

ІНФОРМАТИКА



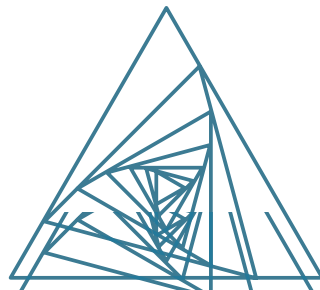
Олена Бондаренко
Василь Ластовецький
Олександр Пилипчук
Євген Шестопапов

ІНФОРМАТИКА

**Підручник для 5 класу
ЗАКЛАДІВ ЗАГАЛЬНОЇ
СЕРЕДНЬОЇ ОСВІТИ**

**Рекомендовано
Міністерством освіти і науки
України**

Харків
Видавництво «Ранок»



ЗМІСТ

Передмова.....	5
Розділ 1. ІНФОРМАЦІЙНІ ПРОЦЕСИ ТА СИСТЕМИ	
§ 1. Інформація та повідомлення. Правила роботи за комп'ютером	9
§ 2. Інформаційні процеси. Інформаційні технології.....	14
§ 3. Використання комп'ютерів	22
§ 4. Комп'ютер як інформаційна система. Складові комп'ютера.....	25
§ 5. Операційна система та її інтерфейс.....	30
§ 6. Операції над файлами та папками.....	34
Практична робота 1. Виконання операцій над файлами та папками.....	39
Розділ 2. МЕРЕЖЕВІ ТЕХНОЛОГІЇ ТА ІНТЕРНЕТ	
§ 7. Комп'ютерні мережі. Локальна комп'ютерна мережа.....	43
§ 8. Глобальна комп'ютерна мережа	47
§ 9. Безпечне використання інтернету. Спілкування в інтернеті.....	53
§ 10. Пошук відомостей в інтернеті. Авторське право.....	58
Практична робота 2. Пошук інформації в інтернеті	64
Розділ 3. КОМП'ЮТЕРНІ ПРЕЗЕНТАЦІЇ	
§ 11. Створення та опрацювання комп'ютерних презентацій.....	67
§ 12. Оформлення комп'ютерних презентацій	74
§ 13. Виступи з показом презентації.....	83
Практична робота 3. Проектування та розробка власної презентації	92
Розділ 4. ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ	
§ 14. Основні об'єкти текстового документа	95*
§ 15. Робота з фрагментами тексту. Редагування та форматування символів.....	100
Практична робота 4. Здійснення форматування символів текстового документа	106
§ 16. Форматування абзаців	108
Практична робота 5. Здійснення форматування абзаців текстового документа	113

* Починаючи зі с. 97 нумерацію сторінок вказано за авторським оригіналом.

Практична робота 6. Створення текстового документа	114
§ 17. Опрацювання об'єктів текстового документа.....	117
Розділ 5. АЛГОРИТМИ ТА ПРОГРАМИ	
§ 18. Алгоритм і його властивості	123
§ 19. Виконавець алгоритмів і система його команд.....	128
§ 20. Способи опису алгоритму. Алгоритмічні структури	132
§ 21. Середовище опису та виконання алгоритмів	139
§ 22. Основні поняття мови програмування Python	144
§ 23. Лінійні алгоритми та програми	149
§ 24. Математичні оператори мови Python.....	153
Практична робота 7. Складання та виконання лінійних алгоритмів.....	157
§ 25. «Черепашача» графіка.....	158
§ 26. Алгоритми створення зображень	163
Практична робота 8. Створення зображень за алгоритмами	168
§ 27. Логічні вирази.....	168
§ 28. Алгоритми і програми з розгалуженнями. Оператор if ...	173
§ 29. Алгоритми і програми з розгалуженнями. Оператор if ... else.....	176
Практична робота 9. Складання та виконання алгоритмів із розгалуженнями	179
§ 30. Алгоритми з повтореннями. Цикл із лічильником	181
§ 31. Алгоритми і програми з циклами з лічильником.....	186
Практична робота 10. Складання та виконання алгоритмів із повтореннями	190
Розділ 6. ПРАКТИКУМ З ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	
Практична робота 11. Створення словника комп'ютерних термінів	194
Практична робота 12. Створення презентації «Передавання інформації за допомогою графічних символів».....	195
Практична робота 13. Створення зображень засобами черепашачої графіки.....	197
Практична робота 14. Складання та виконання алгоритмів із розгалуженнями та повтореннями	199
Практична робота 15. Складання та виконання алгоритмів із повтореннями	201
Комп'ютерний словник	203
Алфавітний покажчик	206

ДОРОГІ П'ЯТИКЛАСНИКИ ТА П'ЯТИКЛАСНИЦІ!

