Найменування	Ціна, грн.	Ціна, хол.
4	Пенал	100	=B4/холінг
5	Рюкзак	850	=B5/холінг
6	Ручка	20	=B6/холінг



РОЗДІЛ 2. Опрацювання табличних даних



Дізнайся більше про використання імен діапазонів.

вводьте з клавіатури). В результаті у формулі замість адреси клітинки **B1** буде відображено її ім'я **холінг**.

5. Натисніть **Enter** та застосуйте до наступних клітинок автозаповнення. В результаті обчислення буде виконано правильно, адже ім'я клітинки має абсолютне посилання.

ВПРАВА 9.1



Завдання. Виконати в таблиці необхідні розрахунки, надавши імена клітинкам.

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Ціни на комунальні послуги									
3		Опалення, 1 м ²				35,65					
4		Водопостачання, 1 м ³				15,36					
5		Електропостачання, 1 квт				1,68					
6		Газопостачання, 1 м ³				6,90					
7		Квартплата, 1 м ²				6,50					
8											
9	№ квар- тири	Площа квартири, м ²	Показники лічильників			Вартість за місяць				Квартплата	Сума до сплати
10			Газо- постачання	Водо- постачання	Електро- постачання	Опалення	Водо- постачання	Електро- постачання	Газо- постачання		
11	1	61	10	3,5	155						
12	2	48	11	2,8	320						
13	3	33	23	0,9	85						
14	4	61	8	4,8	266						
15	5	61	9	5,2	333						
16	6	48	12	1,8	95						
17	7	33	22	2,8	222						
18	8	61	19	10,5	580						
19										Усього	

1. Створіть таблицю за зразком або завантажте файл **Вправа_9_1.xlsx**.
2. Надайте імена клітинкам **E3**, **E4**, ..., **E7** (**Опалення**, **Водопостачання**, тощо).
 - a. Виділіть клітинку **E3**.
 - b. Перейдіть до поля **Ім'я**.
 - c. Введіть ім'я **Опалення** та натисніть **Enter**.
 - d. Аналогічно надайте імена іншим клітинкам.
3. Обчисліть вартість опалення в кожній квартирі за місяць.
 - a. До клітинки **F11** уведіть формулу для розрахунку вартості опалення за місяць: (ціна за 1 м²) × (площа квартири). Для цього виділіть клітинку **F11** та введіть формулу **=Опалення*B12**.
 - b. Виконайте копіювання формули до клітинок знизу, використовуючи дію автозаповнення.
4. До інших клітинок формули введіть самостійно.



ВПРАВА 9.2



Завдання. Створити таблицю відповідно до зразка. Обчислити значення y за формулою:

$$y = \frac{P + 2x}{Q} - \frac{x^2}{(T + 1)},$$

де $Q = 3P$, $T = 2P^2$.

1. Створіть таблицю відповідно до зразка.
2. Надайте клітинкам **A2**, **B2** та **C2** імена **P**, **Q** і **T** відповідно.
3. Обчисліть значення змінних Q і T , увівши необхідні формули до клітинок **B2** і **C2**.
4. До клітинки **B5** введіть формулу для знаходження значення змінної y :
=(P+2*A5)/Q-A5^2/(T+1).
5. Застосуйте до попередньої формули дію автозаповнення, виконавши її копіювання до діапазону **B6:B14**.

	A	B	C
1	P	Q	T
2	0,1		
3			
4	x	y	
5	1		
6	3		
7	5		
8	7		
9	9		
10	11		
11	13		
12	15		
13	17		
14	19		



Контрольні запитання та завдання

1. Що таке ім'я діапазону клітинок?
2. Для чого клітинці надають ім'я?
3. Як створити ім'я діапазону клітинок?
4. Як змінити ім'я клітинки?
5. Де можна переглянути імена клітинок, які вже названі?



Питання для роздумів

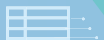
1. Як задати ім'я клітинці тільки для даного аркуша?
- 2*. Чи можна для створення нового імені діапазону використовувати клітинки, яким уже надано імена?
- 3*. Чи може одна й та сама клітинка мати декілька імен?



Завдання для досліджень

1. Визначте, як надати ім'я клітинці за допомогою її контекстного меню.
- 2*. Визначте, як виділити одночасно декілька діапазонів за допомогою їхніх імен.






ПРАКТИЧНА РОБОТА 2. «РОЗВ'ЯЗУВАННЯ ЗАДАЧ НА ОБЧИСЛЕННЯ»



Завдання 1. Створити таблицю за зразком та виконати обчислення.

	A	B	C	D	E	F	G	H	I	J	K
	Найменування морозива	Ціна	Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота	Неділя	Всього	Сума за тиждень
1											
2	Каштан	12,50₴	3	4	10	6	3	6	6		
3	Імперія	13,75₴	5	2	11	5	8	5	2		
4	Ласка	11,80₴	7	6	7	1	9	7	3		
5	Всього морозива										
6	Всього сума										

- Створіть таблицю відповідно до зразка.
Примітка. Щоб змінити напрямок тексту, слід обрати інструмент  **Орієнтація** в групі **Вирівнювання** на вкладці **Основне**.
- Уведіть необхідні формули до клітинок **J2**, **K2**, **C5** і **C6**. Скопіюйте їх до відповідних діапазонів, використовуючи автозаповнення.
- У клітинці **K6** уведіть формулу для обчислення загального прибутку.
- Надайте аркушу ім'я **Завдання_1** та збережіть файл.



Завдання 2. На другому аркуші створити таблицю множення чисел від 1 до 9.

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

- Додайте в попередній книзі новий аркуш та надайте йому ім'я **Завдання_2**.
- Створіть таблицю відповідно до зразка.
- Створіть копію цього аркуша. Надайте йому ім'я **Завдання_3**.
- Перейдіть до аркушу **Завдання_2**.
- Уведіть до клітинки **B2** формулу **=B\$1*\$A2**, що перемножує число в рядку на число у стовпці, які розташовані ліворуч та зверху над даною клітинкою.

Пояснення. У записі формули було враховано, що при копіюванні її праворуч має незмінним залишатися стовець **A**, а при копіюванні вниз сталим є перший рядок.



6. Виконайте копіювання формули з клітинки **B2** до діапазону **C2:J2**, застосувавши автозаповнення.
7. Виділіть діапазон **B2:J2** та скопіюйте його формули до діапазону **B3:J10**, скориставшись автозаповненням. У результаті ви маєте отримати таблицю множення.



Завдання 3*. Створіть таблицю множення, як у завданні 2, використовуючи імена діапазонів.

1. На аркуші **Завдання_3** створіть імена діапазонів множників, надавши імена діапазонам **B1:J1 A2:A10**.
2. До клітинки **B2** уведіть формулу для обчислення добутку відповідних множників.
3. Скопіюйте введену формулу до інших клітинок таблиці множення.



МАТЕМАТИЧНІ ФУНКЦІЇ



Поняття функції

Для виконання більш складних обчислень використовують вбудовані до електронного процесора функції. Пригадаємо, що *функція в Excel* — це заздалегідь створена формула, яка виконує обчислення за введеними величинами в зазначеному порядку.

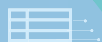
У цьому класі ви вже ознайомилися з деякими найбільш уживаними функціями, такими як **SUM**, **MAX**, **MIN**, **AVERAGE** та **COUNT**, що призначені для знаходження суми, максимального та мінімального значень, середнього арифметичного та знаходження кількості числових значень відповідно.

Нагадаємо, які основні елементи має містити функція:

- функція повинна мати власне ім'я, наприклад, **SUM**, **MAX** тощо;
- в дужках функція має містити аргументи, що відокремлені крапкою з комою.

Аргументом функції може бути число, текст, посилання на клітинку або діапазон клітинок, формула або інша функція.

Кожна функція має свій синтаксис і порядок дій, якого слід дотримуватися, оскільки в протилежному випадку результати обчислень будуть хибними. Використовуючи функції, також треба задавати для них аргументи відповідного типу. MS Excel містить математичні, статистичні, логічні та інші функції.



11.2.

Математичні функції

Функції, які найчастіше використовуються на практиці, наведено в таблиці 11.1. Оскільки значне коло користувачів використовує російський інтерфейс програми MS Excel, у другому стовпці таблиці подано назви функцій для даної версії табличного процесора.

Таблиця 11.1. Перелік основних математичних функцій

Функція	В російському інтерфейсі	Призначення
ABS (число)	ABS	Модуль числа
SQRT (число)	КОРЕНЬ	Квадратний корінь числа
QUOTIENT (число; дільник)	ЧАСТНОЕ	Ціла частина від ділення
MOD (число; дільник)	ОСТАТ	Остача від ділення
PRODUCT (число1; число2; ...)	ПРОИЗВЕД	Добуток чисел
SUMPRODUCT (діапазон1; діапазон2;...)	СУММПРОИЗВ	Визначення суми добутків відповідних діапазонів
SUMIF (діапазон; критерій; діапазон суми)	СУММЕСЛИ	Пошук суми у діапазоні, що задовольняє критерію
INT (число)	ЦЕЛОЕ	Округлення до меншого цілого
ROUND (число; k_розрядів)	ОКРУГЛ	Округлення
COS (число)	COS	Косинус кута
SIN (число)	SIN	Синус кута
TAN (число)	TAN	Тангенс кута
RADIANS (число)	РАДИАНЫ	Перетворення градусів на радіани
DEGREES (число)	ГРАДУСЫ	Перетворення радіанів на градуси

Зверніть увагу, що аргументом тригонометричних функцій має бути значення кута в радіанах.

Для одержання кута в радіанах використовується спеціальна функція **RADIANS**, яка дозволяє перевести значення градусів у радіани, для зворотного переведення також існує відповідна функція **DEGREES**.

11.3.

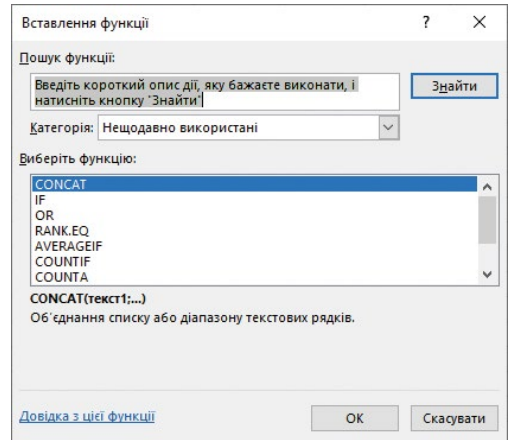
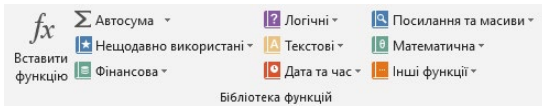
Використання майстра функцій

Функції можна вводити вручну або за допомогою майстра функцій. Викликати майстра функцій можна різними способами.

Найпростіший спосіб — це натиснення кнопки **f(x)**, яка розташовано ліворуч від рядка формул. В результаті її натиснення відкривається вікно **Вставлення функції**.

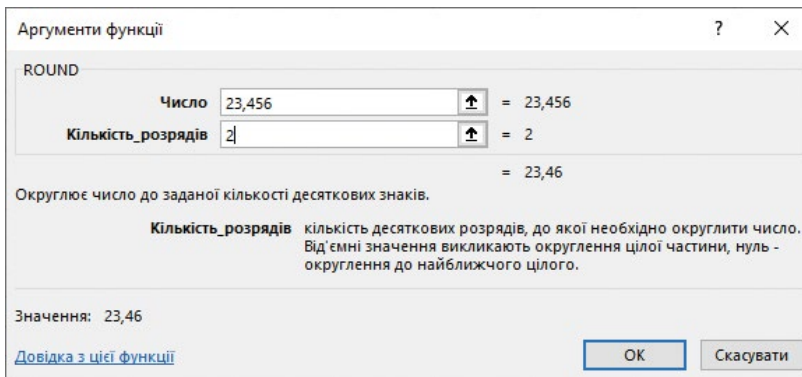


Ще один спосіб виклику — на вкладці **Формули** обрати інструмент **Вставити функцію**. Після виклику діалогового вікна користувач може обрати в полі **Категорія** одну із необхідних категорій функцій. Бажану категорію також можна обрати на вкладці **Формули**.



Розглянемо використання майстра функцій на прикладі округлення числа.

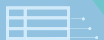
1. Перейдіть на вкладку **Формули**.
2. В категорії **Математична** оберіть функцію **ROUND**. В результаті має з'явитися діалогове вікно, в якому необхідно ввести аргументи даної функції.
3. До поля **Число** введіть 23,456. Дане число має також з'явитися праворуч від поля введення.
4. До поля **Кількість_розрядів** уведіть 2. В цьому ж вікні під значеннями аргументів має з'явитися результат виконання даної функції, а саме, 23,46.



Зверніть увагу, що дане вікно також містить підказки по самій функції та її аргументах.

5. Після натиснення **OK** одержуємо результат обчислення функції.

Вікно **Аргументи функції** має подібний вигляд для різних функцій, але з різною кількістю аргументів.



Щоб відредагувати формули, що містять вбудовані функції, у рядку формул потрібно клацнути на назві функції, а потім на значку **f(x)** (ліворуч від рядка формул), у результаті буде виведено вікно **Аргументи функції** для обраної функції. У цьому вікні ви можете виконати необхідні зміни та натиснути **ОК**.

11.4.

Уведення функцій з клавіатури

Якщо вводити в рядок формул перші літери функції, відкриється список функцій для автозавершення (рис. 11.1).

Для швидкого введення функції необхідно клацнути на ній два рази. В результаті буде вставлено бажану функцію та відкрито дужку для введення аргументів. Також з'явиться підказка щодо переліку аргументів, які має містити функція (рис. 11.2).

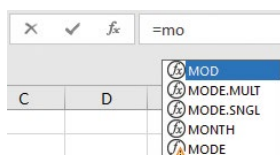


Рис. 11.1

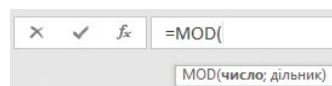


Рис. 11.2

Після введення аргументів потрібно закрити дужку. Якщо формулу завершено, то натискаємо клавішу **Enter**, якщо ні, то вводимо інші знаки для продовження формули. Нагадаємо, що для відокремлення аргументів використовують «;».

Зверніть увагу, що цей спосіб введення функцій доцільно використовувати тоді, коли ви добре знаєте назви функцій, які необхідно застосувати для обчислень.

Розглянемо використання математичних функцій на практиці.

ВПРАВА 11.1. (ФУНКЦІЇ SQRT ТА ABS)

Математичні функції в Excel часто використовуються для обчислення різноманітних виразів. Розглянемо саме таке завдання, в якому необхідно знайти значення виразу при заданих значеннях змінних a , b і c .



Завдання. У клітинках **A2**, **B2**, **C2** дано числові значення. Знайдіть значення інших змінних за вказаними нижче формулами.

$$X = \sqrt{\frac{a}{b}} + |b - c|,$$

$$P = \frac{a + b + c}{2},$$



$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

$$M = \frac{1}{2} \sqrt{2b^2 + 2c^2 - a^2},$$

$$Q = p \cdot S + |b - c|,$$

$$T = \sqrt{|a + b - 2|} - \frac{a - c}{\sqrt{b}}.$$

1. Створіть таблицю відповідно до зразка.
2. Уведіть необхідну формулу до клітинки **B4** для обчислення значення виразу:

$$X = \sqrt{\frac{a}{b}} + |b - c|.$$

В середовищі Excel дана формула буде мати вигляд: **=SQRT(A2/B2)+ABS(B2-C2)**. Де **SQRT** — функція квадратного кореня, **ABS** — функція модуля числа.

Розглянемо введення формули через рядок формул.

- a. Виділіть клітинку **B4**.
 - b. Уведіть знак «=».
 - c. Почніть вводити функцію (введіть символи **sq**).
 - d. Зі списку, що з'явився, оберіть функцію **SQRT**.
 - e. Уведіть аргумент функції **A2/B2**.
 - f. Закрийте дужку та введіть символ «+».
 - g. Почніть вводити функцію модуля (введіть символ **a**).
 - h. Оберіть функцію **ABS**.
 - i. Уведіть аргумент функції **B2-C2** та закрийте дужку.
 - j. Для застосування формули натисніть **Enter**. В результаті ви маєте одержати 1,960769.
3. Формули для обчислення значень у клітинках **B5:B9** уведіть самостійно.

	A	B	C
1	a	b	c
2	12	13	14
3			
4	X		
5	P		
6	S		
7	M		
8	Q		
9	T		

ВПРАВА 11.2. (ФУНКЦІЯ SUMIF)

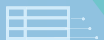
Далі розглянемо завдання, в яких необхідно знайти суму значень, що задовольняють деякому критерію.



Завдання. У таблиці відомі дані про кількість учнів у кожному навчальному закладі та тип навчального закладу (школа, ліцей, гімназія). Знайти загальну кількість учнів у ліцях.

1. Заповніть таблицю відповідно до зразка.
2. Перейдіть на вкладку **Формули**.

	A	B	C
1	Навчальні заклади		
2	№	Тип навчального закладу	Кількість учнів
3	1	Гімназія	400
4	2	Гімназія	360
5	3	Ліцей	240
6	4	Школа	350
7	5	Ліцей	404
8	6	Школа	315
9			
10	Усього в ліцях		



Дізнайтеся більше про використання функцій **MOD** та **QUOTIENT**, виконавши вправу 11.3.



Дізнайтеся більше про використання функції **SUMIF**, виконавши вправу 11.4.



Дізнайтеся більше про використання функції **SUMPRODUCT**, виконавши вправу 11.5.

3. В категорії **Математична** оберіть функцію **SUMIF**.
4. Як перший аргумент укажіть діапазон пошуку слова «ліцей», а саме, діапазон **B3:B8**.
5. Другим аргументом є сам критерій, тобто слово «ліцей». Отже, до поля **Критерій** уведіть слово «ліцей».
6. Як третій аргумент **Діапазон_для_суми** введіть діапазон **C3:C8**. Це діапазон, в якому здійснюється пошук суми.
7. Натисніть **ОК**.



Контрольні запитання та завдання

1. Що таке функція в MS Excel?
2. Які ви знаєте математичні функції?
3. Укажіть способи виклику майстра функції.
4. Укажіть послідовність дій для додавання функції за допомогою майстра функції.
5. Як увести формулу вручну?
6. Які одиниці вимірювання кутів використовуються для тригонометричних функцій?



Питання для роздумів

- 1*. Чи можна в одній формулі ввести одну функцію за допомогою майстра функції, а іншу — вручну?
- 2*. Навіщо в Excel використовують декілька способів виклику майстра функції?
- 3*. Результатом яких математичних функцій завжди буде ціле число?



Завдання для досліджень

- 1*. Дослідіть, скільки вкладень може мати одна функція.



§12.

СТАТИСТИЧНІ ФУНКЦІЇ

12.1.

Поняття функції

Для здійснення статистичного аналізу даних в MS Excel використовують статистичні функції. Знаходяться ці функції в категорії **Статистичні**. Більшість функцій у цій категорії використовує вузьке коло спеціалістів, проте частина з них є досить часто вживаними. Так, до статистичних функцій належать уже відомі вам **MAX**, **MIN**, **AVERAGE** та **COUNT**. В таблиці 12.1 наведено ще деякі функції, які використовують при обчисленнях, з категорії **Статистичні**.

Таблиця 12.1. Функції з категорії Статистичні

Функція	В російському інтерфейсі	Призначення
COUNTA (значення1; значення2;...)	СЧЁТЗ	Визначення кількості непорожніх клітинок
COUNTIF (діапазон; критерій)	СЧЁТЕСЛИ	Визначення кількості клітинок, що відповідають заданому критерію
AVERAGEIF (діапазон; критерій; діапазон_середн)	СРЗНАЧЕСЛИ	Пошук середнього у діапазоні, що відповідає заданому критерію
RANK.AVG (число; посилання; порядок)	РАНГ.СР	Визначення рангу числа в переліку чисел
RANK.EQ (число; посилання; порядок)	РАНГ.РВ	Визначення рангу числа в переліку чисел

12.2.

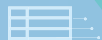
Функція COUNTA

Вам уже знайома функція **COUNT** — вона визначає кількість числових значень у діапазоні. На відміну від неї, функція **COUNTA** визначає кількість не тільки числових, але й будь-яких інших значень, виключаючи порожні клітинки. Розглянемо використання даної функції, виконавши вправу.

ВПРАВА 12.1



Завдання. Відомі дані про кількість учнів у кожному класі (якщо у клітинці числа розділені рискою, це означає, що клас розділено на групи



в заданій кількості учнів). Буква класу визначає його профіль (математичний, фізичний, хімічний та економічний). У діапазоні **F3:F7** визначити кількість класів кожної паралелі, а у діапазоні **B8:E8** кількість класів кожного профілю.

	A	B	C	D	E	F
1	Паралель	Буква класу				Усього класів
2		М	Ф	Х	Е	
3	7	15/14	16/16		24	
4	8		25	20		
5	9	22	15/15	14/15	17/16	
6	10	13/15	15/16	15/14		
7	11	16/16	15/15		15/14	
8	Усього класів					

1. Завантажте файл **Вправа_12_1.xlsx** з матеріалів до даного параграфа.
2. Визначте кількість класів у кожній паралелі.
 - a. Виділіть клітинку **F3**.
 - b. На вкладці **Формули** в меню інструменту та **Інші функції** оберіть категорію **Статистична**.
 - c. Оберіть функцію **COUNTA**.
 - d. В полі аргументу **значення 1** виділіть необхідний діапазон **B3:E3**.
 - e. Натисніть **ОК**. В результаті буде підраховано кількість 7-х класів.
 - f. Скопіюйте одержану формулу нижче, використовуючи автозаповнення.
3. Обчисліть кількість класів у профілі.
 - a. Оберіть клітинку **B8**.
 - b. У рядку формул уведіть **дорівнює** та почніть введення функції. Оберіть функцію **COUNTA** у списку.
 - c. Після дужки вкажіть діапазон **B3:B7**.
 - d. Закрийте дужку та натисніть **Enter**. В результаті кількість математичних класів буде підраховано.
 - e. Знайдіть кількість класів в інших профілях, застосувавши до формул спосіб автозаповнення.

12.3.

Функція COUNTIF

Ще одна досить поширена у використанні функція **COUNTIF** або **СЧЁТЕСЛИ**. Вона визначає кількість клітинок, що відповідають заданому критерію. Розглянемо практичну вправу з її використанням.



ВПРАВА 12.2



Завдання. Дано таблицю, що містить інформацію про результати участі учнів в олімпіаді з математики. Визначити кількість десятикласників, що брали участь в олімпіаді та кількість переможців олімпіади. (Переможцями є учні, що одержали більше 16 балів.)

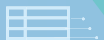
	A	B	C	D	E
1	Прізвище	Клас	Кількість балів		Нижня межа переможців
2	Іванов	10	16		16
3	Шевцов	9	25		
4	Бондаренко	11	30		
5	Ігнатов	10	18		
6	Крохмаленко	11	24		
7	Васильєв	10	13		
8					
9	Кількість десятикласників				
10	Кількість переможців				

- Завантажте файл **Вправа_12_2.xlsx** з матеріалів до даного параграфу.
- Визначте кількість десятикласників.
 - Виділіть клітинку **B9**.
 - Перейдіть до категорії **Статистична**.
 - Оберіть функцію **COUNTIF**.
 - Як перший аргумент установіть діапазон класів, а саме, **B2:B7**.
 - Другим аргументом є критерій. У даному випадку це буде число 10. Уведіть число 10.
 - Натисніть **ОК**. В результаті кількість десятикласників буде знайдено.
- У клітинці **B10** знайдіть кількість переможців.
 - У рядку формул уведіть **дорівнює** та почніть вводити назву функції. Оберіть зі списку функцію **COUNTIF**.
 - Як перший аргумент укажіть діапазон, що містить бали, а саме **C2:C7**. Установіть крапку з комою.
 - Як другий аргумент має бути умова, що значення є більше 16. Запишіть цю умову ("**>16**").

Примітка. Зверніть увагу, що умова, в якій використовується знак порівняння, має бути записана в лапках.

- Закрийте дужку та натисніть **Enter**. В результаті кількість переможців буде знайдено.

Розглянемо випадок, коли нижня межа переможця вказана у певній клітинці (наприклад, **E2**) і може за певних обставин змінюватися. В цьому випадку критерій підрахунку має містити посилання на дану клітинку.



4. Змініть умову визначення кількості переможців наступним чином: **">&E2**. Натисніть **Enter** та переконайтеся, що кількість переможців для введеної до клітинки **E2** умови знайдено.
5. Змініть значення умови у клітинці **E2**. Уведіть, наприклад, 20. Переконайтеся, що кількість переможців буде змінено.

12.4.

Функція AVERAGEIF

Схожою до функції **COUNT** за своїм призначенням є функція **AVERAGEIF** (**СРЗНАЧЕСЛИ**). З її допомогою визначають середнє арифметичне значення у діапазоні, що задовольняє вказану умову. Розглянемо практичне застосування цієї функції, виконавши вправу.

ВПРАВА 12.3



Завдання. В таблиці подано оцінки учнів. Визначити їхній середній бал без урахування оцінок з фізкультури.

	A	B	C	D	E	F	G	H	I
1	№	Прізвище	Математика	Фізика	Хімія	Фізкультура	Інформатика	Історія	Середній бал
2	1	Іванов	11	8	7	11	11	8	
3	2	Шевцов	10	10	11	10	10	11	
4	3	Бондаренко	8	10	7	9	9	9	
5	4	Ігнатов	9	9	6	10	8	11	
6	5	Крохмаленко	9	9	10	9	10	10	
7	6	Новіков	9	8	8	10	9	9	
8	7	Козлов	7	7	7	9	11	9	
9	8	Карпенко	7	9	7	11	9	9	

1. Завантажте файл **Вправа_12_3.xlsx** з матеріалів до даного параграфа.
2. Виділіть клітинку **I2**.
3. В категорії **Статистична** оберіть функцію **AVERAGEIF**.
4. Як перший аргумент укажіть діапазон пошуку критерію **\$C\$1:\$H\$1**. Оскільки цей діапазон буде сталим для кожної наступної формули, фіксуємо його.
5. Як другий аргумент уведіть умову «не дорівнює фізкультура» ("**<>фізкультура**").
6. Третім аргументом уведіть діапазон оцінок учня **C2:H2**.
7. Натисніть **OK**.
8. Скопіюйте формулу нижче, використовуючи автозаповнення. В результаті ви маєте одержати середній бал учнів без урахування оцінок з фізкультури.

12.5.

Функції RANK.AVG та RANK.EQ.

Наступні дві функції **RANK.AVG** та **RANK.EQ** схожі за своїм призначенням. Вони визначають ранг числа в переліку чисел.

Ранг числа — це його порядок відносно інших значень у списку.

Якщо сортувати список, то ранг буде його позицією. Цю функцію часто використовують для визначення рейтингу навчання учнів, виступу спортсменів на змаганнях тощо. Так, учень з найкращим середнім балом буде першим за рейтингом і матиме ранг, що дорівнює одиниці. Відповідно, спортсмен, що має найменший час на дистанції з бігу, також буде першим, і ранг його дорівнюватиме одиниці (рис. 12.1).

	A	B	C	D
1	№	Прізвище	Середній бал	Рейтинг
2	1	Іванов	6,8	9
3	2	Шевцов	7,9	5
4	3	Бондаренко	3,9	10
5	4	Ігнатов	10,3	1
6	5	Крохмаленко	8,4	4
7	6	Новіков	7,2	7
8	7	Козлов	9,6	3
9	8	Карпенко	10,3	1
10	9	Слісарчук	7,3	6
11	10	Кулеш	7	8

	A	B	C	D
1	№	Прізвище	Час	Місце
2	1	Іванов	12,7	9
3	2	Шевцов	11,5	3
4	3	Бондаренко	12,5	7
5	4	Ігнатов	12	5
6	5	Крохмаленко	11,2	2
7	6	Новіков	11,1	1
8	7	Козлов	12,6	8
9	8	Карпенко	12,3	6
10	9	Слісарчук	11,8	4
11	10	Кулеш	12,8	10

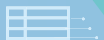
Рис. 12.1. Визначення рейтингу успішності учнів та результатів змагань спортсменів.

Аргументами даних функцій є:

- **число**, для якого визначається ранг;
- **посилання**, що містить діапазон чисел, серед яких визначається ранг;
- **порядок**, який визначає спосіб упорядкування даних.

Якщо порядок дорівнює нулю або його не вказано, то число, що є найбільшим значенням діапазону посилання, отримає значення рангу 1. Якщо порядок дорівнює 1, то значення рангу 1 отримає число, що є найменшим значенням діапазону посилання.

Чим же відрізняються функції **RANK.AVG** та **RANK.EQ**? Відмінність цих функцій у тому, що якщо в діапазоні посилань будуть однакові значення, то перша функція визначить середній ранг, а друга функція визначить вищий ранг. На рис. 12.2 видно, що в першому



РОЗДІЛ 2. Опрацювання табличних даних

випадку це 1,5 (середнє значення між 1 та 2). При застосуванні другої формули для обох випадків цей ранг дорівнює 1. В обох випадках наступне за величиною значення одержує ранг 3.

RANK.AVG(число;посилання;порядок)

	A	B	C	D
1	№	Прізвище	Середній бал	Рейтинг
2	1	Іванов	6,8	9
3	2	Шевцов	7,9	5
4	3	Бондаренко	3,9	10
5	4	Ігнатов	10,3	1,5
6	5	Крохмаленко	8,4	4
7	6	Новіков	9,6	3
8	7	Козлов	7,2	7
9	8	Карпенко	10,3	1,5
10	9	Слісарчук	7,3	6
11	10	Кулеш	7	8

RANK.EQ(число;посилання;порядок)

	A	B	C	D
1	№	Прізвище	Середній бал	Рейтинг
2	1	Іванов	6,8	9
3	2	Шевцов	7,9	5
4	3	Бондаренко	3,9	10
5	4	Ігнатов	10,3	1
6	5	Крохмаленко	8,4	4
7	6	Новіков	9,6	3
8	7	Козлов	7,2	7
9	8	Карпенко	10,3	1
10	9	Слісарчук	7,3	6
11	10	Кулеш	7	8

Рис. 12.2. Відмінність використання функцій **RANK.AVG** та **RANK.EQ**

Розглянемо, як знаходити ранг, виконавши практичну вправу.

ВПРАВА 12.4



Завдання. Відомі результати спортсменів у стрибках у довжину. Визначити рейтинг спортсменів.

	A	B	C	D
1	№	Прізвище	Довжина стрибка, м	Рейтинг
2	1	Іванов	2,35	
3	2	Шевцов	2,22	
4	3	Бондаренко	2,26	
5	4	Ігнатов	2,17	
6	5	Крохмаленко	1,95	
7	6	Новіков	1,89	
8	7	Козлов	1,93	
9	8	Карпенко	2,41	
10	9	Слісарчук	2,38	
11	10	Кулеш	2,15	

1. Завантажте файл **Вправа_12_4.xlsx** з матеріалів до даного параграфа.
2. Виділіть клітинку **D2**.
3. В категорії **Статистична** оберіть функцію **RANK.EQ**.
4. Як перший аргумент укажіть значення довжини стрибка першого спортсмена, а саме, клітинку **C2**.
5. Другим аргументом є посилання на значення довжин стрибків усіх спортсменів, отже, вкажіть діапазон **\$C\$2:\$C\$11**. Діапазон зафіксовано, оскільки він є сталим для кожної наступної формули.
6. Третій аргумент можна не вказувати, оскільки першим має бути результат у спортсмена, що має найбільше значення. Натисніть **ОК**.
7. Скопіюйте формулу нижче, застосувавши автозаповнення. В результаті ви маєте одержати рейтинг спортсменів.

**Контрольні запитання та завдання**

1. Які ви знаєте статистичні функції?
2. Що є спільного та відмінного у функцій **COUNT** та **COUNTA**?
3. З допомогою якої функції можна визначити рейтинг успішності учнів?
4. Чим відрізняються функції **RANK.AVG** та **RANK.EQ**?

**Питання для роздумів**

- 1*. Дано прайс-лист, що містить вартості комп'ютерних деталей. З допомогою якої функції можна визначити кількість деталей, вартість яких більша за 100 грн?
- 2*. З допомогою яких статистичних функцій (крім **MAX**) можна знайти максимальне значення числа з наданого переліку чисел.

**Завдання для досліджень**

- 1*. Дослідіть призначення статистичних функцій **COUNTIFS** та **AVERAGEIFS**. Наведіть можливі приклади їхнього використання.



§13. ЛОГІЧНІ ФУНКЦІЇ



13.1. Використання логічних функцій

Ще один тип розповсюджених функцій — логічні функції. У MS Excel можна вирішувати завдання, виконання яких залежить від певної умови. Частина таких завдань можна реалізувати, використовуючи функції з категорії **Логічні**.

Однією з основних функцій даної категорії є функція **IF** (рос. **ЕСЛИ**), яка має такий синтаксис:

=IF(Лог_вираз;Значення_якщо_істина;Значення_якщо_хибність).

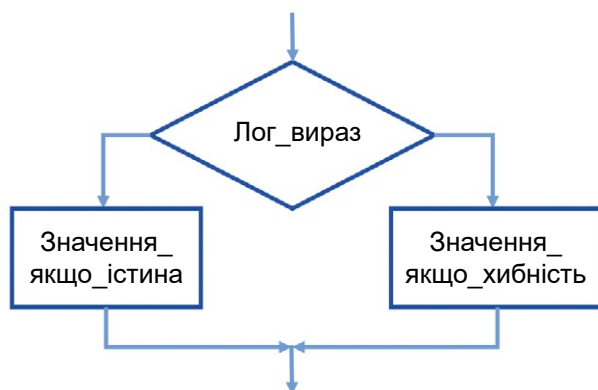
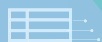


Рис. 13.1. Блок-схема виконання логічної функції IF

Наведена блок-схема демонструє дію логічної функції **IF**. Першим аргументом даної функції є логічний вираз, який містить умову. Якщо умова виконується, буде знайдено значення виразу, вказане у функції як другий аргумент, у разі невиконання умови буде знайдено значення виразу, що є третім аргументом функції.

Розглянемо використання даної функції на практиці.

ВПРАВА 13.1



Завдання. Дано чисельник та знаменник дробу. У клітинці **C2** вивести повідомлення, яким є цей дріб: правильним чи неправильним. (Правильним є дріб, у якого чисельник менший за знаменника.)

	A	B	C
1	1		тип дробу
2	7		правильний

Рис. 13.2

1. Уведіть дані до клітинок **A1**, **A2** та **C1** та виконайте форматування відповідно до зразка (рис. 13.2).
2. Уведіть формулу для визначення типу дробу.
 - a. Виділіть клітинку **C2**.
 - b. На вкладці **Формула**, перейдіть до категорії **Логічні** та оберіть функцію **IF**.
 - c. Як перший аргумент укажіть умову **A1<A2**.
 - d. У полі другого аргументу введіть напис, що має виводитися у разі виконання умови, а саме, **"правильний"**.
 - e. Третій аргумент — це напис, що виводиться, якщо умова не виконується. Тут уведіть напис **"неправильний"**.
 - f. Натисніть **ОК**. В результаті ви маєте одержати повідомлення, що дріб є правильний. Замініть значення чисельника на 8, тепер має з'явитися повідомлення, що дріб є неправильним.

13.2.

Використання складених логічних виразів

Як і в програмуванні, в табличному процесорі можна використовувати складені логічні вирази, для яких використовують логічні зв'язки **AND** (рос. **И**), **OR** (рос. **ИЛИ**) та **NOT** (рос. **НЕ**). У дужках даних функцій логічні вирази записують через крапку з комою.

Для порівняння числових значень використовують знаки порівнянь:

- =, <> — дорівнює, не дорівнює;
- >, < — більше, менше;
- >=, <= — більше або дорівнює, менше або дорівнює.

ВПРАВА 13.2



Завдання. Відомо час, за який учні подолали дистанцію у 100 м. Визначити, яке місце посів кожний учень, та надати відповідно до цього рекомендації: якщо 1-е, 2-е або 3-є місце — «Призер», інакше — «Тренуйся далі».

1. Створіть таблицю відповідно до зразка (рис. 13.3).
2. У діапазоні **C3:C7** визначте рейтинг учнів, скориставшись **RANK.EQ**.
3. У клітинці **D3** введіть формулу для надання рекомендації.
 - a. Зробіть активною клітинку **D3** та викличте майстра функцій.
 - b. Оберіть функцію **IF** з категорії **Логічні**.
 - c. Оскільки в якості логічного виразу ми маємо встановити умову, що значення клітинки **C3** має приймати одне із значень 1, 2 або 3, слід скористатися функцією **OR**. Для цього у діалоговому вікні **Аргументи функції**, перебуваючи у полі **Лог_вираз**, відкрийте список функцій, клацнувши на стрілку в полі, що розташоване ліворуч від рядка формул (рис. 13.4) та оберіть функцію **OR**. (Якщо дана функція у списку не відображається, скористайтеся командою **Інші функції...**)
 - d. Як перший аргумент функції **OR** установіть логічний вираз **C3=1**, як другий аргумент — **C3=2**, як третій — **C3=3**.

	A	B	C	D
1	Таблиця змагань учнів на дистанцію 100 м			
2	Прізвище	Час	Місце	Рекомендації
3	Іваненко	13,5		
4	Петренко	12,4		
5	Сидоренко	13,8		
6	Гаврилюк	12,9		
7	Гук	13,1		

Рис. 13.3

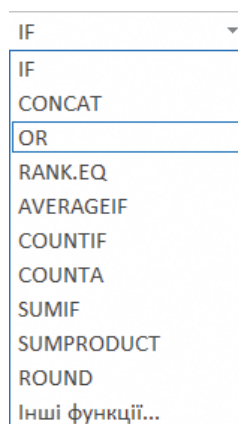


Рис. 13.4



- e. У рядку формул після введеної умови **OR(C3=1;C3=2;C3=3)** поставте крапку з комою. В результаті ви маєте повернутися до вікна функції **IF**.
- f. У полі **Значення_якщо_істина** введіть "Призер".
У полі **Значення_якщо_хибність** — "Тренуйся далі".
- g. Натисніть **ОК**. В результаті має бути сформовано повідомлення для першого спортсмена.
- h. Скопіюйте створену формулу, використовуючи автозаповнення.

13.3.

Вкладені розгалуження

Щоб перевірити більш ніж одну умову, використовують вкладення функції **IF**. Так, наприклад, відомо результати ігор між двома командами «Квант» і «Алмаз». У діапазоні **D2:D6** необхідно визначити для кожної гри, хто переміг, або вивести повідомлення про нічию (рис. 13.5).

Блок-схема розв'язання цього завдання буде мати такий вигляд (рис. 13.6).

	A	B	C	D
1	№ гри	Квант	Алмаз	Хто переміг?
2	1	3	4	
3	2	0	0	
4	3	2	1	
5	4	5	1	
6	5	1	3	

Рис. 13.5.

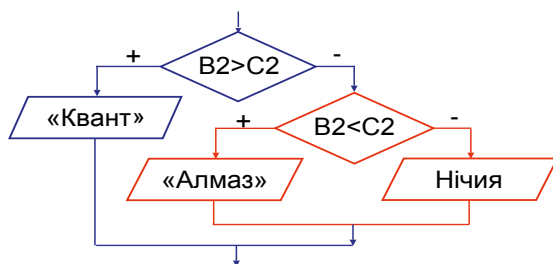


Рис. 13.6.

В MS Excel для реалізації такого алгоритму використовують вкладення функції. Для даного випадку формула має бути такою:

=IF(B2>C2;"Квант";IF(B2<C2;"Алмаз";Нічия))

Червоним кольором тут позначено вкладену функцію **IF**.

ВПРАВА 13.3



Завдання. Вивести інформацію про результати змагань між командами «Квант» та «Алмаз», керуючись даними, поданими вище.

1. Створити таблицю відповідно до зразка (рис. 13.5).
2. До клітинки **D2** ввести формулу, що виводить результат гри між командами.



- a. У клітинці **D2** викличте майстра функції.
- b. Оберіть функцію **IF** з категорії **Логічні**.
- c. Як перший аргумент даної функції введіть умову **B2>C2**.
- d. У полі **Значення_якщо_істина** введіть назву команди «Квант».
- e. Перейдіть до поля **Значення_якщо_хибність**.
- f. У списку функцій оберіть функцію **IF**. В результаті має з'явитися вікно **Аргументи функції** для формування вкладеної функції **IF**.
- g. Як перший аргумент уведіть умову **B2<C2**.
- h. У полі **Значення_якщо_істина** введіть назву команди «Алмаз».
- i. В полі **Значення_якщо_хибність** уведіть повідомлення — Нічия.
- j. Натисніть **ОК**. В результаті має бути сформоване повідомлення для першої гри.
- k. Для одержання повідомлення для усіх ігор застосуйте до формули автозаповнення.



Дізнайтеся про використання функцій із категорій **Текстові** та **Дата та час**.

ВПРАВА 13.4



Завдання. Дано номер автобусного квитка, визначити, чи є цей номер «щасливим». Щасливим вважається той квиток, у якого сума перших трьох цифр номера дорівнює сумі останніх трьох цифр номера.



Контрольні запитання та завдання

1. Для чого використовують функції з категорії **Логічні**?
2. Опишіть синтаксис функції **IF**.
3. Для чого використовують функції **AND**, **OR**, **NOT**?
4. Опишіть синтаксис функції **AND**.



Питання для роздумів

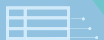
- 1*. У яких випадках використовують вкладені логічні функції?
- 2*. Чи можна використовувати вкладення функції **AND** в **OR** і навпаки?



Завдання для досліджень

- 1*. Скільки логічних виразів може містити функція **OR**?
- 2*. Скільки вкладень може містити функція **IF**?
- 3*. Порівняйте синтаксис функції **IF** у MS Excel з логічним оператором у мові Python.
- 4**. Яке призначення і синтаксис функції **IFERROR (ЕСЛИОШИБКА)**?





§14.

УМОВНЕ ФОРМАТУВАННЯ

При створенні та форматуванні документів з великою кількістю даних зручним є застосування такої можливості Excel, як *умовне форматування*.

Умовне форматування — це створення певного формату клітинки у разі виконання вказаної умови.

Це може бути зміна шрифту, його кольору, заливка клітинки тощо. Створюючи умовне форматування в Excel, широко застосовують візуальні засоби виділення даних, такі як кольорові шкали, гістограми, набори значків тощо.



14.1.

Використання кольорової шкали

	A	B
1	Дата	Температура
2	01.01	5
3	02.01	6
4	03.01	-4
5	04.01	3
6	05.01	0
7	06.01	7
8	07.01	-5

Рис. 14.1

Кольорова шкала — це графічне виділення, що дозволяє зрозуміти розподіл даних. За допомогою кольорової шкали, використовуючи градації двох або трьох кольорів, можна порівнювати діапазони клітинок. Водночас рівень яскравості кольору відповідає найбільшим, середнім або найменшим значенням. Так, на рис. 14.1 найбільше значення позначається синім кольором, найменше — червоним.

Для одного й того ж числового ряду можна застосовувати різні вбудовані кольорові шкали, а також можна створити і власну шкалу.

ВПРАВА 14.1

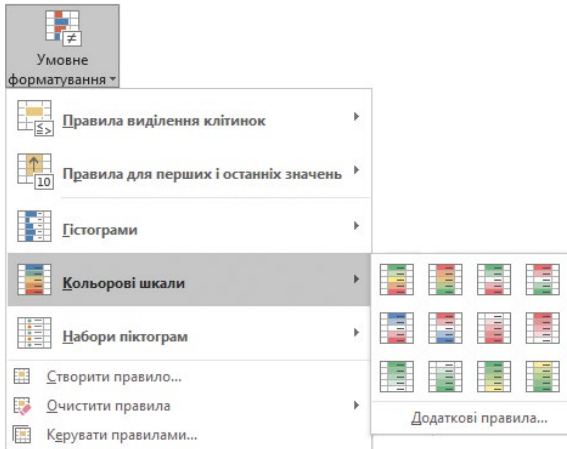


Завдання. У таблиці з числовими даними застосувати умовне форматування з використанням колірної шкали до діапазону **A1:AL33** для одержання деякого зображення.

1. Відкрийте файл **Вправа_14_1.xlsx** з матеріалів до даного параграфа.
2. Виділіть діапазон **A1:AL33**.
3. У групі **Стили** на вкладці **Основне** оберіть **Умовне форматування** ▶ **Кольорові шкали**.
4. Оберіть першу кольорову схему.
5. Поекспериментуйте з вибором інших схем кольорових шкал.



6. Створіть копію даного аркуша та застосуйте до діапазону **A1:AL33** команду **Набори піктограм** з меню інструменту **Умовне форматування**.



14.2.

Використання правил виділених клітинок

Форматування за допомогою правил виділених клітинок застосовують до клітинок з текстом, числовими значеннями та значеннями дати або часу. Дане форматування використовують для спрощення пошуку деяких клітинок у діапазоні з застосуванням оператора порівняння.

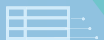
Розглянемо використання даного типу умовного форматування, виконавши практичну вправу.

ВПРАВА 14.2



Завдання. В електронному журналі червоним кольором виділити клітинки з оцінками високого рівня, зеленим кольором — достатнього, жовтим — низького.

1. Введіть дані до таблиці, як на рис. 14.2 (без форматування) або відкрийте файл **Вправа_14_2.xlsx** з матеріалів до даного параграфу.
2. Застосуйте умовне форматування для оцінок високого рівня.
 - a. Виділіть діапазон **B2:G11**, що містить оцінки.
 - b. На вкладці **Основне** в меню інструмента **Умовне форматування** оберіть команду **Правила виділених клітинок** ▶ **Між...**
 - c. У вікні **Між** установіть початкове значення 10, кінцеве 12.
 - d. У наступному полі зі списком оберіть форматування **Блідо-червона заливка з темно-червоним текстом**.



- е. Натисніть **ОК**. В результаті оцінки, що мають високий рівень, повинні бути підсвічені червоним. Звичайно, при зміні оцінки на оцінку нижчого рівня підсвічування буде прибрано.

	A	B	C	D	E	F	G
1	Прізвище та ім'я	01.02	02.02	03.02	04.02	05.02	06.02
2	Андрієць Олександра	3	4		9		11
3	Бондар Станіслав	9	6	8		11	
4	Бондаренко Сергій	8	7	4	9	10	
5	Буянова Ліна		9	9		5	
6	Городний Микола	7		10	8		10
7	Здоровець Олексій	10	4		1	8	
8	Зновяк Владислава	11					7
9	Іванов Володимир	5	6	5	7	9	
10	Іванов Сергій	6			5		8
11	Іванченко Артем	7	2				6

Рис. 14.2

3. Аналогічно встановіть умовне форматування для оцінок достатнього та низького рівнів.

14.3.

Використання гістограм

	A	B
1	Місяць	Кількість опадів у мм.
2	січень	37,2
3	лютий	11,4
4	березень	10
5	квітень	19,5
6	травень	11,7
7	червень	129,3
8	липень	57,1
9	серпень	13
10	вересень	8,7
11	жовтень	86
12	листопад	12,5
13	грудень	21,2

Рис. 14.3

Ще один вид умовного форматування — *гістограми*. Цей вид форматування використовують для порівняння значення клітинки відносно значень інших клітинок. При цьому довжина стовпчика гістограми відповідає значенню клітинки (чим більше значення — тим він довший).

Використовувати такий спосіб умовного форматування доречно до даних, наприклад, такої таблиці, як наведено на рис. 14.3.

Щоб застосувати такий вид умовного форматування для виділеного діапазону, треба обрати команду **Гістограми** з меню інструмента **Умове форматування**. Після чого вибрати бажаний вид гістограми із запропонованих схем.

14.4.

Створення власного правила з використанням формул

Бувають випадки, коли заготовок, що містить інструмент **Умове форматування**, недостатньо для розв'язання поставленої задачі. У вправі 14.3 виконаємо завдання, в якому треба виконати



форматування залежно від значень, що розташовані не в самій клітинці, де здійснюється форматування, а в деякій іншій клітинці.

ВПРАВА 14.3



Завдання. Дано таблицю, що містить середній бал учнів і значення прохідного бала для вступу до деякого ВНЗ. Виділити рожевим кольором середні бали тих учнів, що мають можливість вступити до даного ВНЗ.

1. Створіть таблицю, як на рис. 14.4 (без форматування), або завантажте файл **Вправа_14_3.xlsx** з матеріалів до даного параграфа.
2. Виділіть діапазон клітинок **C2:C9**.
3. В меню інструмента **Умове форматування** оберіть команду **Створити правило...** В результаті має відкритися вікно **Нове правило форматування**.
4. В даному вікні оберіть команду **Використовувати формулу для визначення клітинок для форматування**.

	A	B	C
1	№	Прізвище, ім'я	Середній бал
2	1	Андрієць Олександра	7,5
3	2	Бондар Станіслав	9,8
4	3	Буянова Олеся	10,2
5	4	Городний Микола	6,3
6	5	Здоровець Олексій	11,1
7	6	Іванов Володимир	8,9
8	7	Іванов Сергій	5,9
9	8	Іванченко Артем	9,5
10			
11		Прохідний бал	9,5

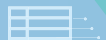
Рис. 14.4

У полі даного вікна необхідно ввести логічний вираз, у разі істинності якого буде виконуватися умове форматування виділених клітинок. Отже, це має бути формула **C2>=\$C\$11**. Тут **C2** є відносним посиланням, оскільки значення середнього бала для кожного учня в діапазоні змінюється, посилання на клітинку **C11** є абсолютним, оскільки для усіх учнів прохідний бал є сталим.

5. Уведіть формулу **C2>=\$C\$11**.
6. Клацніть кнопку **Форматувати...** та встановіть рожевий колір зафарбування клітинки.
7. Натисніть **ОК**. В результаті значення у стовпці **C**, що є більше прохідного балу будуть зафарбовані рожевим кольором.

Примітка. Якщо потрібно змінити створене правило, виділіть клітинки, до яких застосовано умове форматування, та виконайте команду **Умове форматування** ▶ **Керувати правилами...**, оберіть правило та натисніть кнопку **Редагувати правило...**

8. У клітинці **C11** змініть значення прохідного бала та перевірте, як буде змінюватися підсвічення балів учнів.
- 9*. Змініть створене правило таким чином, щоб підсвічувалися не лише результати учнів, а весь рядок з інформацією про учня, як на рис. 14.4.



Контрольні запитання та завдання

1. Для чого використовують умовне форматування?
2. Які способи умовного форматування ви знаєте?
3. Як виконати умовне форматування виділеного діапазону?
4. Який спосіб умовного форматування потрібно застосувати для виділення клітинок, значення яких належать числовому діапазону?



Питання для роздумів

- 1*. Наведіть приклади завдань, для яких доцільно використовувати умовне форматування за допомогою гістограм.
- 2*. Чи можна при створенні власного правила умовного форматування скористатися майстром формул?



Завдання для досліджень

- 1*. Укажіть послідовність дій установлення умовного форматування для клітинок, які містять дати з тими подіями, що відбулися раніше за поточну дату.
- 2*. Дослідіть, як можна встановити умовне форматування для даних, які є більшими (меншими) за середнє арифметичне даних деякого діапазону.



ПРАКТИЧНА РОБОТА 3. «ВИКОРИСТАННЯ ВБУДОВАНИХ ФУНКЦІЙ. УМОВНЕ ФОРМАТУВАННЯ»

ЗАВДАННЯ 1



У заданому шаблоні робочого аркуша **Тестування** реалізувати перевірку відповідей на запитання тестів, оцінити відповіді за 12-бальною шкалою оцінювання. У бланку відповідей правильні відповіді підсвітити зеленим кольором, неправильні — рожевим.

1. Завантажте файл **ПР_3.xls** з матеріалів до даного параграфа.
2. На аркуші **Тестування** виконайте форматування таблиці відповідно до зразка (рис. 15.1).



	A	B	C	D	E	F	G	H	I
1		Уведіть відповіді до бланку відповідей		Бланк відповідей					
2				еталон	г	б	б	в	
3	1	Столиця України...		№ запитання	1	2	3	4	
4	а	Харків		Відповіді					
5	б	Львів		бали					
6	в	Одеса							
7	г	Київ							
8	2	Скільки областей в Україні?		Усього балів:					
9	а	25							
10	б	24							
11	в	22							
12	г	27							
13	3	Найдовше місто України...							
14	а	Київ							
15	б	Кривий Ріг							
16	в	Чернігів							
17	г	Дніпропетровськ							
18	4	Найдовша річка України							
19	а	Десна							
20	б	Південний Буг							
21	в	Дніпро							
22	г	Дністер							

Рис. 15.1.

Пояснення. У клітинках **A3:B22** введено запитання та варіанти відповідей. У клітинках **E2:H2** — правильні відповіді на запитання, згодом вони будуть приховані, щоб той, хто буде давати відповіді на запитання, їх не бачив. У клітинках **E5:H5** залежно від відповіді буде зараховано 1 або 0 балів, а в клітинці **E8** має бути виставлена оцінка за тест.

- До клітинок **E4:H4** введіть відповіді довільним чином.
- У клітинці **E5** введіть формулу для перевірки правильності відповіді та зарахування одного бала, якщо відповідь є правильною: **=IF(E2=E4;1;0)**. Скопіюйте цю формулу до діапазону **F5:H5**, використовуючи автозаповнення.
- У клітинці **I5** полічіть загальну кількість набраних балів за тестування.
- У клітинці **E8** введіть формулу для виставлення оцінки за 12-бальною системою оцінювання. Для цього потрібно загальну кількість балів (**I5**) поділити на кількість запитань і помножити на 12. Результат має бути цілим числом.
- До клітинок **E4:H4** застосуйте умовне форматування: якщо в клітинці відповідь правильна, то вона зафарбована в зелений колір, якщо неправильна, то — в рожевий.
 - Виділіть діапазон **E4:H4**.
 - На вкладці **Основне** в групі **Стилі** оберіть команду **Умовне форматування** ▶ **Створити правило**.



- c. У вікні **Нове правило форматування** оберіть команду **Використовувати формулу для визначення клітинок для форматування**.
 - d. У полі **Форматувати значення, для яких ця формула має значення ІСТИНА** введіть формулу: **E5=1** або **E2=E4**.
 - e. Клацніть кнопку **Форматувати...** та встановіть зелений колір зафарбування клітинки. Натисніть **ОК**.
 - f. Самостійно створіть правило для встановлення умовного форматування, якщо відповідь неправильна.
8. Приховайте проміжні обчислення та правильні відповіді, щоб особа, яка проходить тестування, їх не бачила.
- a. Виділіть клітинки **D2:H2** і **D5:I5**.
 - b. Установіть для них білий колір літер.



Дізнайся більше
про захист
клітинок.

Примітка. Excel у своєму арсеналі має і більш професійний підхід для приховування даних і формул з подальшим захистом аркуша від змін. При бажанні з цією можливістю ви можете ознайомитися самостійно.

- 9*. Змініть формулу в клітинці **E8**, враховуючи, що якщо користувач не надав жодної відповіді, він має замість 0 балів отримати оцінку 1 бал.

ЗАВДАННЯ 2



Розв'язати квадратне рівняння. Відповідь представити у вигляді двох чисел, округлених до двох десяткових знаків, або у вигляді повідомлення «немає розв'язків». Повідомлення про відсутність розв'язків підсвітити жовтим кольором, додатні розв'язки — рожевим, а від'ємні — блакитним.

- 1. Перейдіть до аркуша **Квадратне рівняння**.
- 2. У клітинках **A2:C2** введіть довільні числа, які є коефіцієнтами квадратного рівняння.
- 3. У клітинці **E2** обчисліть дискримінант квадратного рівняння за формулою: $D = b^2 - 4ac$.
- 4. До клітинок **B4** і **B5** уведіть формули для обчислення коренів квадратного рівняння з урахуванням умови: якщо $D < 0$, то рівняння не має коренів.
 - a. До клітинки **B4** введіть формулу:
=IF(E2<0;"немає розв'язків";(-B2+SQRT(E2))/(2*A2)).
 - b. До клітинки **B5** уведіть формулу самостійно.
- 5. Виконайте умовне форматування клітинок **B4** і **B5**, указане в умові завдання.



ЗАВДАННЯ 3



На аркуші **Заробітна платня** заповнити таблицю відповідно до запропонованих умов.

Нарахування за лікарняні дні залежать від стажу роботи: до 3-х років — 50%; від 3-х до 5-ти років — 60%; від 5-ти до 8-ми років — 70 %; більше 8-ми років — 100%.

Врахувати, що податок становить 15%, а відрахування до Пенсійного фонду — 2%.

Увести формули до клітинок, зафарбованих зеленим кольором, та виконати їх копіювання, застосувавши дію автозаповнення.

Проаналізуємо, які саме формули необхідно ввести.

Вартість одного робочого дня слід визначити як частку від ділення ставки працівника на кількість робочих днів у місяці.

Для визначення кількості відпрацьованих днів потрібно відняти від кількості робочих днів у місяці кількість днів за лікарняними.

Нарахування за робочі дні визначається як кількість відпрацьованих днів, помножена на вартість одного робочого дня.

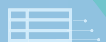
Нарахування за лікарняні дні залежать від стажу роботи, що міститься у стовпчику **С**. Визначається ця сума, як добуток кількості днів за лікарняним на вартість одного робочого дня, помножена на коефіцієнт, який залежить від стажу роботи. Якщо стаж менше трьох років, коефіцієнт дорівнює 0,5, якщо стаж від 3-х до 5-ти років, коефіцієнт буде 0,6, від 5 до 8 років коефіцієнт становить 0,7, якщо стаж становить більше 8 років, то коефіцієнт дорівнює одиниці.

Для визначення загальної суми нарахування слід додати суму, що нарахована за робочі дні, до суми, що нарахована за лікарняні дні.

Для визначення податку помножте загальну суму нарахування на 0,15, оскільки податок становить 15%. Аналогічною є й формула обчислення нарахувань до Пенсійного фонду.

Для визначення суми до видачі слід від загальної суми нарахування відняти податки та відрахування до Пенсійного фонду.

Формули для визначення загальної суми, максимального, мінімального та середнього значень визначте самостійно.



§16.

СТВОРЕННЯ ДІАГРАМ

16.1.

Поняття діаграми


Вивчаючи матеріал попередніх параграфів, ви могли переконатися, що за допомогою електронних таблиць можна легко порівнювати дані. Дізнатися, наприклад, яка температура є найвищою чи, навпаки, найнижчою, скільки дівчат у класі та скільки хлопців, у якому місяці продали більше книжок, яка з річок найдовша тощо.

Однак, є набагато кращий спосіб порівняння даних, більш наочний. Найзручніше порівнювати дані, подані у вигляді графіків або діаграм.

Діаграма (від грец. *διάγραμμα* — креслення) — це графічне зображення, в якому дані подані у вигляді геометричних фігур.

16.2.

Типи діаграм

Найбільшого розповсюдження набула побудова стовпчастих діаграм. *Стовпчасті діаграми*, як правило, використовують для відображення змін даних за певний проміжок часу або демонстрації порівняння різних об'єктів. До того ж категорії в них зазвичай формуються по горизонтальній осі, а значення — по вертикальній. Створюються вони з допомогою інструмента , який розташований на вкладці **Вставлення** в області **Діаграми**. З його допомогою можна вставити звичайні плоскі та об'ємні діаграми, а також стовпчасті діаграми з накопиченням. Так, на рис. 16.1 наведено приклад стовпчастої діаграми, що містить інформацію про кількість переможців олімпіад з 7-го по 11-й клас за 2020-й та 2021-й роки.

Схожими за своїм призначенням є *гістограми*. Їхня відмінність у тому, що вони мають іншу орієнтацію осей. На відміну від стовпчастих діаграм, тут вісь *x* розташована вертикально, а вісь *y* — горизонтально. Горизонтально розташовані осі є більш зручними для порівняння різних величин у межах одного часового періоду. Створюють їх з допомогою того ж самого інструмента, що і стовпчасті діаграми. В меню цього інструмента також є як плоскі, так і об'ємні гістограми різних видів. Як приклад на рис. 16.2 наведено діаграму, що містить інформацію про обсяги продажу техніки протягом певного періоду.



Рис. 16.1



Рис. 16.2

Ще один тип діаграм — *лінійчаті діаграми* (або графіки). Ці діаграми дозволяють демонструвати в єдиному масштабі неперервну зміну даних за рівні проміжки часу. В графіку дані категорій розподіляються по горизонтальній осі, а значення — по вертикальній осі. Створюються вони з допомогою інструмента . З його допомогою можна вставити різні види графіків як з маркерами, так і без них. Також є можливість побудови об'ємної лінійчатої діаграми. Як приклад на рис. 16.3 наведено графік, що містить інформацію про зміни середньої температури повітря протягом тижня.

Схожими за призначенням є *діаграми з областями*. Ці види діаграм демонструють величину змін між кількома рядами даних залежно від часу. Побудувати їх можна, використовуючи меню того ж інструмента, що і для побудови графіків, обравши бажаний вид діаграми з областями. На рис. 16.4 наведено приклад такої діаграми, що відображає динаміку обсягів продажу різних видів фруктів.



Рис. 16.3

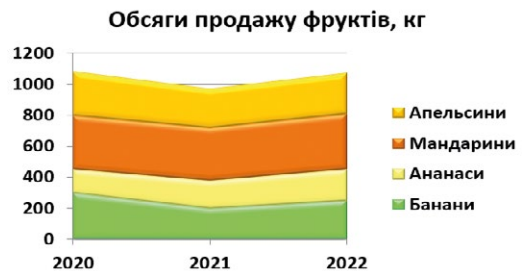


Рис. 16.4

Секторні (або кругові) діаграми відображають величину розміру елементів одного ряду даних пропорційно до суми всіх елементів. Точки даних у секторній діаграмі відображаються у вигляді відсоткової частки від цілого круга. Створюються вони з допомогою інструмента . Цей інструмент також надає можливість створення кільцевих діаграм. Як приклад на рис. 16.5 наведено кругову діаграму, що містить інформацію про розподіл продажу різних видів квітів.

Ще один вид діаграм — *точкові діаграми*. Їх використовують для визначення потенційних зв'язків між двома змінними. На точковій діаграмі значення двох осей об'єднуються в одну точку даних і розміщуються з різними інтервалами. Такі діаграми, як правило, використовують для виведення та порівняння наукових, статистичних або інженерних даних. Завдяки цьому типу діаграм можна будувати графіки функцій, здійснювати графічне розв'язання системи рівнянь тощо. Створюються вони з допомогою інструмента . З його допомогою можна вставити діаграми, що містять лише точки (маркери), діаграми в яких точки з'єднані гладкими або прямими лініями. Як приклад на рис. 16.6 наведено точкову діаграму, що містить інформацію про залежність між кількістю разів виходу реклами та кількістю проданих автомобілів.



Рис. 16.5

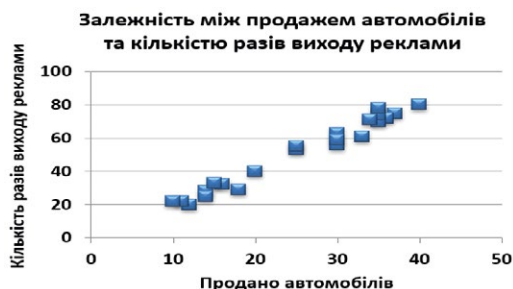


Рис. 16.6

16.3.

Елементи діаграми

Як видно з наведених вище прикладів, діаграми мають різний зовнішній вигляд і різні елементи. Розглянемо елементи на прикладі стовпчастої діаграми (рис. 16.7).

Зовнішня область діаграми має назву *область діаграми*, область, у якій безпосередньо побудовано діаграму, називається *областю побудови*. В області побудови розміщено стовпці, які називаються *елементами даних*. Висота стовпця є пропорційною числовому значенню, яке він відображає. Якщо діаграму побудовано за даними декількох числових рядів, стовпці будуть різних кольорів. Елементи даних, представлені одним кольором, називають *рядом даних*. Заголовки стовпців і рядків таблиці часто використовуються як імена рядів даних, що відображається в легенді діаграми. Так, у наведеному прикладі іменами рядів даних є 8 клас і 9 клас. Про відношення кольору стовпців до того чи іншого числового ряду повідомляє *легенда*. У верхній частині області діаграми розташовано *назву діаграми*.



Шкала значень розташована по вертикалі і має назву *вісь значень*, по горизонталі вказано категорії на *осі категорій*.

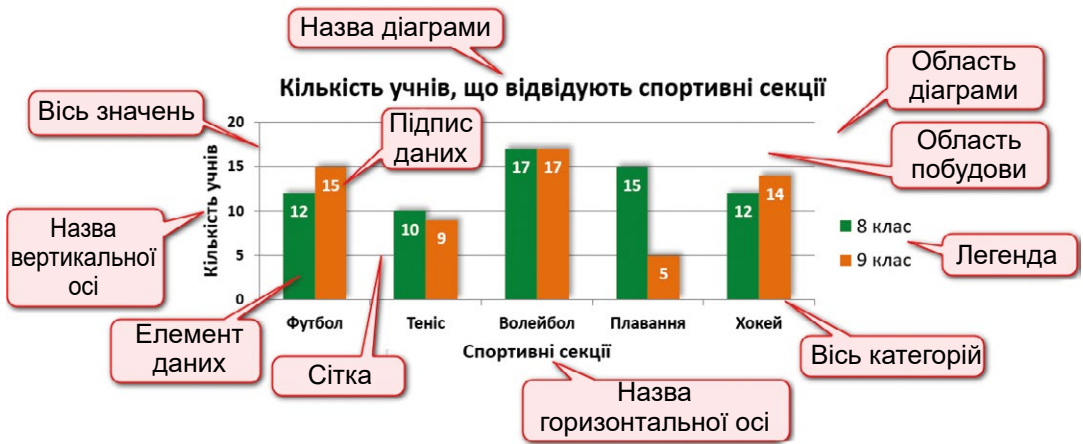


Рис. 16.7. Елементи діаграми

16.4.

Створення діаграми

Розглянемо алгоритм побудови розглянутих та інших, менш уживаних діаграм.

1. Виділити діапазон з даними.
2. На вкладці **Вставлення** клацнути стрілку в нижньому правому куті області **Діаграми**.

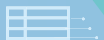
В результаті відкриється діалогове вікно **Вставлення діаграми**. На першій вкладці представлено рекомендовані діаграми, які пропонує Excel на основі виділених числових даних.

3. Обрати вкладку **Усі діаграми**.

Тут представлено перелік можливих для побудови діаграм. При виборі бажаної категорії можна обрати тип діаграми та побачити, який вигляд буде мати побудована діаграма.

4. Обрати бажаний тип діаграми та натиснути **ОК**. В результаті діаграму буде побудовано.

Як уже було зазначено вище, кожний тип діаграм має свої різновиди. Так, діаграми можуть бути плоскими та об'ємними. Залежно від мети подання даних використовують звичайну діаграму або діаграму з накопиченням. У звичайній стовпчастій діаграмі графічні



зображення числових даних з різних наборів розташовано поруч один з одним. У діаграмі з накопиченням — один на одному. Це дає змогу оцінити сумарні дані і внесок кожної складової в загальну суму. Користувач може обрати потрібний різновид з меню інструмента відповідного типу діаграм.

Розглянутий вище алгоритм побудови є універсальним і підійде для побудови будь-якого виду діаграм. Для швидкої побудови розповсюджених видів діаграм використовують відповідні інструменти, що розташовані на вкладці **Вставлення** в області **Діаграми**. Розглянемо їхнє використання, виконавши практичну вправу.


ВПРАВА 16.1. СТВОРЕННЯ ДІАГРАМИ

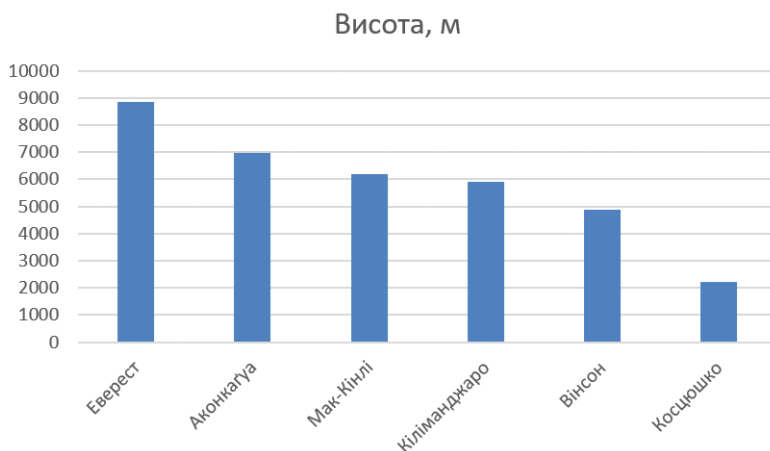


Завдання. Створити діаграму, використовуючи дані з таблиці **Найвищі вершини гір** (рис. 16.8).

	A	B	C
1		Найвищі вершини гір	
2	№	Вершина	Висота, м
3	1	Еверест	8848
4	2	Аконкагуа	6962
5	3	Мак-Кінлі	6194
6	4	Кіліманджаро	5895
7	5	Вінсон	4892
8	6	Косцюшко	2228

Рис. 16.8

1. Створіть таблицю з даними відповідно до зразка (рис. 16.8).
2. Виділіть клітинки **B2:C8**, що містять дані, на основі яких потрібно побудувати діаграму. (У стовпці **B** містяться назви вершин, а у стовпці **C** — їхні висоти.)
3. Оберіть вкладку **Вставлення**.
4. В області **Діаграми** оберіть меню стовпчастого типу діаграм .
5. У випадаючому списку оберіть перший тип діаграми: **звичайна стовпчаста діаграма**. В результаті діаграму буде побудовано з параметрами форматування за умовчанням.





Діаграму можна перемістити на окремий аркуш, а також повернути на початковий аркуш або на будь-який інший.

Як правило, побудовані діаграми вимагають зміни певних налаштувань і форматувань зовнішнього вигляду. Для подальшої роботи з діаграмою користувач може скористатися одним із інструментів в групі вкладок **Знаряддя для діаграм**, що з'являються після побудови або виділення діаграми.

16.5.

Форматування діаграм

Для швидкої зміни макета діаграми потрібно у вкладці **Конструктор** відкрити меню інструмента **Швидкий макет** та встановити бажаний зразок. Цей інструмент призначений для того, щоб можна було до діаграми застосувати наявний уже макет найбільш уживаних виглядів діаграм і не витрачати кожного разу час на налаштування діаграми.

Налаштування всіх елементів діаграми можна зробити також окремо, скориставшись меню інструмента **Додати елемент діаграми** на вкладці **Конструктор**. Меню цього інструмента містить відповідні команди для налаштувань осей, їхніх назв, назви діаграми тощо.

Форматуючи діаграми, не слід забувати про їхній зовнішній вигляд. Один із швидких способів — на вкладці **Конструктор** обрати один із запропонованих зразків в області **Стилі діаграм**. Проте це не єдиний спосіб. Більше можливостей форматування містить вкладка **Формат**. Тут є можливість більш детальної зміни стилів фігур, параметрів шрифтів тощо. Щоб замінити фон одного з елементів діаграми, потрібно виділити об'єкт та на вкладці **Формат** в області **Стилі фігур** обрати бажаний стиль. Для створення власного стилю скористайтеся командами **Заливка фігури**, **Контур фігури**, **Ефекти для фігур**.

ВПРАВА 16.2. ФОРМАТУВАННЯ ДІАГРАМИ

Завдання. Виконати форматування діаграми відповідно до зразка (рис. 16.9).

1. Змініть заголовок діаграми. Для цього клацніть всередині наявного заголовку та введіть необхідний текст. Форматування заголовка здійсніть стандартними засобами форматування шрифту на вкладці **Основне**.

Примітка. Якщо необхідно видалити заголовок, його потрібно виділити та натиснути **Delete**. Щоб знову додати заголовок діаграми, необхідно на вкладці **Конструктор** обрати інструмент **Додати елемент діаграми** ► **Назва діаграми** та бажаний спосіб розташування заголовка.

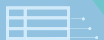



Рис. 16.9. Діаграма найвищих вершин гір

2. Змініть параметри вертикальної осі, встановивши інтервал між її значеннями 2000.
 - a. В контекстному меню осі оберіть команду **Формат осі**. Праворуч має з'явитися вікно форматування осі.
 - b. В групі **Одиниці вимірювання** в полі **Основні** значення 1000 змініть на 2000.
3. Додайте назву вертикальної осі. Для цього на вкладці **Конструктор** оберіть інструмент **Додати елемент діаграми** ▶ **Назви осей** ▶ **Вертикальна для основної осі**.
4. Змініть напрямок підписів даних (зробіть їх горизонтальними).
 - a. У вікні **Формат Осі** оберіть групу  (**Властивості й розмір**).
 - b. У полі **Настроюваний кут** установіть значення 0°. (Якщо написам не вистачатиме місця, збільшіть ширину діаграми.)
5. Змініть стиль та зафарбування стовпців діаграми. Для цього на вкладці **Формат** в області **Стилі фігур** оберіть необхідну заготовку.
6. Додайте у верхній частині стовпців підписи даних. Для цього на вкладці **Конструктор** оберіть команду **Підписи даних** в меню інструменту **Додати елемент діаграми**.
7. Остаточне форматування діаграми виконайте самостійно.

Примітка. Якщо існує необхідність змінити колір одного із стовпців діаграми, необхідно спочатку виділити усі стовпці, а потім клацнути на потрібному стовпці і змінити його налаштування.

ВПРАВА 16.3



Завдання. Створити таблицю, як на рис. 16.10, та побудувати на основі цих даних кругову діаграму (рис. 16.11).

1. Створіть таблицю відповідно до зразка (рис. 16.10).



	A	B
1	Назва океану	Площа, млн кв. км
2	Тихий	179,7
3	Атлантичний	93,4
4	Індійський	74,9
5	Північний Льодовитий	13,0

Рис. 16.10



Рис. 16.11

2. Виділіть необхідний для побудови діаграми діапазон **A1:B5**.
3. Оберіть вкладку **Вставлення**.
4. В області **Діаграми** оберіть інструмент для побудови секторної діаграми. А у випадяючому списку оберіть перший тип (рис. 16.12).
5. Змініть назву діаграми. Для цього клацніть на області діаграми та введіть текст **Розподіл площі Світового океану**.

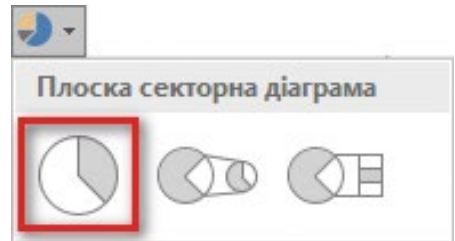


Рис. 16.12

6. Додайте написи даних зовні секторів діаграми у вигляді відсотків.
 - a. В меню інструмента **Додати елемент діаграми** оберіть команду **Підписи даних** ▶ **Зовні скраю**.
 - b. В контекстному меню даних оберіть команду **Формат підписів даних**.
 - c. В області **Вміст підписів** установіть прапорець біля опції **Відсотки**, а інші прапорці приберіть.
 - d. Збільшіть розмір шрифту підписів на вкладці **Основне**.
7. Змініть розташування легенди діаграми, обравши в меню інструмента **Додати елемент діаграми** команду **Легенда** ▶ **Справа**.
8. Змініть кольорове оформлення діаграми та області побудови.



Виконайте вправу 16.4.



Дізнайтеся більше про спарклайни.



Контрольні запитання та завдання

1. Що таке діаграма?
2. Які види діаграм вам відомі?
3. Для чого призначені діаграми?
4. Назвіть основні елементи діаграми.
5. Для чого використовують діаграми з областями; точкові діаграми?
6. Укажіть послідовність дій для створення діаграми.
7. Як виконати форматування елементів діаграми?



Питання для роздумів

- 1*. Чи відбудуться зміни в діаграмі після зміни числових даних, на основі яких побудовано діаграму?
- 2*. Чи можна побудувати діаграму, використовуючи несуміжні стовпці?
- 3*. Який тип діаграм краще підходить для побудови графіків функцій?



Завдання для досліджень

- 1*. Дослідіть, як змінити тип уже побудованої діаграми. (Наприклад, стовпчасту діаграму замінити на секторну.)
- 2*. Дослідіть, як розширити кількість наявних у діаграмі даних. (Наприклад, було побудовано діаграму прибутків за 5 днів, що відповідно містить 5 стовпців. Необхідно кількість днів збільшити на 2, відповідно на діаграмі має бути побудовано 7 стовпців.)

§17.

ВИБІР ТИПУ ДІАГРАМ

Основне завдання побудови діаграм — вибрати правильно її тип для коректного відображення числових даних. Найбільш широко використовують графіки, стовпчасті та кругові діаграми.

Для побудови **графіків** ураховують:

- чи є залежність між двома числовими величинами;
- значення числових величин, що мають бути визначені через однакові проміжки часу.



Отже, використання графіка передбачає демонстрацію деякого неперервного процесу. Також графіки слід застосовувати, коли потрібно відобразити дуже велику кількість значень.

Графік зміни температури протягом тижня



Рис. 17.1. Приклад побудови графіка

Відвідування спортивних секцій учнями школи



Рис. 17.2. Приклад побудови стовпчастої діаграми

Побудова **стовпчастих** діаграми доцільна тоді, коли:

- є залежність між двома числовими величинами;
- дані, що мають бути відображені на діаграмі, не демонструють неперервний процес.

Побудову **секторних (кругових)** діаграм здійснюють, якщо:

- необхідно відобразити дані лише одного ряду;

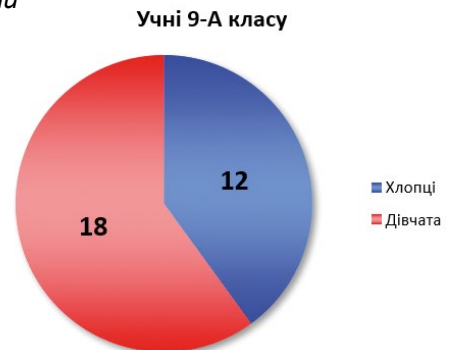
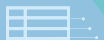


Рис. 17.3. Приклад побудови секторної діаграми



РОЗДІЛ 2. Опрацювання табличних даних

- жодне із значень не є нульовим або від'ємним;
- категорії мають відповідати частинам єдиного цілого.

ВПРАВА 17.1




Завдання. Таблиця містить три набори даних. Проаналізувати представлені дані, обрати необхідний тип діаграм для відображення цих даних і побудувати діаграми.

	A	B	C	D	E	F	G	H
1	Розподіл часу учня протягом доби			Продаж товару за день			Коливання курсу валют	
2								
3	Вид діяльності	Витрати часу, год		Найменування	Кількість, кг		Дата	Значення
4	Навчання	6		Зефір	25,5		01.02	27,58
5	Прийом їжі	2		Тістечко	36,7		02.02	27,67
6	Сон	8		Морозиво	75,4		03.02	27,11
7	Відпочинок	5,5		Мафін	23,3		04.02	27,33
8	Заняття спортом	2,5		Круасан	16,5		05.02	26,82

Рис. 17.4

1. Введіть дані до аркуша електронної таблиці відповідно до зразка (рис. 17.4).
2. Розгляньте дані розподілу часу учня протягом доби. Виберіть тип діаграми.

Оскільки дані містять усього один ряд, не містять від'ємних і нульових значень, а також є складовими єдиного цілого, а саме, доби, то для даного набору даних доцільною буде побудова секторної діаграми.

3. Побудуйте діаграму розподілу часу учня протягом доби.
 - a. Виділіть діапазон **A4:B8**.
 - b. На вкладці **Вставлення** оберіть інструмент для побудови секторної діаграми  та її бажаний вид. В результаті діаграму буде побудовано.
4. Додайте до діаграми заголовки та підписи даних у відсотках.
5. Розгляньте дані продажу товарів за день. Виберіть тип діаграми.

Оскільки між двома числовими величинами немає залежності, дані не є часткою деякого цілого значення, то для їхнього представлення доцільно побудувати стовпчасту діаграму або гістограму.

6. Самостійно побудуйте стовпчасту діаграму продажу товарів за день, виконайте її оформлення, додайте назву та вилучіть легенду.
7. Розгляньте дані коливання курсу валют. Виберіть тип діаграми.

Оскільки є залежність між двома величинами (залежно від дати встановлено певний курс валют), а також значення відтворюють певний процес через однаковий проміжок часу, то для демонстрації цих даних доцільно побудувати графік.



8. Побудуйте графік коливання курсу валют без використання маркерів, оформіть його, надавши йому червоний колір з рожевою підсвіткою, додайте назву і вилучіть легенду.



Виконайте вправу 17.2.



Контрольні запитання та завдання

1. У яких випадках використовують секторні діаграми?
2. Які типи діаграм демонструють неперервну зміну даних за рівні проміжки часу?
3. За яких умов здійснюють побудову стовпчастих діаграм?
- 4*. Чи можна для одного набору даних побудувати різні типи діаграм?



Питання для роздумів

- 1*. Який тип діаграми підійде для відображення набору наступних даних:
 - а) щоденна кількість опадів протягом місяця;
 - б) кількість переможців олімпіади з різних предметів;
 - в) хімічний склад речовин у повітрі?
- 2*. Визначте умови побудови точкової діаграми.
- 3*. Розробіть алгоритм вибору типу діаграми залежно від числових даних.



Завдання для досліджень

- 1*. Визначте, як побудувати діаграми різних типів на одній ділянці побудови, наприклад, стовпчасту діаграму та графік.
- 2*. Дослідіть призначення таких типів діаграм, як бульбашкова, поверхнева, каскадна.

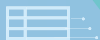


ПРАКТИЧНА РОБОТА 4. «СТВОРЕННЯ ДІАГРАМ»

ЗАВДАННЯ 1



Створити таблицю продажу автомобілів різних марок в автосалоні за два місяці (рис. 18.1) та побудувати на її основі стовпчасту діаграму відповідно до зразка (рис. 18.2).



Марка автомобіля	Січень	Лютий
Mersedes	10	8
Opel	8	12
Fiat	9	6
Audi	4	7

Рис. 18.1

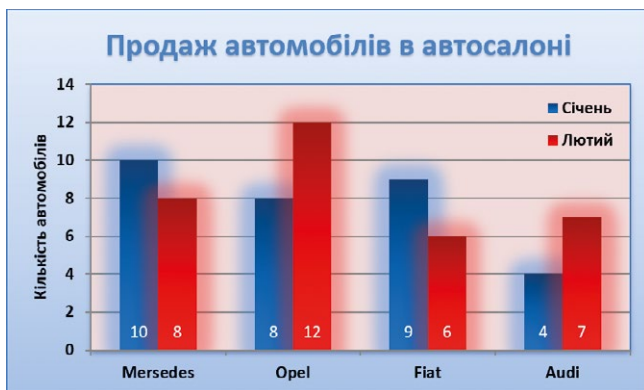


Рис. 18.2.

ЗАВДАННЯ 2



Дано таблицю одержання та відправлення електронних повідомлень деяким користувачем упродовж тижня (рис. 18.3). Побудувати діаграму, що має наочно показати представлені дані.

	A	B	C
1	Дата	Відправлено	Одержано
2	01.02	20	11
3	02.02	10	5
4	03.02	6	8
5	04.02	4	9
6	05.02	10	15
7	06.02	4	6
8	07.02	7	10


Рис. 18.3

1. Оберіть тип діаграми для відображення даних.

Оскільки є залежність між двома величинами (залежно від дати надіслано або відправлено певну кількість листів), а також значення відтворюють певний процес через однаковий проміжок часу, то для демонстрації цих даних доцільно побудувати графіки.

2. Побудуйте діаграму для графічного подання даних.

- a. Виділіть діапазон **A1:C8**.

- b. На вкладці **Вставлення** оберіть інструмент для побудови графіка  та в його меню оберіть вид **Лінійчатa діаграма з маркерами**. У результаті буде побудовано графіки.

3. Додайте назву діаграми.

4. Установіть форматування ліній та маркерів графіків відповідно до зразка (рис. 18.4).

- a. Клацніть на одному графіку та в контекстному меню оберіть команду **Формат ряду даних**.

- b. Перейдіть до категорії **Заливка й контур**, оберіть інструмент **Лінія** та встановіть необхідний колір лінії.

- c. Перейдіть до категорії **Ефекти** та встановіть тінь для лінії.

- d. Перейдіть до категорії **Заливка й контур**, оберіть інструмент **Маркер**.

- e. У групі **Параметри маркера** встановіть необхідний тип та розмір.



Електронне листування протягом тижня

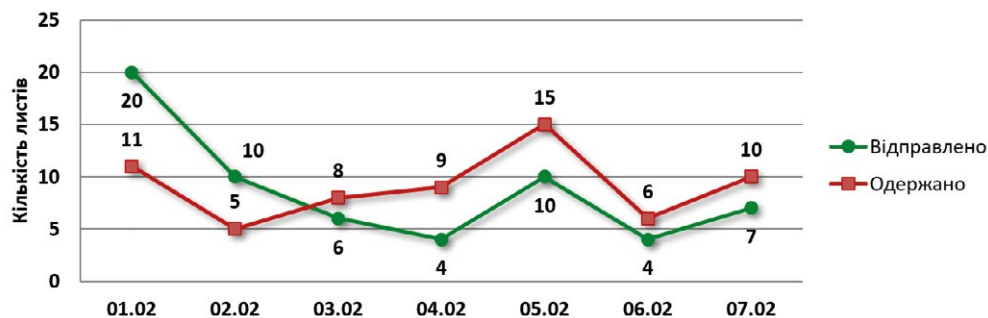


Рис. 18.4

- f. У групах **Заливка** та **Рамка** встановить необхідний колір маркера.
- g. Аналогічно налаштуйте зовнішній вигляд другого графіка.
5. Подальше форматування діаграми завершіть самостійно.

ЗАВДАННЯ 3



Побудувати графік функції $y = x^2$, для x від -10 до 10 з кроком 1 відповідно до зразка рис. 18.5.

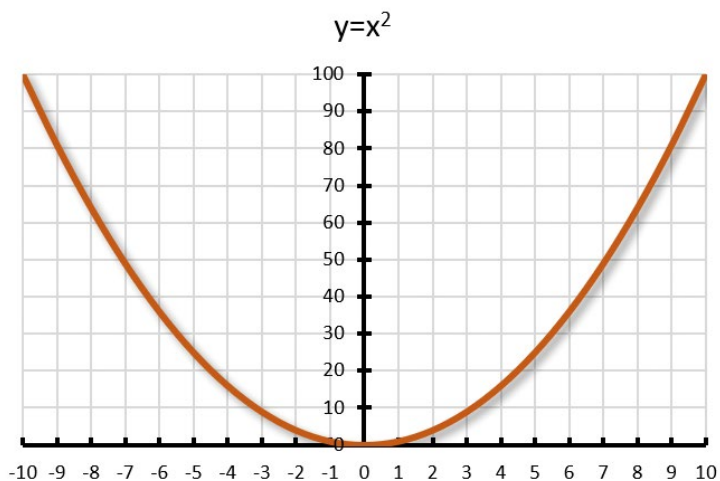


Рис. 18.5

1. У клітинці **A1** запишіть змінну x , а в клітинці **B1** змінну y .
2. До клітинок **A2:A21** уведіть числа від -10 до 10 з кроком 1.
3. До клітинки **B2** введіть формулу для знаходження значення y : **=A2^2**.
4. Виділіть діапазон **A1:B1**.



5. На вкладці **Вставлення** оберіть інструмент для побудови точкової діаграми . А в його меню оберіть вид **Точкова діаграма з плавними лініями**. В результаті графік буде побудовано.
6. Встановіть максимальне та мінімальне значення осей, а також значення основних поділок. Для цього в контекстному меню осі x оберіть команду **Формат осі**. В результаті відкриється вікно **Формат осі**. У розділі **Параметри осі** встановіть необхідні значення у полі **Мінімум** — мінус 10; у полі **Максимум** — 10. Для основних поділок установіть значення 1 у полі **Основні**.
7. Змініть колір та товщину осі. Для цього у цьому ж вікні **Формат осі** оберіть інструмент **Заливка й контур**. Відкрийте групу властивостей **Контур**. Установіть чорний колір та товщину 2 пункти.
8. Додайте на осі поділки. Для цього поверніться до інструменту **Параметри осі**, оберіть групу **Поділки** і у полі **Тип основних поділок** оберіть значення **Усередині**.
9. Аналогічно здійсніть налаштування і також параметрів осі y.
10. Подальше форматування діаграми виконайте самостійно.

ЗАВДАННЯ 4*



Побудувати графік функції $y = \frac{1}{x}$. Для побудови графіка керуватися поясненнями із попереднього завдання.



УПОРЯДКУВАННЯ ДАНИХ У ТАБЛИЦЯХ. ПРОМІЖНІ ПІДСУМКИ

19.1.

Звичайне сортування

Коли користувач має справу з великою кількістю даних, для зручної роботи з ними можна здійснити їхнє сортування.

Сортування — це розташування даних таблиці в потрібному порядку. Наприклад, від меншого до більшого (за зростанням) або навпаки — від більшого до меншого (за спаданням).

MS Excel має можливості сортування не тільки за значенням даних, а й за кольором шрифту або клітинки. Впорядковувати можна текст, числа, дати та час. Як правило, сортування даних здійснюється по стовпцях. До того ж, заголовки стовпців мають бути розташовані тільки в одному рядку.



Назви з пробілами	Назва фільму
	Один в полі воїн.
	Три мушкетери.
	10 негреньят.
Назви, на початку яких є числа	12 мавп.
	13-й район.
	9 днів одного року.
	Аполлон 13.
	Вісім з половиною.
Сортування тексту	Два капітани.
	Двоє під однією парасолькою.
	Один вдома.
	Перший вчитель.

Рис. 19.1

Для сортування текстових даних потрібно звернути увагу, щоб усі клітинки мали текстовий формат і на початку тексту не містили пробілів. Текст з пробілами буде вищим за того, який не містить пробілів. Написи, що на початку містять числа, будуть розташовані над текстовими написами. Це добре видно із наведеного на рис. 19.1 прикладу.

Для найпростішого сортування текстових даних необхідно клацнути в будь-якій клітинці стовпця з даними для сортування, після чого скористатися одним із способів сортування.

1 спосіб

На вкладці **Основне** обрати в меню інструмента **Сортувати й фільтрувати** команду **Сортування від А до Я** (для сортування за алфавітом). В результаті буде здійснено сортування даних у тому стовпці, в якому було виділено клітинку, а значення, що знаходяться у відповідних рядках інших стовпців, будуть також переміщені.

Для сортування у зворотному порядку в меню інструмента **Сортувати й фільтрувати** слід обрати команду **Сортування від Я до А**.

2 спосіб

Перейти на вкладку **Дані**, де обрати необхідний інструмент для сортування **Сортування від А до Я** або **Сортування від Я до А**.

Щоб посортувати дані в межах тільки одного стовпця, спочатку треба виділити його. Обрати інструмент для сортування. В результаті програма запропонує обрати один із варіантів (рис. 19.2).

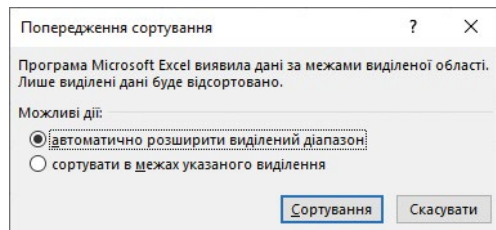
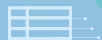


Рис. 19.2



Якщо обрати варіант **сортувати в межах указанного виділення**, то дані, що знаходяться в інших стовпцях, залишаться на місцях і сортування відбудеться лише в межах виділеного стовпця.

19.2.

Настроюване сортування

Excel має ширші можливості сортування. Їх використовують, коли необхідно здійснити сортування спочатку у межах одного стовпця, а потім іншого. Так, на рис. 19.3 наведено приклад, де спочатку сортування зроблено за стовпцем **Видавництво**, а в межах кожного видавництва здійснено сортування за роком видання за спаданням.

Назва	Видавництво	Рік випуску
Фанат футболу	Богдан	2016
П'ятеро дітей та ельф	Богдан	2015
Зірка з ліжка та сніданку	Богдан	2011
Мій тато став зіркою	Видавництво Старого Лева	2015
Нескінчена Історія	Видавництво Старого Лева	2014
Кіт на ім'я Сплет	Видавництво Старого Лева	2013
Ліга непарних шкарпеток	Клуб сімейного дозвілля	2013
Гра в паралельне читання	Клуб сімейного дозвілля	2013
Зворотній бік світла	Клуб сімейного дозвілля	2012
Робінзон Крузо	Країна мрій	2012
Вінн-Пух. Повернення у столітній ліс	Країна мрій	2011
Володар Драконів	Ранок	2015
Про Маринку і Тюпу	Ранок	2011

Рис. 19.3

Також настроюване сортування використовується при сортуванні за кольором клітинки або шрифту.

Розглянемо використання настроюваного сортування, виконавши практичну вправу.

ВПРАВА 19.1



Завдання. Виконати сортування книжок за стовпцем **Видавництво** в алфавітному порядку, а потім за стовпцем **Рік видання** за спаданням.

1. Завантажте файл **Вправа_19_1.xlsx** з матеріалів до даного параграфа.
2. Клацніть у будь-якій клітинці таблиці.
3. Перейдіть на вкладку **Основне**.



4. В меню інструмента **Сортувати й фільтрувати** оберіть команду **Настроюване сортування...**. В результаті буде відкрито вікно **Сортування** (рис. 19.4). Також це вікно можна відкрити, обравши інструмент **Сортувати** на вкладці **Дані**.

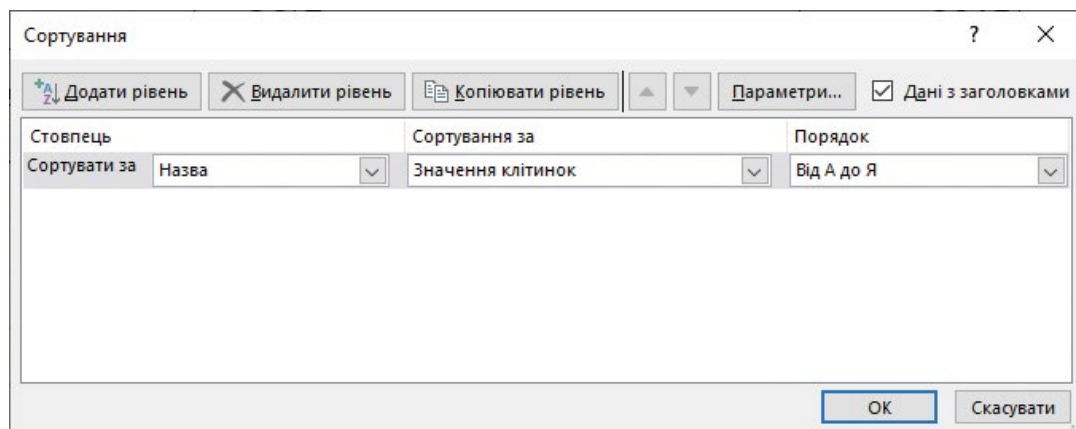


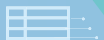
Рис. 19.4

Зверніть увагу на прапорець біля опції **Дані з заголовками**. Його слід установлювати тоді, коли в таблиці перший рядок містить заголовки, а не дані, отже, при сортуванні перший рядок буде залишатися на своєму місці. Якщо у першому рядку містяться дані, для яких потрібно застосувати сортування, прапорець необхідно прибрати.

5. Установіть необхідні параметри сортування.
- У полі **Стовпець** у списку **Сортувати за** оберіть стовпець **Видавництво**.
 - Залиште незмінним **Сортування за** **Значення клітинок**.
 - Також залиште незмінним порядок сортування **Порядок** **Від А до Я**.
 - Щоб здійснити сортування ще за одним стовпцем, натисніть кнопку **Додати рівень**.
 - У списку **Потім за** оберіть стовпець **Рік випуску**.
 - Залиште незмінним сортування за значенням клітинок.
 - Установіть порядок сортування від найбільшого значення до найменшого.
 - Натисніть **ОК**. В результаті дані в таблиці буде відсортовано спочатку за назвами видавництва в алфавітному порядку, а потім у межах кожного видавництва за роками в порядку спадання.



Дізнайтеся більше про настроюване сортування.



19.3.

Проміжні підсумки

Підбиття *проміжних підсумків* — це зручна та наочна можливість провести обробку записів списку для подальшого аналізу. Проміжні підсумки можуть визначати кількість, суму, добуток, середнє, максимальне та мінімальне значення в групі даних.

Так, на рис. 19.5 наведено приклад обчислення проміжних підсумків кількості книжок, що були випущені у кожному році. Із прикладу видно, що у даний список автоматично вставляються рядки, що відображають підсумкові значення за обраними клітинками.

Назва	Видавництво	Рік випуску	Кількість сторінок
Кіт на ім'я Сплет	Видавництво Старого Лева	2013	36
Ліга непарних шкарпеток	Клуб сімейного дозвілля	2013	128
Гра в паралельне читання	Клуб сімейного дозвілля	2013	288
		2013 Кількість	3
Нескінчена Історія	Видавництво Старого Лева	2014	560
		2014 Кількість	1
П'ятеро дітей та ельф	Богдан	2015	280
Мій тато став зіркою	Видавництво Старого Лева	2015	38
Володар Драконів	Ранок	2015	512
		2015 Кількість	3
Фанат футболу	Богдан	2016	32
		2016 Кількість	1
		Загальна кількість	8

Рис. 19.5

Зверніть увагу, що створюючи проміжні підсумки, потрібно, щоб перший рядок містив назви стовпців, а інші рядки містили однотипні дані у кожному стовпці. Так, стовпці **Назва** та **Видавництво** містять текстові дані, а стовпці **Рік випуску** та **Кількість сторінок** — числові. До того ж, таблиця не повинна мати порожніх рядків.

І основне, на що потрібно звернути увагу, — стовпець з даними, за якими здійснюється підсумок, має бути відсортований. З прикладу видно, що значення у стовпці **Рік випуску** відсортовано за зростанням року.

Розглянемо послідовність дій знаходження проміжних підсумків, виконавши практичну вправу.

ВПРАВА 19.2



Завдання. За допомогою проміжних підсумків знайдіть:

- кількості книжок, що були випущені в кожному році;
- максимальні та мінімальні ціни кожного видавництва.



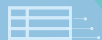
1. Створіть копію файлу **Вправа_19_1.xlsx** та надайте йому ім'я **Вправа_19_2.xlsx**.
2. Створіть копію наявного у книзі аркуша та надайте аркушам імена **Завдання_a** та **Завдання_b**.
3. Відсортуйте за зростанням дані у стовпці **Рік випуску**.
4. Задайте обчислення проміжних підсумків, щоб знайти кількість книжок, що були випущені в кожному році.
 - a. На вкладці **Дані** оберіть інструмент **Проміжні підсумки**. В результаті відкриється вікно **Проміжні підсумки**.
 - b. У групі **При кожній зміні в:** оберіть стовпець, за яким будуть здійснюватися підсумки, а саме, **Рік випуску**.
 - c. У групі **Використовувати функцію** оберіть функцію, за якою будуть підбиватися підсумки. Оберіть функцію **Кількість**.
 - d. У групі **Додати підсумки до:** оберіть стовпці, що містять значення, за якими будуть підбиватися підсумки. Оберіть стовпець **Рік випуску**.
 - e. Натисніть **ОК**. В результаті ви отримаєте проміжні підсумки, у яких відображено кількість книжок у кожному році.

Зауваження. Для видалення проміжних підсумків на вкладці **Дані** оберіть інструмент **Проміжні підсумки**, натисніть кнопку **Видалити все** та потім — **ОК**.

5. Задайте обчислення проміжних підсумків, щоб знайти максимальну ціну книжки кожного видавництва.
 - a. Перейдіть на аркуш **Завдання_b**;
 - b. Виділіть стовпець **Видавництво** та здійсніть сортування його даних в алфавітному порядку.
 - c. На вкладці **Дані** оберіть інструмент **Проміжні підсумки**.
 - d. У групі **При кожній зміні в** оберіть стовпець **Видавництво**.
 - e. Оберіть функцію **Максимум**.
 - f. У групі **Додати підсумки до** вкажіть стовпець **Ціна**.
 - g. Натисніть **ОК**. В результаті буде знайдено максимальну ціну книжки кожного видавництва.
6. Додайте до попередньої таблиці мінімальні значення ціни книжки кожного видавництва.
 - a. Знову оберіть інструмент **Проміжні підсумки**.
 - b. Змініть функцію **Максимум** на **Мінімум**.
 - c. Зніміть прапорець з опції **Замінити поточні підсумки**.
 - d. Натисніть **ОК**. В результаті ви одержите як максимальні, так і мінімальні значення ціни книжки кожного видавництва.



Виконайте вправу 19.3.



Контрольні запитання та завдання

1. Які є способи сортування даних?
2. За якими параметрами можна здійснити сортування в MS Excel?
3. Для чого використовують проміжні підсумки?
- 4*. Як прибрати проміжні підсумки з аркуша?



Питання для роздумів

- 1*. Як виконати сортування за даними усього стовпця разом із першим рядком?
- 2*. Чому при обчисленні проміжних підсумків один із стовпців має бути відсортованим?
- 3*. Наведіть приклади даних, для яких можна застосувати функції обчислення суми та середнього арифметичного при підбитті проміжних підсумків.



Завдання для досліджень

- 1*. Дослідіть, як виконати сортування за кольором клітинки, кольором шрифту.
- 2*. Дослідіть можливості сортування за вбудованими та користувацькими списками.

§20.

ПРОСТІ ТА РОЗШИРЕНІ ФІЛЬТРИ

20.1.

Простий фільтр

Сортування не завжди спрощує пошук, якщо йдеться про великі обсяги даних. Іноді зручно здійснити пошук даних, що відповідають деякому критерію. В таких випадках треба скористатися можливостями фільтрації даних. *Фільтрація* дозволяє виводити на екран лише ті значення, які відповідають заданим умовам відбору для стовпців, не змінюючи сам документ.

Так, на рис. 20.1 наведено приклад, у якому із переліку книжок вибрано лише ті, що видано в 2013 році.