Ви тримаєте в руках підручник з інформатики, призначений саме для вас, учнів і учениць 5 класу. У початковій школі ви дізналися про базові поняття інформатики, навчилися працювати з комп'ютером, створювати графічні й текстові документи, шукати в інтернеті інформаційні матеріали, маєте поняття про алгоритми та їх виконавців.

У цьому навчальному році на уроках інформатики вас очікує чимало цікавого та корисного, а пропонований підручник буде вашим надійним помічником. Як же з ним працювати?

Розділ 1 «Інформаційні процеси та системи» присвячений значною мірою повторенню та закріпленню матеріалу, вивченого в початковій школі, щоб ви змогли успішно засвоїти подальші розділи, розділ 2 «Мережеві технології та інтернет» присвячений пошуку інформації в інтернеті, розділ 3 «Комп'ютерні презентації» — створенню й оформленню цікавих та змістовних презентацій, розділ 4 «Опрацювання текстових даних» — створенню текстових документів із зображеннями й таблицями, а розділ 5 «Алгоритми та програми» — ознайомленню з алгоритмами й основними поняттями мови програмування Python. Розділ 6 пропонує практикум з використання інформаційних технологій.

Розділ підручника складається з параграфів і практичних робіт за темами розділу. Кожен параграф містить теоретичні відомості за темою уроку, приклади використання отриманих знань, питання для самоперевірки, вправу, комп'ютерний тест.

«Питання для самоперевірки» допоможуть з'ясувати, чи зрозуміли ви вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт за комп'ютером. «Вправа» складається з теоретичних і практичних завдань. Виконуючи ці завдання, ви навчитеся краще працювати з комп'ютером. Оцінити свої знання, уміння та навички вам допоможе «Комп'ютерне тестування» з автоматичною перевіркою відповідей. Його можна пройти на інтерактивній освітній платформі.

У підручнику ви знайдете інструкції до практичних робіт. Щоб виконати кожну з них, потрібно повторити матеріал, вивчений на попередніх уроках, — тоді ви зможете успішно застосувати свої знання, виконуючи завдання за комп'ютером.

Бажаємо натхнення та успіхів!

Авторський колектив

У тексті підручника використано такі позначення:



Запам'ятайте



Розгляньте приклад



Ознайомтеся з цікавою інформацією



Зверніть увагу



Знайдіть відповідь у підручнику



Знайдіть відповідь в інтернеті



Виконайте практичне завдання за комп'ютером



Виконайте тестові завдання за комп'ютером



Виконайте завдання підвищеної складності



Виконайте завдання в парах



 Інтернет-підтримка

За QR-кодом
або посиланням
rnk.com.ua/101524

РОЗДІЛ 1

ІНФОРМАЦІЙНІ ПРОЦЕСИ ТА СИСТЕМИ



- § 1. Інформація та повідомлення. Правила роботи за комп'ютером
- § 2. Інформаційні процеси. Інформаційні технології
- § 3. Використання комп'ютерів
- § 4. Комп'ютер як інформаційна система. Складові комп'ютера
- § 5. Операційна система та її інтерфейс
- § 6. Операції над файлами та папками

Практична робота 1. Виконання операцій над файлами та папками

ПОВТОРЮЄМО



З уроків інформатики ви вже знаєте, що відомості про навколишній світ і процеси, які в ньому відбуваються, називають *інформацією*. Людина отримує інформацію у вигляді повідомлень, які сприймає *органами чуття*. За *способом сприйняття* повідомлення поділяють на візуальні, смакові, нюхові, аудіальні й тактильні; за *способом подання* — на текстові, графічні, звукові, мультимедійні.

Дії, які можна виконувати з інформацією: пошук, опрацювання, передавання, зберігання тощо називають *інформаційними процесами*.

Інформацію опрацьовують за допомогою універсального пристрою — *комп'ютера*. Існують найрізноманітніші комп'ютери: від кишенькових до суперкомп'ютерів.

1. Що таке інформація?
2. Як людина сприймає інформацію?
3. Які бувають повідомлення за способом їх сприйняття людиною?
4. У який спосіб можна подати інформацію?
5. Наведіть приклади передавання інформації.
6. Де використовують комп'ютери?



У цьому розділі ви більше дізнаєтеся про будову та роботу комп'ютера; навчитесь безпечно та грамотно працювати з операційною системою — комплексом програм для керування комп'ютером.

§ 1. Інформація та повідомлення. Правила роботи за комп'ютером

Упродовж життя людина сприймає безліч повідомлень, опрацьовує інформацію, зберігає отримані знання або передає їх. Як це відбувається, вивчає особлива наука — інформатика.