	A	B	E	F
1	Назва	Автор	Рік випуску	Кількість сторінок
2	Вінн-Пух. Повернення у столітній ліс	Девід Бенедиктус	2011	208
3	Володар Драконів	Функе К.	2015	512
4	Гра в паралельне читання	Іванцова М.	2013	288
5	Зворотний бік світла	Дара Корній	2012	320
6	Зірка з ліжка та сніданку	Джаклін Вілсон	2011	224
7	Кіт на ім'я Сплет	Скоттон Р.	2013	36
8	Ліга непарних шкарпеток	Вдовиченко Г.	2013	128
9	Мій тато став зіркою	Кирпа Г.	2015	38
10	Нескінчена Історія			
11	Про Маринку і Тюпу			
12	П'ятеро дітей та ельф			
13	Робінзон Крузо			
14	Фанат футболу			

	A	B	E	F
1	Назва	Автор	Рік випуску	Кількість сторінок
4	Гра в паралельне читання	Іванцова М.	2013	288
7	Кіт на ім'я Сплет	Скоттон Р.	2013	36
8	Ліга непарних шкарпеток	Вдовиченко Г.	2013	128

Рис. 20.1

Відфільтровані рядки можна змінювати, редагувати, використовувати для створення діаграми, а от змінювати їх порядок розташування не можна.

Є два типи фільтрації списків. Для створення простих умов відбору використовують *автофільтр*, а для більш складних — *розширений фільтр*.

Автофільтр використовують, коли умова вибору стосується тільки одного стовпця або коли необхідно одночасно врахувати умови декількох стовпців. Для застосування автофільтра необхідно виконати такі кроки:

1. Клацнути на будь-якій клітинці таблиці даних.
2. Застосувати режим автофільтра одним із способів.

1 спосіб

Перейти на вкладку **Основне** та в меню інструмента **Сортувати й фільтрувати** обрати команду **Фільтр**.

2 спосіб

Перейти на вкладку **Дані** та обрати інструмент **Фільтр**.

В результаті у верхньому рядку з'являться стрілки, з допомогою яких можна виконати фільтрацію (рис. 20.2). Для зняття фільтра необхідно знову обрати цей же інструмент.

	A	B	E	F
1	Назва	Автор	Рік випуску	Кількість сторінок
4	Гра в паралельне читання	Іванцова М.	2013	288
7	Кіт на ім'я Сплет	Скоттон Р.	2013	36
8	Ліга непарних шкарпеток	Вдовиченко Г.	2013	128

Рис. 20.2

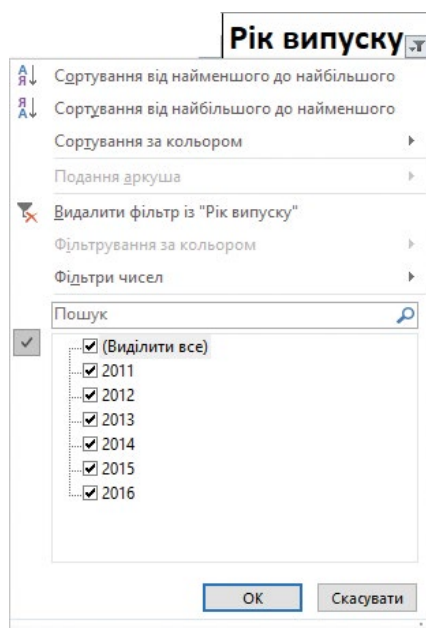
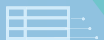


Рис. 20.3

3. Клацнути на кнопці з трикутником у стовпці, який треба відфільтрувати, наприклад, **Рік випуску**.

4. Зі списку, що відкриється, обрати умову фільтрації.

Список фільтрації містить усі дані стовпця (без повторень), команди сортування і команду переходу до текстових або числових фільтрів залежно від даних стовпця (рис. 20.3).

Для вибору одного або декількох значень із запропонованих даних списку необхідно залишити навпроти них прапорці, а інші забрати. Після натиснення кнопки **ОК** буде виведено лише вибрані значення. Якщо натиснути на прапорець **(Виділити все)**, буде знято всі прапорці. Щоб видалити заданий фільтр, необхідно натиснути інструмент **Очистити**.

Для встановлення умов фільтрації використовують команди **Текстові фільтри**, **Фільтри чисел** або **Фільтри дат** залежно від типів даних.

Розглянемо використання автофільтра, виконавши практичну вправу.

ВПРАВА 20.1



Завдання. Вивести дані, що задовольняють указані умови:

- а) вивести усі книжки, в яких кількість сторінок від 100 до 200;
- б) вивести усі книжки, назва яких починається на літеру «п»;
- в) вивести усі книжки, в яких будь-яке слово у назві починається на «п»;
- г) знайти усі книжки видавництва «Богдан», що були випущені після 2014 року.

1. Завантажте файл **Вправа_20_1.xlsx** з матеріалів до даного параграфа.
2. Створіть три копії наявного в книзі аркуша та надайте їм назви **Завдання_а**, **Завдання_б**, **Завдання_в** та **Завдання_г** відповідно.
3. На аркуші **Завдання_а** застосуйте автофільтр для виведення усіх книжок, у яких кількість сторінок від 100 до 200.
 - а. Клацніть на будь-якій клітинці таблиці даних.
 - б. Оберіть команду **Фільтр** на вкладці **Основне** в меню інструмента **Сортувати й фільтрувати**.
 - с. Клацніть стрілку біля стовпця **Кількість сторінок**.



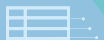
- d. Оберіть команду **Фільтри чисел** ▶ **Між**. У результаті відкриється вікно **Користувацький автофільтр**.
- e. Навпроти поля **Більше або дорівнює** введіть 100, а навпроти поля **Менше або дорівнює** — 200.
- f. Натисніть **ОК**.

При фільтруванні текстових даних зручно використовувати символи «?» і «*», що дозволяють здійснювати пошук за неповними даними. Тут знак «?» визначає один будь-який символ, а «*» — будь-яку кількість довільних символів. Так, запис «a*» буде означати, що необхідно вивести всі дані, що починаються на літеру «a».

Запис «*a*» означає будь-який текст, що містить в собі «a», а запис «?a*» означає будь-який текст, у якому другою є літера «a», оскільки знак питання означає лише будь-який один символ.

4. На аркуші **Завдання_б** знайдіть усі книжки, назви яких починаються на літеру «п».
 - a. Застосуйте до таблиці автофільтр.
 - b. Оберіть стрілку у стовпці **Назва**.
 - c. Оберіть команду **Текстовий фільтр** ▶ **Починається з**.
 - d. Введіть літеру «п» та застосуйте фільтр.
5. На аркуші **Завдання_в** знайдіть усі назви, в яких будь-яке слово починається на «п».
 - a. Застосуйте до таблиці автофільтр.
 - b. Оберіть стрілку у стовпці **Назва**.
 - c. Оберіть команду **Текстовий фільтр** ▶ **Користувацький фільтр**.
 - d. У першому списку оберіть опцію **починається з** та встановіть значення «п».
 - e. Установіть перемикач у положення **АБО**.
 - f. У другому списку оберіть опцію **містить** та встановіть значення «* п*» (рис. 20.4). Це означає, що текст має містити будь-який текст, пробіл, літеру «п» та ще будь-який текст.

Рис. 20.4



- g. Застосуйте фільтр, натиснувши **ОК**.
- 6. На аркуші **Завдання_г** знайдіть книжки видавництва «Богдан», що були випущені після 2014 року.
 - a. Застосуйте до таблиці автофільтр.
 - b. Оберіть стрілку у стовпці **Видавництво**.
 - c. Залиште у списку прапорець лише біля видавництва «Богдан».
 - d. Оберіть стрілку у стовпці **Рік видання**.
 - e. Оберіть команду **Фільтри чисел** ▶ **Більше**.
 - f. Уведіть 2014.
 - g. Застосуйте фільтр, натиснувши **ОК**.

20.2.

Розширений фільтр

Якщо значення деякого параметра має задовольняти умову, що складається більш ніж із двох частин, або частини умови, які стосуються різних параметрів, з'єднані сполучником «або», використовують розширені фільтри. Так, на рис. 20.5 наведено результат фільтрації, коли необхідно знайти книжки, автором яких є Дара Корній, або ті, що були випущені після 2014 року.

Назва	Автор	Рік випуску	Кількість сторінок
Володар Драконів	Функе К.	2015	512
Зворотний бік світла	Дара Корній	2012	320
Мій тато став зіркою	Кирпа Г.	2015	38
П'ятеро дітей та ельф	Несбіт Е.	2015	280
Фанат футболу	Тібо Ж.	2016	32

Рис. 20.5

Розглянемо використання розширених фільтрів, виконавши практичну вправу.

ВПРАВА 20.2



Завдання. Знайти книжки, автором яких є Дара Корній, або ті, що були випущені після 2014 року.

	Н	І
1	Автор	Рік випуску
2	Дара Корній	
3		>2014

Рис. 20.6

1. Відкрийте файл **Вправа_20_2.xlsx** з матеріалів до даного параграфа.

Для використання розширеного фільтру спочатку необхідно створити критерій відбору даних, указавши назви стовпців і значення пошуку.



2. Створіть таблицю з критеріями відповідно до зразка (рис. 20.6).
 - a. Скопіюйте з основної таблиці назви стовпців **Автор** та **Рік видання**, оскільки саме за цими стовпцями має здійснюватися фільтрація.
 - b. Під назвою стовпця **Автор** скопіюйте з таблиці ім'я автора **Дара Корній**.
 - c. Нижче у стовпці **Рік видання** введіть умову **>2014**.

Умова щодо року випуску вказується нижче, оскільки відповідно до завдання ми маємо знайти книжки, що мають указанного автора **або** рік видання після 2014. Якби умова була Дара Корній **та** рік видання після 2014, то умову щодо року було б вказано в цьому ж самому рядку, що й автор.

3. Перейдіть до вкладки **Дані** та оберіть інструмент **Додатково**. В результаті чого має з'явитися діалогове вікно **Розширений фільтр**.
4. У полі **Вихідний діапазон** установіть діапазон, у якому розташована таблиця з даними. Як правило, програма її може визначити самостійно за умовчанням. В даному випадку це діапазон **A1:F24**.
5. У полі **Діапазон умов** укажіть діапазон з критерієм фільтра, а саме, **I1:J3**.
6. Установіть опцію біля перемикача **скопіювати результат до іншого розташування**.
7. У полі **Діапазон для результату** встановіть **A26:F26**, що є першим рядком, починаючи з якого буде вставлено результат фільтрації.
8. Натисніть **ОК**. В результаті мають бути відображені усі дані, що задовольняють критерій відбору.



Дізнайтеся більше про «розумні» таблиці.



Виконайте вправу 20.3.



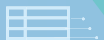
Контрольні запитання та завдання

1. Для чого використовують фільтрацію даних?
2. Як застосувати автофільтр?



Питання для роздумів

- 1*. У яких випадках використовують автофільтр, а в яких — розширений фільтр?
- 2*. Чи можна замість автофільтра скористатися можливостями розширеного фільтра?



Завдання для досліджень



- 1*. Визначте, яким буде критерій для розширеного фільтра, якщо у вправі 20.2 потрібно вивести всі книжки, що мають більше 100 сторінок та видані у видавництві «Богдан» або «Ранок».
- 2*. Дослідіть і поясніть призначення інструментів групи **Знаряддя даних** на вкладці **Дані**.



ПРАКТИЧНА РОБОТА 5.

«УПОРЯДКУВАННЯ ТА ФІЛЬТРАЦІЯ ДАНИХ У ТАБЛИЦЯХ»



Завдання. Використовуючи таблицю з інформацією про прокат мультфільмів у кінотеатрах, вивести дані, застосувавши фільтрацію та проміжні підсумки.

1. Завантажте файл **ПР_5.xlsx** з матеріалів до даного параграфа.
2. Виконайте копіювання даних таблиці ще на сім аркушів.
3. На першому аркуші здійсніть сортування в алфавітному порядку спочатку за назвою мультфільму, а потім — у зворотному порядку за найменшою ціною.
4. На другому аркуші застосуйте до таблиці проміжні підсумки, щоб підрахувати суму касового збору та кількість проданих квитків на кожний мультфільм.

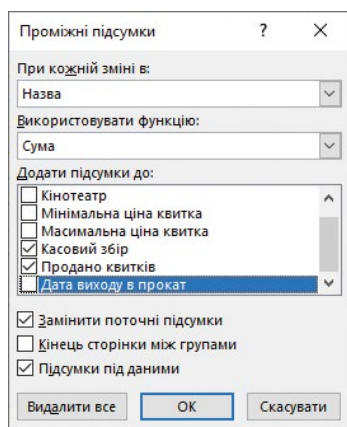


Рис. 21.1

- a. Виділіть стовпець **Назва** та здійсніть сортування його даних в алфавітному порядку.
- b. На вкладці **Дані** оберіть інструмент **Проміжні підсумки**.
- c. Установіть необхідні параметри у вікні **Проміжні підсумки** (рис. 21.1) та натисніть **ОК**.
5. На третьому аркуші визначте мінімальну та максимальну ціну квитка на кожний мультфільм, використовуючи проміжні підсумки.
 - a. Виконайте сортування таблиці за назвою мультфільмів.
 - b. Застосуйте проміжні підсумки за стовпцем **Мінімальна ціна квитка**, визначаючи найменше значення.
 - c. У цій же таблиці застосуйте проміжні підсумки за стовпцем **Максимальна ціна квитка**, визначаючи найбільше значення. Для додавання значень



максимальної ціни квитка в діалоговому вікні **Проміжні підсумки** приберіть прапорець з опції **Замінити поточні підсумки**.

- 6*. На четвертому аркуші визначте суму касового збору та кількість проданих квитків у кожному кінотеатрі, використовуючи проміжні підсумки.
7. На п'ятому аркуші визначте всі мультфільми, які було показано в кінотеатрах «Старт» і «Салют», застосувавши автофільтр.
8. На шостому аркуші виведіть дані про мультфільми, ціна квитків на які становить від 40 грн до 100 грн включно.
9. На сьомому аркуші виведіть відомості про всі мультфільми кінотеатру «Факел» або ті мультфільми, на які було продано більше 1100 квитків.
 - a. У діапазоні **I1:J3** створіть критерій відбору, як на рис. 21.2.
 - b. Застосуйте до даних таблиці розширений фільтр з урахуванням створеного критерію, розташувавши результати фільтрації під таблицею з даними.
- 10*. На восьмому аркуші виведіть інформацію про мультфільми кінотеатру «Boomer», що мають касовий збір менше 20000 грн або на них було продано менше 900 квитків.

Кінотеатр	Продано квитків
Факел	
	>1100

Рис. 21.2



РОЗВ'ЯЗУВАННЯ ЗАДАЧ З МАТЕМАТИКИ ЗАСОБАМИ ТАБЛИЧНОГО ПРОЦЕСОРА

На сьогоднішній день електронні таблиці широко використовуються в різних професійних галузях. Вони можуть стати у пригоді і при розв'язуванні завдань з різних навчальних предметів. У даному параграфі розглянемо використання електронних таблиць для розв'язування математичних задач.

ЗАДАЧА 1



Розв'язати систему рівнянь графічним способом (рис. 22.1).

$$\begin{cases} y = ax^2 + bx + c, \\ y = kx + d, \end{cases}$$

де a, b, c, k і d — довільні числові коефіцієнти. При зміні коефіцієнтів мають змінюватися корені системи рівнянь та зовнішній вигляд графіків функцій.

1. Розташуйте коефіцієнти рівнянь системи у клітинках програми MS Excel так, як це представлено на рис. 22.2.
2. Створіть таблиці даних для побудови графіків. Для змінної x установіть значення від -5 до 5 з кроком $0,5$.
 - a. До клітинки **A6** уведіть значення -5 .

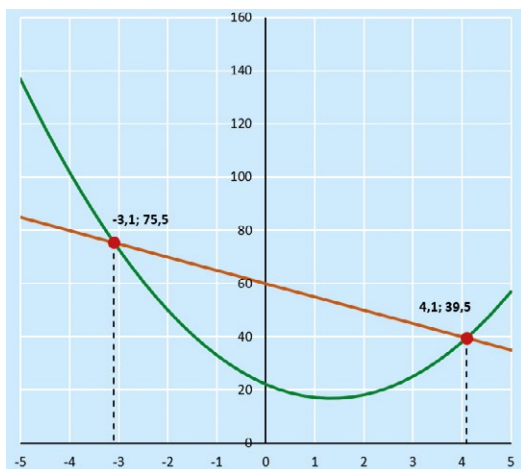
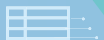


Рис. 22.1

	A	B	C	D	E	F
1	$y=ax^2+bx+c$				$y=kx+d$	
2	a	b	c		k	d
3	3	-8	22		-5	60

Рис. 22.2

- b. На вкладці **Основне** оберіть в меню інструмента **Заповнити** команду **Прогресія**.
 - c. У вікні, що відкриється, встановіть **За стовпцями, крок — 0,5, граничне значення — 5**.
 - d. Натисніть **ОК**.
 3. Для змінної у введіть формулу: **=A\$3*A6^2+B\$3*A6+C\$3** до клітинки **B6**, та виконайте автозаповнення для її копіювання. Посилання на клітинки **A3**, **B3** та **C3** фіксуємо, оскільки вони є сталими.

В результаті дані для побудови першого графіка створено. Графіком другої функції буде пряма, отже, для її побудови достатньо всього дві точки. Для x доцільно обрати максимальне та мінімальне значення змінної x , що містить таблиця даних для побудови першого графіка, а значення змінної y обчислити за формулою.

4. До клітинки **E6** уведіть «-5», а до клітинки **E7** — «5».
 5. У клітинці **F6** уведіть формулу **=E\$3*E6+F\$3**. Тут також зафіксовано посилання на клітинки **E3** та **F3**, оскільки значення коефіцієнтів рівняння є сталими. Застосуйте до формули автозаповнення.
 6. Побудуйте графік першої функції.
 - a. Виділіть діапазон з даними.
 - b. На вкладці **Вставлення** в меню інструмента точкової діаграми оберіть **Точкову діаграму з гладкими лініями**.
 7. Додайте до діаграми графік другої функції.
 - a. На вкладці **Конструктор** оберіть інструмент **Вибрати дані**.
 - b. Натисніть кнопку **Додати**. В результаті з'явиться вікно **Редагування ряду**.
 - c. В даному вікні у полі **значення ряду x** оберіть діапазон **E6:E7**.



- d. В полі **значення ряду у** оберіть діапазон **F6:F7**.
- e. Натисніть **ОК** два рази.

Використовуючи побудовану діаграму, можна вже визначити корені даної системи рівнянь. Це будуть координати точок, що розташовані на перетині побудованих графіків функцій. Для візуалізації даних так, як це представлено на рис. 22.1, необхідно виконати ще додаткові розрахунки. Так, точки, якими позначено перетин діаграм, є маркерами, що містять координати, значення яких є розв'язками системи рівнянь. Отже, для знаходження значень координат цих точок ми маємо розв'язати систему рівнянь. А потім за цими даними побудувати ще один ряд даних. Пунктирні лінії, що з'єднують червоні маркери з віссю x — це також графіки, для яких потрібно спочатку створити таблиці даних, а потім за цими таблицями побудувати графіки.

Розв'яжемо систему рівнянь.

$$\begin{cases} y = ax^2 + bx + c, \\ y = kx + d. \end{cases}$$

Оскільки обидві праві частини дорівнюють y , прирівняємо їх:

$$ax^2 + bx + c = kx + d.$$

Перенесемо усі доданки з правої частини рівняння в ліву:

$$ax^2 + bx + c - kx - d = 0.$$

Зведемо подібні доданки.

$$ax^2 + (b - k)x + c - d = 0.$$

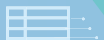
У результаті отримаємо звичайне квадратне рівняння. Розв'яжемо його в середовищі MS Excel. Щоб не плутати назви коефіцієнтів даного квадратного рівняння з коефіцієнтами першого рівняння системи, надамо їм імена $a1$, $b1$, $c1$, де $a1 = a$, $b1 = b - k$, $c1 = c - d$.

- 8. До клітинки **E9** уведіть формулу **=A3** (клітинка має містити значення коефіцієнта $a1$).
- 9. До клітинки **E10** уведіть формулу **=B3-E3** (клітинка має містити значення коефіцієнта $b1$).
- 10. До клітинки **E11** уведіть формулу **=C3-F3** (клітинка має містити значення коефіцієнта $c1$).
- 11. У клітинці **E12** обчисліть дискримінант квадратного рівняння. Для цього введіть формулу **=E10^2-4*E9*E11**.
- 12. У клітинці **D14** уведіть формулу знаходження першого кореня рівняння: **=IF(E12<0;NA();(-E10+SQRT(E12))/(2*E9))**.

В даній формулі враховано, що якщо дискримінант буде від'ємним, то значення клітинки буде мати помилку, що означає — значення не доступне. За умови відображення такої помилки графік будуватися не буде.

	D	E
8	$ax^2 + bx + c = kx + d$	
9	a1	3
10	b1	-3
11	c1	-38
12	D	465
13	x	y
14	4,1	39,5
15	-3,1	75,5

Рис. 22.3



13. У клітинці **D15** уведіть формулу знаходження другого кореня рівняння самостійно.
14. Знайдіть за даними значеннями x відповідні значення y , підставляючи їх у другу функцію. Отже, введіть до клітинки **E14** формулу **=E\$3*D14+\$F\$3**, тут фіксуйте посилання на клітинки **E3** та **F3**, оскільки вони є сталими.
15. Скопіюйте формулу нижче, застосувавши автозаповнення.
16. Зменшіть розрядність одержаних координат точок до одного знака після коми з допомогою інструменту **Зменшити розрядність**.
17. На основі одержаної таблиці даних (рис. 22.3) додайте ряд даних для побудови маркерів.
 - a. Виділіть діаграму на вкладці **Конструктор**.
 - b. Оберіть інструмент **Вибрати дані**.
 - c. Натисніть кнопку **Додати**.
 - d. У полі **значення ряду x** оберіть діапазон **D14:D15**.
 - e. У полі **значення ряду y** оберіть діапазон **E14:E15**.
 - f. Натисніть два рази **ОК**. В результаті ви одержите лінію, що з'єднує вказані точки.
18. Змініть тип діаграми. Для цього оберіть інструмент **Змінити тип діаграми** та встановіть тип **Точкова**, що містить лише маркери.
19. Змініть формат маркерів.
 - a. В контекстному меню маркерів оберіть команду **Формат ряду даних**.
 - b. Оберіть інструмент **Заливка й контур**.
 - c. Оберіть **Маркер**.
 - d. Установіть червоний колір заливки маркера та відсутність контуру маркера.
20. Додайте значення координат маркерів.
 - a. Виділіть маркери.
 - b. На вкладці **Конструктор** в меню інструменту **Додати елемент діаграми** оберіть команду **Підписи даних** ▶ **Над**.
 - c. Для редагування відображених даних в контекстному меню даних оберіть команду **Формат підписів даних**.
 - d. Залиште прапорці біля опцій **Значення x** та **Значення y** і залиште в якості роздільника крапку з комою. При необхідності перемістіть дані для зручного відображення.
- 21*. Додайте пунктирні лінії, що з'єднують червоні маркери з віссю x .
22. Виконайте остаточне форматування побудованої діаграми.

ЗАДАЧА 2



Дано координати вершин трикутника (значення кожної координати належить проміжку від -10 до 10). Побудувати трикутник за його координатами (рис. 22.4), знайти довжини сторін трикутника, периметр та площу.

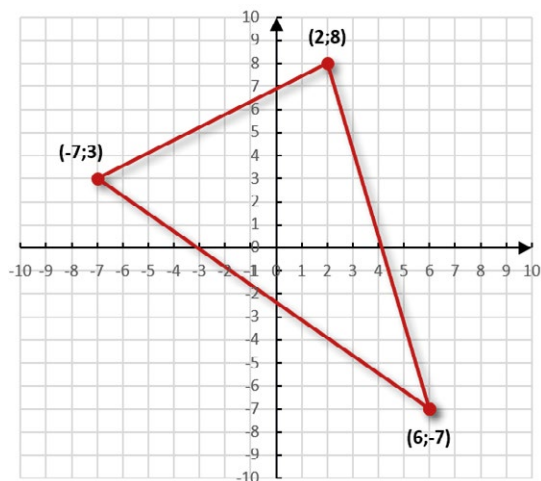


Рис. 22.4

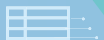
	A	B	C
1		x	y
2	1	6	-7
3	2	-7	3
4	3	2	8
5	1	6	-7

Рис. 22.5

1. Задані координати вершин трикутника розташуйте у клітинках програми MS Excel, як на рис. 22.5.

Зверніть увагу, що в даній таблиці перша координата повторюється. Це потрібно для того, щоб при побудові точкової діаграми відбувалося з'єднання третьої координати з першою. Отже, якщо до діапазону **B2:C5** користувач вводить значення координат самостійно, то у клітинках **B6**, **C6** необхідно ввести формули.

2. До клітинки **B6** уведіть формулу **=B2**, а до клітинки **C6**, відповідно, формулу **=C2**.
3. Побудуйте трикутник.
 - a. Виділіть діапазон **B2:C5**.
 - b. На вкладці **Вставлення** в меню інструмента точкової діаграми оберіть **Точкова діаграма з прямими лініями та маркерами**.
4. Встановіть для осі *x* та *y* мінімальне значення -10, максимальне 10.
 - a. У контекстному меню осі оберіть **Формат осі...**
 - b. У вікні, що відкриється, встановіть необхідні значення.
5. Сформууйте підписи маркерів, що містять координати.
 - a. У клітинці **D1** запишіть його назву **Координати**.
 - b. У клітинці **D2** введіть формулу: **=CONCAT("(",B2;"",C2;"))** та застосуйте до неї автозаповнення для створення підписів іншим координатам.
 - c. В контекстному меню маркерів оберіть команду **Додати підписи даних**.
 - d. Для редагування маркерів в контекстному меню маркерів оберіть **Формат підписів даних**.
 - e. Установіть прапорець біля опції **Значення із клітинок**.
 - f. У вікні, що з'явилося, встановіть діапазон **D2:D4**.



- g. Натисніть **ОК**.
- h. У вікні **Формат підписів даних** заберіть зайві прапорці.
- i. Для зручного відображення створених підписів даних перемістіть їх у бажане місце.



Розв'яжіть задачі 3 і 4.

- 6. Подальше форматування діаграми побудови трикутника ви маєте виконати самостійно.
- 7. Самостійно виконайте обчислення довжин сторін, периметра та площі трикутника у відповідних клітинках за наведеними нижче формулами.

	A	B
7	a	16
8	b	10
9	c	16
10	P	42
11	S	77,5

$$a = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$b = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}$$

$$a = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2}$$

$$P = a + b + c$$

$$S = \sqrt{p(p-a)(p-b)(p-c)}, \text{ де } p = \frac{P}{2}$$

**Висновки**

Залежно від того, яку частину адреси зафіксовано та чи зафіксовано її взагалі, розрізняють **відносні**, **абсолютні** та **змішані** посилання на адреси клітинок.

Щоб виконати математичні обчислення, використовують функції з категорії **Математичні**, для здійснення статистичного аналізу даних використовують функції з категорії **Статистичні**, для виконання завдань з урахуванням умов використовують функції з категорії **Логічні**.

У MS Excel можна побудувати такі основні типи діаграм: стовпчасті, секторні, лінійчаті, точкові, гістограми, діаграми з областями.

Для розташування даних таблиці в заданому порядку використовують **сортування**.

Для автоматичних розрахунків спільних підсумків декількох рядків взаємопов'язаних даних використовують **проміжні підсумки**.

Для пошуку даних за деякою умовою використовують **фільтрацію даних**. Для створення простих умов відбору використовують **автофільтр**, а для більш складних — **розширений фільтр**.



Розділ 3

Бази даних.

Системи керування базами даних



§23.

ПОНЯТТЯ ТА ПРИЗНАЧЕННЯ БАЗ ДАНИХ



23.1.

Поняття бази даних, системи керування базами даних

Раніше ви навчилися працювати із масивами даних у середовищі електронних таблиць. Проте для зберігання та опрацювання декількох пов'язаних між собою масивів даних зручніше використовувати технології баз даних, про які й піде мова далі.

Давайте розберемося, що ж таке база даних.



База даних — це структурована сукупність даних, які відображають стан об'єктів певної предметної області та зв'язки між ними.

Предметною областю є сфера застосування бази даних. Так до предметної області можна віднести лікарню, залізницю, навчальний заклад тощо.

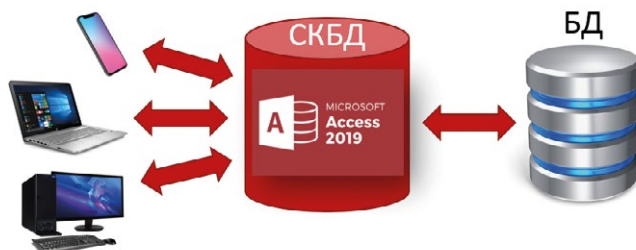
Для опрацювання баз даних з допомогою комп'ютера використовують спеціальні програми, які називають *системами керування базами даних* (скорочено СКБД).

Система керування базами даних — це програма, що забезпечує можливість створення бази даних та виконання різноманітних операцій з даними, які в ній зберігаються.

В якості системи керування базами даних ми з вами будемо знайомитися з програмою Access, що входить до пакета Microsoft Office.



Використання даної програми розглянемо на прикладі версії 2019 року.



До функцій систем керування базами даних належать:

- визначення структури даних;
- маніпулювання даними, а саме пошук, додавання, редагування, видалення тощо;
- збереження даних.

23.2.

Елементи таблиці бази даних

У програмі Access дані представляють у вигляді таблиці.

Стовпці таблиці називають *полями*. Поля в таблицях містять однотипні дані. Так, у представленому на рис. 23.1 прикладі поле **Найменування** має текстовий тип, поля **Ціна, грн** та **Кількість** — числовий тип, поле — **Дата постачання** також числовий, але у форматі дати.

Рядки, що містять інформацію про деякий один об'єкт, називають *записами*.

Записи	Поля			
	Найменування	Ціна, грн	Дата постачання	Кількість
	Стіл письмовий	2000	01.08.2021	3
	Шафа	5000	02.09.2021	5
	Стілець	1200	03.10.2021	12

Рис. 23.1

Розглянемо вимоги до полів та записів таблиці.

1. Кількість полів визначається розробником. Тобто, перед тим, як уводити дані до таблиці, потрібно створити її структуру, отже, розробник створює структуру таблиці, і користувач під час роботи з даними не може змінювати структуру таблиці.



2. Поля повинні мати унікальне ім'я. Тобто у таблиці не можуть бути два поля з однаковими іменами.
3. Поля можуть бути обов'язковими для заповнення або ні. Наприклад, якщо таблиця містить поле **Знижка** для переліку товарів, то для деяких товарів знижка використовується, а деякі продаються без знижки, отже, ця клітинка поля **Знижка** буде порожньою.
4. Таблиці можуть містити скільки завгодно записів, їх кількість обмежується тільки обсягом диска.
5. Поля можуть мати різний формат — це може бути текст, числа, зображення тощо.

Щоб кожен запис бази даних можна було однозначно ідентифікувати, використовують так звані *ключові поля*.

Ключове поле — це поле (або комбінація полів), яке однозначно визначає запис.

6. У таблиці не може бути двох записів з однаковими значеннями ключа.

При створенні структури таблиці бази даних часто в якості ключового поля створюють поле **id** або **код**, що є ідентифікатором запису — у цьому полі кожному запису надається певний унікальний номер.

ВПРАВА 23.1



Завдання. Дано таблицю клієнтів деякого банку, що має поля **Прізвище**, **Ім'я**, **Ідентифікаційний код**, **Номер будинку**, **Реєстраційний номер автомобіля**, **Населений пункт**, **Дата останньої транзакції**, **Номер особового рахунку**. Визначити, яке поле можна вказати в якості ключового.

Поле **Прізвище**, звичайно не може бути ключовим, оскільки серед клієнтів, напевно, є клієнти з однаковими прізвищами. Це ж саме стосується й поля **Ім'я**.

Що стосується поля **Ідентифікаційний код**, то цей код є унікальним для кожної людини, отже, його можна встановити в якості ключового.

Чи може бути ключовим полем номер будинку, в якому проживає клієнт банку?

Звісно ні — номери будинків можуть повторюватися.

А чи можна вказати в якості ключа реєстраційний номер автомобіля? Номер автомобіля є унікальним для кожного автомобіля, отже, на перший погляд це поле можна використати в якості ключа. Але давайте повернемося до умови завдання. Потрібно визначити поле, що може бути ключем для таблиці



Клієнти банку. Для цієї таблиці поле **Реєстраційний номер автомобіля** ключем бути не може, адже не усі клієнти мають автомобіль, а за деякими може бути зареєстровано більше одного автомобіля.

Поле **Населений пункт** ключовим бути не може, оскільки в одному населеному пункті може проживати багато клієнтів.

Чи може бути ключовим поле **Дата останньої транзакції**?

Звісно, ні, оскільки за один день клієнт може виконати більше однієї транзакції.

Ще одне поле, яке може викликати сумніви, — це **Номер особового рахунку**. Оскільки в одному банку їх можна відкрити декілька, то ключовим таке поле бути не може.

Отже, ми визначили, що із запропонованого переліку полів ключовим може бути лише поле **Ідентифікаційний код**.

ВПРАВА 23.2



Завдання. Дано таблицю **Автомобілі**, що містить поля **Колір, Марка, Рік випуску, Пробіг, VIN-код, Тип кузова, Модель, Державний номер**.

Визначити, яке поле можна вказати в якості ключового.

Як вже було зазначено раніше, в базі даних може бути більше ніж одна таблиця, але, на відміну від Excel, ці таблиці пов'язані між собою. Такий вид представлення баз даних називають *реляційні бази даних*.

Реляційні бази даних — це набір простих таблиць, між якими встановлені зв'язки.

Так, на рис. 23.2 представлено приклад схеми реляційної бази даних предметної області **Магазин**. З прикладу видно, що база даних

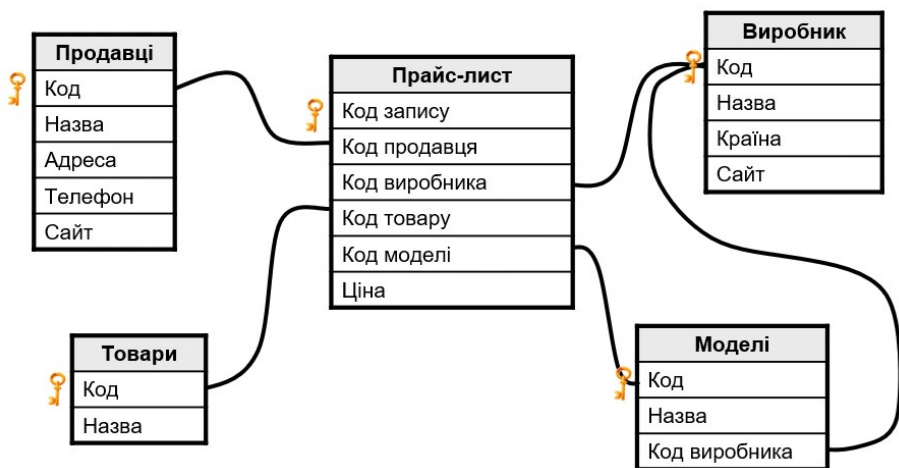


Рис. 23.2. Приклад реляційної бази даних



містить таблицю **Прайс-лист**, що пов'язана з такими таблицями, як **Продавці**, **Товари**, **Виробник** та **Моделі**.

Реляційні бази даних мають велике поширення у світі, отже, саме з їхніми можливостями ми і будемо знайомитися на уроках інформатики.

Крім реляційної бази даних, існують й інші — це *ієрархічні*, *мережеві*, що є вже застарілими, та *об'єктно-орієнтовані* сучасні бази даних. На рис. 23.3, а) наведено приклад структури ієрархічної бази даних. У ній один об'єкт є основним, а інші — підлеглі. На рис. 23.3, б) представлено приклад структури мережевої бази даних, де об'єкти можуть мати значно більше зв'язків, ніж в ієрархічній.

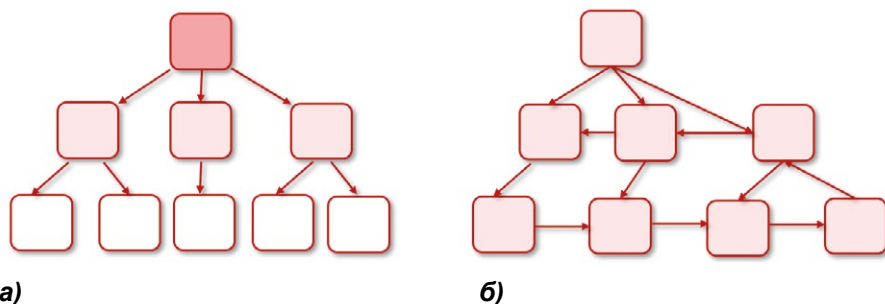


Рис. 23.3. Приклади структур: а) ієрархічної та б) мережевої баз даних

Як уже було зазначено раніше, база даних містить відомості з певної предметної області. Отже, при створенні бази даних розробник має визначити, які відомості буде містити база даних, до яких таблиць їх доцільно додати та яким чином пов'язати ці таблиці між собою. Спробуємо визначити назви таблиць для предметної області, виконавши вправу.

ВПРАВА 23.3



Завдання. Визначити, які таблиці доцільно було б створити для такої предметної області, як **Ліцей**.

Звичайно, ця база даних має містити відомості про ліцеїстів та їхніх учителів, оскільки вони мають містити дещо різну інформацію. Їх доцільно буде розташувати в окремі таблиці **Ліцеїсти** та **Учителі**.

Доцільним буде також створити таблицю **Класи**, що є окремим об'єктом бази даних, пов'язати між собою дані, що мають містити ці таблиці, буде зручно з використанням таблиці **Розклад**.

Це лише невеликий перелік таблиць, що може містити дана предметна область, тут можна додати на власний розсуд значно більше таблиць, що зможуть розширити інформацію про різні об'єкти ліцею — це можуть бути, наприклад, таблиці **Батьки**, **Олімпіади**, **Гуртки**, **Кабінети** тощо.



Самостійно визначте, які ще таблиці можна було б додати до предметної області **Ліцей**.

ВПРАВА 23.4



Завдання. Визначити, які таблиці доцільно було б створити для такої предметної області, як **Залізниця**.

Після визначення переліку таблиць визначають, які дані слід занести до кожної таблиці. Визначимо це у наступній вправі.

ВПРАВА 23.5



Завдання. Визначити перелік полів, що має містити таблиця **Ліцеїсти**.

Це мають бути поля **Прізвище** та **Ім'я**, доцільно також додати поля **Дата народження** та **Стать**. Також доцільним є наявність поля **Клас**, у якому навчається ліцеїст.

Самостійно визначте, які ще можна додати поля до таблиці **Ліцеїсти**.

ВПРАВА 23.6



Завдання. Визначити перелік полів, що має містити таблиця **Поїзд**.



Контрольні запитання та завдання

1. Що таке база даних?
2. Що таке предметна область? Наведіть приклади предметних областей.
3. Що таке система керування базами даних? Які її функції?
4. Які існують види баз даних?
5. Як називаються елементи таблиці в базі даних?
6. Що таке ключове поле?



Питання для роздумів

- 1*. Чому реляційні бази даних мають найбільше розповсюдження?
- 2*. Чи може таблиця бази даних мати в якості ключа декілька полів.



Завдання для досліджень

- 1*. Підготуйте доповідь про історію появи баз даних.
- 2*. Дізнайтеся, в яких галузях використовують системи керування базами даних.



§24.

СТВОРЕННЯ ТАБЛИЦЬ

24.1.

Типи полів

У попередньому параграфі ми познайомилися з основними поняттями бази даних. Визначили, що дані в них зберігаються у вигляді таблиць. У даному параграфі познайомимося з середовищем Access, навчимося створювати та редагувати у ньому таблиці.

Розглянемо створення бази даних предметної області **Книжковий інтернет-магазин**. Отже, створимо базу даних **Інтернет-магазин**, а в ній створимо поки що одну таблицю **Книжки**. Зовнішній вигляд майбутньої таблиці представлено на рис. 24.1.

Код	Назва	Автор	Видавництво	Рік видання	Кількість сторінок	Ціна	Назва	Кількість	Обкладинка	Жанр	Анотація
1	Останній лист	Віктор Янкевич	Богдан	2002	224	99,50 грн.	<input checked="" type="checkbox"/>	3	Ⓜ(1)	роман	Вічна згадка
2	Чорний павук	Гольцфельд Євгенія	Богдан	2019	128	149,00 грн.	<input checked="" type="checkbox"/>	5	Ⓜ(1)	повість	Ця місто
3	Пророкиця	Кокотюха Андрій	Богдан	2016	272	179,00 грн.	<input checked="" type="checkbox"/>	35	Ⓜ(1)	детектив	Поліцей
4	Моя сім'я та інші	Дарелл Джеральд	Богдан	2017	328	199,00 грн.	<input checked="" type="checkbox"/>	100	Ⓜ(1)	повість	Перша і
5	Птахи, звірі та	Дарелл Джеральд	Богдан	2018	320	199,00 грн.	<input type="checkbox"/>		Ⓜ(1)	повість	Друга ч
6	Пастка на полюс	Кокотюха Андрій	Богдан	2018	280	179,00 грн.	<input checked="" type="checkbox"/>	3	Ⓜ(1)	детектив	Темної
7	Вірус	Гелен Фіцджеральд	Фабула	2017	320	150,00 грн.	<input type="checkbox"/>		Ⓜ(1)	детектив	Су завж
8	Чужа	Емі Джентрі	Фабула	2016	320	170,00 грн.	<input checked="" type="checkbox"/>	7	Ⓜ(1)	детектив	«Чужа»
9	Висотка	Баллард Дж. Г.	Фабула	2015	224	150,00 грн.	<input type="checkbox"/>		Ⓜ(1)	фантастика	Джейм
10	Демельза	Вінстон Грем	Фабула	2017	544	200,00 грн.	<input type="checkbox"/>		Ⓜ(1)	історичний роман	«Демел

Рис. 24.1

Перед тим, як створювати таблицю в Access, ми маємо проаналізувати, які дані вона буде містити. З даної таблиці видно, що вона містить поля різних типів. У таких полях, як **Назва**, **Автор**, **Видавництво**, **Жанр** та **Анотація**, представлено текстові дані. Числові дані містять поля **Код**, **Рік видання**, **Кількість сторінок**, **Ціна** та **Кількість**. Розглянемо які типи даних можуть мати поля в таблиці Access. Як вже було зазначено раніше, поля можуть мати різні типи текстовий, числовий, зображення тощо.

За умовчанням для усіх полів в Access встановлюється текстовий тип, а саме, *короткий текст* (короткий текст містить до 255 символів). Також дані можуть бути представлені у вигляді *довгого тексту* (довгий текст містить більше ніж 255 символів).

Дані можуть мати *числовий тип* (числовий тип представлено у вигляді цілих або дробових чисел). Як окремий числовий формат є формат *грошова одиниця*. Ще один числовий формат — це *дата й час*. Одним із зручних числових форматів є *автономерація* — цей формат використовується, коли програма автоматично має



встановити номер запису і користувач не повинен вводити число са-
мостійно.

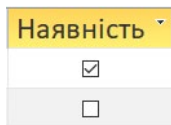


Рис. 24.2.

Ще один тип даних — це тип **так/ні** (це поле є логічним, і в ньому користувачу достатньо встановити прапорець, наявність якого визначає значення **так**, відсутність — **ні**) (рис. 24.2).

Для додавання зображень або інших об'єктів використовують такі типи полів, як *вкладення* або об'єкт *OLE*. Відмінність їх в тому, що з допомогою вкладення можна додати кілька зображень або об'єктів, а з допомогою об'єкта OLE — лише одне зображення або об'єкт.

Ще один тип — *гіперпосилання*. Незавжди здогадатися, що дані цього типу мають містити посилання на відкриття файлів, вебсторінок тощо.

Розглянемо алгоритм створення бази даних.

1. У меню кнопки **Пуск** обрати програму **Access**.
2. У вікні, що відкриється, обрати **Пуста база даних**.
3. Натиснути піктограму у вигляді папки для вибору місця збереження бази даних.
4. Обрати папку та ввести ім'я бази даних, наприклад, **Інтернет-магазин**.
5. Натиснути **ОК**.
6. Натиснути кнопку **Створити**. В результаті базу даних буде створено.

За умовчанням у базі даних створюється нова таблиця. Для подальшого самостійного створення таблиць бази даних слід закрити створену за умовчанням вкладку.

24.2.

Інтерфейс програми Access

Познайомимося з вікном програми Access (рис. 24.3). Інтерфейс програми має спільні елементи з іншими офісними додатками Microsoft. Так, верхня частина Access містить вкладки з інструментами для роботи в середовищі. Знизу розташовано рядок стану. Ліворуч — перелік об'єктів, наявних у програмі. Так, окрім таблиць, що містять дані в Access, використовуються *запити*, *форми*, *звіти* та *макроси*. В центральній частині вікна програми розташовано об'єкт, з яким безпосередньо працює користувач, в даному випадку — це таблиця **Книжки**.

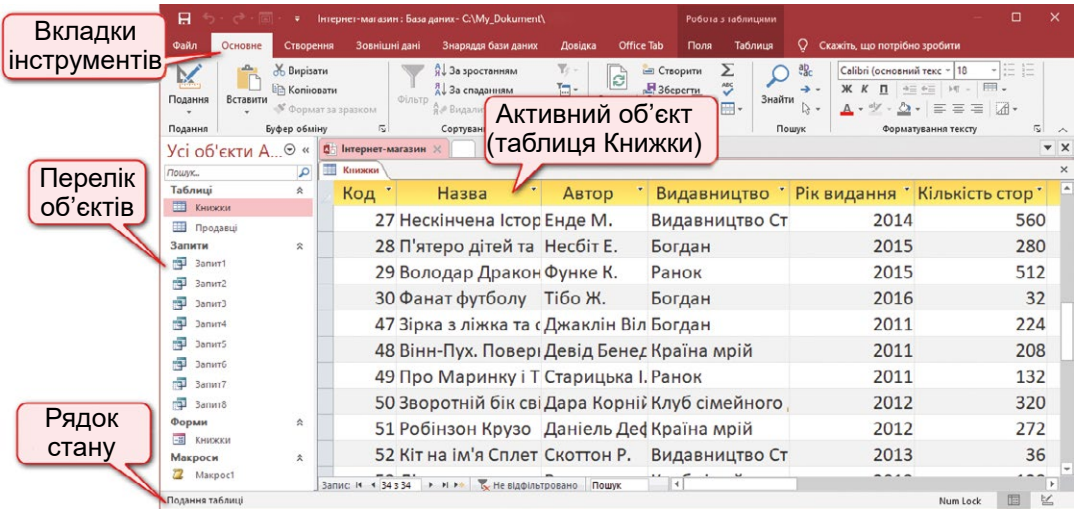


Рис. 24.3.

24.3.

Створення структури таблиці

Для створення структури таблиці необхідно виконати такий алгоритм:

1. На вкладці **Створення** обрати інструмент **Конструктор таблиць**.
2. Увести назви полів та встановити їхній тип.
3. Зберегти структуру таблиці.

Детальніше розглянемо процес створення структури таблиці, виконавши практичну вправу.

ВПРАВА 24.1



Завдання. Створити базу даних **Інтернет-магазин**.

Після чого створити у ній таблицю **Книжки**, як на рис. 24.1.

1. Створіть базу даних **Інтернет-магазин** (пояснення див. у п. 24.1).
2. Створіть структуру таблиці **Книжки**, для чого оберіть вкладку **Створення** ► **Конструктор таблиць**.
3. Створіть поле **Код** (рис. 24.4).
 - a. У стовпці **Ім'я поля** введіть ім'я **Код**.
 - b. У стовпці **Тип даних** оберіть тип **Автономерація**. При виборі цього типу програма сама буде здійснювати нумерацію записів у таблиці.

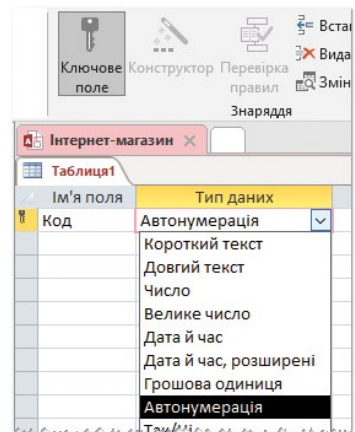


Рис. 24.4



- c. Укажіть, що дане поле буде ключовим, для чого натисніть інструмент **Ключове поле**. Після чого ліворуч від назви поля з'явиться позначка у вигляді ключа.
- 4. Додайте наступне поле **Назва**, тип поля залиште за умовчанням **Короткий текст**.
- 5. Аналогічно створіть поле **Автор**.
- 6. Створіть поле **Видавництво**.
 - a. Уведіть назву поля **Видавництво**.

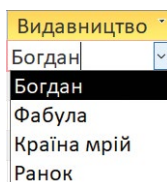


Рис. 24.5

Проаналізуємо, які дані містить це поле. Перелік значень у даному полі в даному прикладі обмежений декількома видавництвами. Щоб при введенні назви видавництва не припуститися помилки, а також кожного разу не вводити назву видавництва доцільно створити список з переліком наявних значень даного стовпця, а при заповненні таблиці даними, обирати потім значення зі списку. Так, на рис. 24.5 представлено створений список, із якого можна обрати назву видавництва.

- b. Для створення такого списку при виборі типу даних оберіть команду **Майстер підстановок**. В результаті відкривається вікно, в якому потрібно обрати одну із запропонованих опцій.

Опція **Поле підстановки** отримає значення з іншої таблиці або запиту передбачає отримання значення з іншої таблиці, але у нас немає в базі даних ніяких таблиць з яких ми можемо взяти значення видавництва. Друга опція — **Я самостійно введу потрібні значення** передбачає самостійне створення списку.

- c. Оберіть другу опцію та натисніть **Далі**.
- d. У стовпці введіть перелік наявних у таблиці видавництв: **Богдан, Фабула, Країна мрій, Ранок**.
- e. Натисніть **Далі** та **Готово**.
- 7. Збережіть структуру таблиці та перегляньте, як виглядатиме створений список видавництв.
 - a. Закрийте вкладку таблиці.
 - b. Натисніть **Так**, погоджуючись із збереженням таблиці.
 - c. Уведіть ім'я таблиці **Книжки** та натисніть **ОК**. В результаті таблицю буде збережено, і її значок має з'явитися у списку ліворуч.
 - d. Відкрийте таблицю **Книжки** подвійним клацанням на піктограмі.
 - e. Клацніть під полем **Видавництво**. В результаті має відкритися створений для заповнення даними список, як на рис. 24.5.
- 8. Поверніться до режиму створення структури таблиці. Для цього на вкладці **Основне** в меню інструмента **Подання** оберіть **Конструктор**.



9. Додайте до щойно створеного списку видавництво **Клуб сімейного дозвілля**.
 - a. Оберіть поле **Видавництво**.
 - b. В області властивостей поля оберіть вкладку **Підстановка**.
 - c. У властивості **Джерело рядків** додайте видавництво **Клуб сімейного дозвілля**, дотримуючись наведеного у списку синтаксису.
 - d. Натисніть **ОК**.
10. Додайте поля **Рік видання** та **Кількість сторінок**, установлюючи для них тип даних **Число**.
11. Додайте поле **Ціна**, встановлюючи для нього тип **Грошова одиниця**.

У режимі таблиці за умовчанням біля числа буде розташовано грошову одиницю у вигляді російського карбованця.

12. Змініть підпис грошової одиниці на **грн**.
 - a. Оберіть властивість **Формат** на вкладці **Загальні**.
 - b. Введіть до властивості **Формат** запис **# ##0,00" грн"**.

Проаналізуємо, що означає вказаний формат грошового підпису **# ##0,00" грн"**. Так, символ решітки тут означає будь-яку цифру. Пробіл між решітками вказує, що треба між цифрами встановити пропуск. 0 — також будь-яка цифра, але вона має бути обов'язково, якщо вона не вказана, то замість неї буде встановлено нуль. Кома вказує, де потрібно встановити розділення десяткового дробу. Напис в лапках — це текст, який буде дописано до числа.

13. Додайте поле **Наявність**. Це поле буде логічного типу, отже, у списку **Тип даних** слід обрати **так/ні**.
14. Додайте поле **Кількість**, що має числовий тип.
15. Додайте поле **Обкладинка** — воно має виводити зображення обкладинки книжки. Встановіть для нього тип **Вкладення**.
16. Змініть подання на табличне та подивіться, як воно буде відображено в таблиці. Як у назві, так і в рядку із значеннями має бути відображено скріпку. Поряд зі скріпкою у рядку значень буде відображено число нуль (рис. 24.6). Це означає, що тут розміщено 0 вкладень.
17. Установіть підпис для поля **Обкладинка**, для чого поверніться у режим конструктора та у вкладці **Загальні** для властивості **Підпис** уведіть значення **Обкладинка**.
18. Створіть поле **Жанр**. Для нього оберіть **майстер підстановок** та введіть список жанрів.



Рис. 24.6



19. Створіть поле **Анотація**, що має містити короткий зміст книжки. Для нього встановіть тип **Довгий текст**.
20. Збережіть структуру таблиці.

24.4.

Уведення даних до таблиці

Другий етап роботи з таблицею — це заповнення її даними. Для цього існує декілька способів.

1. Ввести дані з клавіатури.
2. Скопіювати дані з документа.
3. Імпортувати дані з іншого документа (текстового файлу, електронної таблиці, іншої бази даних тощо).

Розглянемо усі ці способи, виконавши практичну вправу.

ВПРАВА 24.2



Завдання. Увести дані до таблиці **Книжки**.

1. Завантажте файли зображень та даних з матеріалів до даного параграфа.
2. Уведіть перший запис із використанням клавіатури.
 - a. Відкрийте таблицю **Книжки**.
 - b. До поля **Назва** введіть **Останній лист**, до поля **Автор** — **Віктор Янкевич**. До поля **Видавництво** оберіть значення **Богдан** зі списку. Рік видання введіть 2002, кількість сторінок — 224, ціна — 99,50 грн. Укажіть, що книга є в наявності у кількості 3 шт.
 - c. Додайте зображення обкладинки. Для цього клацніть два рази на скріпці у полі **Обкладинка**. У діалоговому вікні, що з'явилося оберіть файл зображення обкладинки, що міститься на диску комп'ютера та натисніть **ОК**.

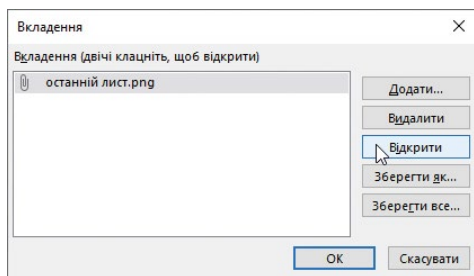


Рис. 24.7

В результаті біля скріпки має з'явитися число 1, що означає наявність вкладення одного файлу. Але саме зображення тут не відображається, його можна побачити, клацнувши на скріпці два рази та натиснувши кнопку **Відкрити** (рис. 24.7). Також зображення можуть відображатися при поданні даних з таблиці у вигляді форм.

Що стосується даних з поля анотації, то ми їх вручну вводити не будемо. Оскільки вони занадто довгі. Для додавання анотацій



скористаємося текстовим файлом **книжки_анотації.docx**, що містить назви книжок та анотації до них.

3. Додайте анотацію до першої книжки з файлу **книжки_анотації.docx**.
 - a. Виділіть в текстовому документі клітинку, що містить анотацію до книжки.
 - b. Перейдіть до таблиці **Книжки** у базі даних.
 - c. Установіть курсор до клітинки у полі **Анотація**.
 - d. В контекстному меню клітинки оберіть команду **Вставити**. В результаті текст анотації буде додано.

Якщо потрібно додати дані до однієї клітинки, то такий спосіб є доцільним. А якщо таких даних не менше десятка? А то й сотні або тисячі? Тоді доцільно буде скопіювати відразу усі дані з текстового документа.

4. Додайте до таблиці Access усі назви книжок, що містить текстовий документ.
 - a. Виділіть у текстовому документі назви усіх книжок, які необхідно вставити.
 - b. Перейдіть до таблиці **Книжки** у базі даних.
 - c. Наведіть курсор на клітинку починаючи з якої потрібно додати назви. Курсор має набути вигляду широкого білого плюса.
 - d. Клацніть на даній клітинці та натисніть **Ctrl+V** для вставлення даних. В результаті нові назви з'являться в таблиці.
5. Додайте до таблиці Access анотації до книжок, що містить текстовий документ.
 - a. Виділіть анотації у текстовому документі.
 - b. Перейдіть до таблиці в Access.
 - c. Виділіть білим широким плюсом увесь діапазон для вставлення анотацій.
 - d. Натисніть **Ctrl+V**. В результаті усі анотації буде додано.

Чому ж для вставлення назв ми виділяли лише одну клітинку, а для додавання анотацій увесь діапазон вставлення? Це відбувалося тому, що в першому випадку ми додавали назви в рядку створення нових записів і після вставлення автоматично утворювалися 9 нових записів. У другому випадку записи були вже створені і, якщо б ми робили вставлення лише в одну клітинку наявного запису, то анотацію було б додано лише одну. Саме тому необхідно було вказати весь діапазон клітинок для додавання анотацій.

6. Самостійно введіть з клавіатури інші дані перших десяти записів цієї таблиці (рис. 24.1).



Зручним способом уведення даних є *імпорт даних*. У більшості випадків імпорт даних здійснюють з електронних таблиць Excel.

7. Імпортуйте дані ще про 12 книжок із документа **книжки.xlsx** до таблиці **Книжки** в Access.
 - a. Перейдіть до середовища Access.
 - b. Оберіть вкладку **Зовнішні дані**.
 - c. В меню інструменту **Нове джерело даних** оберіть команду **З файла ▶ Excel**. В результаті відкривається діалогове вікно **Отримати зовнішні дані – Таблиця Excel**.
 - d. Натисніть кнопку **Огляд** та оберіть файл **книжки.xlsx**, що містить дані.
 - e. Натисніть **Відкрити**.
 - f. Установіть опцію **Додати копію записів до таблиці Книжки**. Якщо необхідно було б створити нову таблицю з даними, слід було залишити перемикач у початковому положенні.
 - g. Натисніть **ОК**. В результаті відкривається діалогове вікно, що містить у першому рядку заголовки полів таблиці, нижче — рядки з даними.
 - h. Натисніть **Далі** та **Готово**.
 - i. У наступному вікні натисніть **Закрити**.
 - j. Відкрийте таблицю **Книжки** та переконайтеся, що дані з таблиці Excel додано.

Зауважимо, що в такий спосіб неможливо додати файли зображень.

Іноді існує необхідність отримати дані, що містить таблиця Access у вигляді електронної таблиці. Для цього дані експортують із Access в Excel. Розглянемо процес експорту на прикладі таблиці **Книжки**.

8. Виконайте експорт даних із середовища Access до Excel.
 - a. У середовищі Access відкрийте вкладку **Зовнішні дані**.
 - b. В області **Експорт** оберіть інструмент **Excel**. В результаті відкриється діалогове вікно **Експорт — таблиця Excel**.
 - c. Оберіть в даному вікні місце розташування та ім'я файла, до якого буде занесено дані, та натисніть **ОК**, а в наступному вікні — **Закрити**.
 - d. Відкрийте експортовану таблицю в Excel, щоб подивитися, які дані було отримано.

Видалення записів

Для видалення запису з таблиці Access необхідно його виділити (клацнути праворуч на сірому полі від запису) та в контекстному меню обрати команду **Видалити запис**. Також можна обрати відповідну команду на вкладці **Основне**.

**Контрольні запитання та завдання**

1. Які існують типи полів в Access?
2. Опишіть алгоритм створення структури таблиці.
3. Які існують способи введення даних до таблиці?
4. Який тип поля необхідно встановити для додавання зображення?
5. Для чого використовують **Майстер підстановок**?

**Питання для роздумів**

- 1*. Чому поля не можуть мати однакові назви?
- 2*. Чи можна до клітинки із поля, створеного з допомогою **Майстра підстановок** додати декілька значень зі списку?

**Завдання для досліджень**

- 1*. Дослідіть, що таке обчислювальний тип поля в Access.
- 2*. Дослідіть, які типи файлів можна додати до поля, що має тип **Вкладення**.
- 3*. Дослідіть, як у середовищі Excel можна експортувати дані із Access до Excel.

**§25. СТВОРЕННЯ ЗВ'ЯЗКІВ МІЖ ТАБЛИЦЯМИ**

Як було зазначено раніше, бази даних можуть мати декілька таблиць, пов'язаних між собою. Розглянемо, як утворюються зв'язки між таблицями, а також їхню реалізацію в середовищі Access.

В попередньому параграфі ми розпочали створення таблиці **Книжки** у базі даних **Інтернет-магазин**. Інтернет-магазин — це по суті платформа для продажу певного товару, у якого є продавець. Отже, додамо до бази даних ще одну таблицю **Продавці** (рис. 25.1). Дана таблиця буде містити *назву продавця, його рейтинг, контактний телефон та дату реєстрації* на сайті інтернет-магазину.

Визначимо, які типи даних будуть містити поля даної таблиці. Неважко здогадатися, що поле **Продавець** — це короткий текст. Оскільки назви фірм-продавців є унікальними і не можуть повторюватися, то це поле також можна встановити в якості ключового для даної таблиці.



Продавець	Рейтинг	Контактний телефон	Дата реєстрації
AB-Books	4,1	+38-054-43-33	16.05.2012
Books and Toys	4,7	+38-068-71-13	17.10.2019
GOODSHOCK	5	+38-068-72-12	12.05.2017
GSS	3,8	+38-068-72-13	24.09.2020
Персей	4,5	+38-068-72-11	21.02.2018

Рис. 25.1. Таблиця Продавці

Наступне поле — **Рейтинг**. Рейтинг продавців, як правило, визначають покупці, встановлюючи кількість зірок за якість обслуговування. Припустимо, що в нашому інтернет-магазині максимальна кількість зірок — 5. Отже, рейтинг буде приймати числа від 0 до 5 включно і може бути дробовим числом, оскільки він визначається загальною сумою встановлених зірок покупцями, поділену на кількість покупців. Відповідно рейтинг матиме тип **Число**, що буде представлено десятковим дробом від 0 до 5.

Поле **Контактний телефон** хоча і представлений у вигляді цифр, проте числа, що ми отримуємо із цих цифр, не використовуються для будь-яких обчислень чи порівнянь, отже, для цього поля доцільно встановити текстовий тип, проте сформувавши для його даних певну маску введення.

Останнє поле — **Дата реєстрації**. Незавжди здогадатися, що це буде тип **Дата й час**.

ВПРАВА 25.1



Завдання. Імпортувати таблицю **Продавці**. Змінити властивості полів.

1. Завантажте на комп'ютер файл **Продавці.xlsx** з матеріалів до даного параграфа.
2. Відкрийте базу даних **Інтернет-магазин**.
3. Імпортуйте таблицю **Продавці** з файла **Продавці.xlsx**.
 - a. На вкладці **Зовнішні дані** оберіть **Нове джерело даних** ▶ **З файла** ▶ **Excel**.
 - b. Натисніть кнопку **Огляд** та оберіть файл **Продавці.xlsx**.
 - c. Залишіть перемикач у положенні **Імпортувати дані джерела до нової таблиці в поточній базі даних**.
 - d. Натисніть **ОК** та **Далі**.
 - e. Переконайтеся, що встановлено прапорець **Перший рядок містить заголовки стовпців**. Якщо прапорець не встановлено, встановіть його та натисніть **Далі** два рази.



- f. У наступному вікні встановіть перемикач у положення **Вибрати власний первинний ключ Продавець**.
 - g. Натисніть **Готово** та **Закрити**. В результаті таблицю буде імпортовано.
 - h. Перегляньте результат імпорту в режимі таблиці.
4. Змініть у полі **Рейтинг** формат подання числа таким чином, щоб число мало одну цифру після коми.
- a. Відкрийте таблицю **Продавці** в режимі конструктора.
 - b. Оберіть поле **Рейтинг**.
 - c. На вкладці **Загальне** для властивості **Розмір поля** встановіть **Одинарне значення**.
5. Змініть формат відображення контактного телефону.
- a. Оберіть поле **Контактний телефон**.
 - b. Оберіть вкладку **Загальне** та для властивості **Маску введення** введіть таку маску **“+38-0”###-##-##**. (Символи вказані в лапках, будуть відображатися перед введенням номера телефону, далі користувач має ввести шість будь-яких цифр).

Встановлення умов на введення даних

Аби при введенні даних до таблиць не виникало помилок, в Access є можливість установлювати додаткові умови на введення даних.

Так, при спробі ввести два однакових імені продавців відразу виникне помилка. Це відбувається тому, що поле **Продавець** є ключовим і Access пильно слідкує за тим, щоб значення в ключових полях не повторювалися. Також виникне помилка при введенні текстових даних до поля, що має числовий тип. Обмеження можна встановити і на введення даних до інших типів полів. Розглянемо, як установити обмеження на введення даних, виконавши практичну вправу.

ВПРАВА 25.2



Завдання. Установити умову на введення даних до поля **Рейтинг** від 0 до 5 включно.

Так, ми знаємо, що поле **Рейтинг** — це число, що може приймати значення від 0 до 5. Отже, встановимо обмеження на введення цього значення.

- 1. Відкрийте таблицю **Продавці** в режимі конструктора.
- 2. Оберіть поле **Рейтинг**.
- 3. На вкладці **Загальні** оберіть властивість **Правило перевірки**.
- 4. Клацніть три точки біля цієї властивості та введіть умову **between 0 and 5**. Це означає, що значення в цьому полі має лежати в проміжку від 0 до 5 включно.



5. Для властивості **Текст перевірки** введіть повідомлення, що має з'являтися у випадку, якщо користувач неправильно введе значення. Введіть, наприклад, «Значення має бути від 0 до 5».
6. Перейдіть в режим таблиці та введіть до поля **Рейтинг**, наприклад, число 6. Натисніть **Enter**. В результаті з'явиться вікно з повідомленням про помилку.
7. Відмініть зміни в даній клітинці.

25.1.

Створення зв'язків

В результаті виконаної роботи в базі даних ми маємо дві таблиці **Продавці** та **Книжки**. Нам необхідно зв'язати ці таблиці, щоб було зрозуміло, в якого продавця можна купити ту чи іншу книжку.

Щоб зв'язати між собою ці таблиці, ми маємо в таблиці **Книжки** додати поле **Продавець**, в якому вказати назву продавця із таблиці **Продавці**. Поле **Продавець**, яке ми створимо в таблиці **Книжки**, називають *зовнішнім ключем*. Це ж саме поле з таблиці **Продавці** називають *первинним ключем*.

ВПРАВА 25.3



Завдання. Додати до таблиці **Книжки** продавців.

1. Відкрийте таблицю **Книжки** в режимі конструктора.
2. Створіть поле **Продавець**.
 - a. У стовпці **Ім'я поля** введіть поле **Продавець**.
 - b. Аби при введенні продавців не виникало помилок, оберіть **Майстер підстановок**.
 - c. Залиште опцію **поле підстановки має отримати значення з іншої таблиці або запиту**, оскільки значення ми візьмемо з уже наявної таблиці **Продавці**.
 - d. Натисніть кнопку **Далі**.
 - e. Оберіть таблицю **Продавці** і ще раз натисніть **Далі**.
 - f. У списку полів оберіть поле **Продавець** і натисніть **Далі** два рази.
 - g. Переконайтеся, що у вікні відображено саме той список, що потрібен, натисніть **Далі** та **Готово**.
3. Відкрийте таблицю **Книжки** та заповніть поле **Продавці** довільним чином, обираючи їх зі списку.

Використовуючи підстановку значень ключового поля з однієї таблиці до іншої, ми автоматично пов'язали дві таблиці. Де ж можна побачити, які таблиці і як саме пов'язані?



Для відображення та редагування зв'язків між таблицями використовують *схему даних*. Для роботи з нею бажано закрити вкладки усіх об'єктів бази даних, перейти на вкладку **Знаряддя бази даних** та обрати інструмент **Зв'язки**. На рис. 25.2 видно, що тут відображено структуру таблиць **Книжки** та **Продавці**, а також зв'язок між полями **Продавець** цих таблиць. Для редагування зв'язків необхідно клацнути на лінії зв'язку два рази. В результаті відкриється вікно **Редагування зв'язків**.

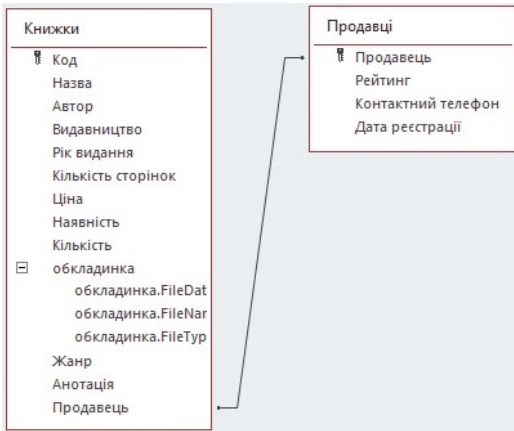


Рис. 25.2

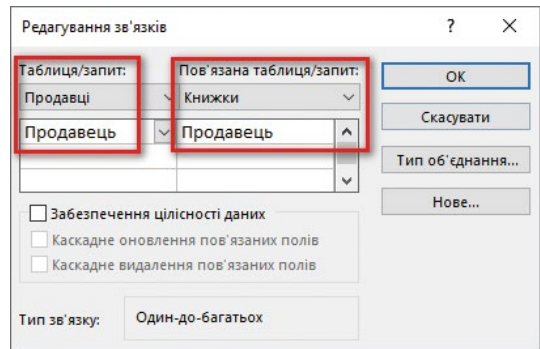


Рис. 25.3

В даному вікні (рис.25.3) вказано, що поле **Продавець** із таблиці **Продавці** пов'язано із полем **Продавець** таблиці **Книжки**. Також вказано тип зв'язку **Один-до-багатьох**. Цей тип зв'язку вказує на те, що один продавець може продавати багато книжок, а одну й ту саму книжку може продавати лише один продавець. Розберемося, яким є призначення прапорців, що містить дане вікно.

Перший прапорець **Забезпечення цілісності даних**. При його встановленні в схемі даних відразу з'являються позначення типу зв'язку **Один-до-багатьох**, де **один** позначається зі сторони таблиці **Продавець**, а багато (знак нескінченності) біля таблиці **Книжки**.

У випадку встановлення цієї опції зовнішній ключ таблиці **Книжки** не може містити значень, яких не містить ключове поле таблиці **Продавець**.

Якщо спробувати до таблиці **Книжки** ввести в якості продавця, наприклад, **Еталон**, та перейти до наступного запису, в результаті з'явиться помилка, як на рис. 25.4. Це відбувається тому, що в таблиці **Продавець** назва **Еталон** відсутня. Якщо це ж значення ввести без встановлення прапорця біля опції **Забезпечення цілісності даних**, то



помилка не виникне, отже, ми спокійно зможемо ввести назву неіснуючого продавця.

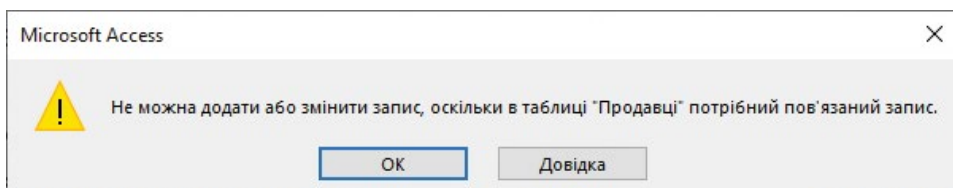


Рис. 25.4

Якщо після неіснуючої назви зробити спробу встановити опцію **Забезпечення цілісності даних**, також з'явиться помилка, яка інформує про те, що неможливо встановити дану опцію, оскільки у підлеглий таблиці розміщено дані, яких немає в основній таблиці. А саме: таблиця **Книжки** містить продавця, що відсутній в самій таблиці **Продавець**. Доки дані не буде змінено, доти програма не дозволить установити дану опцію.

Наступна опція, яку можна встановити у вікні редагування зв'язків, — це **Каскадне оновлення пов'язаних полів**. У цьому випадку при зміні значень у первинному ключі відбуваються відповідні зміни в зовнішньому ключі.

Якщо після встановлення опції **Каскадне оновлення пов'язаних полів** у таблиці **Продавці** змінити назву одного з продавців, наприклад, дописати до продавця **GSS** ще **-X** та оновити дані в таблиці, то після відкриття таблиці **Книжки** назву даного продавця буде автоматично змінено.

Ще є одна опція — **Каскадне видалення пов'язаних полів**. Установлення її дозволяє при видаленні запису із таблиці, що має первинний ключ, видалити відповідні записи із таблиці, що містить зовнішній ключ.

Так, якщо у таблиці **Продавці** видалити запис із продавцем **GSS-X**, то після відкриття таблиці **Книжки** усі записи з даним продавцем буде автоматично видалено.

Отже, при створенні та редагуванні зв'язків необхідно чітко розуміти необхідність установлення розглянутих опцій у вікні редагування в залежності від поставленої задачі.

ВПРАВА 25.4



Завдання. Установити між таблицями **Книжки** та **Продавці** зв'язок, що буде забезпечувати цілісність даних, каскадне оновлення та видалення пов'язаних полів.



Отже, для створення у базі даних зв'язку **Один-до-багатьох** в деякій таблиці **В** необхідно створити зовнішній ключ, який приймає начення первинного ключа таблиці **А** (рис. 25.5).



Рис. 25.5

**Контрольні запитання та завдання**

1. Що таке первинний та зовнішній ключ таблиці?
2. Який засіб використовують для відображення зв'язків в Access?
3. Що відбудеться, якщо до ключового поля ввести два однакові значення?
4. Як установити умову на введення даних між двома числовими значеннями?
5. Що означає зв'язок між таблицями **Один-до-багатьох**?
- 6*. Опишіть послідовність дій для створення зв'язку **Один-до-багатьох** у середовищі Access.

**Питання для роздумів**

- 1*. При введенні текстових даних до числового поля виникає помилка, а при введенні числових даних до текстового поля помилки не виникає. Як ви думаєте, чому?
- 2*. В яких випадках під час редагування зв'язку при встановленні опції **Забезпечення цілісності даних** може з'явитися помилка?

**Завдання для досліджень**

- 1*. Дослідіть, як установити умову на введення даних між двома числовими даними, що не включено до діапазону, наприклад, значення має бути більше 0, але менше 7.
- 2*. Дослідіть, що відбудеться з даними підлеглої таблиці, якщо в головній таблиці змінити значення ключового поля при різних положеннях опцій **Каскадне оновлення пов'язаних полів** та **Каскадне видалення пов'язаних полів**.





§26.

ПРАКТИЧНА РОБОТА 6. «СТВОРЕННЯ ТАБЛИЦЬ ТА ЗВ'ЯЗКІВ МІЖ НИМИ»



Завдання. Створити базу даних предметної області **Готель**, що містить таблиці **Відвідувачі** та **Номери готелю**, які мають таку схему даних (рис. 26.1):

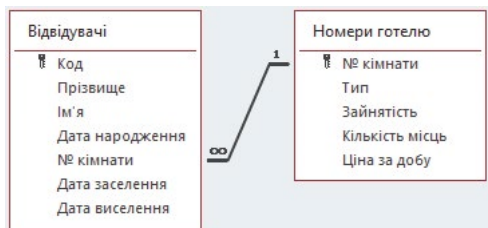


Рис. 26.1

1. Завантажте файл **Готель.xlsx** з матеріалів до даного уроку.
2. Створіть нову базу даних з назвою **Готель**.
3. Створіть структуру таблиці **Номери готелю** з такими полями:
 - **№ кімнати** (Тип Даних — **Число**, **Розмір поля** — **Ціле число**, встановіть дане поле в якості ключового);
- **Тип** (зробіть з використанням **Майстра підстановки** даних зі списку: Стандарт, Люкс, Сімейний, Економ);
- **Зайнятість** (тип так/ні);
- **Кількість місць** (тип установіть самостійно);
- **Ціна за добу** (тип та підпис грн установіть самостійно).
4. Заповніть таблицю даними аркуша **Номери готелю** з файла **Готель.xlsx** (дані можна скопіювати або імпортувати до таблиці).
5. Імпортуйте до бази даних таблиці **Відвідувачі** з файла **Готель.xlsx**. У процесі імпорту вкажіть **Дозволити Access додати первинний ключ**.
6. В таблиці **Відвідувачі** врахуйте, що дата виселення має бути більшою, ніж дата заселення.
 - a. Перейдіть у режим таблиці.
 - b. На вкладці **Поля** в меню інструменту **Перевірка** оберіть команду **Правило перевірки**.
 - c. У вікні **Побудовник виразів** введіть вираз: **[Дата заселення] < [Дата виселення]** та натисніть **ОК**.

Зауваження. Для швидкого і безпомилкового введення такого виразу у стовпці **Категорії виразів** слід клацнути два рази поле **Дата заселення**, поставити знак «<», після чого клацнути два рази поле **Дата виселення**.

7. Самостійно на власний розсуд додайте ще інформацію про трьох відвідувачів.
8. Створіть зв'язок між таблицями.
 - a. Відкрийте схему даних.
 - b. Натисніть інструмент **Додати таблицю**.



- c. Оберіть у списку обидві таблиці, утримуючи **Ctrl**, та натисніть **Додати**, а потім **Закрити**. В результаті до схеми даних буде додано дві таблиці.
 - d. Установіть зв'язок між таблицями, натиснувши поле **№ кімнати** у таблиці **Номери готелю** та переміщуючи його, утримуючи клавішу миші, на поле **№ кімнати** таблиці **Відвідувачі**. В результаті має відкритися вікно **Редагування зв'язків**.
 - e. Виконайте підсилення зв'язку. Для чого встановіть усі прапорці у вікні.
9. В результаті підсилення зв'язку може виникнути помилка. Це може відбутися тому, що поля **№ кімнати** в різних таблицях мають різні формати чисел. Для виправлення цієї помилки закрийте схему даних та змініть числовий формат у таблиці **Відвідувачі**.
- a. Відкрийте таблицю **Відвідувачі** в режимі конструктора.
 - b. Оберіть поле **№ кімнати**.
 - c. На вкладці **Загальні** для властивості **Розмір поля** встановіть значення **Ціле число**.
10. Ще раз створіть зв'язок між таблицями із встановленням обмеження цілісності даних, каскадним оновленням та видаленням даних.



§27.

СОРТУВАННЯ ТА ФІЛЬТРАЦІЯ ДАНИХ



27.1.

Сортуння даних

У попередніх параграфах ми створили базу даних інтернет-магазину, що складається з двох таблиць. Далі розглянемо, як використовувати дані, що містять таблиці. Саме в даному параграфі познайомимось з тим, як здійснювати сортуння та фільтрацію даних.

Із сортунням та фільтрацією даних ви вже знайомі, вивчаючи роботу в електронних таблицях Excel. Дії, що виконуються в Access, є подібними. Спочатку розглянемо, як виконати в таблиці звичайне сортуння.

Сортуння записів в Access здійснюють у порядку зростання чи спадання значень певного поля. (В межах одного поля, як це було в Excel, сортуння виконати неможливо.) Найчастіше сортуння здійснюють, коли необхідно визначити найбільші або найменші значення, встановити написи в алфавітному порядку тощо. Сортуння також доцільно застосувати у разі потреби групування записів



за певним параметром (розташувати поряд записи з однаковими чи близькими значеннями).

ВПРАВА 27.1



Завдання. Виконати сортування даних в таблицях бази даних **Інтернет-магазин**.

1. Виконайте сортування за зростанням по полю **Назва** в таблиці **Книжки**.
 - a. У базі даних **Інтернет-магазин** відкрийте таблицю **Книжки**.
 - b. Клацніть на заголовок поля **Назва**.
 - c. На вкладці **Основне** в області **Сортування й фільтр** оберіть інструмент **За зростанням**. У результаті значення в стовпці має бути відсортовано. Звичайно, разом із ними мають бути відсортовані й дані з інших полів.

Ще один спосіб сортування — натиснути стрілку вниз біля назви поля та обрати бажаний спосіб сортування.

Щоб скасувати сортування, слід натиснути інструмент **Видалити сортування**. В результаті таблиця набуде початкового вигляду.

На відміну від Excel, інструменту розширеного сортування тут немає. Але для того, щоб сортування було здійснено спочатку за даними одного поля, а потім іншого, достатньо послідовно виконати сортування в бажаних полях.

2. Виконайте сортування за зростанням спочатку за роками, а в межах повторення років — за авторами книжок. В результаті автори тих книжок, що мають однаковий рік видання, будуть упорядковані в алфавітному порядку.

27.2.

Фільтрація даних

Далі розглянемо, як здійснюється в Access фільтрація даних. Пригадаємо, що *фільтрація* — це пошук інформації за вказаною умовою. Існує декілька видів фільтрації даних. Найпростіший пошук — це коли значення, яке необхідно знайти, міститься в клітинці.

ВПРАВА 27.2



Завдання. Виконати фільтрацію даних в таблицях бази даних **Інтернет-магазин**.

1. Знайдіть усі книжки, що було видано в 2017 році.
 - a. Відкрийте таблицю **Книжки**.
 - b. Клацніть на стрілці біля поля **Рік видання**.



- c. Приберіть прапорець з опції **Виділити все** та встановіть прапорець біля 2017.
- d. Натисніть **ОК**. В результаті мають залишитися лише книжки, що були видані у 2017 році.
- e. Для видалення фільтра зніміть виділення з інструменту **Застосувати фільтр**.

Для фільтрації даних за значеннями двох полів необхідно виконати ці ж самі дії, але для двох полів.

- 2. Знайдіть усі книжки видавництва **Богдан**, що мають жанр **Повість** або **Роман**.
 - a. Клацніть стрілку біля поля **Видавництво** та залиште прапорець біля видавництва **Богдан**. Натисніть **ОК**. В результаті мають залишитися усі книжки видавництва **Богдан**.
 - b. Клацніть стрілку біля поля **Жанр**. У цьому полі залиште прапорці біля полів **Роман** та **Повість**.

При фільтрації текстових даних можна застосувати текстові фільтри, що є аналогічними до тих, що ми розглядали в програмі Excel. Це може бути текст, який починається, закінчується або містить деякі символи. Або навпаки не містить указані у фільтрі символи.

Аналогічно можна застосувати фільтри і до числових даних, які є більшими, меншими, ніж задані, лежать у вказаному проміжку або не дорівнюють заданому числу.

- 3. Знайдіть усі книжки, що в імені або прізвищі автора містять сукупність символів «дж».
 - a. Клацніть стрілку біля поля **Автор**.
 - b. У меню команди **Текстові фільтри** оберіть команду **Містить**.
 - c. Уведіть символи «дж» та натисніть **ОК**.
- 4. Знайдіть усі книжки, яких наявності 10 і більше.
 - a. Клацніть стрілку біля поля **Кількість**.
 - b. У меню команди **Фільтри чисел** оберіть **Більше**.
 - c. Уведіть число 10 та натисніть **ОК**.
- 5. Знайдіть усі книжки, вартість яких перевищує 150 грн, та ті, що були видані до 2017 року.
 - a. Клацніть стрілку біля поля **Ціна**.
 - b. В меню команди **Фільтри чисел** оберіть **Більше**.
 - c. Уведіть число 150 та натисніть **ОК**. В результаті ви одержите перелік книжок, ціна яких 150 грн та більше.

Оскільки нам потрібні тільки ті книжки, в яких ціна більша за 150 грн, змінимо умову фільтрації.



- d. Ще раз клацніть стрілку біля поля **Ціна**.
 - e. В меню команди **Фільтри чисел** оберіть **Більше**.
 - f. Уведіть умову >150 та натисніть **ОК**. В результаті одержуємо перелік книжок, ціна яких більше за 150 грн.
 - g. Клацніть стрілку біля поля **Рік видання**.
 - h. В меню команди **Фільтри чисел** оберіть **Менше**.
 - i. Уведіть умову <2017 та натисніть **ОК**.
6. Самостійно знайдіть книжки, назва яких починається на "в".
- 7*. Самостійно знайдіть книжки, що мають жанр **Роман** і були випущені у 2012 році.
- 8*. Самостійно знайдіть книжки, вартість яких менше 100 грн, видавництво **Ранок** або **Богдан**, видані після 2014 року.

27.3.

Розширений фільтр

А що ж робити, коли необхідно знайти дані, для яких виконується не обидві умови, а одна із умов? Так, наприклад, необхідно знайти усі книжки, вартість яких перевищує 150 грн або ті, що були видані до 2017 року. Так, на рис. 27.1 у синіх рамках обведено дані, які має бути відображено. Дані, що знаходяться в зелених рамках, відповідають першій умові (ціна більша за 150 грн), дані, розміщені в червоних рамках, задовольняють другу умову (рік видання менший від 2017-го).

Код	Продавець	Назва	Автор	Видавництво	Рік видання	Кількість сторінок	Ціна
1	Books and Toys	Останній лист	Віктор Янкевіч	Богдан	2002	224	99,50 грн.
2	AB-Books	Чорний павук	Гольфелм Є	Богдан	2019	128	149,00 грн.
3	AB-Books	Пророцця	Кокотюха Ан,	Богдан	2016	272	179,00 грн.
4	GOODSHOCK	Моя сім'я та інші	Дарелл Джеф	Богдан	2017	328	199,00 грн.
5	GOODSHOCK	Птахи, звірі та ро	Дарелл Джеф	Богдан	2018	320	199,00 грн.
6	Персей	Пастка на полоза	Кокотюха Ан,	Богдан	2018	280	179,00 грн.
7	Персей	Вірус	Гелен Фіцджі	Фабула	2017	320	150,00 грн.
8	AB-Books	Чужа	Емі Джентрі	Фабула	2016	320	170,00 грн.
9	Books and Toys	Висотка	Баллард Дж.	Фабула	2015	224	150,00 грн.
10	Books and Toys	Демельза	Вінстон Грем	Фабула	2017	544	200,00 грн.
19	AB-Books	Зірка з ліжка та с	Джаклін Вілс	Богдан	2011	224	21,00 грн.
20	AB-Books	Вінн-Пух. Поверн	Девід Бенеді	Країна мрій	2011	208	55,00 грн.

Рис. 27.1

Пошук такої інформації можна здійснити з допомогою використання *розширеного фільтру*. До того ж розширений фільтр містить у собі також засоби сортування. Розглянемо принцип використання такого фільтра, виконавши вправу.



ВПРАВА 27.3



Завдання. В таблиці **Книжки** бази даних **Інтернет-магазин** знайти усі книжки, що були випущені до 2017 року, або ті, що коштують більше за 150 грн.

1. Відкрийте таблицю **Книжки**.
2. На вкладці **Основне** в області **Сортуння й фільтр** оберіть в меню інструменту **Додатково** команду **Розширений фільтр/сортуння**. В результаті відкриється вікно конструктора.
3. Оберіть поля, за якими необхідно зробити фільтрацію. Для цього клацніть два рази на поле **Ціна**, а потім два рази — на **Рік видання**. В результаті поля буде додано до конструктора.
4. Для поля **Ціна** виконайте сортуння за зростанням.
5. Укажіть умову фільтрації по полю **Ціна**. Для цього в рядку **Критерії** вкажіть умову **>150**.
6. Також указуємо, що маємо отримати ще й дані про книжки видані, після 2017 року. Для цього в рядку **Або** навпроти поля **Рік видання** вкажіть умову **<2017**. Отже, вікно конструктора має бути заповнено, як на рис. 27.2.
7. Натисніть кнопку **Застосувати фільтр**.

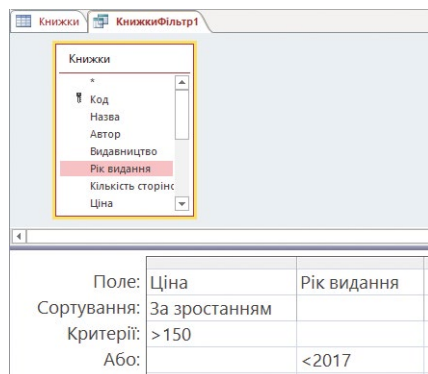


Рис. 27.2

Розглянемо ще один вид фільтрів — це пошук тексту за частиною його вмісту.

Для цього використовують оператор **Like «критерій»** або **Схоже «критерій»** в українській версії Access.

В якості критерію вказується маска для пошуку. В масці можна використовувати символи підстановки: «*», що означає будь-яку кількість довільних символів, та «?», що означає один довільний символ.

Так, для пошуку всіх записів, що починаються літерою «а», можна вказати оператор **Like «a*»**, для пошуку всіх записів, що містять літеру «а» буде застосовано оператор **Like «*a*»**.

ВПРАВА 27.4



Завдання. Знайти усі книжки, в назві яких є слова, що починаються на літеру «п». Отже, це мають бути книжки, в яких назва починається на «п», а також ті, в яких є й інші слова, що починаються на «п».



1. На вкладці **Основне** в області **Сортування й фільтр** в меню інструмента **Додатково** оберіть команду **Розширений фільтр/сортування**.
2. Додайте до конструктора поле **Назва**.
3. Для пошуку всіх книжок, що починаються на літеру «п», введіть умову **Like «п*»**.
4. Натисніть **Enter**.

Якщо ви працюєте в українській версії Access, то програма автоматично замінить слово **Like** його українською версією на слово **Схоже**.

5. В рядку **Або** додайте ще один критерій пошуку **Like «* п*»**. Даний критерій дозволить знайти усі інші слова, що починаються літерою «п». Він означає, що рядок буде містити деякий текст, потім символ пробілу, а далі ще будь-який текст.
6. Застосуйте фільтр.



Контрольні запитання та завдання

1. Як виконати сортування за зростанням (спаданням) по одному полю?
2. Як скинути результати сортування та (або) фільтрації?
3. Як виконати фільтрацію числових даних, що є більшими, ніж задане число?
4. Як виконати фільтрацію, коли необхідно врахувати умову одночасно для двох полів?
5. Як у розширеному фільтрі вказати умову на текстові дані, що мають в собі містити вказані символи?



Питання для роздумів

- 1*. В яких випадках використовують розширений фільтр?
- 2*. Подумайте, чи не достатньо було б у вправі 27.4 для пошуку ввести який небудь один критерій. **Like «п*»** чи **Like «* п*»**.



Завдання для досліджень

- 1*. Дослідіть можливості використання інструменту **Змінити фільтр** в меню інструменту **Додатково** на вкладці **Основне**.
- 2*. Дослідіть, яку умову слід установити в розширеному фільтрі таблиці **Книжки** для виведення усіх книжок, що є в наявності.





§28. ЗАПИТИ

28.1. Створення та редагування запитів

Для пошуку інформації за заданою умовою ми використовували фільтри. Проте фільтри не можна зберегти як окрему таблицю для довготривалого використання. Для пошуку та збереження інформації за заданою умовою використовують *запити*. Саме про них і піде мова в даному параграфі.

Крім того, що запити можна зберігати як окремі об'єкти в базі даних, з їхньою допомогою можна здійснювати пошук інформації за даними не лише однієї таблиці.

Запити — це спеціальні об'єкти, призначені для вибірки даних із таблиць бази за заданою умовою. При зміні даних у вихідних таблицях дані автоматично будуть змінюватися і в запитах.

Для створення запитів можна використовувати вбудований конструктор, подібний до конструктора розширених фільтрів. Так, на рис. 28.1 наведено конструктор створення запиту, що має вивести інформацію про назву та автора книжок, що були видані у 2017 році. Розглянемо процес створення такого запиту, виконавши вправу.

Поле:	Назва	Автор	Рік видання
Таблиця:	Книжки	Книжки	Книжки
Сортування:			
Відображення:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Критерій:			2017

Рис. 28.1

ВПРАВА 28.1



Завдання. Створити запити у базі даних **Інтернет-магазин**.



Запит 1. Вивести назву та автора книжок, які було видано у 2017 році.

1. Відкрийте базу даних **Інтернет-магазин**.
2. На вкладці **Створення** оберіть інструмент **Макет запиту** — в результаті має відкритися вікно **Відображення таблиці**.
3. Оберіть таблицю **Книжки** та натисніть кнопки **Додати** та **Закрити**. В результаті таблицю буде додано до конструктора.



4. Додайте до конструктора поля **Назва** та **Автор**, клацаючи два рази на відповідні назви полів таблиці. Це поля, які необхідно вивести відповідно до умови завдання.
5. Також додайте поле **Рік видання**. За цим полем буде створено критерій пошуку.
6. У полі **Рік видання** навпроти рядка **Критерії** введіть 2017.
7. Запустіть запит на виконання, натиснувши інструмент **Запуск** на вкладці **Конструктор**.

Відповідно до умови завдання в запиті ми маємо вивести інформацію лише за полями **Назва** та **Автор**, отже, поле **Рік видання** необхідно приховати.

8. Поверніться у режим конструктора запиту та зніміть прапорець у полі **Рік видання** навпроти рядка **Відображення** (рис. 28.1).
9. Ще раз запустіть запит на виконання. Тепер буде відображено лише ті поля, що відповідають указаному критерію.
10. Збережіть запит.
 - a. Закрийте вкладку запиту і натисніть **Так** на запит про збереження запиту.
 - b. Введіть ім'я запиту, наприклад **Запит_1** та натисніть **ОК**. В результаті запит буде збережено і відображено в групі запитів.



Запит 2. Вивести інформацію про назву, автора книжок та кількість сторінок, що мають жанр повість та кількість сторінок більшу за 200.

1. Створіть новий запит в режимі макета та додайте до нього таблицю **Книжки**.
2. До конструктора запиту додайте поля **Назва**, **Автор**, **Кількість сторінок** та **Жанр**.
3. Для поля **Кількість сторінок** додайте умову **>200**.
4. Для поля **Жанр** укажіть умову **повість**.
5. Виконайте запит та переконайтеся, що вказані умови враховано.
6. Поверніться до режиму конструктора та заберіть видимість даних для поля **Жанр**.
7. Виконайте запит іще раз.
8. Збережіть запит під назвою **Запит_2**.




Запит 3. Вивести інформацію про назву, видавництво та ціну книжок у жанрі детективу та з ціною від 100 грн до 180 грн.

1. Створіть новий запит в режимі макета та додайте до нього таблицю **Книжки**.
2. До конструктора запиту додайте поля **Назва**, **Видавництво**, **Ціна** та **Жанр**.



3. Для поля **Ціна** в рядку **Критерій** уведіть умову **>=100 and <=180**.
 - a. У разі введення великої умови доцільно скористатися побудовником виразів. Відкрийте побудовник виразів, обравши інструмент **Конструктор** або натиснувши **F2**.
 - b. У вікні побудовника введіть умову **>=100 and <=180** (або **between 100 and 180**) та натисніть **ОК**.
4. Для поля **Жанр** укажіть умову **детектив**.
5. Приберіть відображення даних для поля **Жанр**.
6. Виконайте запит та переконайтеся, що вказані умови враховано.
7. Збережіть запит під назвою **Запит_3**.

 **Запит 4.** Вивести інформацію про назву, видавництво та ціну книжок у жанрі детективу або з ціною від 100 грн до 180 грн.

Зверніть увагу, що на відміну від попереднього запиту, де для виведення записів необхідно було врахувати обидві умови, в даному випадку має спрацьовувати одна із таких умов. Для створення такого запиту критерій для поля **Жанр** треба розмістити на рядку нижче, ніж для поля **Ціна**. В даному випадку для умови буде спрацьовувати сполучення **або**.

Отже, для врахування декількох умов одночасно їх розташовують в одному рядку. Для врахування однієї з умов критерії розміщують в різних рядках (рис. 28.2).

Поле:	Назва	Видавництво	Ціна	Жанр
Таблиця:	Книжки	Книжки	Книжки	Книжки
Сортування:				
Відображення:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Критерій:			Between 100 And 180	"детектив"
Або:				

та

Поле:	Назва	Видавництво	Ціна	Жанр
Таблиця:	Книжки	Книжки	Книжки	Книжки
Сортування:				
Відображення:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Критерій:			Between 100 And 180	
Або:				"детектив"

або

Рис. 28.2

1. Створіть копію об'єкта **Запит_3** та надайте йому ім'я **Запит_4**.
 - a. У контекстному меню об'єкта **Запит_3** оберіть команду **Копіювати**.
 - b. У контекстному меню області об'єктів оберіть команду **Вставити**.
 - c. Уведіть ім'я **Запит_4** та натисніть кнопку **ОК**.



2. Відкрийте **Запит_4** в режимі конструктора.
3. Перемістіть у стовпці **Жанр** критерій «детектив» із рядка **Критерії** до рядка **Або**.
4. Збережіть запит.

Як було зазначено раніше, для створення запитів можна скористатися більше, ніж однією таблицею. Розглянемо такий випадок.



Запит 5. Створити запит, у якому вивести назви усіх книжок, що є в наявності, але тільки тих продавців, рейтинг яких перевищує 4,4.

1. Створіть новий запит в режимі макета та додайте обидві таблиці **Книжки** та **Продавці**. Зверніть увагу, що між таблицями в запиті відображено зв'язок. Для правильного відображення даних у запиті його *не потрібно* змінювати.
2. Додайте поля **Назва** та **Наявність** із таблиці **Книжки**.
3. Із таблиці **Продавці** додаємо поле **Рейтинг**.
4. Для поля **Наявність** укажіть умову **1** та перейдіть до іншого поля. В результаті критерій зміниться на **Істина**. Отже, можна було ввести в якості критерію і слово **Істина**, але для спрощення запису, як ми вже переконалися, можна ввести і число **1**.
5. В рядку критерії для поля **Рейтинг** уведіть умову **>4,4**.
6. Виконайте запит.
7. Оскільки поле **Наявність** завжди виводить однакові значення, його виведення є недоцільним. Отже, перейдіть у режим конструктора та приберіть прапорець для поля **Наявність** у рядку **Відображення**.
8. Збережіть запит під назвою **Запит_5**.



Запит 6. Вивести всю інформацію про продавців, що продають книжки видавництва Богдан.

1. Створіть новий запит в режимі макета та додайте обидві таблиці **Книжки** та **Продавці**.
2. Для виведення усієї інформації з таблиці **Продавці** не потрібно додавати до конструктора кожне поле. Для цього клацніть два рази на поле із зображенням зірочки.
3. Додайте поле **Видавництво** із таблиці **Книжки**.
4. В якості критерію відбору для поля **Видавництво** введіть **Богдан**, приберіть відображення даного поля в запиті (рис. 28.3).
5. Запустіть запит на виконання. В результаті буде відображено вся інформація про продавців книжок видавництва Богдан.

Але зверніть увагу, що деякі записи в запиті повторюються (рис. 28.4). Це відбувається тому, що ці продавці мають у продажу декілька книжок видавництва Богдан. Змінимо налаштування запиту таким чином, аби записи в запиті не повторювалися.



Поле: Продавці.* Видавництво
 Таблиця: Продавці Книжки
 Сортування:
 Відображення: ☒ ☐
 Критерії: "Богдан"

Рис. 28.3

Продавець	Рейтинг	Контактний телефон	Дата реєстрації
Персей	4,5	+38-068-72-11	21.02.2018
Персей	4,5	+38-068-72-11	21.02.2018
GOODSHOCK	5	+38-068-72-12	12.05.2017
GOODSHOCK	5	+38-068-72-12	12.05.2017
GOODSHOCK	5	+38-068-72-12	12.05.2017

Рис. 28.4

- Відкрийте запит у режимі конструктора.
- Оберіть інструмент **Аркуш властивостей**.
- Для властивості **Унікальні значення** встановіть значення **Так**.
- Виконайте запит ще раз і переконайтеся, що тепер усі записи виведено без повторень.
- Збережіть запит під назвою **Запит_6**.

28.2.

Запити з параметрами

Якщо критерій пошуку має ввести сам користувач, під час виконання запиту використовують так званий *запит з параметром*. Так, наприклад, якщо є необхідність вивести назви усіх книжок, які були видані у вказаному користувачем році (рис. 28.5).

Введення значення параметра ? X

Уведіть рік видання

2015

OK Скасувати

Назва	Рік видання
Висотка	2015
Володар Драконів	2015
П'ятеро дітей та ельф	2015

Рис. 28.5

Запит з параметром — це спеціальний тип запиту, що дозволяє користувачеві самому ввести критерій відбору даних на етапі запуску запиту.

ВПРАВА 28.2



Завдання. Створити запити з параметром у базі даних **Інтернет-магазин**.



Запит 7. Вивести назви усіх книжок, які були видані у вказаному користувачем році.

- Створіть новий запит в режимі макета та додайте до нього таблицю **Книжки**.
- До конструктора запиту додайте поля **Назва** та **Рік видання**



Поле:	Назва	Рік видання
Таблиця:	Книжки	Книжки
Сортування:		
Відображення:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Критерій:		[Уведіть рік видання]

Рис. 28.6

3. Аби при виконанні запиту **Рік видання** його міг вводити користувач, у рядку **Критерій** поля **Рік видання** запишіть повідомлення **[Уведіть рік видання]** (рис. 28.6).

4. Запустіть запит на виконання. В результаті програма виведе вікно із запитом на введення року видання.

5. Введіть, наприклад, 2018.

6. Якщо виведено записи із повтореннями, то для виведення унікальних записів у вікні властивостей запиту для властивості **Унікальні значення** встановіть значення **Так**.

При створенні таких запитів зверніть увагу, що критерій пошуку не повинен співпадати з назвою одного з полів таблиць. Так, наприклад, замість фрази **Уведіть рік видання** не можна написати просто **Рік видання**, оскільки в цьому випадку при запуску запиту вікно запиту виведено не буде, а буде виведено усі записи таблиці.

7. Збережіть запит під назвою **Запит_7**.



Запит 8. Знайти всю інформацію про книжки за фрагментом назви книжки, який введе користувач.

1. Створіть новий запит в режимі макета та додайте до нього таблицю **Книжки**.

2. З даної таблиці додайте усі поля обравши поле із зірочкою.

3. Також додайте поле **Назва** для формування критерію відбору.

4. Зніміть навпроти даного поля прапорець у рядку **Відображення**, оскільки ми вже зазначили його відображення, обравши поле із зірочкою.

5. У рядку **Критерій** поля **Назва** відкрийте побудовник виразів.

6. У побудовнику введіть умову: **Схоже «*» & [Уведіть фрагмент назви для пошуку] & «*»** та натисніть **ОК**.

7. Запустіть запит на виконання. В результаті має з'явитися вікно повідомлення, що пропонує ввести фрагмент назви для пошуку.

8. Уведіть, наприклад, фрагмент **«вір»** та натисніть **ОК**. В результаті буде виведено усі книжки, що у своїй назві мають вказаний фрагмент.

Розберемо детальніше, що означає введений нами критерій пошуку.

Схоже «*» & [Уведіть фрагмент назви книжки] & «*»

Команда **Схоже** буде здійснювати пошук фрази, що буде відповідати масці, вказаній після неї. Замість напису **[Уведіть фрагмент назви для пошуку]** буде підставлено введені користувачем символи.



Зірочки, що розташовані в лапках на початку і в кінці фрази означають будь-яку кількість будь-яких символів. А з допомогою символів & (амперсент) довільні символи і символи введені користувачем з'єднуються між собою. Отже, ми отримаємо текст, що містить спочатку будь-які символи, далі — символи, введені користувачем, а потім ще будь-які символи.

9. Збережіть запит під назвою **Запит_8**.



Виконайте вправу 28.3.



Контрольні запитання та завдання

1. Що таке запит?
2. Як створити запит у режимі макета?
3. Чи можна створити запит за даними декількох таблиць?
4. Що таке запит з параметром?



Питання для роздумів

- 1*. В яких випадках записи у запиті можуть повторюватися? Як позбутися цієї проблеми?
- 2*. Який існує раціональний спосіб виведення усіх полів таблиці в запиті?
- 3*. Які переваги пошуку та відбору даних в середовищі Access над Excel?



Завдання для досліджень

- 1*. Розгляньте можливості майстра побудови запитів.
- 2**. Дослідіть, що таке запити на створення таблиці, додавання, оновлення та видалення записів.



ПРАКТИЧНА РОБОТА 7. «СТВОРЕННЯ ЗАПИТІВ»



Завдання. Створити запити в базі даних **Готель**.



Запит 1. Вивести всю інформацію про вільні номери готелю.



Запит 2. Вивести номери вільних кімнат, ціна за добу в яких від 1000 грн до 1500 грн.



- ☒ **Запит 3.** Вивести номери та тип кімнат, в яких кількість місць більша за 2 та ціна за добу менша від 1500 грн.
- ☒ **Запит 4.** Вивести номери та тип кімнат, в яких кількість місць більша за 2 або ціна за добу менша від 1500 грн.
- ☒ **Запит 5.** Вивести всю інформацію про відвідувачів, що проживають в номерах класу економ.
- ☒ **Запит 6.** Вивести прізвище та дату народження відвідувачів, які за добу проживання в готелі сплачують від 1500 грн до 2000 грн.
- ☒ **Запит 7.** Вивести всю інформацію про відвідувачів, що проживають в номері, який введе користувач.
- ☒ **Запит 8.** Вивести номер кімнати, ціну за добу та зайнятість номерів, що мають кількість місць стільки або більше, ніж увів користувач.
- ☒ **Запит 9*.** Вивести прізвище, ім'я, дату народження та номер кімнати відвідувача, що народився в тому місяці, який уведе користувач.



Висновки

База даних — це структурована сукупність даних, які відображають стан об'єктів певної предметної області та зв'язки між ними.

Система керування базами даних (СКБД) — це програма, що забезпечує можливість створення бази даних та виконання різноманітних операцій з даними, які в ній зберігаються. Однією із таких з таких СКБД є MS Access.

Дані в базі даних зберігаються у вигляді *таблиць*. Стовпці таблиці називають *полями*, рядки — *записами*.

Створення та редагування структури таблиці здійснюється в *режимі конструктора*. Введення та редагування даних таблиці здійснюється в *режимі таблиці*.

Для створення у базі даних зв'язку **Один-до-багатьох** в деякій таблиці **В** необхідно створити зовнішній ключ, який приймає значення первинного ключа таблиці **А**.

Access має можливості сортування даних у полі за зростанням та за спаданням. Для створення простих умов відбору використовують *фільтр*, а для більш складних — *розширений фільтр*.

Запити — це спеціальні об'єкти, призначені для вибірки даних з таблиць бази за заданою умовою.



Розділ 4

Алгоритми та програми



§30.

ПОНЯТТЯ ОДНОВИМІРНОГО МАСИВУ

30.1.

Поняття масиву

У даному розділі ми продовжимо вивчення мови Python, дізнаємося про поняття та використання масивів.

Масив — це впорядкований набір елементів одного типу зі спільним ім'ям.

Так, на рис. 30.1 наведено приклад масиву. Тут **A** — це спільне ім'я масиву, *i* — змінна індексів елементів масиву, де *індекс* — це порядковий номер елемента масиву, відповідно він може бути лише цілим числом. Числа, наведені в другому рядку, є елементами масиву.

Індекс	i	0	1	2	3	4
Ім'я	A	4	-9	7	3	8
		Елементи масиву				



Пригадайте основні поняття та оператори мови Python.

Рис. 30.1. Приклад масиву

У мові Python для створення масивів використовують *списки*. Проте між масивами та списками є принципова різниця, яка полягає в тому, що елементи одного масиву мають бути одного типу, наприклад, цілі числа, а в списках елементи можуть бути будь-якого



типу. Так, список може містити як числа, так і значення текстового типу.

Приклади списків:

```
A=[13, 25, 81, 15]
B=[3, 'Hello', 8, 5]
```

30.2. Способи введення та виведення списків

1 спосіб

Існує декілька способів створення списків. Найпростіший спосіб — це присвоєння змінній елементів списку, де елементи списку записуються в квадратних дужках через кому.

```
змінна = [елемент_1, елемент_2, ..., елемент_n]
```

Вивести такий список доволі просто у вікні консолі. Для цього використовується оператор **print**, аргументом якого є змінна, що має ім'я списку. Нижче наведено приклад створення та виведення списку планет сонячної системи.

```
#створення списку
planet=['Меркурій','Венера','Земля','Марс','Юпітер',
        'Сатурн','Уран','Нептун']
#виведення списку
print(planet)
```

Якщо необхідно вивести не весь список, а один із його елементів, в якості аргументу оператора **print** вказують ім'я списку, а в квадратних дужках — номер цього елемента. Наприклад, `print(planet[2])`. Отже, після виконання цієї команди буде виведено значення **Земля**.

З даного прикладу неважко здогадатися, що нумерація елементів у списку починається з нуля.

ПРАКТИЧНЕ ЗАВДАННЯ 1

1. Уведіть програмний код створення та виведення списку планет.
2. Запустіть програму та перевірте, як вона працює.
3. Змініть команду виведення списку на команду виведення 2-го елемента списку.
4. Перевірте, як буде працювати програма тепер.

Для виведення кожного елемента списку з нового рядка потрібно скористатися оператором циклу:



```
#виведення списку
for i in range(len(planet)):
    print(planet[i])
```

Тут **len** — функція визначення розмірності списку, відповідно **len(planet)** визначає розмірність списку **planet** — в даному випадку це список з восьми елементів, отже, значення функції **len(planet)=8**.

Аргументом оператора **print** є звернення до елемента масиву, в даному випадку **planet[i]**, де **planet** — ім'я списку, **i** — його індекс, тобто порядковий номер елемента списку. Оскільки нумерація елементів списку починається з нуля, цикл буде виконуватися від 0-го до 7-го елемента включно.

З метою виведення списку у стовпець можна скористатися і більш спрощеною конструкцією:

```
for i in planet:
    print(i)
```

Тут **i** — це поточне значення елемента списку. Останній спосіб зручно використовувати, коли в задачі використовується лише один список.

ПРАКТИЧНЕ ЗАВДАННЯ 2

1. Змініть команду виведення списку планет на команди виведення елементів цього ж списку у стовпець.
2. Запустіть програму та перевірте, як вона працює.

2 спосіб

Уводити елементи до списку може також користувач під час виконання програми. Для цього слід скористатися оператором **input**.

Так, розглянемо приклад уведення користувацького масиву, що складається з п'яти цілих чисел.

```
print("Уведіть масив із п'яти чисел")
#створення масиву
mas=[]
#введення елементів масиву
for i in range(5):
    a=int(input('mas['+str(i)+']='))
    mas.append(a)
#виведення масиву
print(mas)
```



Тут команда `mas=[]` означає створення порожнього списку.

Далі використовується оператор циклу `for i in range(5):`, з допомогою якого можна ввести п'ять елементів списку.

Наступна команда `a=int(input('mas['+str(i)+']=''))` дозволяє ввести ціле число та присвоїти його значення змінній `a`. Так, в результаті виконання цієї команди буде сформовано напис `mas[i]=`, де замість `i` буде вказано поточний номер елемента списку. Після введення числа та натиснення клавіші **Enter** змінна `a` одержить уведене користувачем число. Тут функція `str` переводить порядковий номер, що є цілим числом, у текстовий тип. Це потрібно для формування текстового напису, що є запитом на введення елемента списку. Відповідно функція `int` переводить уведене користувачем значення в цілий тип.

Для додавання в кінець списку значення елемента використовують метод `append`. Так, у команді `mas.append(a)` в кінець списку `mas` додається значення змінної `a`.

ПРАКТИЧНЕ ЗАВДАННЯ 3

1. Уведіть програмний код створення та виведення списку з п'яти цілих чисел.
2. Запустіть програму. Введіть з її допомогою список чисел 7, 3, 5, 4, 1.

3 спосіб

Ще один спосіб введення елементів масиву — заповнення його за формулою. Так, нижче наведено приклад заповнення масиву за формулою $2i^2+1$, де i — порядковий номер елемента.

```
#введення елементів масиву
for i in range(5):
    a=2*i**2+1
    mas.append(a)
```

Як бачимо, відмінність даного коду від попереднього лише у команді присвоєння значення змінній `a`. Саме їй присвоюється вираз, що відповідає вказаній в умові формулі.

ПРАКТИЧНЕ ЗАВДАННЯ 4

1. Змініть попередній програмний код на створення списку за формулою $2i^2+1$.
2. Запустіть програму та перевірте, як вона працює.



4 спосіб

Ще один спосіб створення масиву — це заповнення його випадковими числами. Його використовують для економії часу на введення елементів масиву, коли значення елементів не є важливими.

Нижче наведено приклад такого масиву, що має 10 елементів. Значення елементів належать проміжку від 1 до 100.

```
A=[3, 23, 45, 65, 78, 12, 43, 55, 77, 98]
```

Для створення подібного списку можна застосувати такий код:

```
from random import*
mas=[]
for i in range(10):
    a = randint(1,100)
    mas.append(a)
print(mas)
```

Так, у першому рядку для використання випадкових чисел слід виконати імпорт модуля **random**. Як і в попередніх прикладах, потрібно створити порожній список **mas**, в операторі циклу змінній *a* надати значення випадкового числа, що лежить в проміжку від 1 до 100, і з використанням методу **append** одержане значення додати в кінець списку.

ПРАКТИЧНЕ ЗАВДАННЯ 5

1. Змініть попередній програмний код на створення списку 10 випадкових чисел у проміжку від 1 до 100.
2. Запустіть програму та перевірте, як вона працює.

30.3.

Створення масиву засобами об'єктно-орієнтованого програмування

Введення та виведення елементів масиву можна організувати також засобами об'єктно-орієнтованого програмування.

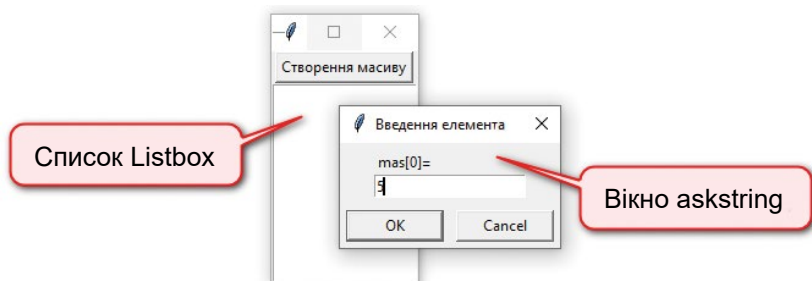
1 спосіб

Для введення елементів списку зручно використовувати вікна введення даних **askstring**. Так, після натискання у вікні програми кнопки **Створити масив** має з'явитися вікно, в якому слід увести вказаний елемент масиву.

Виводити елементи масиву краще всього до такого елемента керування, як *список*, що створюється з допомогою конструктора **Listbox**.



Розглянемо, як увести масив із використанням вікон введення даних **askstring** та виведенням його до елемента керування **Listbox**, виконавши практичне завдання.



ПРАКТИЧНЕ ЗАВДАННЯ 6

1. Уведіть команди створення інтерфейсу графічного вікна.

```
from tkinter import*
#імпорт модуля вікон введення даних
from tkinter.simpledialog import*
window=Tk()
#кнопка
Btn=Button(text="Створення масиву")
Btn.pack()
#список
Lbox=Listbox()
Lbox.pack()
```

2. Додайте команди обробника події натискання кнопки **Створення масиву**.

```
mas=[]
def btn_click():
    #введення елементів
    for i in range(5):
        a = int(askstring('Введення елемента',
            'mas['+str(i)+'']='))
        mas.append(a)
    #виведення масиву
    for i in range(5):
        Lbox.insert(END,mas[i])
```



Тут команда

```
a = int(askstring('Введення елемента', 'mas['+str(i)+'']='))
```

створює вікно введення даних, у якому в якості заголовка використовується напис **Введення елемента**, а в якості запрошення утворюється повідомлення щодо введення елемента за номером, указаним у дужках. Значення, введене користувачем у даному вікні, буде перетворено на цілий тип завдяки функції **int** та присвоєно змінній **a**.

Виведення масиву здійснюється в окремому циклі, що має такий же діапазон, що і цикл для введення елементів масиву. Для додавання елементів до об'єкта **Lbox** використовується метод **insert**, в якому перший аргумент **End** указує позицію вставлення елементу масиву, а саме, кінець, другий аргумент — значення елемента масиву **mas**, що розташований у масиві за вказаним індексом **i**.

3. Додайте команду виклику обробника події в конструкторі кнопки:

```
Btn=Button(text="Створення масиву", command=btn_click)
```

4. Запустіть програму та перевірте, як вона працює.

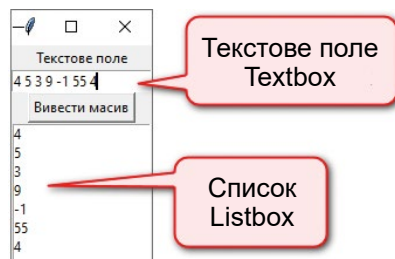
Після запуску програми та натискання кнопки **Створити масив** з допомогою вікна введення даних ви послідовно введете елементи масиву. Після введення останнього елемента усі вони будуть відображені у списку **Listbox**.

- 5*. Спробуйте змінити програму таким чином, аби кожний елемент масиву з'являвся у списку відразу після його введення до вікна **askstring**.

2 спосіб

Для введення елементів масиву в об'єктно-орієнтовному програмуванні також використовують *текстові поля* (**Textbox**).

Розглянемо введення елементів масиву через текстове поле, виконавши практичне завдання.



ПРАКТИЧНЕ ЗАВДАННЯ 7

1. Уведіть команди створення інтерфейсу графічного вікна.

```
from tkinter import*
window=Tk()
mas=[]
#напис
Lbl=Label(text='Текстове поле')
```

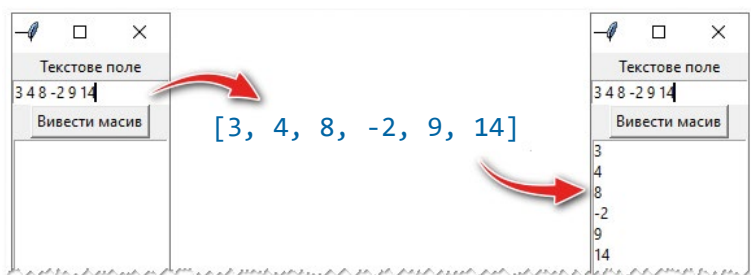


```
Lb1.pack()
#текстове поле
Ent=Entry()
Ent.pack()
#кнопка
Btn=Button(text="Вивести масив")
Btn.pack()
#список
Lbox=Listbox()
Lbox.pack()
```

2. Додайте команди створення порожнього списку **mas** та обробника події натиснення кнопки **Створення масиву**.

```
mas=[]
def btn_click():
    #введення елементів
    a = Ent.get()
    mas = a.split()
    #виведення масиву
    for i in mas:
        Lbox.insert(END,i)
```

Розглянемо обробник події натиснення кнопки. Значення, введені користувачем до текстового поля, мають бути зчитані, перетворені на список та виведені до елемента керування списком у вигляді окремих елементів.



В результаті виконання команди `a = Ent.get()` завдяки методу **get** змінна `a` одержить значення, введені користувачем до текстового поля, а саме, числа, введені через пробіл.

Наступна команда `mas = a.split()` з допомогою методу **split** перетворює рядок даних, записаних через пробіл, на список, значення якого присвоюється змінній **mas**.



3. Запустіть програму на виконання.
4. В текстовому полі введіть через пробіл елементи списку.
5. Натисніть кнопку **Створити список**. У результаті введені до текстового поля елементи списку буде відображено в елементі **Listbox**.



Контрольні запитання та завдання

1. Що таке масив?
2. Які існують способи введення масиву користувачем?
3. Як вивести масив до вікна консолі у рядок; у стовпець?
4. Як ввести елементи масиву випадковим чином?



Питання для роздумів

- 1*. Яким може бути програмний код для виведення списку п'яти випадкових чисел, наприклад, `[0.5, 12.3, 3.6, 7.8, -5.4]`?
- 2*. Яким може бути програмний код для виведення двох списків, елементи яких утворюються з використанням формул $2i - 3i^2$ та $2i^3 + 4i^2 - 5i$ відповідно?



Завдання для досліджень

- 1*. Дізнайтеся, як увести елементи списку з використанням оператора **input** без використання циклу.
- 2*. Дізнайтеся, як увести елементи списку з використанням текстового поля, розділюючи елементи списку комою.



§31. ДІЇ ЗІ СПИСКАМИ

Ми вже познайомилися із способами створення списків, навчилися їх виводити. Розглянемо, які дії можна виконувати зі списками, а також функції та методи опрацювання списків.

31.1.


Об'єднання та виділення списків

Одна із дій, яку можна виконати зі списками, — це *об'єднання списків*.



Так, якщо ми маємо список `a=[1, 2, 3]` та список `b=[4, 5]`, то для їхнього об'єднання достатньо до списку **a** додати список **b**. В результаті ми одержимо новий список **c**, в якому спочатку будуть розташовані елементи списку **a**, а потім елементи списку **b**.

```
a=[1,2,3]
b=[4,5]
c=a+b
print(c)
```




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

Ще одна дія — це *виділення елементів*. Загальний вигляд цієї дії має такий синтаксис:

`Ім'я_списку(початок:кінець:крок)`

Усі параметри цієї дії є необов'язковими. Так на рис. 31.1 наведено приклад коду та його виконання для виділення зі списку елементів з 2-го по 5-й включно.

```
a=[1,3,5,7,9,11,13,15,17]
b=a[2:6]
print(b)
```




```
[5,7,9,11]
```

Рис. 31.1

А на рис. 31.2 наведено приклад виділення зі списку елементів, що мають непарні індекси. Зверніть увагу, що кінцевий індекс в даному виразі пропущений.

```
a=[1,3,5,7,9,11,13,15,17]
b=a[1::2]
print(b)
```



```
[3,7,11,15]
```

Рис. 31.2

31.2.

Методи опрацювання списків

Вам уже знайомий метод **append** — з його допомогою елемент можна додати в кінець списку.

Якщо ж до списку необхідно додати відразу декілька елементів, то для цього використовують метод **extend**.

Так, на рис. 31.3 наведено приклад додавання чисел 4 та 5 у кінець даного списку **a**.



```
a=[1,2,3]
a.extend([4,5])
print(a)
```



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

Рис. 31.3

Якщо необхідно елемент додати не в кінець списку, а до вказаної позиції використовують метод **insert**.

На рис. 31.4 наведено приклад додавання числа 0 у 3-тю позицію до списку.

```
a=[1,1,1,1,1,1,1,1]
a.insert(3,0)
print(a)
```



```
[1,1,1,0,1,1,1,1]
```

Рис. 31.4

Наступні методи призначені для видалення елементів. Так, метод **pop** видаляє елемент за вказаним індексом (рис. 31.5).

```
a=[1,1,0,1,1,1,1,1]
a.pop(2)
print(a)
```



```
[1,1,1,1,1,1,1]
```


Рис. 31.5

Видалити елементи можна також із використанням оператора **del**, що має такий синтаксис:

```
del ім'я_списку[номер_позиції]
```

У квадратних дужках можна вказати номер елемента або діапазон елементів для видалення. На рис. 31.6 наведено приклад видалення елементів, що знаходяться від 2-ї до 5-ї позиції включно.

```
a=[1,2,3,4,5,6,7,8]
del a [2:6]
print(a)
```



```
[1,2,7,8]
```

Рис. 31.6

Для видалення першого елемента, що має вказане в якості аргументу значення, використовується метод **remove**. На рис. 31.7 наведено приклад видалення першого елемента, що має значення нуль.



```
a=[1,0,1,0,1,0,1,0]  
a.remove(0)  
print(a)
```



```
[1,1,0,1,0,1,0]
```

Рис. 31.7

Якщо ж необхідно очистити від значень увесь список, використовують метод **clear**.

Якщо необхідно знайти номер першого входження елемента, який має вказане значення, використовують метод **index**. Так, на рис. 31.8 наведено приклад пошуку номера першого входження елемента, що має значення 5.

```
a=[3,4,2,5,3,5]  
b=a.index(5)  
print(b)
```



```
3
```

Рис. 31.8.

Для визначення кількості елементів, що мають вказане значення використовують метод **count**. На рис. 31.9 наведено приклад пошуку кількості входжень елемента, що має значення 5.

```
a=[3,4,2,5,3,5]  
b=a.count(5)  
print(b)
```



```
2
```

Рис. 31.9

31.3. Функції роботи зі списками

Крім розглянутих методів роботи зі списками є ще й функції. Функції, які ми будемо використовувати при подальшому створенні програм, наведено в таблиці 31.1.

Таблиця 31.1

Функція	Призначення	Приклади
sum(список)	Обчислення суми елементів	sum([4, 7, 3, 6])= 20
max(список)	Визначення максимального елемента	max([4, 7, 3, 6])= 7
min(список)	Визначення мінімального елемента	min([4, 7, 3, 6])= 3



Розглянемо використання функцій та методів роботи зі списками при створенні програм.

ВПРАВА 31.1



Завдання. На змаганнях з фігурного катання спортсмен одержує оцінку свого виступу від 10-и суддів. Оцінка, яку може виставити один суддя, визначається кількістю балів від 0 до 6. Після одержання усіх оцінок найкраща та найгірша оцінки відкидаються, і спортсмену нараховується сумарний бал оцінок, що залишилися. Створити вихідний масив оцінок від 10-ти суддів, заповнюючи його елементи випадковим чином дійсними числами від 0 до 6 з одним десятковим знаком. Вивести:

- одержаний список оцінок;
- максимальну та мінімальну оцінки;
- список оцінок, серед яких буде визначено результат;
- сумарний бал, який одержить спортсмен.

1. Уведіть команди створення та виведення масиву оцінок суддів.

```
from random import*
graid=[]
for i in range(10):
    a=round(uniform(0,6),1)
    graid.append(a)
print('Оцінки суддів:')
print(graid)
```

У першому рядку здійснено імпорт модуля **random**, оскільки ми створюємо список із випадкових чисел. У наступному рядку створюємо порожній список **graid**, до якого буде додано оцінки суддів. Наступна команда — *оператор циклу*, для введення 10 елементів списку. З допомогою команди **a=round(uniform(0,6),1)** змінній **a** надається дійсне випадкове число в проміжку від 0 до 6, яке округлюється до одного десяткового знака за допомогою функції **round**. Наступна команда **graid.append(a)** додає значення одержаного випадкового числа в кінець списку **grade**. Останні дві команди виводять повідомлення та список оцінок суддів.

2. Запустіть програму на виконання та переконайтеся, що створено список із 10 дійсних випадкових чисел.
3. Додайте команду знаходження максимальної оцінки суддів:
max_graid=max(graid)
Самостійно визначте місце її розташування в програмі.
4. Аналогічно додайте команду знаходження мінімальної оцінки суддів.
5. Із допомогою оператора **print** виведіть знайдені значення.



6. Додайте команду видалення зі списку **graid** максимального значення:
`graid.remove(max_graid)`
7. Аналогічно додайте команду видалення зі списку **graid** мінімального значення.
8. Додайте команду виведення одержаного списку.
9. Додайте команду обчислення сумарного балу спортсмену:
`s=round(sum(graid),1)`
У даній команді функція **sum** використовується для знаходження суми елементів масиву **graid**, а функція **round** — для округлення значення суми до одного десяткового знака.
10. Додайте команду для виведення значення суми.
11. Запустіть програму та перевірте правильність її виконання.

ВПРАВА 31.2



Завдання. Дано два списки, кожен з яких містить середньодобову температуру протягом тижня (рис. 31.10). Об'єднати ці списки в один, знайти максимальну, мінімальну та середню температури за два тижні. Створити та вивести новий список, що містить температуру лише у вихідні дні.

	понеділок	вівторок	середа	четвер	п'ятниця	субота	неділя
тиждень 1	12.5	13.8	14.9	25.3	17.8	21.4	23.4
тиждень 2	22.3	18.7	19.2	21.5	16.1	26.2	22.9



Рис. 31.10

1. Уведіть команди створення та об'єднання списків:
`week1=[12.5, 13.8, 14.9, 25.3, 17.8, 21.4, 23.4]`
`week2=[22.3, 18.7, 19.2, 21.5, 19.1, 26.2, 22.9]`
`week=week1+week2`
2. Додайте команду виведення об'єднаного списку.
3. Додайте команди знаходження і виведення максимального та мінімального значень списку **week**.

Середнє значення знайдемо за формулою $Sa=s/k$, де s — сума значень, k — кількість. Для знаходження суми скористаємося функцією **sum**, а кількість є відомою, $k=14$. Для округлення результату середнього значення до одного десяткового знака скористаємося функцією **round**.



4. Додайте команду знаходження середнього значення списку **week**:

```
Sa=round(sum(week)/14,1)
```

5. Додайте команду виведення середнього значення списку **week**.
6. Додайте команди створення списку **weekend**, що містить значення температур у вихідні дні, а саме, в суботу та неділю:

```
weekend=[]  
weekend.extend([week1[5],week1[6],week2[5],week2[6]])
```

Тут метод **extend** використовується для додавання до списку відразу декількох елементів.

7. Додайте команду виведення списку **weekend**.
8. Запустіть програму та перевірте правильність її виконання.
9*. Змініть команду створення списку **weekend** таким чином, щоб до нього додавалися не окремі елементи списків, а їхній діапазони.
10*. Також спробуйте реалізувати цю команду таким чином, аби отримати список **weekend** не з двох даних списків **week1** та **week2**, а зі списку **week**.



Контрольні запитання та завдання

1. Як виконати об'єднання списків?
2. Які існують методи додавання елементів до списку?
3. Назвіть можливі способи видалення елемента зі списку.
4. Як знайти максимальний та мінімальний елементи списку?
- 5*. Як знайти середнє значення елементів списку?



Питання для роздумів

- 1*. Укажіть, як можна перевірити, чи є в списку певний елемент.
- 2*. Визначте, яким має бути вираз для виділення елементів, що мають парні індекси.
- 3*. Яким має бути команда знаходження середнього арифметичного між максимальним та мінімальним елементами списку?



Завдання для досліджень

- 1*. Дізнайтеся, які ще існують методи роботи зі списками.





§32.

ПРАКТИЧНА РОБОТА 8.
«ДІЇ З МАСИВАМИ»

У даній практичній роботі розглянемо використання функцій та методів роботи зі списками при створенні об'єктно-орієнтованих програм.

ЗАВДАННЯ 1



Дано список прізвищ учасників олімпіади з інформатики та результати їхньої участі в олімпіаді, що мають значення від 0 до 100 у вигляді десяткового дробу з одним десятковим знаком. Визначити:

- 1) найвищий бал та прізвище учня, що має найвищий бал;
- 2) найнижчий бал та прізвище учня, що має найнижчий бал;
- 3) середній бал усіх учасників олімпіади.

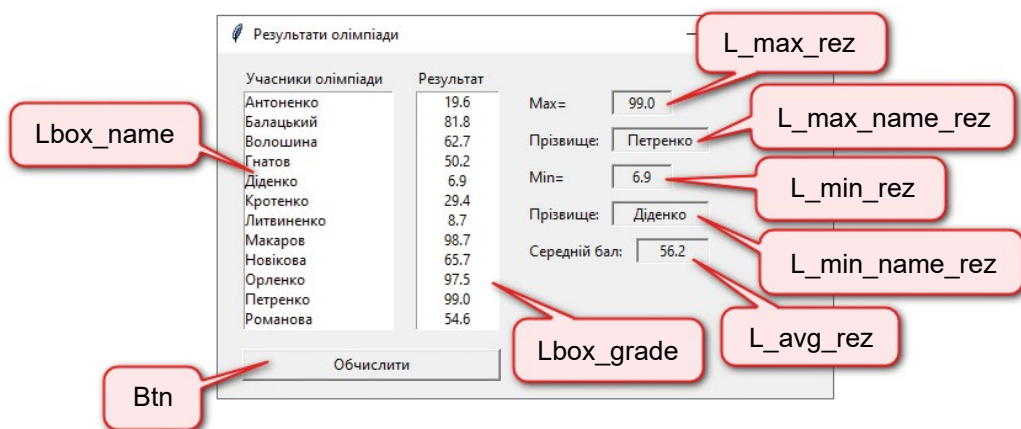


Рис.32.1

1. Завантажте та відкрийте файл **Завдання_1.py** із матеріалів до даного параграфа.
Даний файл містить список учасників олімпіади та елементи інтерфейсу для введення та виведення даних. Аби ви могли орієнтуватися в наданому програмному коді, він містить деякі коментарі, а на рис. 32.1 у виводах підписано імена елементів керування, що використовуються для введення та виведення даних у програмі.
2. Додайте команди виведення масиву **name** до елемента керування **Lbox_name**:

```
#розмірність списку  
size=len(name)
```



```
for i in range(size):
    #додавання елементу масиву в кінець Lbox_name
    Lbox_name.insert(END,name[i])
```

Створення масиву результатів олімпіади

Створимо масив результатів олімпіади, що має ім'я **grade** та виведемо даний масив до елемента керування **Lbox_grade**. Для прискорення введення кількості балів, одержаних за участь в олімпіаді, заповнимо список балів випадковим чином.

3. Додайте команду імпорту модуля **random**.

4. Додайте команду створення порожнього масиву **grade**.

Подальше створення масиву та виведення значень його елементів продовжимо виконувати у тілі того ж самого циклу, що і для виведення елементів масиву **name**.

5. Додайте команду створення дійсного випадкового числа, що належить проміжку від 0 до 100:

```
a=round(uniform(0,100),1)
```

6. Уведіть команду додавання значення змінної *a* в кінець списку **grade**:

```
grade.append(a)
```

7. Уведіть команду додавання утвореного елемента масиву **grade** в кінець елемента керування **Lbox_grade**:

```
Lbox_grade.insert(END,grade[i])
```

8. Запустіть програму та переконайтеся, що обидва списки створено.

Знаходження максимального і мінімального балів та відповідних прізвищ

Далі знайдемо максимальний бал результатів олімпіади та виведемо прізвище учасника, що має цей результат. Оскільки цей результат буде знайдено після натиснення кнопки **Обчислити**, створимо обробник події натискання кнопки **btn_click**.

9. Уведіть обробник події кнопки для знаходження максимального результату.

```
def btn_click():
    #знаходження максимального значення
    max_grade=max(grade)
    #виведення максимального значення до графічного вікна
    L_max_rez['text']=max_grade
```

10. Додайте команду визначення позиції входження максимального значення до списку результатів: `max_grade_i=grade.index(max_grade)`



Тут метод **index** дозволяє визначати номер входження елемента керування за його значенням. **grade** — це масив, у якому здійснюється пошук, **max_grade** — значення, яке необхідно знайти.

11. Оскільки прізвище учасника олімпіади, що одержав найкращий результат, знаходиться під тим же номером, що і його результат в масиві **grade**, то для виведення прізвища введіть таку команду: `L_max_name_rez['text']=name[max_grade_i]`

Дана команда означає, що властивість тексту елемента керування **L_max_name_rez** буде замінено на значення елемента масиву **name** розташоване під номером **max_grade_i**, де **name** — це список учасників олімпіади, **max_grade_i** — номер розташування найкращого результату.

12. Додайте виклик обробника події в конструкторі кнопки.
13. Запустіть програму на виконання, натисніть кнопку **Обчислити** та переконайтеся, що максимальний результат та прізвище учасника, що має цей результат, знайдено.



Додатково виконайте
Завдання 2.

14. Самостійно додайте команди обчислення та виведення мінімального результату та учасника, що має цей результат, а також середнє значення усіх учасників олімпіади.



§33.

ЗНАХОДЖЕННЯ ПІДСУМКОВИХ ВЕЛИЧИН

Досить часто при роботі з масивами виникає необхідність обчислення суми, добутку та кількості елементів масиву, що задовольняють деяку умову. Як знаходити такі величини, розглянемо далі, розв'язуючи прикладні задачі.

ВПРАВА 33.1



Завдання. Дано прибутки та збитки деякої фірми за кожний місяць поточного року. Визначити:

- суму прибутків та суму збитків;
- прибутковою чи збитковою була робота фірми за підсумками року і якою була сума прибутку чи збитку;
- кількість місяців, коли фірма мала збитки;
- кількість місяців, коли прибуток був більший за 50000.

Створення масиву прибутків та збитків

Для зручності створення масиву прибутків та збитків заповнимо його випадковими числами в діапазоні від -100000 до 100000 .



1. Уведіть команду імпорту модуля **random**.
2. Додайте команду створення порожнього списку **d**, у якому будемо формувати масив доходів фірми.
3. Додайте оператор циклу для створення масиву 12 випадкових чисел, що відповідають кількості місяців у році.

```
for i in range(12):
    a=randint(-100000,100000)
    d.append(a)
    print(d[i])
```

4. Запустіть програму на виконання та переконайтеся, що буде виведено масив із 12-ти випадкових чисел, що належать діапазону від -100000 до 100000 .

Сума прибутків та сума збитків

Наступним етапом є визначення суми прибутків та суми збитків. Суму прибутків позначимо змінною *sp*, а суму збитків — *sz*.

5. Додайте команду встановлення початкового значення для суми прибутків, рівного 0.
6. Додайте оператор циклу знаходження суми прибутків:

```
for i in range(12):
    if d[i]>0:
        sp=sp+d[i]
```

Тут оператор розгалуження перевіряє умову, чи є значення елемента масиву додатним. Якщо умова виконується, то значення суми прибутку **sp** збільшується на значення елемента масиву **d[i]**.

7. За межами тіла циклу додайте команду виведення значення суми прибутку.
8. Аналогічно додайте команди обчислення суми збитків **sz**.
 - a. Перед попереднім циклом установіть початкове **sz=0**.
 - b. У тілі циклу, в якому було знайдено суму прибутків, додайте команди обчислення суми збитків:

```
if d[i]<0:
    sz=sz+d[i]
```

9. Додайте команду виведення суми збитків **sz**.
10. Запустіть програму та перевірте правильність її виконання.

Визначення прибутковості чи збитковості роботи фірми

Далі визначимо, прибутковою чи збитковою була робота фірми за підсумками року і якою була сума прибутку чи збитку.



11. Надайте змінній r значення різниці між сумою прибутку та збитку $r = sp - abs(sz)$.
12. Додайте оператор розгалуження: якщо значення r є додатним, то виведіть повідомлення, що фірма є прибутковою, інакше — збитковою:

```
if r>0:
    print('Фірма є прибутковою, прибуток фірми за рік
          становить:', r)
else:
    print('Фірма є збитковою, збиток фірми за рік
          становить:', abs(r))
```

Тут функція **abs** дозволяє вивести абсолютне значення r .

13. Запустіть програму та перевірте правильність її виконання.

Визначення кількості місяців, коли фірма мала збитки

Далі визначимо кількість місяців, коли фірма мала збитки. Кількість збитків позначимо змінною kz . Пошук кількості збиткових місяців виконаємо в тілі того ж циклу, в якому виконували обчислення суми прибутків та збитків.

14. Перед оператором тіла циклу додайте команду $kz=0$, що визначає початкове значення змінної kz .
15. У тілі оператора розгалуження, що перевіряє чи є значення елемента масиву від'ємним, додайте команду збільшення значення змінної kz на 1.

```
if d[i]<0:
    sz=sz+d[i]
    kz=kz+1
```

16. За межами тіла циклу додайте команду виведення значення кількості збиткових місяців.
17. Запустіть програму та перевірте правильність її виконання.

Визначення кількості місяців, коли фірма мала прибуток більший за 50000

Далі визначимо кількість місяців, коли фірма мала прибуток більший за 50000. Кількість таких місяців позначимо змінною kp . Пошук кількості прибуткових місяців виконаємо в тілі того ж циклу.

18. Додайте команду $kp=0$, для встановлення початкового значення змінної kp .
19. У тілі циклу додайте оператор розгалуження: якщо значення елемента масиву більше за 50000, то збільшіть значення змінної kp на 1:

```
if d[i]>50000:
    kp=kp+1
```



20. Додайте команду виведення значення змінної kp після виконання тіла циклу.
21. Запустіть програму та перевірте правильність її виконання.
22. Самостійно додайте команди визначення:
 - кількості місяців коли прибуток становив від 50000 до 100000;
 - суми, одержаної в літні місяці;
 - прибутковості чи збитковості роботи фірми зимою.

33.1.**Знаходження суми та кількості елементів****Знаходження суми**

Отже, для знаходження та виведення суми елементів, що задовольняють указану умову, використовується така загальна конструкція:

```
#встановлення початкового значення
s=0
#використання оператора циклу
for i in range():
    #перевірка умови
    if умова:
        #знаходження суми
        s=s+ім'я_списку[номер_елементу]
#виведення значення суми
print(s)
```

Спочатку сумі надається початкове значення 0. Оскільки сума обчислюється в циклі, наступним кроком є оператор циклу. Далі здійснюється перевірка умови. У разі виконання умови обчислюється сума за таким правилом: до попереднього значення суми додається значення елемента масиву. Виведення значення суми відбувається за межами тіла циклу.

Знаходження кількості

Для знаходження кількості елементів, що задовольняють указану умову, буде застосовано таку конструкцію:

```
#встановлення початкового значення
k=0
#використання оператора циклу
for i in range():
    #перевірка умови
    if умова:
        #знаходження кількості
```



```
k=k+1
#виведення значення кількості
print(k)
```

Дана конструкція є аналогічною до знаходження суми. Відмінною є лише формула знаходження кількості.

33.2. Обчислення добутку елементів

Іноді при розв'язуванні задач є необхідність обчислення добутку елементів масиву, що задовольняють умову.

Для знаходження добутку використовують конструкцію, що також схожа на знаходження суми:

```
#встановлення початкового значення
p=1
#використання оператора циклу
for i in range():
    #перевірка умови
    if умова:
        #знаходження добутку
        p=p*ім'я_списку[номер_елемента]
#виведення значення добутку
print(p)
```

Тут відмінною є команда встановлення початкового значення: замість нуля, змінній, що має обчислити добуток, установлюється значення 1, оскільки при множенні на нуль добуток завжди буде нульовим, а при множенні на 1 ми отримаємо те саме число. Також тут є відмінною дія обчислення добутку. Замість знака додавання звичайно тут використовується знак множення.

ВПРАВА 33.2



Завдання. В поліклініці з понеділка починають робити щеплення. Кожен день фіксують кількість проведених щеплень. Заповнити масив проведення щеплень випадковими числами, врахувавши, що кількість днів щеплень має ввести користувач, у неділю щеплення не відбувається та кількість щеплень за один день від 20 до 50 осіб. Визначити:

- кількість щеплень, зроблених у парні дні, та кількість щеплень, зроблених у непарні дні;
- кількість днів, коли було зроблено від 30 до 40 щеплень за день;
- середню кількість щеплень, що було зроблено у робочі дні.



Створення та виведення масиву кількості щеплень

1. Уведіть команду імпорту модуля **random**.
2. Для створення масиву кількості проведених щеплень додайте команду створення порожнього списку **vac**.
3. У наступному рядку додайте команду введення кількості днів проведення щеплення:

```
n=int(input('Введіть, скільки днів проводиться щеплення:'))
```

4. Для створення масиву уведіть оператор циклу, що буде виконуватися **n** раз:

```
for i in range(n):  
    if (i+1)%7==0:  
        a=0  
    else:  
        a=randint(20,50)  
    vac.append(a)
```

У тілі циклу оператор розгалуження здійснює перевірку, чи ділиться порядковий номер дня на 7, тобто чи припадає цей день на неділю. Зверніть увагу, що тут до *i* додається 1, оскільки нумерація індексів циклу починається з 0, а порядок днів з одиниці, отже, індекс відрізняється від порядкового номера дня на 1. В лівій частині виразу обчислюється остача від ділення порядкового номера дня на 7. Отже, в умові перевіряється, чи ділиться число на 7 без остачі. Якщо вказана умова виконується, то змінній *a* присвоюється значення 0. Тут змінна *a* — це кількість щеплень у поточний день. У протилежному випадку змінній *a* надаємо значення цілого випадкового числа у проміжку від 20 до 50. За межами оператора розгалуження додаємо значення змінної *a* в кінець масиву **vac**.

5. За межами оператора циклу додайте команду виведення масиву **vac**.
6. Запустіть програму на виконання, введіть кількість днів щеплення, наприклад, 50. Переконайтеся, що виведено масив чисел, де кожний сьомий елемент є нульовим, а інші числа є випадковими числами від 20 до 50.

Кількість щеплень, зроблених у парні та непарні дні

Далі визначимо кількість щеплень, зроблених у дні, що мають парний порядковий номер, та кількість щеплень у непарні дні. Так, для даного прикладу це елементи масиву, що мають індекс 1, 3, 5 тощо, відповідно вони будуть мати парні порядкові номери днів 2, 4, 6 і так далі. Значення, яке необхідно знайти, будемо надавати змінній *sp*, в якості суми елементів масиву **vac**. Відповідно інші елементи



будуть розташовані на місцях, що мають непарний порядковий номер. Суму цих елементів надамо змінній *sn*.

7. У програмному коді встановіть початкові значення для змінних *sp* та *sn*, рівні нулю.
8. Для знаходження суми додайте оператор циклу для виконання *n* ітерацій:

```
for i in range(n):  
    if (i+1)%2==0:  
        sp=sp+vac[i]  
    else:  
        sn=sn+vac[i]
```

У тілі циклу оператор розгалуження здійснює перевірку парності порядкового дня. Як умова здійснюється перевірка, чи ділиться порядковий номер дня без остачі на 2. Якщо умова виконується, обчислюється значення змінної *sp* — кількість щеплень у парні дні. Якщо умова не виконується, обчислюємо значення змінної *sn* — кількість щеплень у непарні дні.

9. За межами тіла циклу додайте команду виведення знайдених значень.
10. Запустіть програму та перевірте правильність її виконання.

Кількість днів, коли було зроблено від 30 до 40 щеплень за день

Далі обчислимо кількість днів, коли було зроблено від 30 до 40 щеплень. Результат надамо змінній *k*. Дане значення знайдемо у тілі того ж самого циклу, що і для попереднього завдання.

11. Перед оператором циклу встановіть початкове значення для змінної *k*, рівне 0.
12. Для перевірки умови додайте вказівку розгалуження, в якій скористайтеся подвійною умовою, суть якої полягає в тому, що значення елемента масиву **vac** є більшим або рівним числу 30 та меншим або рівним числу 40. Якщо умова виконується, збільшіть значення змінної *k* на 1.
13. За межами тіла циклу додайте команду виведення знайденого значення.
14. Запустіть програму та перевірте правильність її виконання.

Середня кількість щеплень, що було зроблено у робочі дні

Знайдемо ще одне значення. А саме, середню кількість щеплень, що було зроблено у робочі дні. Тобто необхідно врахувати усі дні, крім неділь. Результат обчислення округлимо до цілого значення та надамо змінній *avg*. Для знаходження середньої кількості щеплень у робочі дні необхідно розділити загальну кількість щеплень (*s*) на кількість робочих днів:


$$avg = s / (\text{кількість робочих днів})$$

Кількість робочих днів можна визначити за формулою — від загальної кількості днів (n) відняти кількість неділь (k_Sun):

$$avg = s / (n - k_Sun)$$

У даній формулі залишаються невідомими загальна кількість щеплень (s) та кількість неділь (k_Sun). Отже, переходимо до їхнього пошуку.

15. У тілі програми додайте команду знаходження кількості неділь:

```
k_Sun=vac.count(0)
```

Для знаходження кількості неділь скористаємося методом **count**, який визначає у масиві кількість входжень вказаних у дужках значень.

16. Додайте команду визначення загальної кількості щеплень:

```
s=sum(vac)
```

17. Додайте команду обчислення середньої кількості щеплень у робочі дні:

```
avg=round(s/(n-k_Sun))
```

18. Додайте команду виведення знайденого значення.

19. Запустіть програму та перевірте правильність її виконання.

20. Самостійно додайте команди визначення:

- кількості днів, коли було зроблено більше ніж 45 щеплень;
- у які дні було зроблено максимальну та мінімальну кількість щеплень;
- кількості днів, коли щеплення не робили.



Контрольні запитання та завдання

1. Як знайти суму, кількість та добуток елементів масиву, що задовольняють указаній в умові?
2. Чим відрізняється структура коду знаходження суми від знаходження добутку?



Питання для роздумів

- 1*. Як знайти середнє значення елементів списку, що задовольняють указану умову?
- 2*. Якою має бути оператор циклу для знаходження суми та кількості усіх парних елементів масиву?





ПОШУК ЕЛЕМЕНТІВ У МАСИВІ

В одновимірному масиві можна виконувати наступні види пошуків:

- пошук усіх елементів з певними властивостями;
- пошук першого елемента з певними властивостями;
- пошук останнього елемента з певними властивостями.

Існує декілька реалізацій цих алгоритмів пошуку — розглянемо найбільш уживані, створивши програми до відповідних практичних вправ.

ВПРАВА 34.1. ПОШУК УСІХ ЕЛЕМЕНТІВ ІЗ ПЕВНИМИ ВЛАСТИВОСТЯМИ



Завдання. Дано масив **a**, що має 10 цілих випадкових чисел, що належать проміжку $[-10, 10]$. Знайти у ньому всі непарні елементи та їхні індекси.

Загальний алгоритм розв'язання задач такого типу:

1. Уведення даних до масиву **a**;
2. Перегляд елементів масиву **a**, перевірка кожного елемента на відповідність умові та виведення елемента і його індексу, якщо умова виконується.

Програмний код

```
#імпорт модуля випадкових чисел
from random import*
#створення масиву
a=[]
for i in range(10):
    a.append(randint(-10,10))
#виведення масиву a
print(a)
#виведення непарних елементів та їх індексів
for i in range(10):
    if a[i]%2!=0:
        print(a[i],i)
```

1. Запустіть програму та перевірте правильність її виконання.

ВПРАВА 34.2. ПОШУК ПЕРШОГО ЕЛЕМЕНТА З ПЕВНИМИ ВЛАСТИВОСТЯМИ



Завдання. Дано масив **t**, що містить середньодобові температури протягом місяця. Визначити перший день, коли температура була від'ємною.



Загальний алгоритм розв'язання такого типу:

1. Уведення даних до масиву **t**.
2. Встановлення початкового значення для $n1 = -1$, де $n1$ — індекс елемента, що задовольняє умові.
3. Перевірка з **кінця** кожного елемента масиву **t** на відповідність умові. Якщо умова виконується $t1 = \text{значення елемента масиву}$, $n1 = \text{індекс даного елемента}$. Після завершення перегляду у змінних **t1** та **n1** залишаться значення **першого** елемента із заданою властивістю.
4. Перевірка значення змінної **n1**: якщо $n1 = -1$, то таких елементів немає, інакше виведення на екран значення змінних **t1** та **n1**.

Уведення програмного коду

Для спрощення введення даних створимо масив із 30-ти випадкових чисел в проміжку $[-10, 10]$.

1. Уведіть команду імпорту модуля випадкових чисел та встановіть початкове значення для $n1 = -1$.
2. Створіть масив із 30 дійсних випадкових чисел в проміжку $[-10, 10]$, значення випадкових чисел округліть до одного десяткового знака.

#створення масиву

```
t=[]
for i in range(30):
    t.append(round(uniform(-10,10),1))
```

3. Додайте команду виведення масиву **t**.
4. Додайте оператор циклу, що здійснює пошук першого від'ємного елемента.

```
for i in range(29,-1,-1):
    if t[i]<0:
        t1=t[i]
        n1=i+1
```

Зверніть увагу на аргументи функції **range**. Крок **-1** вказує на те, що тіло циклу буде виконуватися від останнього 29-го елемента до 0-го включно.

5. Додайте оператор розгалуження, що перевіряє наявність від'ємного елемента та здійснює виведення результату в залежності від виконання умови.

```
if n1==-1:
    print('Таких днів немає')
```



else:

```
print("Перший день, коли температура була від'ємною ",n1,  
      "-й день місяця. Температура у цей день  
      становила",t1)
```

6. Запустіть програму та перевірте правильність її виконання.

ВПРАВА 34.3. ПОШУК ОСТАННЬОГО ЕЛЕМЕНТА З ПЕВНИМИ ВЛАСТИВОСТЯМИ



Завдання. Дано масив t , що містить середньодобові температури протягом місяця. Визначити останній день, коли температура була додатною.

Загальний алгоритм розв'язання задач такого типу:

1. Уведення даних до масиву t ;
2. Встановлення початкового значення для $n2 = -1$, де $n2$ — індекс елемента, що задовольняє умові.
3. Перевірка з **початку** кожного елемента масиву t на відповідність умові. Якщо умова виконується $t2 = \text{значення елемента масиву}$, $n2 = \text{індекс даного елемента}$. Після завершення перегляду у змінних $t2$ та $n2$ залишаться значення **останнього** елемента із заданою властивістю.
4. Перевірка значення змінної $n2$: якщо $n2 = -1$, то таких елементів немає, інакше — виведення на екран значення змінних $t2$ та $n2$.

Уведення програмного коду

Розглянемо цей тип задачі з використанням коду попередньої програми, додавши необхідні команди.

1. Додайте до коду програми із вправи 34.2 оператор циклу, що здійснює пошук останнього додатного елемента.

```
for i in range(30):  
    if t[i]>0:  
        t2=t[i]  
        n2=i+1
```

У даному випадку пошук здійснюється з початку масиву, отже, оператор циклу буде мати структуру, аналогічну до введення масиву.

2. Додайте оператор розгалуження, що перевіряє наявність додатного елемента та здійснює виведення результату в залежності від виконання умови.



```
if n2==-1:
    print('Таких днів немає')
else:
    print("Останній день, коли температура була додатною ",n2,
          "-й день місяця. Температура у цей день",
          "становила",t2)
```

3. Запустіть програму та перевірте правильність її виконання.

ВПРАВА 34.4



Завдання. Створити програми для розв'язання задач.

- a. Дано масив із 20 цілих чисел. Виведіть на екран індекси тих елементів, що закінчуються цифрою 0.
- b. Дано масив із n дійсних чисел. Знайдіть усі елементи (значення та індекси), що більші за середнє арифметичне всіх елементів масиву.
- c. Дано масив із 10 цілих чисел. Знайдіть усі максимальні елементи масиву та їхні індекси.
- d. Дано масив із n цілих чисел. Знайдіть мінімальний елемент масиву та його індекс. Якщо таких елементів декілька, знайдіть елемент з найменшим індексом.
- e. Дано масив із 15 цілих чисел та дано ціле число x . Знайдіть у масиві останній елемент, що дорівнює x , якщо він є у масиві.



Контрольні запитання та завдання

1. Які види пошуків виконують у масивах?
2. Опишіть алгоритм пошуку всіх елементів, що задовольняють указану умову.
- 3*. Чим відрізняється пошук останнього елемента від пошуку першого елемента, що задовольняє указану умову?



Питання для роздумів

- 1*. Яким може бути алгоритм пошуку всіх від'ємних елементів масиву?
- 2*. Яким може бути алгоритм пошуку першого парного елемента масиву?
- 3*. Яким може бути алгоритм пошуку індексу останнього нульового елемента масиву?





ПОШУК ЕЛЕМЕНТІВ У МАСИВІ. РОЗВ'ЯЗУВАННЯ ПРИКЛАДНИХ ЗАДАЧ

ЗАДАЧА 1



Дано список прізвищ учасників олімпіади з інформатики та результати їхньої участі в олімпіаді, що мають значення від 0 до 100 у вигляді десяткового дробу з одним десятковим знаком. Визначити:

- список переможців олімпіади, а саме, тих, що мають результат більший за середній бал;
- учасників, що отримали більше 50 балів;
- учасників, що мають кількість балів, меншу від n , де n — значення, введене користувачем з використанням вікна введення даних.

Задачу з учасниками олімпіади ми вже розглядали в одному із попередніх параграфів. Отже, скористаємося вже наявними там кодами, необхідними для розв'язання задачі.

1. Завантажте файл-заготовку **Задача_35_1.py** із матеріалів до даного параграфу. Файл-заготовка містить команди створення інтерфейсу, списків учнів та результати їхньої участі в олімпіаді, а також обчислення середнього бала усіх учасників.
2. Запустіть програму на виконання та проаналізуйте її програмний код.
3. Додайте команди обробника події натиснення кнопки **Переможці олімпіади**.

```
def btn1_click():  
    #встановлення напису над третім списком  
    L3['text']='Переможці олімпіади'  
    #очищення списку від попередніх написів  
    Lbox_rez.delete(0,END)  
    for i in range(12):  
        #перевірка умови, якщо поточний результат більший  
        за середнього значення  
        if grade[i]>avg_grade:  
            #якщо умова виконується, додати прізвище  
            учасника в кінець списку  
            Lbox_rez.insert(END,name[i])
```

Додайте виклик обробника події в конструкторі кнопки **Переможці олімпіади**.

- 5*. Обробники подій до кнопок **Більше 50** та **Менше n** створіть самостійно. Додайте команди виклику цих обробників подій у відповідних конструкторах.



ЗАДАЧА 2



Відома вартість квитків у купейному вагоні на поїзд Київ-Одеса у різні дні тижня. Визначити:

- дні, коли вартість квитка є найнижчою;
- дні, коли вартість квитка є найвищою;
- дні, коли вартість квитка є нижчою від n , де n — ціна, яку введе користувач;
- дні, коли вартість квитка є нижчою, ніж середня вартість на квиток.

1. Завантажте файл-заготовку **Задача_35_2.py** із матеріалів до даного параграфу. На рис. 35.1 позначено елементи інтерфейсу програми.

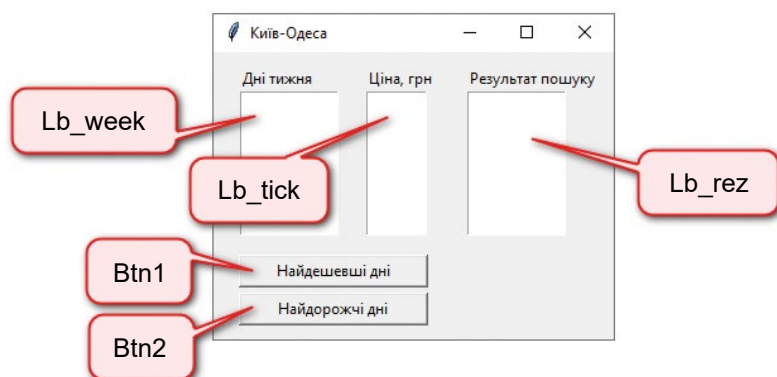


Рис.35.1

2. Додайте команди виведення масивів до елементів керування **Lb_week** та **Lb_tick**. Місце додавання коду встановіть самостійно.

```
for i in range(7):
    #виведення списку днів тижнів
    Lb_week.insert(END,week[i])
    #виведення списку вартості квитків
    Lb_tick.insert(END,ticket[i])
```

3. Додайте обробник події натиснення кнопки **Найдешевші дні**.

```
def Btn1_click():
    #очищення списку від попередніх записів
    Lb_rez.delete(0,END)
    #обчислення мінімальної вартості квитка
    t_min=min(ticket)
    for i in range(7):
        #якщо поточна вартість квитка є мінімальною,
        #додати значення дня тижня до списку результатів
```



```
if ticket[i]==t_min:  
    Lb_rez.insert(END,week[i])
```

4. Додайте команду виклику обробника події в конструкторі кнопки **Найде-шевші дні**.
5. Інші завдання даної задачі виконайте самостійно, додавши необхідні елементи інтерфейсу та команди для обчислення.

ЗАДАЧА 3



Перед початком роботи кондуктор отримує рулон автобусних квитків. Усі квитки пронумеровані шестицифровими числами. Кожен рулон складається з 1000 квитків з номерами від ###000 до ###999, тобто три перших цифри номера квитків з одного рулону збігаються. Нехай даний рулон починається числом 123. Вивести усі «щасливі» квитки з даного рулону. (Щасливим є квиток, у якого сума перших трьох цифр дорівнює сумі останніх трьох цифр.)

Аналіз умови задачі

Оскільки ми маємо для кожного квитка порівняти суму перших трьох цифр та суму останніх трьох цифр, то потрібно визначити їхні значення. Сума перших трьох цифр для кожного квитка є сталою і дорівнює $1+2+3=6$. Для знаходження суми останніх трьох цифр необхідно знаходити значення кожної цифри, а потім їхню суму у тілі циклу.

Створення програмного коду

1. Уведіть команди створення масиву **n**, що містить номери квитків від 123000 до 123999 включно.

```
n=[]  
for i in range(123000,124000):  
    n.append(i)
```

2. Додайте оператор циклу, в якому буде здійснено стільки ітерацій, скільки містить рулон квитків.

```
for i in range(1000):
```

3. У тілі оператора циклу додайте команди визначення 4-ї, 5-ї та 6-ї цифр.

```
#4-та цифра  
a4=n[i]%1000//100  
#5-та цифра  
a5=n[i]%100//10  
#6-та цифра  
a6=n[i]%10
```




4. Додайте команду обчислення суми останніх трьох цифр номера квитка.

```
#сума цифр  
s=a4+a5+a6
```

5. Перевірте, чи дорівнює 6 знайдена сума. Якщо умова виконується, номер квитка є щасливим, виведіть його значення.

```
if s2==6:  
    print(n[i])
```

- 6*. Самостійно змініть програмний код таким чином, аби перші три цифри номерів квитків у ролоні вводив користувач.

ВПРАВА 35



Створіть програми для розв'язання задач.

- a. Дано масив середньодобових температур протягом року. Визначити кількість днів, коли середньодобова температура була нульовою. Знайти, скільки разів середньодобова температура з від'ємної змінювалася на додатну.
- b. Магазин на честь відкриття першим 100 покупцям видавав лотерейні квитки, що мають випадкове число від 1000 до 9999. Якщо у квитку сума цифр менша або дорівнює 10, то покупець одержує виграш 100 грн, якщо перша та остання цифри збігаються — виграш 200 грн, якщо виконуються обидві умови, — виграш 500 грн. Визначити, скільки покупців одержали виграш по 100 грн, по 200 грн та по 500 грн. Визначити загальну суму виграшу, сплачену магазином.

Зауваження. Оскільки в лотерейних квитках числа як правило не повторюються, то для створення послідовності випадкових чисел без повторень скористайтеся наступною конструкцією:

```
from random import*  
x=[]  
a=[]  
#створення послідовності чисел від 1000 до 9999 включно  
#з кроком 1  
for i in range(1000,10000):  
    x.append(i)  
#вибір із послідовності x 100 чисел в довільному порядку  
a=sample(x,100)
```



СОРТУВАННЯ МАСИВІВ

У даному параграфі ми розглянемо, як здійснити сортування елементів списків. Елементи списку можуть бути впорядковані за зростанням та за спаданням. Приклад різних способів сортування списку наведено в таблиці 36.1.

Таблиця 36.1.

Даний список	Список, упорядкований за зростанням	Список, упорядкований за спаданням
4	2	8
2	3	6
8	4	4
6	6	3
3	8	2

Як і більшість сучасних мов програмування, Python має стандартний метод сортування списків **sort**. На рис. 36.1 наведено приклад використання даного методу:

```
a=[4,2,8,6,3]  
a.sort()  
print(a)
```



```
[2,3,4,6,8]
```

Рис. 36.1

Для сортування в протилежному порядку для даного методу використовують параметр **reverse**, що встановлює порядок сортування. Значення **True** вказує на сортування за спаданням, **False** — за зростанням. Отже, за умовчанням цей параметр має значення **False**. На рис. 36.2 наведено приклад сортування списку за спаданням.

```
a=[4,2,8,6,3]  
a.sort(reverse=True)  
print(a)
```



```
[8,6,4,3,2]
```

Рис. 36.2

Розглянемо використання методу **sort** при розв'язанні задач.



ВПРАВА 36.1



Завдання. Дано список середніх балів 14 учнів, що визначаються дійсним випадковим числом, заданим у діапазоні [4:12] балів.

- Впорядкуйте та виведіть середні бали учнів за рейтингом (від найбільшого до найменшого).
- До класу приєднайте ще n учнів із середнім балом, що має ввести користувач, та додайте нові результати учнів до списку.
- Виведіть усі результати за рейтингом.

1. Імпортуйте модуль `random` та додайте команду створення порожнього списку `grade`.
2. Створіть масив, що має список середніх балів 14 учнів, що визначаються дійсним випадковим числом у діапазоні [4:12] балів.

```
for i in range(14):
    grade.append(round(uniform(4,12),1))
```

3. Додайте команду виведення даного списку.
4. Додайте команду сортування списку за спаданням:

```
grade.sort(reverse=True)
```

5. Додайте команду виведення відсортованого списку.
6. Додайте результати нових учнів:

```
#додавання кількості нових учнів
n=int(input('Введіть кількість нових учнів '))
for i in range(n):
    #введення середнього балу учня
    k=float(input('Введіть середній бал '+str(i+1)+'-го
                  учня '))
    #додавання середнього балу до масиву
    grade.append(k)
```

7. Додайте команду сортування нового списку за спаданням та виведення даного списку.
8. Запустіть програму та перевірте правильність її виконання.

ВПРАВА 36.2



Завдання. Дано список учасників олімпіади та їхні результати. У третьому списку вивести переможців олімпіади в алфавітному порядку та їхні результати в наступному форматі: прізвище–результат (рис. 36.3).

1. Завантажте файл-заготовку **Вправа_36_2.py** із матеріалів до даного параграфа.
2. Запустіть програму на виконання та проаналізуйте її програмний код.
3. Додайте команди обробника події натиснення кнопки **Переможці олімпіади**.



Результати олімпіади		
Учасники олімпіади	Результат	Переможці
Новікова	30.4	Антоненко-71.5
Романова	82.4	Балацький-55.2
Волошина	77.0	Волошина-77.0
Орленко	17.5	Кротенко-66.8
Діденко	35.8	Макаров-92.9
Кротенко	66.8	Романова-82.4
Литвиненко	26.4	
Макаров	92.9	
Антоненко	71.5	
Гнатов	25.5	
Петренко	27.7	
Балацький	55.2	
Середній бал: 50.8		
Переможці олімпіади		

Рис.36.3

- a. Уведіть заголовок обробника події:

```
def btn1_click():
```
 - b. Додайте команду створення порожнього масиву **rez**.
 - c. Додайте оператор циклу для 12 ітерацій:

```
for i in range(12):
```
 - d. У тілі циклу сформууйте умову: якщо результат учасника більший за середній бал, то до списку **rez** додати елемент у форматі прізвище–результат:

```
if grade[i]>avg_grade:  
    rez.append(name[i]+'-'+str(grade[i]))
```
 - e. Додайте команду сортування списку **rez** в алфавітному порядку.
 - f. Додайте команду визначення розмірності списку **rez**. Результат надайте змінній **size**.
 - g. Додайте команди виведення масиву **rez** до елемента керування **Lbox_rez**.
4. Додайте команду виклику обробника події в конструкторі кнопки.
 5. Запустіть програму та перевірте правильність її виконання.

**ВПРАВА 36.3**

Дано масив **a**, що містить 20 цілих випадкових чисел з проміжку [1;10]. Виділіть з даного масиву масив **b**, що містить перші 10 елементів масиву **a**, та масив **c**, що містить останні 10 елементів масиву **a**. Впорядкуйте масив **b** за зростанням, а масив **c** за спаданням. Виведіть відсортовані масиви.

**Контрольні запитання та завдання**

1. З допомогою якого методу можна відсортувати списки?
2. Наведіть приклад списку, а також результат його сортування за зростанням та за спаданням.
3. Як виконати сортування списку за спаданням?

**Питання для роздумів**

- 1*. Як ви думаєте, чи можливо з використанням методу **sort** у вправі 36.2 виконати сортування учасників олімпіади за їхніми результатами і вивести відсортований список у форматі **результат–прізвище**?
2. Визначте значення елементів списків, одержаних у результаті виконання фрагментів програм:
 1.

```
a=[4,-3,12,0,5]
a.sort()
print(a)
```
 2.

```
a=[4,-3,12,0,5]
a.sort(reverse=True)
print(a)
```

**Завдання для досліджень**

- 1*. Дізнайтеся про можливості використання методу **sorted**.

**§37.****СОРТУВАННЯ МАСИВІВ МЕТОДОМ ВИБОРУ**

Ми познайомилися зі стандартним методом **sort**, що дозволяє здійснювати сортування списків. Проте його не завжди можна використати для розв'язання задач, в яких потрібно застосувати сортування.



Так, наприклад, якщо у попередній вправі про результати участі в олімпіаді є необхідність відсортувати список учасників за рейтингом, то окремо результати учасників метод **sort** відсортує, але разом із цим відсортувати прізвища учасників не вийде.

Для виконання сортування існують десятки методів сортування. Розглянемо найбільш уживані з них.

Нехай потрібно впорядкувати масив, що містить 5 елементів за зростанням:

$$a[0] \leq a[1] \leq \dots \leq a[4].$$

Алгоритм сортування

- Відшукати максимальний елемент з послідовності **a[0]...a[4]**.
- Максимальний елемент із цієї послідовності поміняти місцями з **a[4]**.
- Відшукати максимальний елемент із послідовності **a[0]...a[3]**.
- Максимальний елемент із цієї послідовності поміняти місцями з **a[3]**.
- ...
- Максимальний елемент із послідовності **a[0]..a[1]** поміняти місцями з **a[1]**.

Так, наприклад, якщо ми маємо список **a=[3,5,1,2,7]**, його сортування вибором максимального елемента буде мати вигляд, як на рис. 37.1, де синьою рамкою виділено діапазон пошуку максимального елемента, в блакитній клітинці вказано максимальне значення, стрілкою позначено місце переміщення максимального елемента. Відсутність стрілки вказує на те, що елемент залишається на своєму місці.

0	1	2	3	4
3	5	1	2	7
3	5	1	2	7
3	2	1	5	7
1	2	3	5	7
1	2	3	5	7

Рис.37.1

Нижче наведено програмний код, що відповідає описаному алгоритму:

```
1. a=[3,5,1,2,7]
2. print(a)
3. for i in range(4):
```



```
4.     #пошук максимального елемента
5.     m=max(a[0:5-i])
6.     #визначення номера максимального елемента
7.     n=a.index(m)
8.     #обмін елементів місцями
9.     a[n],a[4-i]=a[4-i],a[n]
10.    print(a)
```

Аналіз програмного коду

Неважко здогадатися, що в перших двох рядках здійснюється створення списку та виведення його на екран. У третьому рядку створено оператор циклу, що містить на одну ітерацію менше ніж кількість елементів у списку.

У четвертому рядку здійснюється пошук максимального елемента в заданому діапазоні.

Так, для $i=0$ діапазон буде $[0:5]$, що означає до 4-го елемента включно. Для $i=1$ діапазон буде змінено на $[0:4]$, що означає до 3-го елемента включно. Таким чином, з допомогою вказаної конструкції $i-5$ значення діапазону буде зменшуватися. Останній діапазон, у якому буде здійснено пошук максимального $[0:2]$, він буде сформований для $i=3$.

Команда `n=a.index(m)` дозволяє визначити номер знайденого максимального елемента. Цей номер потрібен для того, щоб далі вказати елементи, з якими номерами необхідно обміняти місцями.

Для обміну місцями елементів списку використовують наступну конструкцію:

```
a[n],a[4-i]=a[4-i],a[n].
```

Неважко здогадатися, що вона означає поміняти місцями значення елементів, що розташовані під номерами n та $4-i$.

Як ви думаєте, як має змінитися даний програмний код для сортування його елементів за спаданням?

Для цього існує принаймні два варіанти. Перший і найпростіший — це пошук мінімальних значень замість максимальних. Також для зворотного сортування можна використати діапазон пошуку максимального значення та замінювати його не на останній елемент, а на перший.

До речі, у такий спосіб можна сортувати і текстові дані. При визначенні максимального значення серед двох слів більшим вважається те, що стоїть далі за алфавітом.

Так, наприклад, `max('python', 'pascal') = 'python'`. Розглянемо використання даного методу сортування текстових та числових даних, виконавши вправу.



ВПРАВА 37.1



Завдання. Програма містить дані про найбільш відомі озера України (рис. 37.1). В першому списку розташовано перелік їхніх назв, у другому — середня глибина озера, в третьому — площа.

Озера	Глибина, м	Площа, га
Синевир	9	7
Світязь	7.2	2750
Ялпуг	2.6	14900
Бребенескул	3	0.61
Кугурлуй	1	8200
Кагул	2	9200
Кунігунда	4	0.08

За назвою За глибиною За площею

Рис.37.1

Створити обробники подій для кнопок, що мають здійснити відповідні сортування. Врахуйте, що при сортуванні даних одного списку відповідні елементи інших списків мають бути також переміщені.

Кнопка **За назвою** здійснює сортування озер за назвою в алфавітному порядку.

Кнопка **За глибиною** здійснює сортування глибин озер за спаданням.

Кнопка **За площею** здійснює сортування площ озер за спаданням.

1. Завантажте файл **Вправа_37_1.py** із матеріалів до даного уроку.
2. Створіть обробник події натискання кнопки **За назвою**.
 - a. Створіть команду обробника події натискання кнопки та додайте оператор циклу сортування масиву **name** в алфавітному порядку методом вибору:

```
def btn1_click():  
    for i in range(6):  
        m=max(name[0:7-i])  
        n=name.index(m)  
        name[n],name[6-i]=name[6-i],name[n]
```

- b. Оскільки відповідно до умови дані, що розміщені в інших списках, мають переміщуватися на ті ж місця, що й імена під час сортування, у тілі цього ж циклу додайте команди обміну елементами списків **depth** та **area**:

```
depth[n],depth[6-i]=depth[6-i],depth[n]  
area[n],area[6-i]=area[6-i],area[n]
```

- c. Додайте команди очищення списків **Lbox_name**, **Lbox_depth**, **Lbox_area**:

```
Lbox_name.delete(0,END)  
Lbox_depth.delete(0,END)  
Lbox_area.delete(0,END)
```

- d. Додайте оператор циклу виведення відсортованих списків до елементів керування **Lbox_name**, **Lbox_depth**, **Lbox_area**:



```
for i in range(7):
    Lbox_name.insert(END, name[i])
    Lbox_depth.insert(END, depth[i])
    Lbox_area.insert(END, area[i])
```

3. Додайте команду виклику обробника події в конструкторі кнопки **За назвою**.
4. Обробники подій натискання кнопок **За глибиною** та **За площею** створіть самостійно.

ВПРАВА 37.2



Завдання. Дано список учасників олімпіади та їхні результати. У третьому списку вивести переможців олімпіади у форматі **результат–прізвище**, впорядкувавши їх за рейтингом у порядку спадання (рис. 37.3).

Результати олімпіади		
Учасники олімпіади	Результат	Переможці
Новікова	57.7	76.4 - Литвиненко
Романова	58.7	73.9 - Діденко
Волошина	65.1	65.1 - Волошина
Орленко	29.7	58.7 - Романова
Діденко	73.9	57.7 - Новікова
Кротенко	54.8	54.8 - Кротенко
Литвиненко	76.4	50.4 - Гнатов
Макаров	38.0	
Антоненко	19.1	
Гнатов	50.4	
Петренко	18.6	
Балацький	22.9	
Середній бал: 47.1		
Переможці олімпіади		

Рис.37.3

1. Завантажте файл-заготовку **Вправа_37_2.py** із матеріалів до даного параграфа.
2. Створіть обробник події натискання кнопки **Переможці олімпіади**.
 - a. Уведіть команду заголовка обробника події:

```
def btn1_click():
```

- b. Додайте команди створення двох порожніх списків **rez_n** для розташування у ньому прізвищ переможців та **rez** для розташування їхніх результатів.



- c. Додайте оператор циклу для формування списків прізвищ та результатів переможців:

```
for i in range(12):
    if grade[i]>avg_grade:
        rez.append(grade[i])
        rez_n.append(name[i])
```

- d. Додайте команду визначення довжини одного із утворених масивів **rez** або **rez_n**, оскільки вони мають однакову розмірність, результат надайте змінній *size*.
- e. Додайте команду сортування масиву **rez** за спаданням методом вибору:

```
for i in range(size-1):
    m=min(rez[0:size-i])
    n=rez.index(m)
    rez[n],rez[size-1-i]=rez[size-1-i],rez[n]
```

- f. Аби разом з результатами під час сортування було переміщено і прізвища переможців олімпіади, у тілі циклу сортування масиву **rez** додайте команду обміну елементами **rez_n**:

```
rez_n[n],rez_n[size-1-i]=rez_n[size-1-i],rez_n[n]
```

- g. Додайте команди виведення списків **rez** та **rez_n** до елемента керування **Lbox_rez** у форматі **результат–прізвище**:

```
for i in range(size):
    Lbox_rez.insert(END,str(rez[i])+ ' - '+rez_n[i])
```

3. Додайте команду виклику обробника події в конструкторі кнопки.
4. Запустіть програму та перевірте правильність її виконання.

ВПРАВА 37.3



Створити програми розв'язання задач, використовуючи для сортування метод вибору.

- a. Дано цілочисельний масив з 20 елементів. Відсортуйте 10 перших елементів масиву за зростанням, а 10 останніх елементів — за спаданням.
- b. Дано цілочисельний масив з 15 елементів. Відсортуйте 5 перших елементів, 5 других елементів та 5 останніх елементів за спаданням. Знайдіть суму кожної п'ятірки чисел.
- c. Дано дійсний масив з 10 елементів. Відсортуйте за спаданням усі елементи цього масиву, що розташовані між першим та останнім від'ємними елементами.



- d*.** Дано цілочисельний масив з 10 елементів. Відсортуйте за спаданням усі елементи цього масиву, що розташовані між першим парним та останнім непарним елементами.
- e*.** Дано масив, що містить n елементів. Після впорядкування його за зростанням визначте:
- скільки елементів масиву змінили своє початкове розташування;
 - порядкові номери елементів масиву, які залишилися стояти на своїх місцях;
 - на якому місці в масиві знаходився елемент до сортування, який зараз розташований на k -му місці.



Контрольні запитання та завдання

1. Опишіть алгоритм сортування вибором.
2. Наведіть приклад завдання коли для сортування списку неможливо або не доцільно використовувати метод **sort**.
- 3*. Як змінити команди сортування методом вибору за зростанням для сортування цим же методом за спаданням?



Питання для роздумів

- 1*. Установіть послідовність, в якій змінюється список $a = [5, 4, 7, 9, 6]$ у процесі сортування за спаданням методом вибору.

№	0	1	2	3	4
	9	7	6	5	4
	5	6	7	9	4
	9	6	7	5	4
	5	4	7	9	6

- 2*. Яким буде програмний код сортування методом бульбашок, якщо замість оператора циклу з параметром використати цикл з умовою?



Завдання для досліджень

- 1*. Дізнайтеся, які ще існують методи сортування.





§38.

**СОРТУВАННЯ МАСИВІВ МЕТОДОМ
БУЛЬБАШОК. СКЛАДНІСТЬ АЛГОРИТМУ**

38.1.

Сортування методом бульбашок

Одним із найдавніших відомих методів сортування є сортування «бульбашками». Принцип цього методу полягає у попарному порівнянні елементів та обміну їх місцями, доки менші елементи не перестануть «спливати» до початку списку, а більші елементи не будуть «тонуть» в кінець списку. Метод бульбашок ґрунтується на порівнянні та перестановці сусідніх чисел.

На рис. 38.1 візуально показано, як відбувається порівняння та обмін елементів масиву на прикладі сортування за зростанням списку [7, 4, 3, 5, 1].

I ітерація	II ітерація	III ітерація	IV ітерація																																																																																
<table><tr><td>7</td><td>4</td><td>3</td><td>5</td><td>1</td></tr><tr><td>4</td><td>7</td><td>3</td><td>5</td><td>1</td></tr><tr><td>4</td><td>3</td><td>7</td><td>5</td><td>1</td></tr><tr><td>4</td><td>3</td><td>5</td><td>7</td><td>1</td></tr><tr><td>4</td><td>3</td><td>5</td><td>1</td><td>7</td></tr></table>	7	4	3	5	1	4	7	3	5	1	4	3	7	5	1	4	3	5	7	1	4	3	5	1	7	<table><tr><td>4</td><td>3</td><td>5</td><td>1</td><td>7</td></tr><tr><td>3</td><td>4</td><td>5</td><td>1</td><td>7</td></tr><tr><td>3</td><td>4</td><td>5</td><td>1</td><td>7</td></tr><tr><td>3</td><td>4</td><td>1</td><td>5</td><td>7</td></tr><tr><td>3</td><td>4</td><td>1</td><td>5</td><td>7</td></tr></table>	4	3	5	1	7	3	4	5	1	7	3	4	5	1	7	3	4	1	5	7	3	4	1	5	7	<table><tr><td>3</td><td>4</td><td>1</td><td>5</td><td>7</td></tr><tr><td>3</td><td>4</td><td>1</td><td>5</td><td>7</td></tr><tr><td>3</td><td>1</td><td>4</td><td>5</td><td>7</td></tr><tr><td>3</td><td>1</td><td>4</td><td>5</td><td>7</td></tr></table>	3	4	1	5	7	3	4	1	5	7	3	1	4	5	7	3	1	4	5	7	<table><tr><td>3</td><td>1</td><td>4</td><td>5</td><td>7</td></tr><tr><td>1</td><td>3</td><td>4</td><td>5</td><td>7</td></tr></table>	3	1	4	5	7	1	3	4	5	7
7	4	3	5	1																																																																															
4	7	3	5	1																																																																															
4	3	7	5	1																																																																															
4	3	5	7	1																																																																															
4	3	5	1	7																																																																															
4	3	5	1	7																																																																															
3	4	5	1	7																																																																															
3	4	5	1	7																																																																															
3	4	1	5	7																																																																															
3	4	1	5	7																																																																															
3	4	1	5	7																																																																															
3	4	1	5	7																																																																															
3	1	4	5	7																																																																															
3	1	4	5	7																																																																															
3	1	4	5	7																																																																															
1	3	4	5	7																																																																															

Рис.38.1

З наведеного прикладу видно, що для реалізації даного алгоритму доцільно буде використати вкладений цикл. В зовнішньому циклі буде визначати номер ітерації з допомогою змінної j . У внутрішньому циклі будемо перевіряти умову, де індекс елемента позначимо змінною i .

Так, для наведеного прикладу оператор зовнішнього циклу буде мати вигляд:

```
for j in range(1,5):
```

Розглянемо діапазон значень для внутрішнього тіла циклу для кожної ітерації:

- для $j=1$, i належить інтервалу $[0,4]$;
- для $j=2$, i належить інтервалу $[0,3]$;
- для $j=3$, i належить інтервалу $[0,2]$;
- для $j=4$, i належить інтервалу $[0,1]$.

Отже, оператор внутрішнього циклу буде мати вигляд:

```
for i in range(0,5-j):
```



Для порівняння елементів доцільно скористатися оператором розгалуження:

```
if a[i]>a[i+1]:
```

Таким чином буде здійснюватися порівняння поточного елемента масиву з наступним. У разі виконання умови елементи необхідно поміняти місцями, використовуючи команду:

```
a[i],a[i+1]=a[i+1],a[i]
```

Отже, для сортування списку $a=[7, 4, 3, 5, 1]$ за зростанням методом «бульбашок» програмний код буде таким:

```
a=[7,4,3,5,1]
print(a)
for j in range(1,5):
    for i in range(5-j):
        if a[i]>a[i+1]:
            a[i],a[i+1]=a[i+1],a[i]
print(a)
```

ПРАКТИЧНЕ ЗАВДАННЯ 1.



Уведіть наведений вище код та перевірте правильність його виконання. Спробуйте самостійно змінити програму, аби сортування здійснювалося за спаданням елементів.

Нижче наведено алгоритм сортування масиву, що містить n елементів.

Алгоритм сортування масиву для n елементів:

1. Порівняти i -й елемент списку з $i+1$ -м.
2. Поміняти елементи місцями:
 - а) для сортування за зростанням, якщо i -й елемент *більший*;
 - б) для сортування за спаданням, якщо i -й елемент *менший*.
3. Повторити описані вище дії від початку списку до останнього невідсортованого елемента включно.

ВПРАВА 38.1



Завдання. Виконати завдання вправи 37.1, застосувавши при сортуванні метод бульбашок.

1. Завантажте файл-заготовку **Вправа_38_1.py** із матеріалів до даного параграфа.
2. Створіть обробник події натискання кнопки **За глибиною** для сортування елементів за спаданням.



- a. Створіть команду обробника події натискання кнопки та додайте оператор циклу сортування масиву **depth** за спаданням методом бульбашок:

```
def btn2_click():  
    for j in range(1,7):  
        for i in range(7-j):  
            if depth[i]<depth[i+1]:  
                depth[i],depth[i+1]=depth[i+1],depth[i]
```

- b. Оскільки згідно з умовою відповідні дані, що розміщені в інших списках, мають переміщуватися на ті ж місця, що й глибини озер під час сортування, у тілі цього ж циклу додайте команди обміну елементами списків **name** та **area**.

```
name[i],name[i+1]=name[i+1],name[i]  
area[i],area[i+1]=area[i+1],area[i]
```

- c. Додайте команди очищення елементів керування **Lbox_name**, **Lbox_depth**, **Lbox_area**.
- d. Додайте оператор циклу виведення відсортованих списків до елементів керування **Lbox_name**, **Lbox_depth**, **Lbox_area**.
3. Додайте команду виклику обробника події в конструкторі кнопки **За глибиною**.
4. Обробники подій натискання кнопок **За назвою** та **За площею** створіть самостійно.

ВПРАВА 38.2



Завдання. Дано список переможців олімпіади з інформатики (рис. 38.2). Створити обробник події для кнопки **Призові місця**, що має:

- виконувати сортування призерів олімпіади за рейтингом, використовуючи метод «бульбашок»;
- виводити у списку **Lbox_rez** ступінь диплома, одержаного переможцем.

Врахуйте, що кількість дипломів розраховується у співвідношенні 1:2:3 від загальної кількості переможців.

1. Завантажте файл-заготовку **Вправа_38_2.py** із матеріалів до даного параграфа.

У наданому файлі розташовано інтерфейс програми. Елемент **Lbox_name** містить прізвища переможців. **Lbox_grade** містить результати 13 переможців, що визначається випадковим числом від 50 до 100.

Створення обробника події для кнопки

2. Створіть команди сортування списку **grade** за спаданням методом бульбашок. Зауважте, що даний список має стільки ж елементів, скільки і список **name**.



Результати олімпіади		
Переможці олімпіади	Результат	Ступінь диплому
Кротенко	95.8	I
Волошина	94.6	I
Романова	91.8	II
Петренко	90.2	II
Антоненко	86.9	II
Орленко	78.5	II
Новікова	77.8	III
Гнатов	72.3	III
Балацький	62.8	III
Макаров	57.6	III
Литвиненко	55.4	III
Шевченко	52.9	III
Діденко	51.6	III

Призові місця

Рис.38.2

```
def btn1_click():
    for j in range(1,12):
        for i in range(13-j):
            if grade[i]<grade[i+1]:
                grade[i],grade[i+1]=grade[i+1],grade[i]
```

- У тілі внутрішнього циклу додайте команду обміну місцями елементів масиву **name**.
- Додайте команди очищення елементів керування **Lbox_grade** та **Lbox_name**.
- Додайте команди виведення елементів списків **grade** та **name** до елементів **Lbox_grade** та **Lbox_name** відповідно.
- Додайте команду обчислення кількості дипломів першого ступеня:

```
k1=int(13/6)
```

Із пропорції 1:2:3 неважко здогадатися, що таких дипломів буде шоста частина від загальної кількості призерів. Функція **int** дозволить одержати цілу частину від ділення.

- Додайте команду обчислення кількості дипломів другого ступеня:

```
k2=int(13/3)
```

- Додайте команду обчислення кількості дипломів третього ступеня. Як правило, дипломи третього ступеня одержують усі призери олімпіади, що не отримали дипломи першого та другого ступенів.

```
k3=13-k2-k1
```



9. Додайте команди виведення напису «I» навпроти учасників, що отримали диплом I ступеня:

```
for i in range(k1):  
    Lbox_rez.insert(END, 'I')
```

10. Додайте команди виведення напису «II» навпроти учасників, що отримали диплом II ступеня.
11. Додайте команди виведення напису «III» навпроти учасників, що отримали диплом III ступеня.
12. Додайте команду виклику обробника події.
13. Запустіть програму та перевірте правильність її виконання.
- 14*. Самостійно змініть код таким чином, аби всі дії в ньому виконувалися при зміні кількості переможців у списку **name**.

38.2. Складність алгоритму

При створенні алгоритму програмісти часто кажуть про його складність, швидкодію тощо. Як визначити, який алгоритм є більш складним чи більш швидким? За кількістю команд? Чи є якісь інші критерії визначити це?

Існує декілька способів вимірювання складності алгоритму. Програмісти, як правило, звертають увагу на швидкість алгоритму або об'єм пам'яті, що використовується під час його виконання.

Отже, **складність алгоритму** — це кількісна характеристика, що відображує споживані алгоритмом ресурси під час свого виконання.

Відповідно складність залежить від *розмірів* вхідних даних. Звичайно, масив з 10 елементів буде оброблено значно швидше, ніж аналогічний зі 100 елементів.

Часова складність алгоритму характеризується продуктивністю алгоритму, що визначається кількістю елементарних операцій, необхідних для виконання алгоритму. Вважається, що кожна елементарна операція виконується за однаковий час. Часову складність оцінюють для найгіршого випадку і визначають як максимальний час, необхідний для обробки алгоритмом будь-якої множини з n елементів.

Часову складність визначають за виразом $O(f(n))$. Цей вираз означає, що час роботи алгоритму зростає в залежності від вхідних даних не швидше, ніж деяка константа, помножена на $f(n)$.



38.3.

Поширені складності алгоритмів

$O(1)$ — константна складність.

Час виконання цього алгоритму не залежить від розміру вхідних даних. Проте одиниця у формулі не означає, що алгоритм виконується за одну операцію або потребує дуже мало часу. Він може виконуватися як мікросекунду, так і цілий день.

Наприклад, дано масив цілих чисел. Необхідно визначити кількість елементів масиву.

Алгоритм розв'язання даної задачі буде наступним:

```
a=[2,5,7,8,10]
k=len(a)
```

Незалежно від розміру масиву в даному випадку буде виконано дві операції створення масиву та визначення його довжини.

$O(n)$ — лінійна складність

У даному випадку складність алгоритму лінійно зростає при збільшенні вхідних даних. Якщо кількість вхідних даних збільшити у 2 рази, то і час виконання алгоритму буде збільшено у 2 рази. Такі алгоритми легко «впізнати» за наявністю циклу по кожному елементу масиву. Наприклад, обчислення суми парних елементів масиву:

```
s=0
a=[2,5,7,8,10]
for i in range(5):
    if a[i]%2==0:
        s=s+a[i]
```

Лінійну складність має також алгоритм сортування масиву вибором, що визначається наявністю одного циклу під час сортування.

```
a=[3,5,1,2,7]
for i in range(4):
    m=max(a[0:5-i])
    n=a.index(m)
    a[n],a[4-i]=a[4-i],a[n]
```

$O(n^2)$ — квадратична складність



Час роботи такого алгоритму зростає пропорційно квадрату кількості елементів, що обробляються. Отже, при збільшенні вхідних даних у 2 рази, час виконання алгоритму збільшується у 4 рази. Такі алгоритми легко розпізнати за наявністю вкладених циклів. Яскравим прикладом алгоритму, що має квадратичну складність, є сортування масиву методом бульбашок.

```
a=[7,4,3,5,1]
for j in range(1,5):
    for i in range(5-j):
        if a[i]>a[i+1]:
            a[i],a[i+1]=a[i+1],a[i]
```

ВПРАВА 38.3



Завдання. Створити програми розв'язання задач із вправи 38.3, використовуючи для сортування метод бульбашок.



Контрольні запитання та завдання

1. Опишіть алгоритм сортування методом бульбашок.
2. Якою є складність алгоритму сортування вибором; сортування методом бульбашок?
3. Що таке складність алгоритму?
4. Які існують види складності?
- 5*. Як змінити команди сортування методом бульбашок за зростанням для сортування цим же методом за спаданням?



Питання для роздумів

- 1*. Установіть послідовність, в якій змінюється список $a=[5, 4, 7, 9, 6]$ у процесі сортування за спаданням методом бульбашок.

№	0	1	2	3	4
	9	7	6	5	4
	5	4	7	9	6
	7	9	5	6	4
	5	7	9	4	6
	7	5	9	6	4
	5	7	9	6	4
	5	7	4	9	6
	7	9	6	5	4

- 2*. Яким буде програмний код сортування методом бульбашок, якщо оператор циклу з умовою замінити циклом з параметром?



Завдання для досліджень

1. На сайті **youtube.com** відшукайте та перегляньте відеоролики, що демонструють різні методи сортування з допомогою танців.
- 2*. Проаналізуйте алгоритм сортування масиву методом бульбашок із встановленням прапорця **flag**. Поясніть призначення цього прапорця.

```
a=[3,5,1,8,9]
flag = True
while flag:
    flag = False
    for i in range(4):
        if a[i] > a[i + 1]:
            a[i], a[i + 1] = a[i + 1], a[i]
            flag = True
print(a)
```

- 3*. Ознайомтеся із сортуванням методом вставки.



§39.

ВІЗУАЛІЗАЦІЯ ДАНИХ

39.1.

Створення графічних примітивів

Для зручного представлення табличних даних використовують їхню візуалізацію. Дані можна представити у вигляді різноманітних діаграм та графіків. Для їхньої побудови, як правило, використовують графічні примітиви — такі, як відрізки, прямокутники тощо.

Так, на рис. 39.1 представлено приклад гістограми, побудованої в графічному вікні. Вона складається з таких графічних примітивів, як відрізок — вісь гістограми та прямокутників — стовпців з даними. Розглянемо, як створити такі елементи в графічному вікні.



Рис.39.1



У 8-му класі ми вже виводили в графічному вікні зображення із файла. Напевно, ви пам'ятаєте, що для цього необхідно було спочатку створити графічне полотно з використанням конструктора **Canvas**. Для створення графічних примітивів також потрібно створити це полотно та розташувати в графічному вікні.

Так, наприклад, це можуть бути такі команди:

```
canvas=Canvas(tk,width=350,height=250)
canvas.pack()
```

Далі розглянемо створення графічних примітивів.

Створення відрізка

Для створення відрізка використовують метод **create_line**. Цей метод має таку структуру:

```
полотно.create_line(x1,y1,x2,y2,fill=колір,  
                    arrow=напря́м_стрілки)
```

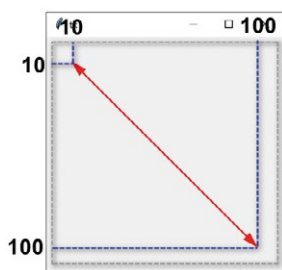


Рис.39.2

Тут **x1, y1** — координати початку відрізка; **x2, y2** — координати кінця відрізка; **fill** — колір лінії (за умовчанням чорний); **arrow** — напрям стрілок кінців відрізка. Може приймати такі значення: **NONE** — стрілки відсутні (за умовчанням), **FIRST** — на початку, **LAST** — в кінці, **BOTH** — з обох боків.

Так, на рис. 39.2 наведено приклад побудови відрізка з використанням команди:

```
canvas.create_line(10,10,100,100,fill=RED,arrow=BOTH)
```

Створення прямокутника

Для створення прямокутника використовують метод **create_rectangle**. Цей метод має таку структуру:

```
полотно.create_rectangle(x1,y1,x2,y2,  
                          fill=колір,  
                          outline=колір)
```

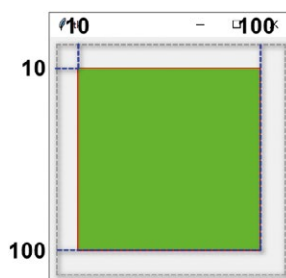


Рис.39.3

Тут **x1, y1** — координати верхнього лівого кута прямокутника; **x2, y2** — координати нижнього правого кута прямокутника; **fill** — колір заливки (за умовчанням чорний); **outline** — колір межі (за умовчанням чорний).

Так, на рис. 39.3 наведено приклад побудови прямокутника з використанням команди:

```
canvas.create_rectangle(10,10,100,100,  
                        fill="green",outline="red")
```



Створення написів

Крім відрізків та прямокутників, у гістограмі присутні підписи. Їх виводять з допомогою методу **create_text**. Цей метод має таку структуру:

```
полотно.create_text(x,y,fill=колір,text=текст)
```

Тут **x, y** — координати нижнього лівого кута напису; **fill** — колір тексту (за умовчанням чорний); **text** — текст, який необхідно вивести. Наприклад,

```
полотно.create_text(10,10,text="Гістограма",fill="red")
```

ВПРАВА 39.1. ПОБУДОВА ГІСТОГРАМИ



Завдання. Побудувати гістограму відвідування спортивних секцій учнями школи відповідно до зразка (рис. 39.1).

Створення графічного вікна, полотна та осі x

1. Уведіть команди створення графічного вікна із заголовком **Спортивні секції**.
2. Додайте команди створення та розташування у вікні графічного полотна:

```
canvas=Canvas(tk,width=350,height=250)
canvas.pack()
```

3. Додайте команду виведення зображення осі **x**.

```
canvas.create_line(10,200,330,200,arrow=LAST)
```

4. Запустіть програму на виконання та переконайтеся, що вісь **x** побудована в нижній частині вікна та має стрілку, спрямовану праворуч.
5. Додайте команди створення двох списків: **sport**, що містить перелік спортивних секцій, та **k** — кількості відвідувачів у секціях:

```
sport=['Футбол','Волейбол','Плавання','Теніс','Гімнастика']
k=[80, 120, 40, 160, 100]
```

Побудова ряду даних (прямокутників)

Спочатку розглянемо, яким чином можна визначити координати першого прямокутника. Для побудови прямокутника необхідно скористатися кодом, що має таку конструкцію:

```
полотно.create_rectangle(x1,y1,x2,y2,
                        fill=колір,outline=колір)
```

Тут **x1, y1** — це координати верхнього лівого кута. Нехай **x1=20**. Висота прямокутника — це перше значення списку **k**, а саме, **k[0]**.



Вісь **x** розташована на відстані 200 рх. Отже, **y1=200-k[1]**. Ширина стовпця дорівнює 40 рх, отже, **x2=20+40=60**. Оскільки нижній правий кут лежить на осі **x**, то **y2=200**. Таким чином, для побудови першого прямокутника необхідно скористатися командою:

```
canvas.create_rectangle(20,200-k[0],60,200,fill="orange")
```

Врахуємо, що відступ між стовпцями 20 пікселів, отже, для побудови другого стовпця це буде команда:

```
canvas.create_rectangle(80,200-k[1],120,200,fill="orange")
```

Аналогічно можна вивести команду і для інших прямокутників. Але, оскільки між прямокутниками однакова відстань, то для побудови прямокутників краще скористатися оператором циклу:

```
x=20
for i in range(5):
    canvas.create_rectangle(x,200-k[i],x+40,200,fill="orange")
    x=x+60
```

Тут **i** — лічильник, що визначає номер прямокутника. **x** має початкове значення 20 і на кожному кроці змінюється на 60.

6. Додайте розглянутий вище код. Запустіть програму та перевірте правильність побудови прямокутників.

Створення написів

7. У тілі циклу додайте команду виведення написів під стовпцями. Самостійно визначте місце розташування команди у програмному коді.

```
canvas.create_text(x+20,215,text=sport[i])
```

У даній команді вказано, що написи буде зміщено на 20 пікселів по осі **x**, відносно початку розташування прямокутників, а також опущено на 15 пікселів униз. Написи секцій буде взято із масиву **sport**.

8. У тілі циклу додайте команду виведення значень даних над стовпцями. Самостійно визначте місце розташування команди у програмному коді.

```
canvas.create_text(x+20,190-k[i],text=k[i])
```

9. Запустіть програму та перевірте правильність її виконання.

ВПРАВА 39.2. ПОБУДОВА ГРАФІКА

Коли дані, що містить масив, змінюються через однакові проміжки часу, для їхнього візуального відображення доцільно будувати графіки. Процес побудови графіків у Python розглянемо, виконуючи практичну вправу.



Завдання. Побудувати графік випадіння опадів за кожний місяць по Україні відповідно до зразка (рис. 39.4). При натисненні кнопки **Створити список** має бути сформований список опадів на кожний місяць, де значенням є випадкове число від 40 до 120. При натисненні кнопки **Побудувати графік** має здійснюватися побудова графіка за даними зі списку.

Створення списку опадів

1. Завантажте файл **Вправа_39_2.py** із матеріалів до даного параграфу. У наданому файлі здійснено імпорт необхідних модулів та створено елементи інтерфейсу програми: список **Lbox** та дві кнопки **Btn1** та **Btn2**.
2. Створіть обробник події натиснення кнопки **Створити список**.

```
def Btn1_click():
    p=[]
    #очищення списку від попередніх значень
    Lbox.delete(0,END)
    for i in range(12):
        #створення елемента, що є випадковим числом
        p.append(randint(40,120))
        #додавання елемента масиву до списку Lbox
        Lbox.insert(END,p[i])
```

3. В конструкторі кнопки **Btn1** додайте команду виклику обробника події **Btn1_click**.
4. Запустіть програму та перевірте створення списку.

Створення осей

5. Додайте команди створення та розташування у вікні графічного полотна:

```
canvas=Canvas(tk,width=300,height=250)
canvas.place(x=80,y=10)
```

6. Додайте команди створення координатних осей:

```
canvas.create_line(30,160,290,160,arrow=LAST) #x
canvas.create_line(30,160,30,10,arrow=LAST) #y
```

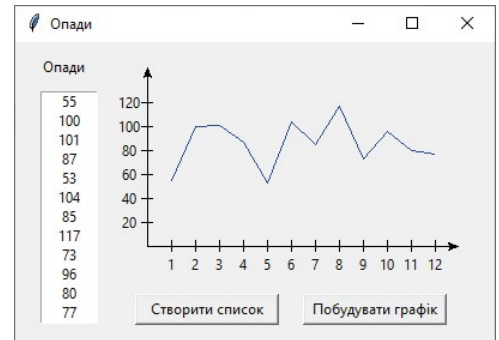


Рис.39.4



З програмного коду видно, що початок координат зміщено на 30 праворуч відносно початку координат графічного полотна. Цей відступ потрібен для розташування біля осі **y** штрихів та підписів.

7. Додайте команди створення штрихів та підписів на осі **x**.

```
#штрихи та підписи на осі x
x=[]
for i in range(12):
    x.append(20*i+50)
    #створення штрихів
    canvas.create_line(x[i],155,x[i],165)
    #створення підписів
    canvas.create_text(x[i],175,text=i+1)
```

У даному коді створено масив **x**, що буде містити значення координат штрихів по осі **x**. Значення по **x** будуть змінюватися через кожні 20 рх, отже, перша координата відносно полотна буде знаходитися на відстані 30 (тут розташовано початок координат) + 20 = 50 рх. Отже, загальна формула для значень масиву **x** має бути **20*i+50**, де **i** змінюється від 0 до 11 включно.

8. Додайте команди створення штрихів та підписів на осі **y**.

Оскільки максимальне значення, можливе по осі **y** — це 120, а ціна поділки дорівнює 20, то для створення штрихів та підписів достатньо 6-ти ітерацій. Масив **y** буде мати такі значення [20,40,...,120].

Отже, для його створення введіть такий код:

```
y=[]
for i in range(6):
    y.append(20*(i+1))
```

9. Нижче у тілі циклу додайте команди створення штрихів та підписів по осі **y**.

```
canvas.create_line(25,160-y[i],35,160-y[i])
canvas.create_text(15,160-y[i],text=y[i])
```

10. Запустіть програму на виконання та перевірте правильність побудови осей.

Побудова графіка

11. Створіть обробник події кнопки **Побудувати графік**.

Оскільки графік складається з відрізків, то для його побудови доцільно використовувати метод **create_line**. Також неважко помітити, що таких відрізків, що сполучають 12 точок, буде 11. Для даного



методу $x1=x[i]$, $y1=160-p[i]$, $x2=x[i+1]$, $y2=160-p[i+1]$. Отже, $x[i]$ та $p[i]$ — це поточні елементи масивів, а $x[i+1]$ та $p[i+1]$ — це наступні елементи масивів.

Враховуючи попередні пояснення, уведіть наведений нижче код:

```
def Btn2_click():
    for i in range(11):
        canvas.create_line(x[i],160-p[i],x[i+1],
                           160-p[i+1],fill="blue")
```

12. В конструкторі кнопки **Btn2** додайте команду виклику обробника події **Btn2_click**.
13. Оскільки створення масиву **p** відбувалося в іншому обробнику подій, змінна **p** має бути оголошена глобальною. Додайте оголошення змінної **p** в якості глобальної в тілі обробника події **Btn1_click**:

```
global p
```

14. Запустіть програму на виконання та перевірте правильність побудови графіка. Сформуйте список ще раз, натиснувши кнопку **Створити список** та побудуйте графік, натиснувши кнопку **Побудувати графік**. В результаті новий графік буде накладатися на попередній. Аби цього не відбувалося, потрібно очистити полотно від попереднього зображення.

Щоб очистити полотно від зображення, використовують метод **delete**, що має таку конструкцію: `полотно.delete(ALL)`.

15. Додайте команду `canvas.delete(ALL)` до обробника події кнопки **Btn2_click** перед оператором циклу та перевірте, як буде працювати програма. В результаті натиснення кнопки **Побудувати графік** буде видалено все, що було побудовано до графіка.

Для видалення лише одного об'єкта, а в нашому випадку перед побудовою нового графіка ми маємо видаляти попередній, в команді створення відрізка графіка необхідно додати властивість **tags=ім'я**, а в якості параметра методу **delete** вказати ім'я даного тегу, який ми бажаємо видаляти.

16. В команді створення відрізка графіка додайте властивість **tags**.

```
canvas.create_line(x[i],160-p[i],x[i+1],160-p[i+1],
                   fill="blue",tags="g")
```

17. Змініть параметр методу **delete** наступним чином:

```
canvas.delete("g")
```

18. Запустіть програму та перевірте правильність її виконання.



ВПРАВА 39.3



Завдання. Дано масив даних з прибутками та збитками магазину за один тиждень.

1. За даними масиву побудуйте графік. (Дані для масиву заповніть випадковим чином в діапазоні $[-10000; 10000]$).
- 2*. Над даними, що мають прибуток, розташуйте напис.
- 3**. Дані, що мають збиток, позначте червоним квадратним маркером.

ВПРАВА 39.4



Завдання. На змаганнях з фігурного катання спортсмен одержує оцінку свого виступу від 10 суддів. Оцінка, яку може виставити один суддя, визначається кількістю балів від 0 до 6. Після одержання усіх оцінок найкраща та найгірша оцінки відкидаються, і спортсмену нараховується сумарний бал оцінок, що залишилися.

За результатами оцінок суддів побудуйте стовпчасту діаграму. Стовпчики зафарбуйте синім кольором. (Масив оцінок суддів заповніть випадковим чином дійсними числами від 0 до 6 з одним десятковим знаком.)

Стовпці, що містять дані з максимальним та мінімальним результатами, зафарбуйте червоним кольором.



Контрольні запитання та завдання

1. Які графічні примітиви використовуються для побудови гістограм та графіків?
2. Як побудувати в графічному вікні відрізок? Які параметри використовують для його побудови?
3. Для чого використовують метод `create_rectangle`?
3. З допомогою якого методу можна видалити об'єкти з полотна?
- 4*. Як видалити один із об'єктів з полотна?



Питання для роздумів

- 1*. Яким має бути програмний код для побудови секторної діаграми?



Завдання для досліджень

- 1*. Ознайомтеся з методами побудови інших графічних примітивів (еліпса, багатокутника тощо).





§40. ДВОВИМІРНІ МАСИВИ

40.1.

Поняття двовимірного масиву

Ми вже познайомилися з опрацюванням одновимірних масивів. У мові Python є можливість опрацювання масивів, що містять більше одного рядка. Такі масиви називають *двовимірними масивами* або *таблицями*. В математиці двовимірні масиви називають *матрицями*.

Двовимірні масиви містять дані одного типу, де кожний елемент визначається двома числами: номером рядка та номером стовпця, на перетину яких його розташовано. Так, на рис. 40.1 наведено приклад цілочисельного двовимірного масиву, що містить 3 рядки та 4 стовпці.

рядки \ стовпці	0	1	2	3
	0	1	2	3
0	5	6	3	2
1	1	0	4	5
2	3	2	5	7

Рис. 40.1

Зверніть увагу, що нумерація як рядків, так і стовпців починається з нуля. Двовимірні масиви у мові Python реалізуються через списки, вкладені до списку. Отже, наведений вище приклад буде представлено таким чином: `[[5,6,3,2],[1,0,4,5],[3,2,5,7]]`. Для наочності списки можна подавати у наступному вигляді:

```
[[5,6,3,2],
 [1,0,4,5],
 [3,2,5,7]]
```

З наведеного прикладу видно, що весь масив обмежується квадратними дужками. Також квадратними дужками обмежуються рядки масиву.



40.2. Створення двовимірного масиву

Розглянемо способи створення двовимірних масивів.

1 спосіб

Найпростіший спосіб — це присвоєння змінній списку із вкладеними списками. Нижче наведено загальний вигляд такого списку:

```
змінна=[рядок_1],[рядок_2],..., [рядок_n]]
```

Так, нижче наведено приклад створення та виведення цілочисельного списку, що містить 3 рядки та 4 стовпці.

```
#створення списку
mas=[[0,1,0,1],[0,1,0,1],[0,1,0,1]]
#виведення списку
print(mas)
```

Якщо необхідно вивести не весь список, а один із його елементів, команда виведення буде мати таку загальну конструкцію:

```
print(ім'я_списку[номер_рядка][номер_стовпця]).
```

Так, для виведення елемента масиву **mas**, що знаходиться у другому рядку та нульовому стовпці, необхідно скористатися командою:

```
print(mas[2][0])
```

ПРАКТИЧНЕ ЗАВДАННЯ 1

1. Уведіть програмний код створення та виведення масиву **mas**, наведеному в якості прикладу.
2. Запустіть програму та перевірте, як вона працює.
3. Додайте команду виведення елемента, що знаходиться у другому рядку та в нульовому стовпці.
4. Перевірте, як буде працювати програма тепер.

Для виведення кожного рядка списку з нового рядка потрібно скористатися оператором циклу:

```
for i in mas:
    print (i)
```

5. Змініть команду виведення списку на команди, представлені вище, та перевірте, як буде здійснюватися виведення списку цим способом.

Зручним також буде представлення елементів двовимірного масиву без зайвих символів, а саме:



```
0 1 0 1
1 0 1 0
0 1 0 1
```

Для такого подання слід скористатися вкладеним циклом:

```
for i in range(3):
    for j in range(4):
        print(a[i][j],end="\t")
    print()
```

Тут *i* — номер поточного рядка, *j* — номер поточного стовпця, *\t* — символ табуляції, що дозволяє зробити відступ між елементами рядка.

6. Змініть команду виведення списку на команди, представлені вище, та перевірте, як буде здійснюватися виведення списку цим способом.

2 спосіб

Для заповнення масиву користувачем, за формулою або випадковим чином спочатку створюють масив, що має визначену кількість рядків та стовпців. Цей масив заповнюють деякими значеннями, наприклад, нулями. Після цього додають програмний код, в якому замінюють ці значення на ті, що вказані в умові завдання.

Так, для створення масиву, що має нульові значення, можна скористатися такою загальною конструкцією:

```
mas=[[0]*кількість_стовпців for i in range(кількість_рядків)]
```

ПРАКТИЧНЕ ЗАВДАННЯ 2

1. Уведіть команди створення та виведення масиву, що має нульові значення з 3-х рядків та 5-ти стовпців.
2. Запустіть програму та перевірте, як вона працює.

3 спосіб

Коли значення елементів списку не має особливого значення, їх можна заповнити випадковим чином. Так, нижче наведено приклад створення двовимірного масиву, що має 5 рядків та 5 стовпців цілими випадковими числами від 0 до 9.

```
from random import*
mas=[[0]*5 for i in range(5)]
for i in range(5):
    for j in range(5):
```



```
mas[i][j]=randint(0,9)
print(mas[i])
print()
```

ПРАКТИЧНЕ ЗАВДАННЯ 3

1. Уведіть програмний код створення та виведення двовимірного масиву цілих випадкових чисел із діапазону [1,20], що має 4 рядки та 3 стовпці.

4 спосіб

Уводити елементи до списку може також користувач під час виконання програми. Для цього в попередній програмі слід замінити команду присвоєння на наступну:

```
a=input('mas['+str(i)+']['+str(j)+'[=')'
```

ПРАКТИЧНЕ ЗАВДАННЯ 4

1. Уведіть програмний код створення та виведення двовимірного масиву, що має 3 рядки та 4 стовпці, значення до якого має ввести користувач.

40.3.

Пошук підсумкових значень, що задовольняють умову

Над елементами двовимірних масивів можна виконувати ті ж самі операції, що і над елементами одновимірних масивів. Розглянемо пошук підсумкових значень, що задовольняють задану умову, виконавши практичну вправу.

ВПРАВА 40.1



Завдання. Заповнити двовимірний масив цілих чисел розміром 3x4 випадковими числами з інтервалу [10, 90]. Знайти у цьому масиві суму парних чисел та кількість тих, що закінчуються цифрою 5.

1. Уведіть команди створення масиву:

```
from random import*
mas=[[0]*4 for i in range(3)]
for i in range(3):
    for j in range(4):
        mas[i][j]=randint(10,90)
        print(mas[i][j],end='\t')
    print()
```




Як і у випадку опрацювання одновимірних масивів для знаходження суми або кількості елементів, що задовольняють вказану умову, у програмі створюються змінні, яким надають початкове значення, рівне 0.

2. Перед оператором циклу додайте команди встановлення початкових значень змінним для знаходження суми та кількості:

```
s=0
```

```
k=0
```

3. У тілі попереднього циклу додайте команди обчислення суми парних елементів:

```
if mas[i][j]%2==0:  
    s=s+mas[i][j]
```

Так, у першому рядку здійснюється перевірка, чи є поточний елемент масиву парним, якщо умова виконується, то даний елемент додається до значення змінної *s*. Після чого здійснюється перехід до перевірки наступного елемента.

4. Самостійно додайте команду виведення значення суми.
5. Запустіть програму та переконайтеся, що суму знайдено правильно.
6. У тілі цього ж циклу додайте команди обчислення кількості чисел, що закінчуються цифрою 5:

```
if mas[i][j]%10==5:  
    k=k+1
```

Так, у першому рядку здійснюється перевірка, чи закінчується поточний елемент масиву на 5, якщо умова виконується, то значення змінної *k* збільшується на 1.

7. Самостійно додайте команду виведення значення кількості.
8. Запустіть програму та переконайтеся, що кількість знайдено правильно.

ВПРАВА 40.2



Завдання. Заповнити двовимірний масив розміром 3x5 випадковим чином 0 та 1. З'ясувати, чого у масиві більше: 1 чи 0.

ВПРАВА 40.3



Завдання. Заповнити двовимірний масив розміром 5x3 випадковими цілими трицифровими числами. Знайти кількість паліндромів у цьому масиві (наприклад, 121, 767).



40.4. Квадратні масиви

Особливої уваги при роботі з двовимірними масивами заслуговують квадратні масиви. Квадратним є масив, кількість рядків та стовпців у якого є однаковою. У такому масиві виділяють головну та побічну діагоналі (рис. 40.2).

	0	1	2	3	4
0	5	6	3	2	0
1	1	0	4	5	3
2	3	2	5	7	6
3	2	7	1	8	1
4	7	9	1	3	4

Головна діагональ

Побічна діагональ

Рис. 40.2.

З наведеного прикладу добре видно, що індекси елементів головної діагоналі по рядках та стовпцях збігаються. А саме, головна діагональ містить такі елементи масиву: **mas[0][0]**, **mas[1][1]** і т. д. Отже, якщо **i** — це індекс, яким позначаються рядки масиву, то поточний елемент головної діагоналі можна позначити **mas[i][i]**. Значення індексів в елементів побічної діагоналі також можна виразити через індекс рядка **i**, а саме, **mas[n+1-i][i]**, де **n** — кількість рядків (стовпців) масиву.

ВПРАВА 40.4



Завдання. Створити квадратний масив розмірністю **nхn**, усі елементи якого дорівнюють 0, а елементи, розташовані на діагоналях, — одиниці.

1. Уведіть команди введення змінної **n** та створення масиву, що має лише нульові елементи.

```
n=int(input('Введіть розмірність масиву '))
mas=[[0]*n for i in range(n)]
```

2. Додайте команди присвоєння одиниці елементам, розташованим по діагоналях.

```
for i in range(5):
    #присвоєння значень елементам головної діагоналі
    mas[i][i]=1
```



```
#присвоєння значень елементам побічної діагоналі
mas[i][6-i]=1
```

3. Додайте команди виведення масиву.

ВПРАВА 40.5



Завдання. Створити квадратний масив розмірністю $n \times n$, усі елементи якого дорівнюють 0, а елементи, розташовані на зовнішніх сторонах масиву, — одиниці.

ВПРАВА 40.6



Завдання. Створити та вивести квадратний масив, що містить випадкові числа в діапазоні $[-10, 10]$ розмірністю $n \times n$. Знайдіть середнє арифметичне елементів головної діагоналі та кількість від'ємних елементів побічної діагоналі.

1. Самостійно уведіть команди:
 - імпорту модуля **random**;
 - введення змінної n ;
 - створення масиву, що має лише нульові елементи;
 - заповнення масиву випадковими числами в діапазоні $[-10, 10]$ та виведення його у вікні консолі.
2. Додайте команди знаходження середнього арифметичного елементів головної діагоналі та кількості від'ємних елементів побічної діагоналі:

```
s=0
k=0
for i in range(n):
    #обчислення суми елементів головної діагоналі
    s=s+mas[i][i]
    #знаходження кількості
    if mas[i][n+1-i]<0:
        k=k+1
```

3. Додайте команди виведення середнього арифметичного елементів головної діагоналі та кількості від'ємних елементів побічної діагоналі.

40.5.

Перегляд кожного рядка або стовпця

Далі розглянемо приклади складання програм за відомими алгоритмами знаходження суми, кількості, добутку тощо. Але ці алгоритми застосовують до елементів кожного рядка (стовпця) двовимірного масиву. Причому, якщо потрібно щось знайти у кожному стовпці, то



масив переглядають по стовпцях (1 стовпець, 2 стовпці, ...), а якщо потрібно щось знайти у кожному рядку, то масив переглядають по рядках (1 рядок, 2 рядок,...).

При перегляді масиву по рядках (це звичайний перегляд, з уведенням елементів масиву) зовнішній цикл буде здійснюватися по номерах рядків (першому індексу), а внутрішній цикл по номерах стовпців (другому індексу). При перегляді масиву по стовпцях зовнішній цикл буде здійснюватися по номерах стовпців (другому індексу), а внутрішній цикл — по номерах рядків (першому індексу).

ВПРАВА 40.7



Завдання. Дано масив 5x5, елементами якого є випадкові числа в діапазоні [0,9]. Створити масиви **Sr** та **Sc**, елементами якого будуть суми рядків та суми стовпців заданого масиву відповідно.

Приклад роботи програми

Вхідні дані:	Вихідні дані:
1 2 3 4 2	Sr=[12, 13, 14, 12, 21]
3 4 2 1 3	Sc=[16, 18, 14, 13, 11]
4 5 2 1 2	
5 3 2 1 1	
3 4 5 6 3	

1. Уведіть команди створення двовимірному масиву **mas** з розмірністю 5x5.
2. Додайте команди створення масиву **Sr**, елементи якого є суми рядків масиву **mas**.

```
Sr=[]
for i in range(5):
    s=0
    for j in range(5):
        s=s+mas[i][j]
    Sr.append(s)
print(Sr)
```

Зверніть увагу, що в задачах даного типу початкове значення суми встановлюється всередині зовнішнього циклу, перед внутрішнім циклом.

3. Додайте один із способів команд створення масиву **Sc**. Елементами даного масиву є суми стовпців масиву **mas**. (Червоним у наведених кодах показано відмінність від попереднього коду.)



1 спосіб	2 спосіб
<pre>Sc=[] for j in range(5): s=0 for i in range(5): s=s+mas[i][j] Sc.append(s) print(Sc)</pre>	<pre>Sc=[] for i in range(5): s=0 for j in range(5): s=s+mas[j][i] Sc.append(s) print(Sc)</pre>

ВПРАВА 40.8



Завдання. Дано масив 4x3, елементами якого є випадкові числа в діапазоні [-10,10]. Створити масиви **vid** та **pos**, елементами якого будуть кількість від'ємних та кількість додатних елементів у кожному рядку відповідно.

Приклад роботи програми

Вхідні дані: 1 -2 3 -4 -2 -3 -4 -2 -1 -3 4 5 2 1 2 5 -3 -2 0 1 3 -4 5 -6 3	Вихідні дані: vid=[3, 5, 0, 2, 2] pos=[2, 0, 5, 2, 3]
--	--



Контрольні запитання та завдання

1. Що таке двовимірний масив?
2. Як представлено двовимірні масиви у середовищі Python?
3. Яким є синтаксис команди звернення до елемента двовимірного масиву?
4. Що таке квадратний масив? Як називаються діагоналі квадратного масиву?
5. Як створити двовимірний масив із нульових елементів?



Питання для роздумів

- 1*. Як знайти добуток елементів головної діагоналі квадратного масиву?
- 2*. Як створити двовимірний масив з розмірністю **nхm**, що містить 0 та 1, заповнені випадковим чином?



Завдання для досліджень



- 1*. Дізнайтеся, як знайти максимальний (мінімальний) елемент двовимірного масиву.

§41.

ПРАКТИЧНА РОБОТА 9. «ДІЇ З ДВОВИМІРНИМИ МАСИВАМИ»



Завдання. На змаганнях з фігурного катання 10 спортсменів одержують оцінку свого виступу від 5 суддів. Оцінка, яку може виставити один суддя, визначається кількістю балів від 0 до 6. Визначити сумарний бал кожного учасника змагань, установити рейтинг учасників, визначити номер судді, який виставив найбільший бал кожному учаснику. Дані та їхню обробку представити в графічному вікні (рис. 41.1). У виносках наведеного рисунка подано імена елементів керування, що будуть використовуватися у коді програми.

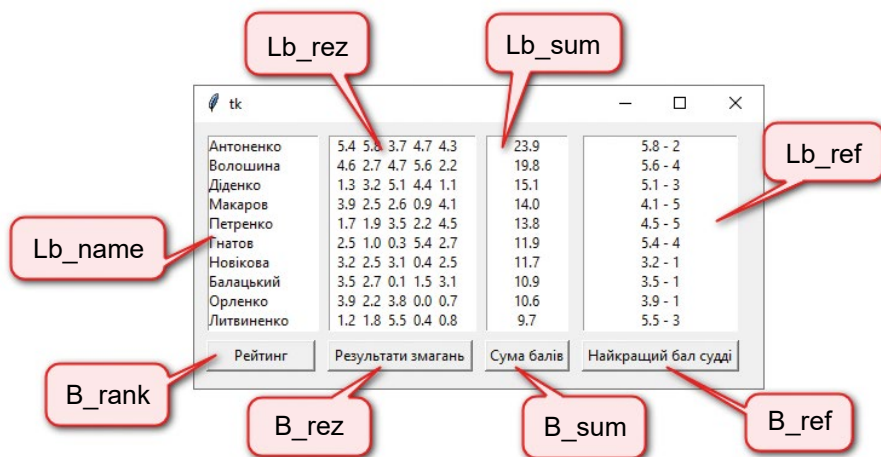


Рис. 41.1

Для виконання завдання скористайтеся файлом-заготовкою **ПР_9.py** із матеріалів до даного параграфу. Даний файл містить команди створення деяких елементів інтерфейсу програми, список **name** із прізвищами учасників змагань.

1. Завантажте та відкрийте файл **ПР_9.py**.
2. Додайте команди виведення списку **name** до елемента керування **Lb_name**. Самостійно визначте місце їхнього розташування:



```
for i in name:
    Lb_name.insert(END,i)
```

Створення та виведення масиву результатів змагань

3. Додайте команду імпорту модуля **random**.
4. Після команди створення графічного вікна додайте команди створення масиву оцінок 10-м учасникам від 5-ти суддів.

```
#заповнення нульового масиву розмірністю 10x5.
rez=[[0]*5 for i in range(10)]
#створення масиву випадковими числами
for i in range(10):
    for j in range(5):
        rez[i][j]=round(uniform(0,6),1)
```

5. Нижче додайте обробник події кнопки **Результати змагань**.

```
def b_rez_click():
    for i in range(10):
        r=''
        for j in range(5):
            r=r + ' ' + str(rez[i][j])
        Lb_rez.insert(END,r)
```

Тут, у тілі зовнішнього циклу формується порожній рядок **r**, який заповнюється та накопичується у тілі внутрішнього циклу елементами поточного рядка масиву **rez** (оцінками від різних суддів одного учасника змагань). Після виконання внутрішнього циклу у тілі зовнішнього циклу здійснюється виведення рядка **r** до елемента керування **Lb_rez**.

6. В конструкторі кнопки **B_rez** додайте виклик обробника події **b_rez_click**.
7. Запустіть програму та перевірте правильність її виконання.

Обчислення та виведення масиву суми балів

8. Додайте команди створення та виведення до графічного вікна елементів **Lb_sum** та **B_sum** (див. рис. 41.1).
9. Додайте команду створення порожнього списку **suma** для знаходження суми оцінок кожного учасника змагань:

```
suma=[]
```

10. Додайте команди обробника події кнопки **Обчислення суми**.



```
def b_sum_click():  
    #очищення Lb_sum від попередніх даних  
    Lb_sum.delete(0,END)  
    for i in range(10):  
        #обчислення та округлення суми поточного рядка  
        s=round(sum(rez[i]),1)  
        #додавання значення суми до масиву  
        suma.append(s)  
        #виведення елемента масиву до Lb_sum  
        Lb_sum.insert(END,suma[i])
```

11. В конструкторі кнопки **B_sum** додайте виклик обробника події **b_sum_click**.
12. Запустіть програму та перевірте правильність її виконання.

Сортування масивів за рейтингом

Для встановлення рейтингу учасників необхідно відсортувати елементи масиву **suma** за спаданням та вивести відсортований масив в елементі **Lb_sum**. Під час сортування елементи масиву мають бути переміщені. Відповідно до цього необхідно перемістити і елементи масиву **name**, а також рядки масиву **rez**.

13. Додайте команди створення та розташування в графічному вікні кнопки **B_rank**.
14. Додайте команди обробника події кнопки **Рейтинг**.
 - a. Уведіть заголовок обробника події **b_rank_click**, а у тілі обробника події команди очищення списків **Lb_name**, **Lb_rez** та **Lb_sum**.
 - b. У тілі обробника додайте команди сортування списку **suma** методом вибору:

```
for i in range(10):  
    #обчислення мінімального значення масиву  
    sum_min=min(suma[0:10-i])  
    #знаходження номера мінімального значення  
    n=suma.index(sum_min)  
    suma[n],suma[9-i]=suma[9-i],suma[n]
```

- c. Нижче додайте команду виведення відсортованого списку до елемента **L_sum**:

```
for i in range(10):  
    Lb_sum.insert(END,suma[i])
```

- d. Самостійно додайте команду переміщення елементів масиву **name** відповідно до переміщення елементів масиву **suma** під час його



сортування, а також команду виведення даного масиву до елемента **Lb_name**.

Для аналогічного переміщення рядків масиву **rez** достатньо звернутися до їхніх індексів. Отже, команда переміщення набуде такого вигляду:

```
rez[n],rez[9-i]=rez[9-i],rez[n]
```

- e. Додайте наведену вище команду до тіла циклу сортування масиву.
- f. Додайте команди виведення оновленого масиву **rez** у тілі циклу виведення масивів **name** та **suma**.

```
r= ''
for j in range(5):
    r=r + ' ' + str(rez[i][j])
    Lb_rez.insert(END,r)
```

- 15. В конструкторі кнопки **B_rank** додайте виклик обробника події **b_rank_click**.
- 16. Запустіть програму та перевірте правильність її виконання.

Визначення номерів суддів, які поставили найбільший бал

Для визначення номерів суддів, які поставили найбільший бал, необхідно знайти максимальне значення кожного рядка та його номер у масиві **rez**. Знайдені значення вивести у форматі: **максимальний бал — номер судді**.

- 17. Додайте команди створення та розташування в графічному вікні елементів керування **Lb_ref** та **B_ref**.
- 18. Додайте команди обробника події кнопки **Найкращий бал судді**.
- 19. В конструкторі кнопки **B_ref** додайте виклик обробника події **b_ref_click**.
- 20. Запустіть програму та перевірте правильність її виконання.
- 21*. Самостійно обчисліть та визначте середній бал кожного учасника змагань.
- 22*. Визначте та виведіть медіану оцінки для кожного учасника. Для цього потрібно відсортувати оцінки кожного учасника за зростанням та вивести 3-й елемент масиву.
- 23*. Порівняйте значення, одержані в п.21 та п.22. Як ви думаєте, чому деякі значення не співпадають?



Висновки

Масив — це впорядкований набір елементів одного типу зі спільним ім'ям. У мові Python для створення масивів використовують *списки*.

Для виконання дій зі списками використовують методи та функції, наведені в таблиці.

Функція	Призначення
sum (список)	Обчислення суми елементів
max (список)	Визначення максимального елемента
min (список)	Визначення мінімального елемента
len (список)	Визначення розмірності списку
del (діапазон)	Видалення елементів
Метод	Призначення
append	Додавання значення в кінець списку
extend	Додавання значень в кінець списку
insert	Додавання значення до вказаної позиції списку
pop	Вилучення елемента з вказаної позиції
remove	Вилучення першого входження вказаного елемента
sort	Сортування списку
clear	Очищення списку
index	Пошук номера першого входження елемента
count	Пошук кількості входжень елемента
split	Перетворення текстового рядка з пробілами на список

Для знаходження суми, кількості та добутку елементів, що задовольняють умову, використовують загальні конструкції, наведені в таблиці нижче.

Знаходження суми	Знаходження кількості	Знаходження добутку
<pre>s=0 for i in range(): if умова: s=s+ім'я_списку[i]</pre>	<pre>k=0 for i in range(): if умова: k=k+1</pre>	<pre>p=1 for i in range(): if умова: p=p*ім'я_списку[i]</pre>

Для сортування масивів використовуються методи сортування вибором, бульбашками тощо.

Для обміну елементів масиву $a[i]$ та $a[j]$ місцями використовується така конструкція:

```
a[i],a[j]=a[j],a[i]
```



Розділ 5

3D-графіка



§42.

ПОНЯТТЯ 3D-МОДЕЛЮВАННЯ

42.1.

Що таке 3D-моделювання

3D-моделювання являє собою процес використання програмного забезпечення для створення математичного подання тривимірного об'єкту або форми. Створений об'єкт називається *3D-моделлю*. Такі моделі використовуються у різних галузях, таких, як кіноіндустрія, відеоігри, архітектура, наука та медицина тощо.

3D-моделювання — це технологія побудови об'ємної моделі об'єкта в тривимірному просторі на базі професійних графічних програм.

Що ж таке 3D-зображення? Це зображення, кожна точка якого в тривимірному просторі має три координати: x , y — координати точки на площині та z — координата, що визначає глибину зображення. На рис. 42.1 точка A має координати у просторі (x, y, z) .

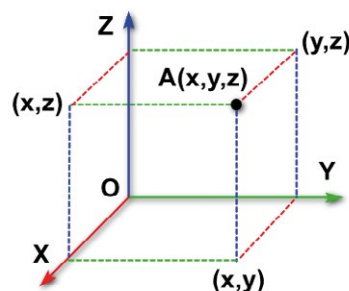


Рис. 42.1. Точка у просторі, задана координатами

Етапи створення 3D-зображення

1. **Побудова 3D-моделі.** На комп'ютері створюється модель, що передає форму об'єкта, який необхідно уявити.
2. **Текстурування.** Встановлення для поверхонь моделей текстур та матеріалів, що надають поверхням моделей реалістичний вигляд.
3. **Освітлення.** Встановлення та налагодження джерел світла.



4. Рендеринг. Одержання зображення за побудованою 3D-моделлю з використанням комп'ютерної програми.

42.2.

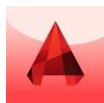
Програми для 3D-моделювання

Розглянемо найбільш розповсюджені програми, що дозволяють здійснювати 3D-моделювання.

3Ds MAX та AutoCAD



3Ds MAX від компанії Autodesk вважається найпотужнішою програмою завдяки широкому функціоналу, універсальності, сумісності з більшістю плагінів і наявності безлічі готових моделей. З її допомогою можна робити практично все, що стосується 3D-моделювання та візуалізації. Нею користуються дизайнери інтер'єрів, творці ігор, архітектори та мультиплікатори. Програма є платною, але існують і безкоштовні версії: пробна на 30 днів або версія для студентів і викладачів на 3 роки.



Крім 3Ds MAX у компанії Autodesk є ще одна дуже популярна програма AutoCAD, що застосовується для створення 2D і 3D-креслень та проектування.

Cinema 4D



Програма Cinema 4D від MAXON — один з найсильніших конкурентів 3DsMax. Програма має інтуїтивно зрозумілий, логічний інтерфейс; новачки можуть користуватися продуктом нарівні з професіоналами; широкий набір функцій; має безкоштовну демо-версію. Оскільки програма є менш популярною, ніж 3Ds MAX, для неї є менше готових 3D-моделей, проте Cinema 4D значно краще підходить для тривимірної анімації.

Blender



Головною перевагою Blender є те, що він безкоштовний. Він має дещо складний інтерфейс, але за своїм функціоналом майже не поступається конкурентам — 3D Max та Cinema 4D. За останні роки Blender став однією з найпопулярніших програм для створення тривимірної графіки. Перевагами над іншими програмами є гнучкість інтерфейсу та часті оновлення; програма підтримується популярними ОС; має всі базові інструменти, що використовуються в інших професійних редакторах; у розпорядженні програми є і базові об'єкти, такі, як куби, кільця, циліндри тощо; програму можна поліпшити, розширивши її функціонал різними плагінами.



TinkerCAD

Програма цікава тим, що не вимагає додаткових інструментів, крім веббраузера. За допомогою даної програми можна створювати 3D-об'єкти, а потім відправляти їх на сервер або комп'ютер файлами у форматі .stl. Програма є безкоштовною для некомерційного використання, проте функціонал у неї є менш потужним, аніж у розглянутих вище програм.



Paint 3D

Найпростіший і найдоступніший варіант для 3D-моделювання. Програма є безкоштовною та вбудованою у Windows 10 за замовчуванням. Отже, користувачу не доведеться нічого шукати, завантажувати і встановлювати. В Paint 3D можна не тільки редагувати, а й створювати власні 3D-моделі. Незважаючи на простоту і доступність, користуватися Paint 3D можуть не тільки новачки. Для професіоналів це непоганий інструмент, за допомогою якого можна швидко і без проблем вносити невеликі правки.



Звичайно, професійне якісне програмне забезпечення коштує недешево. Для роботи з ним потрібно довго і наполегливо вчитися і практикуватися. Але це зовсім не означає, що новачкам потрібно відразу викласти значну суму на покупку ліцензії. У багатьох програм є безкоштовні пробні версії, а також студентські та викладацькі ліцензії на кілька років, якими можна скористатися на період навчання.

42.3.

Інтерфейс програми Blender

Знайомство з 3D-моделюванням продовжимо з використанням програми Blender на прикладі української версії 2.91.2. Зауважимо, що після версії 2.7 інтерфейс програми та використання гарячих клавіш зазнало великих змін, отже, розглянуті далі пояснення будуть актуальними для використання, починаючи з версії Blender 2.8.

Розглянемо декілька елементів інтерфейсу програми Blender (рис. 42.2). При запуску програми центральною її частиною є *сцена*. За умовчанням на сцені відображено три об'єкти — *куб*, *камера* та *освітлення*. Усі об'єкти, розміщені на сцені, дублюються у вікні **Колекція Сцени (Scene Collection)**, що розташовано праворуч. У центрі куба розташовано *курсор*, що, як правило, визначає місце встановлення нових об'єктів. Ліворуч встановлено *панель інструментів*.



Дізнайтеся, як змінити мову інтерфейсу та кольорову схему Blender.

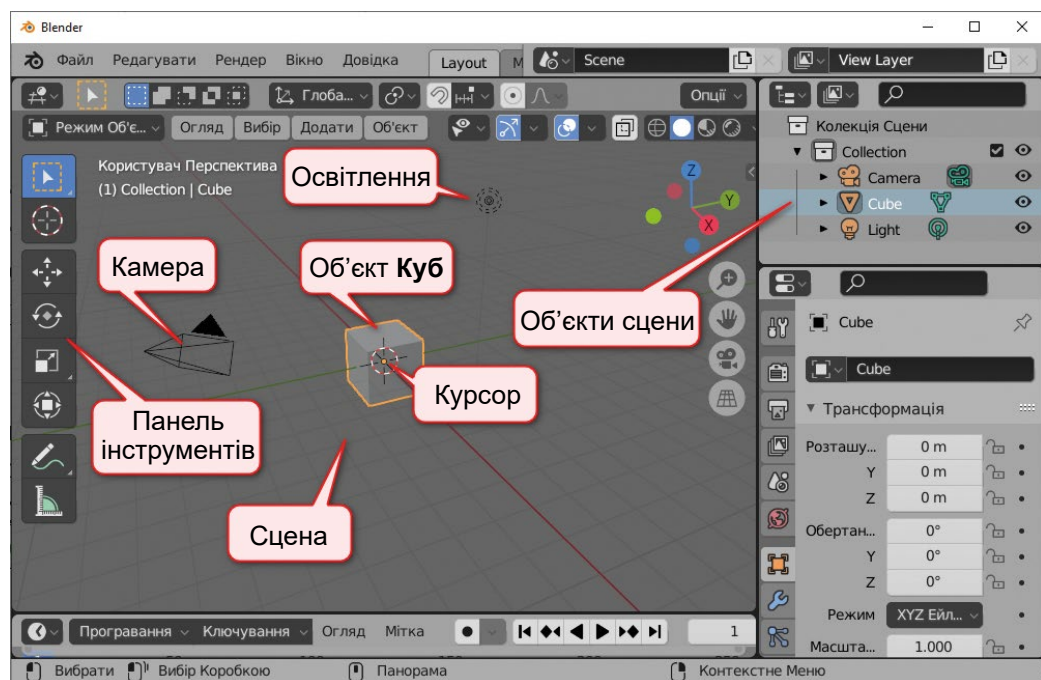


Рис. 42.2. Елементи інтерфейсу програми Blender

ПРАКТИЧНЕ ЗАВДАННЯ 1

1. Запустіть програму Blender. В результаті на фоні програми має з'явитися вікно заставки.
2. Клацніть у довільному місці вікна програми, аби заставка зникла.
3. Приховуйте на сцені камеру та освітлення, натиснувши значок ока біля об'єктів **Camera** та **Light** у вікні **Колекція Сцени (Scene Collection)**.



42.4. Керування видом сцени



Рис. 42.3. Елементи керування сценою

За умовчанням сцена має тривимірний вигляд. У разі необхідності можна змінити її відображення на вид спереду, справа, зверху. Здійснюється це у Blender з допомогою миші, гарячих клавіш та кнопок елемента інтерфейсу. В таблиці 42.1 вказано дії зміни виду сцени з використанням миші.

Керувати сценою можна і з використанням кнопок інтерфейсу (рис. 42.3). Зверніть увагу, що осі у Blender позначено відповідними



кольорами, що використовуються і для подальших дій як зі сценою, так і з об'єктами програми.

Таблиця 42.1. Дії керування сценою з допомогою миші

Дія мишею	Зміна сцени
Прокручування коліщатка	Масштабування сцени
Рух миші при натисненому коліщатку	Обертання сцени
Натиснення Shift та подальший рух миші при натисненому коліщатку	Переміщення сцени

Так натиснення на позначення осі **X**, **Y** або **Z** буде відображати вид справа, позаду та зверху відповідно. Натиснення на кружечки відповідних до даних осей кольорів відображає види зліва, спереду та знизу. Натиснення миші та рух курсору в області рисунка з осями здійснюватимуть довільне обертання сцени. У верхньому правому куті сцени буде відображено поточне її положення.

Натиснення миші на піктограмах із зображенням лупи або руки з подальшим рухом по сцені призводить до масштабування або переміщення відповідно.



Дізнайтеся, як керувати видом сцени з допомогою гарячих клавіш.

ПРАКТИЧНЕ ЗАВДАННЯ 2

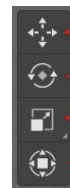
1. Виконайте дії масштабування, обертання та переміщення сцени з використанням миші.
2. Виконайте дії масштабування, обертання та переміщення сцени з використанням кнопок інтерфейсу.

42.5.

Переміщення, обертання та масштабування об'єкта

Основні дії, які можна виконувати з 3D-об'єктами у Blender — це *переміщення, обертання та масштабування*. Їх виконують двома способами: з допомогою елементів інтерфейсу та з використанням гарячих клавіш. Усі зазначені дії виконуються з виділеними об'єктами. Для виділення об'єкта достатньо клацнути на ньому. Для виділення кількох об'єктів слід обвести по них курсором, утримуючи ліву кнопку миші, або клацати по них, утримуючи **Shift**. Видалення виділеного об'єкту здійснюється після натиснення клавіші **Delete**.

Для виконання переміщення, обертання та масштабування використовують відповідні інструменти (рис. 42.4).



Переміщення
Обертання
Масштабування
Усі дії

Рис. 42.4



Після виділення об'єкта при виборі одного з інструментів навколо об'єкта з'являються відповідні покажчики, натиснення та утримання яких призводить до виконання дій відповідно до обраного інструменту (рис. 42.5).

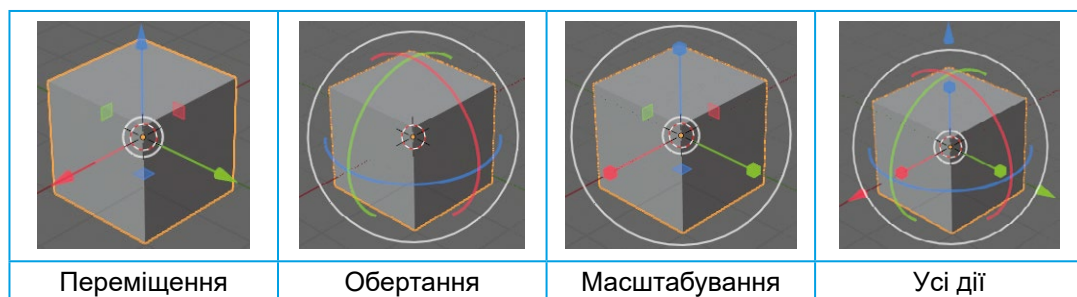


Рис. 42.5

Так, при натисненні синьої стрілки при виборі інструменту переміщення буде здійснено переміщення об'єкта по осі **Z**, а вибір синього чотирикутника переміщує об'єкт по усіх осях крім **Z**. Вибір білої лінії виконує дії у будь-яких напрямках.

ПРАКТИЧНЕ ЗАВДАННЯ 3

1. Виділіть об'єкт **Куб**.
2. Оберіть інструмент **Переміщення (Move)**.
3. Здійсніть переміщення прямокутника, використовуючи послідовно усі можливі маркери, що з'являються при виборі інструменту.
4. Аналогічно виконайте дії обертання та масштабування, обравши відповідні інструменти.
5. Дослідіть можливості інструменту, що виконує усі дії.

Розглянуті вище дії можна здійснювати, використовуючи гарячі клавіші:

- G** (grab) — переміщення;
- R** (rotate) — обертання;
- S** (scale) — масштабування.

Відповідну клавішу необхідно натиснути та відпустити, після чого виконати пересування курсором миші, для завершення дії слід клацнути лівою кнопкою миші.

Наведені гарячі клавіші можна використовувати з іншими клавішами. Так, при послідовному натисненні **G** та **Z** буде здійснюватися переміщення об'єкта по осі **Z**. Натиснення **G**, а потім **Shif+Z** здійснить переміщення об'єкта по осях **X** та **Y** (усі осі, крім **Z**).



ВПРАВА 42.1



Завдання. Створіть 3D-модель стола відповідно до зразка (рис. 42.6).



Перегляньте відео побудови 3D-моделі стола.

1. Створіть стільницю відповідно до зразка.
 - a. Виділіть об'єкт **Куб**.
 - b. Зменшіть висоту куба, змінивши його розмір по осі **Z**. Для цього натисніть послідовно клавіші **S** та **Z**. Після чого пересуньте курсор миші до необхідного положення. Для закінчення дії зменшення натисніть ліву кнопку миші.
 - c. Збільшіть довжину та ширину стільниці, змінивши її розмір по осях **X** та **Y**. Для цього натисніть послідовно клавіші **S** та **Shift+Z**. Після чого пересуньте курсор миші до необхідного положення. Для закінчення дії натисніть ліву кнопку миші.
 - d. Перемістіть стільницю вгору по осі **Z**. Для цього натисніть послідовно клавіші **G** та **Z** та виконайте переміщення.
2. Створіть ніжку для стола.
 - a. Додайте новий куб до сцени. Для цього виконайте команди **Додати (Add) ► Сіть (Mesh) ► Куб (Cube)**.
 - b. Самостійно змініть розміри куба, збільшуючи розмір по осі **Z** та зменшуючи розміри по осях **X** та **Y**.
3. Розташуйте ніжку під стільницею в одному з кутів (рис. 42.7).
 - a. Установіть вигляд сцени **Справа**. Для цього натисніть **X** в області керування осями (рис. 42.8).
 - b. Перетягніть ніжку, як на рис. 42.9. Для цього натисніть послідовно **G** та **Shift+X**.
 - c. Встановіть вигляд сцени **Позаду**. Для цього натисніть **Y** в області керування осями.
 - d. Перетягніть ніжку, як на рис. 42.9. Для цього натисніть послідовно **G** та **X**. В результаті ніжку буде розташовано в одному з кутів стільниці у чому можна переконатися, змінивши сцену на 3D-вигляд.
4. Створіть копію ніжки та перемістіть її до іншого кута стільниці.
 - a. Поверніться до вигляду сцени **Позаду**.
 - b. Виділіть ніжку стола.
 - c. Для копіювання ніжки натисніть **Shift+D**.
 - d. Для переміщення ніжки по горизонталі натисніть **X** та виконайте переміщення праворуч. (Тут для переміщення **G** натискати не

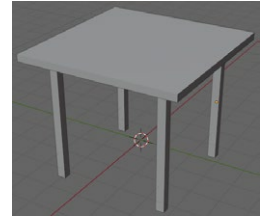


Рис. 42.6



Рис. 42.7

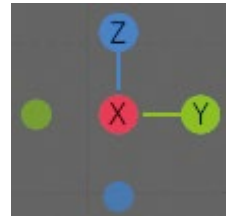


Рис. 42.8

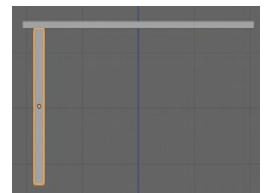


Рис. 42.9



потрібно, оскільки після копіювання ніжки вона знаходиться у стані початку переміщення).

5. Створіть копії утворених ніжок та розташуйте їх під стільницею навпроти вже побудованих.
 - a. Виділіть дві ніжки, навівши на них рамку курсору.
 - b. Оберіть вигляд сцени **Знизу**.
 - c. Для копіювання натисніть **Shift+D**.
 - d. Для переміщення ніжок по горизонталі натисніть **Y**. Виконайте переміщення ніжок у необхідне місце.
6. Виконайте групування елементів стола.
 - a. В області колекція сцени викличте контекстне меню та оберіть команду **Нова колекція**. В результаті має утворитися нова колекція (папка у вигляді коробки).
 - b. Клацніть два рази на назві нової колекції та змініть її ім'я на **Стіл**.
 - c. Виділіть усі об'єкти, що належать до моделі стола, утримуючи **Shift**, та перетягніть їх до утвореної колекції **Стіл**.
7. Перегляньте одержане зображення стола, змінивши сцену на 3D-вигляд.



Виконайте вправу 42.2.

8. Збережіть файл. Для цього в меню **Файл** оберіть команду **Зберегти**. Оберіть папку для збереження та введіть ім'я **Стіл**. Натисніть кнопку **Зберегти файл Blender**.



Контрольні запитання та завдання

1. Що таке 3D-моделювання?
2. Перелічіть етапи створення 3D-зображення.
3. Назвіть програми, що використовуються для створення та редагування 3D-моделей.
4. Які способи використовуються для керування видом сцени у Blender? Які це можуть бути види?
5. Які способи використовуються для переміщення, масштабування та обертання об'єктів у Blender?



Питання для роздумів

1. Чому сцена у Blender розбита на клітинки?
- 2*. Як ви думаєте, чому переважна більшість професіоналів при роботі в 3D-редакторі використовує гарячі клавіші, а не графічні елементи інтерфейсу програми?





Завдання для досліджень

- 1*. Визначте сфери застосування 3D-моделювання. Більш детально підготуйте доповідь про одну із них.
- 2*. Визначте, яким є призначення камери та освітлення, що відображені на сцені під час завантаження Blender.
- 3*. Дослідіть, що відбуватиметься з об'єктом, якщо під час виконання дій переміщення, масштабування та обертання утримувати клавішу **Ctrl**.

§43.

ВЕРШИНИ, РЕБРА, ГРАНІ. ЕКСТРУДУВАННЯ ФОРМИ ОБ'ЄКТА

43.1.

Режим редагування

Дії переміщення, масштабування та обертання 3D-об'єктів у Blender ми здійснювали над суцільним об'єктом. Усе це відбувалося в об'єктному режимі. Проте у суцільному 3D-об'єкті можна виділити окремі елементи: такі, як вершини, ребра та грані (рис. 43.1).

Для виконання дій з цими елементами фігури необхідно змінити об'єктний режим на режим редагування. Для цього можна натиснути клавішу **Tab** або обрати необхідний режим у випадаючому списку, що розташований ліворуч у заголовку редактора (рис. 43.2).

Такі дії, як переміщення, масштабування та обертання в режимі редагування виконуються так само, як і в об'єктному режимі. Проте усі ці операції застосовуються по відношенню до вершин, ребер або граней. Щоб вибрати, з якими саме елементами необхідно виконувати дії, слід обрати відповідний режим виділення (рис. 43.3). Після чого виділяти елементи об'єкта, клацаючи на них курсором. Утримання клавіші **Shif** дозволить виділити послідовно декілька елементів.

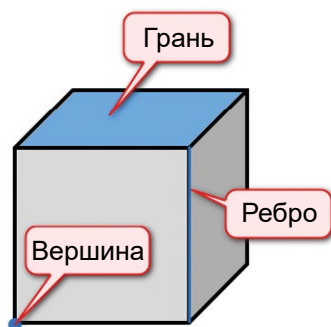


Рис. 43.1. Елементи куба

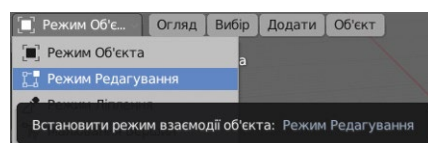


Рис. 43.2

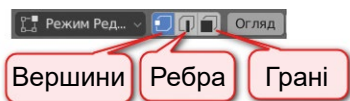


Рис. 43.3



При переході до режиму редагування усі елементи об'єкта є виділеними. Для скасування виділення достатньо клацнути поза межами об'єкта. Для виділення усіх елементів об'єкта достатньо натиснути **A**.

43.2. Додавання 3D-об'єктів

У Blender, крім куба, можна додавати й інші об'єкти, їх також називають *сітками* або *мешоб'єктами*. Для додавання об'єкту в меню **Додати (Add)** слід обрати команду **Сіть (Mesh)**, після чого обрати один із об'єктів (рис. 43.4)

Після додавання об'єкта у нижньому лівому куті програми з'являється вікно для налагодження елементів доданого об'єкта (панель може з'явитися у згорнутому вигляді). Так, при додаванні об'єкта **Циліндр** у даному вікні можна змінити кількість його граней та розміри (рис. 43.5).

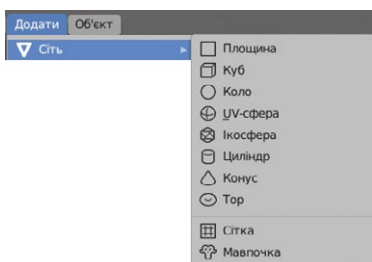


Рис. 43.4

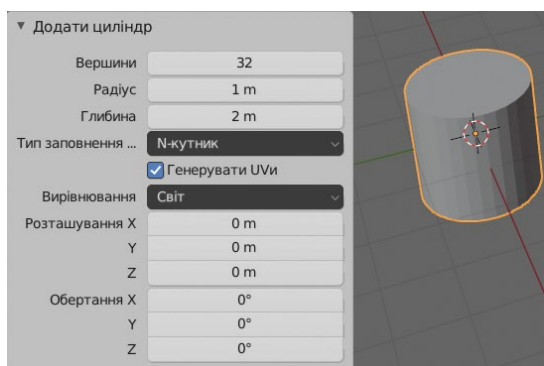


Рис. 43.5

43.3. Екструдкування

Екструдкування (*extrude* за англ. — видавлювати) дозволяє створювати нові вершини, ребра та грані шляхом видавлювання їх із наявних елементів об'єктів. Елемент, до якого застосовується екструдкування, не дублюється, а переноситься на нове місце.

Так, на рис. 43.6 наведено приклади екструдкування вершини, ребра та грані куба.

Екструдкування виконується в режимі редагування і здійснюється після натиснення клавіші **E** та наступного руху курсору в необхідне місце, завершуючись натисненням миші в обраному місці. Наступне натиснення клавіш **X**, **Y** або **Z** призводить до екструдкування за відповідними осями. Для різних способів екструдкування

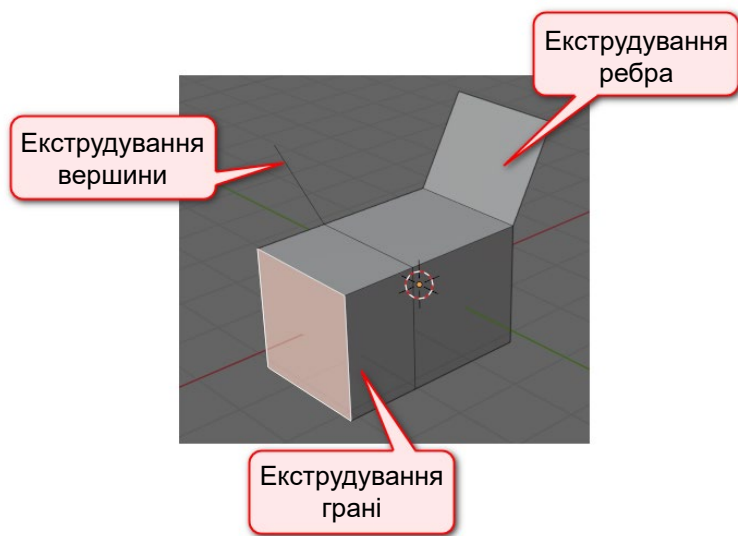


Рис. 43.6

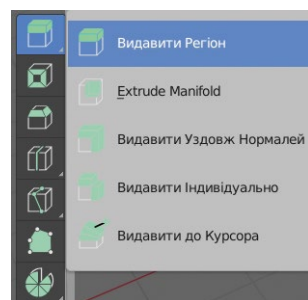


Рис. 43.7

в режимі редагування ліворуч відображається панель екструдкування (рис. 43.7).

Крім стандартного екструдкування, у режимі редагування можна виконати й інші дії з елементами об'єктів.

Так, з використанням гарячої клавіші **I** або інструменту **Вставити грані (Insert Faces)** здійснюється вставлення граней (рис. 43.8).

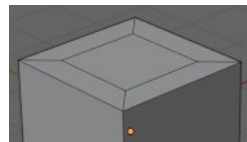


Рис. 43.8

Комбінація клавіш **Ctrl+R** призводить до утворення площин, що перетинають об'єкт. Після натиснення **Ctrl+R** утворюється одна площина. Прокручування коліщатком утворює рівномірно ще необхідну кількість площин. Після натиснення миші можна перетягнути утворені площини у необхідне місце і натиснути ще раз мишею для завершення дії. Після виконаних дій на об'єкті буде утворено нові грані (рис. 43.9). Для виконання описаної дії можна також скористатися інструментом **Замкнутий розріз (Loop Cat)**.

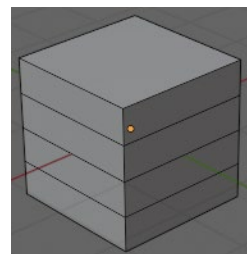


Рис. 43.9

Ще один корисний інструмент **Фаска (Bevel)**, що також викликається комбінацією **Ctrl+B**. З його допомогою утворюються фаски, а саме, згладжуються гострі кути у фігур (рис. 43.10). Для створення фасок лише для кутів використовується комбінація **Ctrl+Shift+B**.

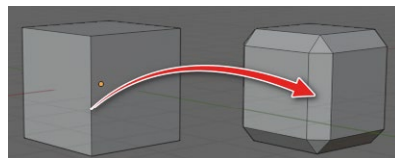



Рис. 43.10



43.4.

Розділення граней

Чим меншою є грань деякого об'єкта, тим більш реалістично він виглядає. Для розділення граней на менші за розміром у Blender є багато різних інструментів. Розглянемо один із найбільш уживаних, а саме, використання модифікатора **Поділ поверхні (Subdivision Surface)**. Для його застосування необхідно:

- в об'єктному режимі виділити об'єкт;
- у вікні праворуч обрати інструмент ;
- відкрити список **Додати модифікатор (Add Modifier)**;
- обрати модифікатор **Поділ поверхні (Subdivision Surface)**;
- у вікні налаштувань модифікатора встановити кількість рівнів поділу у списку **Levels Viewport** (від 1 до 6);
- застосувати модифікатор, обравши команду **Застосувати (Apply)** у верхньому списку даного модифікатора.

Після розділення граней можна додатково ще виконати їхнє згладжування. Для цього слід викликати контекстне меню об'єкта та обрати у ньому команду **Згладжене тонування (Shade Smooth)**.

При розділенні граней слід уважати на те, що чим більший поділ на грані має об'єкт, тим більше пам'яті для його опрацювання вимагає програма. Отже, комп'ютер, обробляючи такі зображення, почне працювати повільніше.

ВПРАВА 43



Завдання. Створити зображення чашки та блюдечка відповідно до зразка (рис. 43.11).



Рис. 43.11



Перегляньте відео створення чашки та блюдечка.

1. Відкрийте файл **Стіл.blend**, створений у вправі 42.1.
2. Приховайте усі об'єкти, що відображаються на сцені. Для цього натисніть піктограму ока навпроти колекції **Стіл** в області відображення колекцій.

Створення блюдечка

3. Створіть колекцію **Блюдечко**.
4. Додайте до сцени об'єкт **Коло**. Для цього в меню **Додати (Add)** оберіть **Сіть (Mesh)** ▶ **Коло (Circle)**.
5. Перейдіть до режиму редагування (клавіша **Tab**).
6. Виконайте екструджування вершин кола до центра (рис. 43.12). Для цього послідовно натисніть **E**, а потім **S** та протягніть курсором до центра кола. Завершіть дію, клацнувши мишею.

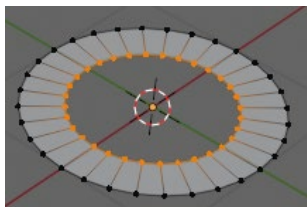


Рис. 43.12

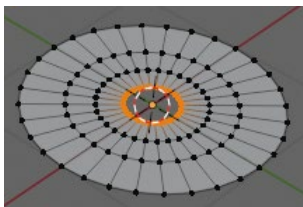


Рис. 43.13

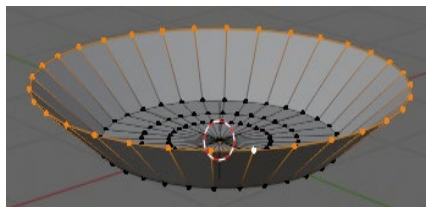



Рис. 43.14

7. Аналогічно виконайте ще такі дві дії екструдування (рис. 43.13).
8. З'єднайте останні утворені вершини до центра. Для цього натисніть **M** та у меню, що з'явиться, оберіть команду **До центра**.
9. Виділіть зовнішні вершини кола. Для цього натисніть **Alt** та оберіть одне із зовнішніх ребер кола.
10. Виконайте екструдування зовнішніх вершин кола та підійміть їх по осі **Z** (рис. 43.14).
 - a. Виконайте екструдування із рівномірним розтягуванням вершин (поспільно натисніть **E** та **S**).
 - b. Підійміть виділені вершини по осі **Z**. Для цього поспільно натисніть **G** та **Z**, перетягніть вершини по осі **Z**. Завершіть дію, натиснувши кнопку миші.
11. Виконайте потовщення стінок блюдечка.
 - a. Перейдіть до об'єктного режиму (клавіша **Tab**).
 - b. У вікні праворуч оберіть інструмент .
 - c. Відкрийте список **Додати модифікатор (Add Modifier)**.
 - d. Оберіть модифікатор **Потовщення (Wireframe)**.
 - e. У вікні налаштувань модифікатора встановіть значення параметра **Товщина** 0.1 m.
 - f. Застосуйте модифікатор, обравши команду **Застосувати (Apply)** у верхньому списку даного модифікатора.
12. Виконайте згладження стінок блюдечка.
 - a. Додайте модифікатор **Поділ поверхні (Subdivision Surface)**.
 - b. У вікні налаштувань модифікатора у списку **Levels Viewport** встановіть значення 2.
 - c. Застосуйте даний модифікатор.
 - d. Виконайте додаткове згладження, обравши в контекстному меню об'єкта команду **Згладжене тонування (Shade Smooth)**.
13. Зробіть видимим колекцію **Стіл** та розташуйте блюдечко на столі. У разі необхідності виконайте його масштабування.

Створення чашки

14. Створіть колекцію **Чашка**.
15. Додайте до сцени об'єкт **Циліндр**. Для цього в меню **Додати (Add)** оберіть **Сіть (Mesh) ▶ Циліндр (Cylinder)**.

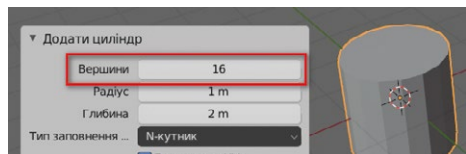


Рис. 43.15

16. Установіть кількість граней 16 у вікні, що з'являється після побудови циліндра (рис.43.15).
17. Перейдіть до режиму редагування.
18. Оберіть режим виділення **Вибір граней**.
19. Виділіть верхню грань.
20. Видаліть виділену грань, натиснувши **Delete**, після чого у вікні, що відкриється, оберіть **Грані**.

21. Створіть фаску на нижній грані (рис. 43.16). Для цього виділіть її, натисніть **Ctrl+B**, протягніть курсор, утворюючи фаску та натисніть кнопку миші.
22. Додайте п'ять ребер по висоті чашки та зробіть її опуклою (рис. 43.17).
 - a. Натисніть **Ctrl+R** та розташуйте курсор на середині чашки.
 - b. Прокрутіть коліщатко миші уперед до одержання п'яти ребер.
 - c. Для підтвердження дії натисніть два рази ліву кнопку миші.
 - d. У вікні налаштувань для параметра **Згладженість** установіть значення 1.
23. Виконайте потовщення стінок чашки, застосувавши в об'єктному режимі модифікатор **Потовщення (Wireframe)** із значенням параметра **Товщина** 0.1 m.

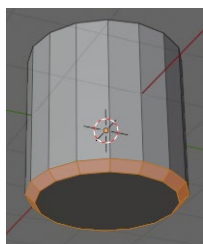


Рис. 43.16

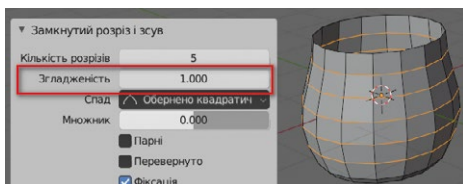


Рис. 43.17

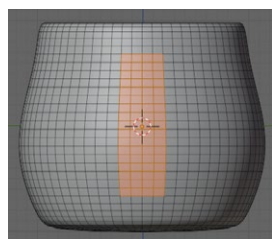


Рис. 43.18

24. Виконайте згладження чашки, застосувавши модифікатор **Поділ поверхні (Subdivision Surface)** із значенням 2 параметра, у списку **Levels Viewport** установіть значення 2.
25. Виконайте додаткове згладження обравши в контекстному меню об'єкта команду **Згладжене тонування (Shade Smooth)**.
26. Створіть у чашці ручку.
 - a. Перейдіть у режим редагування.
 - b. Установіть вигляд сцени **Справа**. Для цього натисніть **X** в області керування осями.
 - c. Оберіть режим виділення граней та виконайте виділення, як на рис. 43.18. Для цього проведіть курсором, утримуючи його над необхідними гранями (для додавання граней до вже виділених утримуйте клавішу **Shift**).

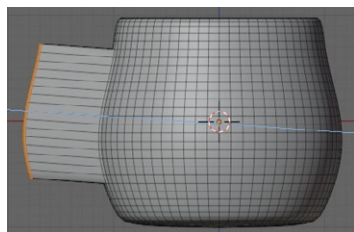


Рис. 43.19

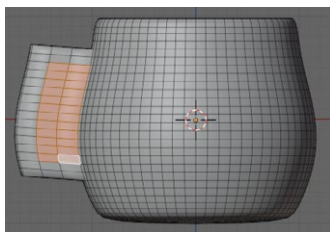


Рис. 43.20

- d. Установіть вигляд сцени **Позаду**. Для цього натисніть **Y** в області керування осями.
 - e. Виконайте екструдування виділеної області, натиснувши **E** (рис. 43.19).
 - f. Виконайте розділення ручки двома площинами. Для цього натисніть **Ctrl+R**. Прокрутіть коліщатко вперед, аби утворилися дві площини, та клацніть після цього два рази мишею.
 - g. Перейдіть у режим виділення граней та виділіть частину, яку необхідно буде видалити у ручки. Виділення виконайте з обох сторін для виду позаду та спереду (рис. 43.20) (при виділенні несуміжних граней утримуйте **Shift**).
 - h. Оберіть меню **Контур** та виконайте команду **З'єднати петлі ребер**. В результаті ручку чашки буде створено.
27. Зробіть видимим колекції **Стіл** та **Блюдечко**. Розташуйте чашку на блюдечку. У разі необхідності виконайте її масштабування.



Контрольні запитання та завдання

1. Що таке екструдування?
2. Чим режим об'єкта відрізняється від режиму редагування?
3. Які елементи можна виділити в режимі редагування?
4. Для чого здійснюється розділення граней?
5. Як створити фаску?



Питання для роздумів

- 1*. Яка дія виконується після натиснення клавіші **M** у режимі редагування?
- 2*. Чому при створенні об'єкта **Коло** утворюються ребра?





Завдання для досліджень

- 1*. Визначте, як у режимі редагування увімкнути можливість виділення усіх елементів.
- 2*. Дослідіть можливості інструментів екструдування для об'єктів куб та сфера.
- 3*. Дослідіть можливості модифікатора **Фаска (Bevel)**. Які є відмінності з однойменним інструментом, що використовується в режимі редагування?

§44.

МАТЕРІАЛИ ТА ТЕКСТУРИ ОБ'ЄКТІВ. РЕНДЕРИНГ ТРИВИМІРНОЇ СЦЕНИ

44.1.

Матеріали об'єктів

Для зміни кольору у середовищі Blender неможливо змінити колір об'єкта, як у звичайному графічному редакторі. В 3D-редакторах для цього використовують матеріали, які, крім кольору об'єкта, відображають також матеріал, із якого він створений. Це може бути пластик, залізо, скло тощо. Отже, крім кольору матеріал може містити такі властивості, як прозорість, відбиття світла, шорсткість тощо.

У Blender один і той же матеріал може бути застосований до безлічі об'єктів. З іншого боку, до одного об'єкта можуть бути застосовані декілька матеріалів.

Створення та налагодження властивостей матеріалів здійснюється на вкладці **Material Properties**. З допомогою кнопки **+** створюють

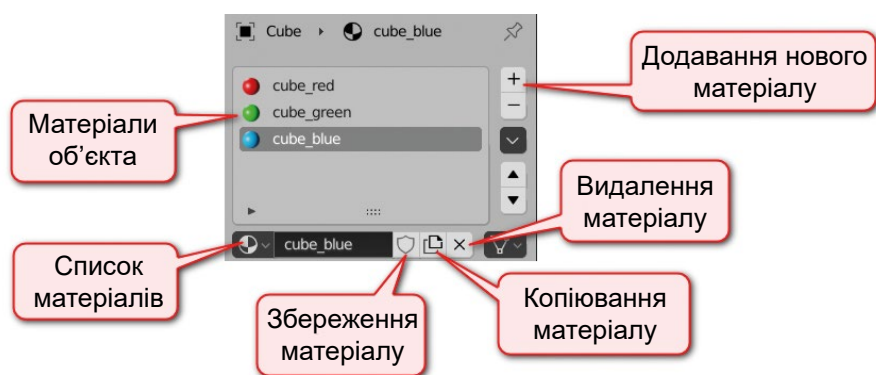


Рис. 44.1



нові матеріали. У нижньому лівому куті розміщено список усіх матеріалів, що використовуються в 3D-зображенні. Кожен об'єкт може мати декілька матеріалів — вони відображені у центральному списку на рис. 44.1. У даному вікні також можна створити копію виділеного матеріалу з допомогою відповідного інструменту. Для перейменування матеріалу достатньо клацнути на його назві два рази. Для видалення матеріалу достатньо натиснути інструмент у вигляді хрестика. А інструмент із зображенням щита зберігає матеріал, якщо він навіть не використовується в жодному об'єкті.

Нижче під даним вікном розміщено панель **Передогляд (Preview)** (рис. 44.2). Що дозволяє показати, як буде виглядати матеріал на різних об'єктах, що розташовані праворуч у даному вікні.

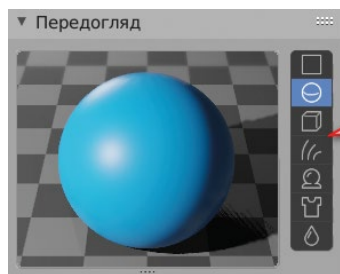


Рис. 44.2

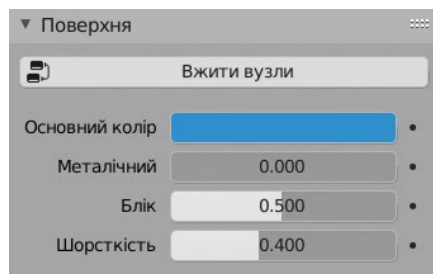


Рис. 44.3

Наступною є панель **Поверхня (Surface)**. Для доступу до спрощених налаштувань слід натиснути кнопку **Вжити вузли (Use Nodes)** (рис. 44.3).

На цій панелі з допомогою властивості **Основний колір (Base Color)** можна встановити колір матеріалу та прозорість. Властивість **Металічний (Specular)** установлює металевий блиск, що дозволяє налаштувати ступінь віддзеркалення поверхні об'єкта. Властивість **Блік (Roughness)** установлює відблиск на об'єкті. Чим значення більше, тим більшим є відблиск, зменшення даного значення надає поверхні матовий вигляд.

Декілька матеріалів до об'єкта створюють та застосовують у режимі редагування. У режимі редагування на панелі матеріалів з'являються кнопки **Присвоїти (Assign)**, **Вибрати (Select)** та **Зняти вибір (Deselect)**. Кнопка **Присвоїти** призначає матеріал виділеним граням об'єкта.

ПРАКТИЧНЕ ЗАВДАННЯ 1



Завдання. Установити різний колір для граней куба.

1. Створіть новий документ, виділіть куб та оберіть режим редагування.



2. Відкрийте вікно **Material Properties**. В результаті у даному вікні буде відображено вже створений за умовчанням матеріал.
3. Перейменуйте матеріал на **cube_red**. Для чого клацніть два рази на назві матеріалу та введіть нову назву.
4. На панелі **Поверхня** натисніть кнопку **Вжити вузли** та оберіть червоний колір матеріалу.
5. Створіть новий матеріал, натиснувши кнопку **+** та введіть ім'я **cube_green**.
6. На панелі **Поверхня** оберіть зелений колір матеріалу.
7. Перейдіть в режим вибору граней та оберіть будь-яку грань куба.
8. Виділіть матеріал **cube_green** та натисніть кнопку **Присвоїти**.
9. Аналогічно розфарбуйте різними кольорами й інші грані куба.

44.2.

Текстури об'єктів

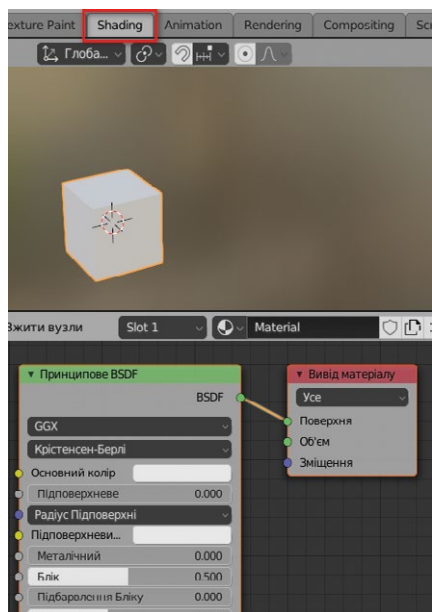


Рис. 44.4

Текстури дозволяють робити матеріали більш реалістичними, схожими на ті, з яких складаються об'єкти з реального світу. У випадку з 3D-об'єктами у Blender текстура накладається на поверхню матеріалу. Отже, не можна використовувати текстуру до об'єкта, що не має матеріалу. Проте з матеріалом може бути пов'язано декілька текстур.

Для налаштування текстур необхідно перейти до редактора **Shading**. Так, після вибору даного редактора для об'єкта куб у нижній частині вікна з'являються два вузли, що відображають параметри налаштувань матеріалу об'єкта (рис. 44.4). Дані вузли мають кольорові точки, через які відбувається з'єднання вузлів для додавання та налаштування текстур.

Розглянемо застосування вбудованої до редактора текстури, виконавши практичне завдання.

ПРАКТИЧНЕ ЗАВДАННЯ 2

1. Додайте шахову текстуру. Для цього в меню заголовка оберіть команду **Додати (Add) ▶ Текстура (Texture) ▶ Шахова текстура (Checker Texture)**. В результаті буде додано вузол шахової текстури.
2. З'єднайте вузол шахової текстури через жовті точки **Колір (Color) ▶ Основний колір (Base Color)**. В результаті куб буде мати шахову текстуру (рис. 44.5).



- Змініть у вузлі **Шахова текстура** другий колір на чорний, а масштаб на 8. Накладання на текстуру зображення із файла розглянемо у вправі 44.

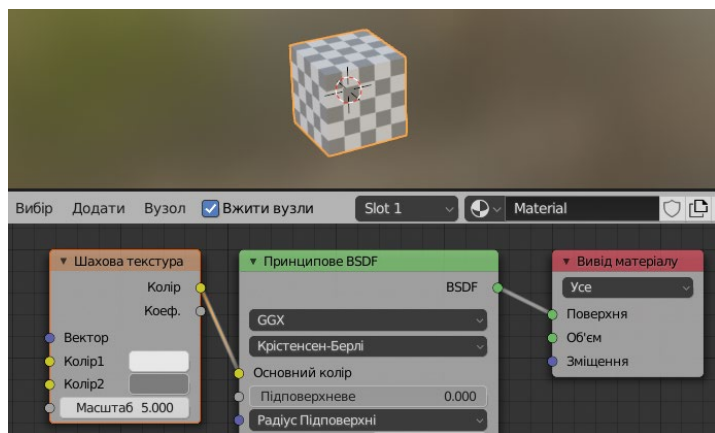



Рис. 44.5.

44.3.

Рендеринг тривимірної сцени

Аби побачити як виглядає кінцеве зображення потрібно виконати його *рендеринг* (візуалізацію). Під час рендерингу будуть відображені частини сцени, що видно з камери. Для рендерингу необхідно натиснути клавішу **F12**. В результаті чого буде відкрито окреме вікно з обраним зображенням.

Якщо зображення не потрапляє у межі видимості камери, то положення камери можна налаштувати на вигляд зображення. Для цього необхідно виділити камеру та натиснути піктограму у вигляді камери  або натиснути клавішу **0** на цифровій клавіатурі. Після чого слід переміщувати камеру після натиснення клавіші **G**.

ВПРАВА 44



Завдання. Змініть матеріал у стола та в чашок відповідно до зразка, поданого на рис. 44.8.

Створення матеріалів для чашки та блюдечка

- Відкрийте файл, створений у вправі 43.
- Створіть три копії чашок і блюдечок та розташуйте їх по периметру стола. Для копіювання натискайте **Alt+D**, аби відбувалося пов'язане копіювання.
- Приховуйте відображення групи чашок.
- Виділіть перше блюдечко.



5. Встановіть вигляд сцени **Зверху**. Для цього натисніть **Z** в області керування осями.
6. Перейдіть у режим редагування.
7. Перейдіть на вкладку **Material Properties**.
8. Створіть новий матеріал, назвіть його **блюдечко**.
9. На панелі **Поверхня** натисніть **Вжити вузли**, оберіть білий колір матеріалу, для властивості **Шорсткість** установіть значення **0**.
10. Масштабуйте перше блюдечко та виділіть у ньому два крайні кільця. Для цього перейдіть у режим виділення граней. Після чого утримуйте **Alt+Shift** та клацайте мишею на необхідні грані.
11. Додайте до матеріалів блюдечка новий матеріал, надайте йому ім'я **кільце**.
12. На панелі **Поверхня** для матеріалу **кільце** оберіть світло-синій колір, для властивості **Шорсткість** установіть значення **0** та натисніть кнопку **Присвоїти**.
13. Перейдіть у режим об'єкта та зробіть видимими зображення чашок.
14. Самостійно виконайте їхнє розфарбування відповідно до зразка (рис. 44.6), використовуючи вже створені матеріали для блюдечка.

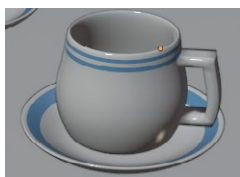


Рис. 44.6

Створення текстури для стола

15. Завантажте файл **wood.jpg** з матеріалів до даного параграфа.
16. Приховайте зображення блюдечок та чашок.
17. Виділіть стільницю стола та перейдіть на вкладку **Shading**.
18. Додайте команду **Додати** ▶ **Текстура** ▶ **Текстура з зображення**.
19. У вузлі **Текстура зображення** натисніть **Відкрити** та оберіть файл **wood.jpg** із файла на диску вашого комп'ютера.

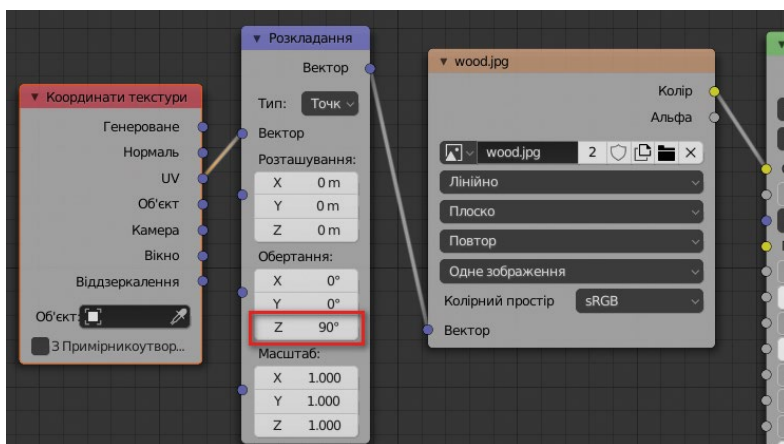


Рис. 44.7



20. З'єднайте точки **Колір** та **Основний колір**. В результаті до стільниці має бути застосовано текстуру дерева.
21. Надайте створеному матеріалу назву **стільниця**.
22. Виділіть ніжку стола та застосуйте до неї матеріал **стільниця**. В результаті матеріал додано, але текстуру дерева, що відображено на ніжці, необхідно повернути.
23. Створіть копію даного матеріалу та надайте йому ім'я **ніжка**.
24. Додайте вузли **Розкладання** (Додати ▶ Вектор ▶ Розкладання) та **Координати текстури** (Додати ▶ Ввід ▶ Координати текстури).
25. З'єднайте точки так, як показано на рис. 44.7 та змініть кут повороту текстури по осі **Z** у вузлі **Розкладання** на **90°**.
26. Виділіть інші ніжки стола та застосуйте до них матеріал **ніжка**.
27. Виконайте налаштування камери та освітлення і виконайте рендеринг отриманого зображення. В результаті ви маєте отримати зображення, подібне до рис. 44.8.



Рис. 44.8



Контрольні запитання та завдання

1. Що таке матеріал об'єкта?
2. Чим матеріал об'єкта відрізняється від текстури?
3. Як створити новий матеріал на основі вже наявного?
4. Чи може об'єкт мати декілька матеріалів, текстур?
- 5*. В яких режимах Blender можна побачити текстуру?
6. Що таке рендеринг і як його виконати?



Питання для роздумів

- 1*. Чому текстура не відображається в режимі об'єкта або редагування?



Завдання для досліджень

- 1*. Дізнайтеся, де можна побачити зміну властивості металевого блиску.





§45.

ТЕКСТОВІ ОБ'ЄКТИ ТА ЇХНЄ РЕДАГУВАННЯ.
ПОНЯТТЯ ПРО 3D-ДРУК

45.1.

Додавання та редагування тексту

Для додавання тексту в Blender необхідно обрати команду **Додати (Add) ▶ Текст**. В результаті у положенні курсору буде додано напис **Text**. Текст додається у вигляді плоскої кривої, що відображено у вікні колекції.

Для редагування тексту необхідно перейти у режим редагування. Видалення тексту здійснюється, як і в звичайних текстових редакторах, з допомогою клавіш **Backspace** або **Delete**. Перехід між символами здійснюється з допомогою клавіш переміщення курсору та різних комбінацій клавіш, а також із допомогою команд вікна **Пересунути курсор**, що з'являється у нижньому лівому куті вікна під час переміщення по символах. Уведення тексту відбувається з поточної позиції курсору. За умовчанням у Blender встановлено вбудований шрифт **Bfont**, отже, текст у ньому вводиться лише латинськими символами. Дії редагування та форматування тексту відбуваються лише з виділеними символами. Для їхнього виділення слід утримувати клавішу **Shift** та натискати відповідну клавішу переміщення курсору. Для копіювання, переміщення та вставлення тексту виконуються ті ж самі дії, що і в більшості текстових редакторів.

Основні дії для роботи з текстом у Blender містяться на вкладці **Текст** (рис. 45.1).

З наведеного меню видно, що воно містить базові команди редагування та форматування шрифту, також є можливість вставки тексту із зовнішнього файлу.

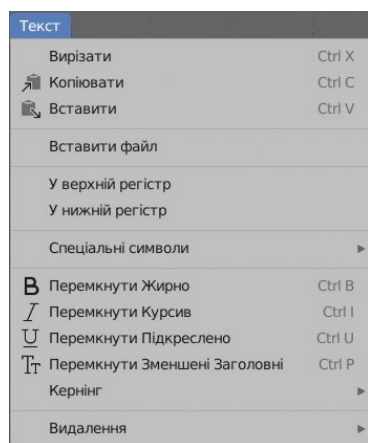


Рис. 45.1


ПРАКТИЧНЕ ЗАВДАННЯ 1

1. Створіть новий файл. Видаліть у ньому об'єкт **Куб**.
2. Додайте до сцени текст. Для цього виконайте команду **Додати ▶ Текст**.
3. Встановіть вигляд сцени **Зверху** та перейдіть у режим редагування.
4. Видаліть наявний там текст та введіть цифри 12345.
5. Виділіть створений рядок цифр, скопіюйте його до буфера обміну та вставте нижче ще чотири рази.
6. У першому рядку видаліть цифру 1, у другому — 2 і т. д.



45.2.

Форматування тексту

Для форматування тексту використовують вкладку **Object Data Properties** .

Так, на панелі **Геометрія (Geometry)**, змінюючи властивість **Витиснути (Extrude)**, тексту можна надати 3D-вигляд, змінюючи його висоту по осі **Z** (рис. 45.2).



Рис. 45.2

Властивості **Глибина (Depth)** та **Роздільність (Resolution)** з панелі **Фаска (Bevel)** надають символам згладженості.

На панелі **Шрифт (Font)** є можливість завантаження бажаного шрифту із встановлених на комп'ютері. Зверніть увагу на те, що тут необхідно завантажити чотири види одного шрифту, що відповідають звичайному, напівжирному, похилому та відразу напівжирному та похилому. Для цього слід обрати інструмент у вигляді папки, після чого у списку, що відкриється, обрати бажаний шрифт.

На панелі **Трансформація (Transform)** з допомогою властивості **Розмір (Size)** змінюють розмір шрифту, а властивість **Косування (Shear)** робить шрифт похилим. **Позиція підкреслення (Underline Position)** та **Товщина підкреслення (Underline Thickness)** встановлюють параметри лінії підкреслення, якщо для шрифту встановлено стиль підкреслений.

Панель **Абзац (Paragraph)** дозволяє виконати вирівнювання тексту та встановити інтервали між символами, словами та рядками.

Для тексту, як і для 3D-об'єктів можна застосувати матеріали. Крім того, текст також можна перетворити на 3D-об'єкт. Для цього текст необхідно виділити в режимі об'єкта та в меню **Об'єкт (Object)** обрати команди **Convert To ► Сіть (Mesh)**.

ВПРАВА 45



Завдання. Створити зображення годинника відповідно до зразка (рис. 45.3).

Створення основи циферблата

1. Створіть новий документ та видаліть у ньому об'єкт **Куб**.
2. Додайте об'єкт **Коло**, встановіть кількість вершин – 12.
3. Перейдіть у режим редагування, вид зверху.
4. Виконайте екструджування виділених вершин до центра три рази (**E** потім **S**).
5. Для з'єднання вершин до центра натисніть **M ► До центра**.
6. Перейдіть до тривимірного режиму та застосуйте до виділеної вершини пропорційне редагування (рис. 45.4). Для цього викличте режим



Рис. 45.3

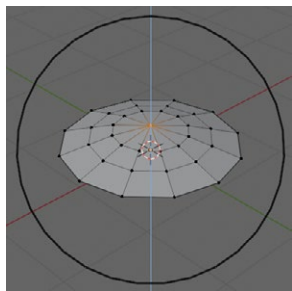


Рис. 45.4

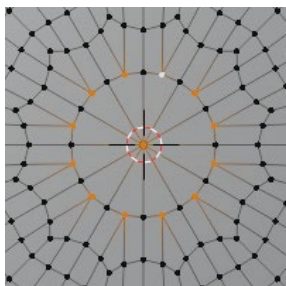


Рис. 45.5



Рис. 45.6

пропорційного редагування (клавіша **O**) та застосуйте переміщення вершини вгору (**G**, а потім **Z**). Коліщатком миші змінюйте радіус пропорційного редагування (чим більший радіус, тим пропорційнішим відбувається редагування).


7. Відмініть режим пропорційного редагування, знову натиснувши **O**.
8. Перейдіть до режиму об'єкту та застосуйте модифікатори **Потовщення** (0,05 m) та **Поділ поверхні** (Level View=2).
9. Також застосуйте згладжене тонування.
10. У режимі редагування перейдіть до виду зверху.
11. Виділіть через одну вершини центральних граней (утримуючи **Ctrl**) та перемістіть їх так, щоб утворилося коло (**G**, а потім **S**) (рис. 45.5).
12. Застосуйте до поверхні циферблата матеріали відповідно до зразка.

Створення стрілок

13. Приховайте зображення циферблата.

14. Додайте об'єкт **Площина**.
15. Перейдіть у режим редагування та змініть місце розташування вершин, аби утворився необхідний чотирикутник (рис. 45.6).
16. Виділіть усі вершини (клавіша **A**) та застосуйте до них екструдкування всередину (**E**, а потім **S**).
17. Підніміть отримані вершини трохи вгору (**G**, а потім **Z**) (рис. 45.6).
18. Застосуйте до поверхні стрілки матеріали відповідно до зразка.
19. Створіть копію стрілки та виконайте її масштабування для утворення меншої стрілки.
20. Зробіть видимим зображення циферблата та розташуйте на ньому стрілки.
21. Додайте до сцени об'єкт **Циліндр**, зменшіть його розміри, розташуйте по центру циферблата, прилаштуйте на ньому стрілки. Надайте циліндру матеріал відповідно до зразка.

Створення чисел на циферблаті

22. Приховайте наявні на сцені об'єкти.
23. Додайте до сцени текст.
24. Перейдіть у режим редагування, видаліть наявний текст та введіть цифру 3.
25. Відкрийте вкладку **Object Data Properties** .
26. На панелі **Геометрія** змініть властивість **Витиснути**, надавши цифри 3D-вигляд.



27. Надайте цифрі згладженість, змінивши властивість **Зсув** на панелі **Фаска**.
28. На панелі **Шрифт** завантажте шрифт comic.ttf, установлений на комп'ютері.
29. Надайте цифрам матеріал бордового кольору.
30. Зробіть видимим циферблат та розташуйте число 3 на циферблаті. У разі необхідності для щільного розташування на циферблаті використовуйте дію обертання (клавіша **R**).
31. Створіть копію даного числа та розташуйте його на місці чисел 6, 9 та 12. В режимі редагування змініть число 3 на необхідні числа.
32. Створіть ще одну копію даного числа та розташуйте її на місці числа 1.
33. Зменшіть утворену копію, змінивши властивість **Розмір** на панелі **Трансформація**.
34. Змініть колір матеріалу одиниці на чорний.
35. Створіть копію цифри 1 та розташуйте її у місця розташування інших чисел.
36. У режимі редагування змініть значення цих чисел.

45.3.

Поняття про 3D-друк

Після створення 3D-моделі у редакторі виникає питання: а що далі можна з цією моделлю робити? Один із варіантів ми вже розглянули — це виконати рендеринг і використовувати 3D-модель як цифрове зображення. А чи можна створену модель перетворити на матеріальний об'єкт? З нещодавніх часів це цілком реально. Адже все більше вдосконалюється ціла індустрія 3D-друку. Спробуємо розібратися за яким принципом працюють принтери, що здійснюють 3D-друк.

Після створення 3D-моделі у редакторі вона зберігається в форматі STL-файла. Після цього 3D-принтер, на який виводиться файл для друку, формує реальний виріб. Сам процес друку — це сукупність циклів нанесення на робочий стіл принтера шару витратних матеріалів з переміщенням робочого столу вниз на рівень готового шару і видаленням з поверхні стола відходів. Цикли безперервно слідують один за одним: на перший шар матеріалу наноситься наступний, елеватор знову опускається і так до тих пір, поки на робочому столі не з'явиться готовий виріб.

Переваги 3D-друку перед звичними, ручними способами побудови моделей — висока швидкість, простота і відносно невелика вартість. Наприклад, для створення 3D-моделі або будь-якої деталі



Рис. 45.7. Приклад роботи 3D принтера



вручну може знадобитися досить багато часу — від кількох днів до місяців. Адже сюди входить не тільки сам процес виготовлення, а й попередні роботи — креслення і схеми майбутнього виробу, які все одно не дають повного бачення остаточного результату. В результаті значно зростають витрати на розробку, збільшується термін від розробки виробу до його серійного виробництва. 3D-технології ж дозволяють повністю виключити ручну працю і необхідність робити креслення і розрахунки на папері, адже програма дозволяє побачити модель у всіх ракурсах уже на екрані й усунути виявлені недоліки не



Ознайомтеся з принципом роботи 3D-принтера у відеоролику.

в процесі створення, як це буває при ручному виготовленні, а безпосередньо при розробці. При цьому можливість помилок, властивих ручній роботі, практично виключається.



Контрольні запитання та завдання

1. Як додати текст до сцени?
2. Як виконати редагування тексту?
3. Які можливості форматування тексту в Blender?
4. Як текст перетворити на 3D-об'єкт?
5. Як відбувається 3D-друк?



Питання для роздумів

- 1*. Як ви думаєте, чи будь-яку тривимірну модель можна надрукувати на 3D-принтері?
- 2*. У яких галузях доцільно використовувати 3D-принтери?



Завдання для досліджень

- 1*. Дізнайтеся, як спрямувати текст у Blender за направляючою лінією.
- 2*. Дізнайтеся, як 3D-модель, створену в Blender, роздрукувати на 3D-принтері.
- 3*. Дізнайтеся, які бувають види 3D-принтерів. Підготуйте доповідь про принципи їхньої роботи.





§46.

АНІМАЦІЯ

46.1.

Знайомство з редактором Timeline

У програмі Blender, окрім створення 3D-зображень, можна також створювати анімації. Для анімації об'єктів можна використовувати як переміщення, обертання та масштабування об'єктів, так і зміну їхньої форми, рух по траєкторії тощо.

Одним із головних понять створення анімації в Blender є *ключовий кадр*. У ключових кадрах створюються зміни, що мають відбуватися між ними. Так, якщо на першому ключовому кадрі об'єкт має розмір 100%, а на наступному ключовому кадрі розмір 50%, то під час відтворення анімації зменшення об'єкта буде здійснюватися плавно, а не різко перескакувати з одного розміру на інший. Звичайно, наскільки це плавно буде відбуватися залежить від часу, який призначений на виконання дії між ключовими кадрами.

Отже, при створенні анімації з'являється ще й четвертий вимір — це час. Для його налаштування в Blender використовується редактор Timeline, що містить шкалу часу. Розміщено даний редактор у нижній частині вікна Blender (рис. 46.1).

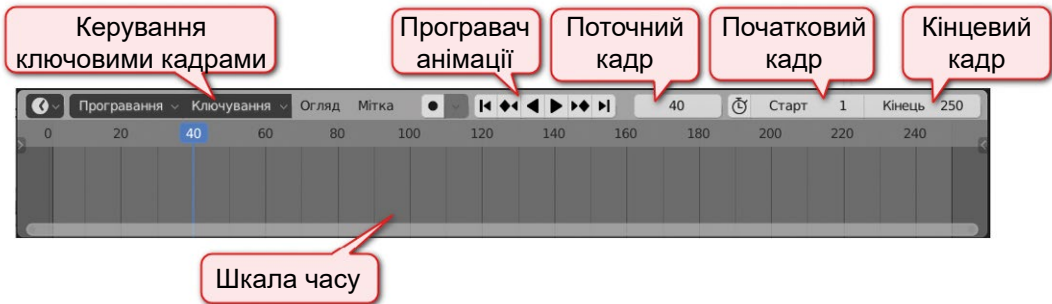


Рис. 46.1. Редактор Timeline

Редактор Timeline дозволяє переміщатися по кадрах, створювати ключові кадри, переглядати анімацію тощо. *Кадр* — це дуже короткий відрізок часу, довжина якого залежить від того, скільки у редакторі налаштовано відображень кадрів у секунду. Звичайно, чим більшим є значення кількості кадрів за секунду, тим коротшим за часом буде відображення кадру. Анімація при цьому буде більш якісною, проте вага вихідного файлу є більшою.



Для переміщення по шкалі часу достатньо перемістити нижню смугу прокручування. Також це можна зробити, утримуючи коліщатко миші. Масштабування шкали часу здійснюється прокручуванням коліщатка миші.

Шкала, у межах якої за умовчанням буде відбуватися анімація, зафарбована більш світлим кольором і за умовчанням це з 1-го по 250-й кадр. Ці значення можна змінити у полях **Старт** та **Кінець** у верхній правій частині редактора Timeline. В цій же області вказано значення поточного кадру, на шкалі поточний кадр вибирається лівою кнопкою миші і підсвічується синім кольором.

Для запуску анімації з поточного кадру достатньо натиснути **Пробіл**. Вона буде виконуватися до останнього кадру і потім розпочнеться виконуватися спочатку. Для завершення анімації можна знову натиснути **Пробіл**. Тоді анімацію буде завершено на поточному кадрі. Якщо зупинити анімацію натисненням **Esc**, то буде здійснено повернення до кадру, з якого було розпочато відображення анімації. Для керування анімацією також можна скористатися кнопками для програвання анімації (рис. 46.1).

Для створення ключового кадру достатньо натиснути **I**, після чого у списку, що з'явиться, обрати тип ключа (переміщення, масштабування, обертання тощо). Також для створення ключа можна скористатися списком **Ключування (Keying)** (рис. 46.2).

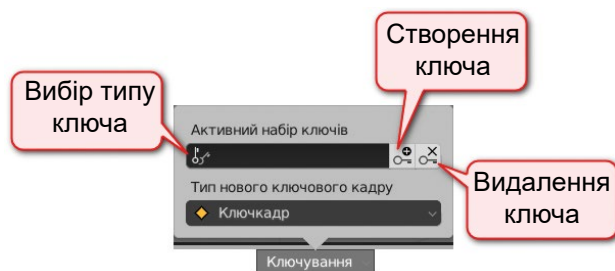


Рис. 46.2. Список **Ключування**

У даному вікні біля значка з ключами необхідно обрати тип ключа: **Location** (переміщення), **Rotation** (обертання), **Scale** (масштабування) тощо. Якщо виконується сукупність різноманітних дій, слід обрати тип ключа **Location, Rotation & Scale**. Після чого слід створити ключ, натиснувши значок зі знаком **+**. В результаті біля створеного ключового кадру на шкалі часу буде відображено жовтий ромб. Видалення ключа здійснюється натисненням значка **x**. Зверніть увагу, що видалити можна лише ключ, коли у списку ліворуч вибрано саме той тип ключ, якого є необхідність видалити.



При створенні ключового кадру фіксується поточна позиція та форма об'єкта. Для створення анімації об'єкта необхідно перейти до наступного кадру, наприклад, до 30-го та виконати деякі дії з об'єктом, після чого знову створити ключ. Тепер при спробі програти анімацію, починаючи з першого ключового кадру, плавно будуть відбуватися дії, що були виконані у другому ключовому кадрі.

Зверніть увагу, що кожний об'єкт має свої ключові кадри. Отже, якщо виділити інший об'єкт, який не змінювався у часі, то на шкалі часу ключі відображені не будуть.

ПРАКТИЧНЕ ЗАВДАННЯ 1



Завдання. Створіть анімацію руху, зміни масштабу та обертання куба.

1. Створіть новий документ.
2. Виділіть об'єкт **Куб**.
3. Відкрийте редактор **Timeline** (збільшіть його розмір, потягнувши курсор миші вгору).
4. Оберіть перший кадр та відкрийте список **Ключування**.
5. Оберіть тип ключа **Location, Rotation & Scale** та натисніть **+** для створення ключа.
6. Оберіть 30-й кадр, підійміть об'єкт **Куб** вгору по осі **Z** (**G**, а потім **Z**).
7. Створіть ключовий кадр (достатньо натиснути **I**).
8. Перейдіть до 60-го кадру, виконайте масштабування куба (клавіша **S**), а потім поворот (клавіша **R**).
9. Створіть ключовий кадр (клавіша **I**).
10. Перейдіть до 90-го кадру, поверніть куб у початкове положення (на вкладці **Object Properties** встановіть значення, як на рис. 46.3.)
11. Створіть ключовий кадр (клавіша **I**).
12. Установіть кінцеве значення анімації 90 у полі **Кінець**.
13. Запустіть анімацію, натиснувши клавішу **Пробіл**.

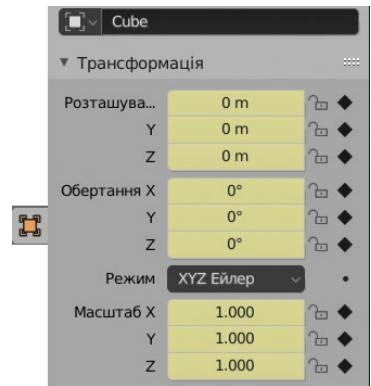



Рис. 46.3

46.2.

Збереження файла анімації

Для відтворення анімації файл зберігають у форматі відео. Перед його рендерингом слід налаштувати параметри зберігання відео. Здійснюється це на вкладці **Output Properties** .

На панелі **Розміри (Dimensions)** можна встановити розміри анімації, а також частоту кадрів. На панелі **Вивід (Output)** доцільно вказати папку для збереження відеофайла, а також обрати тип файла. Частіше всього для збереження відео обирають тип **AVI JPEG**.



Для запуску рендеру слід обрати команду **Рендер (Render) ▶**






Перегляньте відео створення анімації руху стрілок годинника.

Рендер Анімації (Render Animation) (Ctrl+F12). Візуалізація анімації займе певний час. Номер поточного кадру, для якого здійснюється візуалізація, буде відображено у верхньому лівому куті вікна редактора.

ВПРАВА 46.



Завдання. Завдання. Створіть анімацію, що імітує рух стрілок годинника.

1. Відкрийте файл годинника, створений у вправі 45.
2. Виділіть годинникову стрілку та поверніть її у положення 12 годин (**R**, а потім **Shift+Z**).
3. Перейдіть до 1-го кадру редактора **Timeline**.
4. Відкрийте список **Ключування**, оберіть тип ключа **Rotation** та натисніть **+** для створення ключа.
5. Оберіть 240-й кадр.
6. Перейдіть на вкладку на вкладці **Object Properties**  та на панелі **Трансформація** встановіть кут обертання по осі **Z=-360**.
7. Створіть ключовий кадр, натиснувши **I**.
8. Виділіть хвилинну стрілку.
9. Перейдіть до 1-го кадру редактора **Timeline**.
10. Відкрийте список **Ключування**, оберіть тип ключа **Rotation** та натисніть **+** для створення ключа.
11. Оберіть 240-й кадр.
12. Перейдіть на вкладку на вкладці **Object Properties**  та на панелі **Трансформація** встановіть кут обертання по осі **Z=-360*12** (за той самий час, що годинникова стрілка робить 1 оберт, хвилинна виконує 12 обертів).
13. Створіть ключовий кадр, натиснувши **I**.
14. Запустіть анімацію на виконання, натиснувши **Пробіл**. В результаті хвилинна стрілка має робити 12 обертів, поки годинна робить один оберт. Напевно ви помітили, що при повторенні анімації з'являється деяке уповільнення стрілок. Далі приберемо цей ефект.
15. Виділіть одразу обидві стрілки та клацніть на редакторі **Timeline**. У ньому виділіть усі ключові кадри, натиснувши **A**.
16. Викличте контекстне меню редактора **Timeline** та оберіть команду **Режим інтерполяції (Interpolation Mode) ▶ Лінійно (Linear)**.
17. Ще раз запустіть анімацію. Тепер уповільнення повинно бути відсутнім.
18. Збережіть файл анімації у відеофайлі.
 - a. Відкрийте вкладку **Output Properties**  для налаштування збереження файла.
 - b. Оберіть папку у якій необхідно зберегти файл та встановіть тип файла для збереження **AVI JPEG**.
 - c. Виконайте рендеринг анімації (**Ctrl+F12**).



Контрольні запитання та завдання

1. Що таке кадр в анімації?
2. Що таке ключовий кадр в анімації?
3. Яка панель використовується для створення анімації в Blender?
4. Як зберегти анімацію у відеофайлі?



Питання для роздумів

- 1*. Як ви думаєте, що відбудеться, якщо після створення декількох ключових кадрів прибрати проміжний кадр?



Завдання для досліджень

- 1*. Дізнайтеся, як виконати анімацію руху об'єкта за траєкторією.
- 2**. Дізнайтеся, що таке ріггінг в анімації.



Висновки

3D-моделювання — це технологія побудови об'ємної моделі об'єкта в тривимірному просторі на базі професійних графічних програм.

Створення 3D-зображення відбувається за такими етапами: *побудова 3D-моделі; текстурування; освітлення; рендеринг.*

Рендеринг — це одержання зображення за побудованою 3D-моделлю з використанням комп'ютерної програми.

Основні дії, які можна виконувати з 3D-об'єктами в Blender — це *переміщення, обертання та масштабування*. В режимі об'єкта ці дії виконуються над суцільним об'єктом, а в режимі редагування — над його вершинами, ребрами та гранями.

Екструдкування – процес створення нових елементів об'єктів шляхом видавлювання їх із наявних.

Для надання поверхні об'єкта реалістичного виду використовують матеріали та текстури.

Використовуючи технології 3D-друку, можна цифрову модель перетворити на матеріальну.

Одним із головних понять створення анімації в Blender є ключовий кадр. У ключових кадрах створюються зміни, що мають відбуватися між ними.



Зміст



РОЗДІЛ 1

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ІНФОРМАЦІЙНА БЕЗПЕКА

§1.	Класифікація програмного забезпечення	5
§2.	Системне програмне забезпечення	11
§3.	Ліцензії на програмне забезпечення, їхні типи.	18
§4.	Поняття інсталяції та деінсталяції програмного забезпечення. Сумісність програм.	23
§5.	Стиснення та архівування даних.	26
§6.	Практична робота 1. «Архівування та розархівування даних» . .	32
§7.	Інформаційна безпека	34

РОЗДІЛ 2

ОПРАЦЮВАННЯ ТАБЛИЧНИХ ДАНИХ

§8.	Абсолютні, відносні та мішані посилання	47
§9.	Імена клітинок та їхніх діапазонів	51
§10.	Практична робота 2. «Розв'язування задач на обчислення» . .	56
§11.	Математичні функції.	57
§12.	Статистичні функції	63
§13.	Логічні функції.	69
§14.	Умове форматування.	74
§15.	Практична робота 3. «Використання вбудованих функцій. Умове форматування»	78
§16.	Створення діаграм	82
§17.	Вибір типу діаграм	90
§18.	Практична робота 4. «Створення діаграм»	93
§19.	Упорядкування даних у таблицях. Проміжні підсумки	96
§20.	Прості та розширені фільтри	102
§21.	Практична робота 5. «Упорядкування та фільтрація даних у таблицях»	108



§22. Розв'язування задач з математики засобами табличного процесора	109
---	-----

РОЗДІЛ 3

БАЗИ ДАНИХ. СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ

§23. Поняття та призначення баз даних	115
§24. Створення таблиць	121
§25. Створення зв'язків між таблицями	129
§26. Практична робота 6. «Створення таблиць та зв'язків між ними»	136
§27. Сортування та фільтрація даних	137
§28. Запити	143
§29. Практична робота 7. «Створення запитів»	149

РОЗДІЛ 4

АЛГОРИТМИ ТА ПРОГРАМИ

§30. Поняття одновимірного масиву	151
§31. Дії зі списками	159
§32. Практична робота 8. «Дії з масивами»	166
§33. Знаходження підсумкових величин	168
§34. Пошук елементів у масиві	176
§35. Пошук елементів у масиві. Розв'язування прикладних задач	180
§36. Сортування масивів	184
§37. Сортування масивів методом вибору	187
§38. Сортування масивів методом бульбашок. Складність алгоритму	194
§39. Візуалізація даних	201
§40. Двовимірні масиви	209
§41. Практична робота 9. «Дії з двовимірними масивами»	218

РОЗДІЛ 5

3D-ГРАФІКА

§42. Поняття 3D-моделювання	223
§43. Вершини, ребра, грані. Екструдювання форми об'єкта	231
§44. Матеріали та текстури об'єктів. Рендеринг тривимірної сцени	238
§45. Текстові об'єкти та їхнє редагування. Поняття про 3D-друк	244
§46. Анімація	249