Інформатика — це наука про властивості інформації та методи її створення й опрацювання з використанням комп'ютерів.

Що таке інформація

Основні поняття інформатики — *інформація та повідомлення*.

Всесвіт, що нас оточує і частиною якого є ми самі, складається з об'єктів (предметів, процесів і явищ). Та поряд із матеріальною складовою Всесвіту існує й нематеріальна його складова — її називають інформацією (рис. 1.1). Нині інформація є важливим ресурсом і набуває дедалі більшого значення в житті суспільства. Але що ж таке інформація?



Рис. 1.1



Інформація — це відомості про об'єкти Всесвіту.

У різних сферах людської діяльності застосовують також інші означення цього терміна. Так, у побуті інформацією часто вважають лише ті відомості, які для людини є новими.

У Законі України «Про інформацію» сказано, що інформація — це будь-які відомості чи дані, які можна зберегти на матеріальних носіях або відобразити в електронному вигляді.

Люди отримують і передають інформацію за допомогою повідомлень.



Повідомлення — це форма подання інформації.

Без повідомлень не можна сприйняти інформацію.

Одну й ту саму інформацію можна передати різними повідомленнями, а одне й те саме повідомлення може нести різну інформацію.

Людина отримує повідомлення за допомогою органів чуття: зору, слуху, нюху, смаку, дотику (рис. 1.2).



Рис. 1.2

Пригадаємо, як розрізняють повідомлення та інформацію за способом сприйняття та способом подання (рис. 1.3).

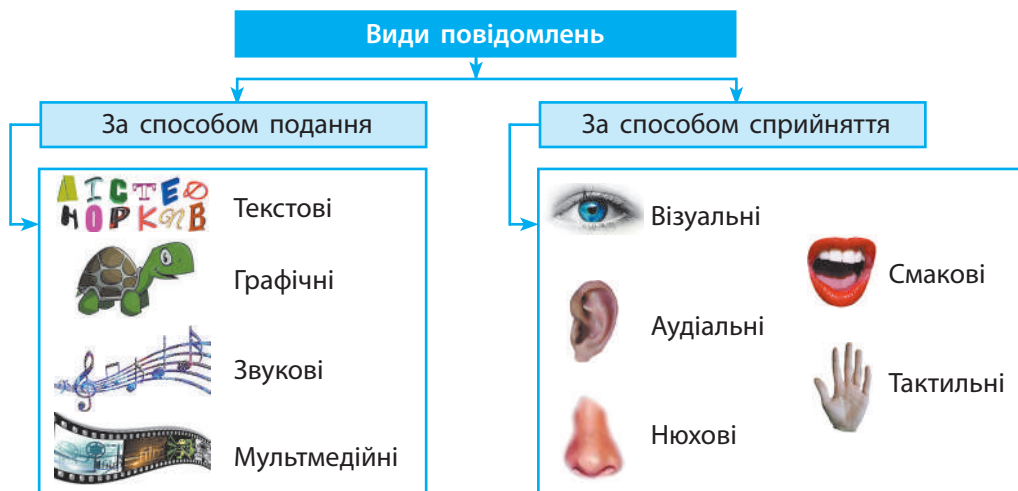


Рис. 1.3

Людина має й інші органи чуття. Наприклад, завдяки вестибулярному апарату ви навіть із заплющеними очима легко визначаєте положення свого тіла в просторі (нахили, повороти) (рис. 1.4). Подібними «органами» конструктори обладнують сучасних роботів, яких вони навчили ходити на двох ногах, стрибати і навіть робити сальто.



Рис. 1.4

Чи знаєте ви, що технічні пристрої отримують повідомлення за допомогою різних датчиків і сенсорів?

1 Світильник із *датчиком руху* (рис. 1.5, а) допомагає економити електроенергію, адже він засвічується лише тоді, коли в межах дії датчика руху з'являється рухомий об'єкт (людина, автомобіль тощо). А *датчик диму* (рис. 1.5, б), установлений у приміщенні і приєднаний до системи сигналізації, забезпечує від пожежі.

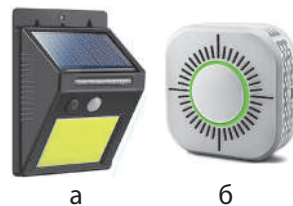


Рис. 1.5

Деякі тварини мають органи чуття для сприйняття інформації, недоступної для людини.

2 Змії сприймають теплове (інфрачервоне) випромінювання. Кити можуть сприймати наднизькі звуки (інфразвуки), а кажани — надвисокі (ультразвуки), яких людина не чує.

Класифікація інформації

Інформацію можна класифікувати за різними ознаками. Розглянемо таблицю:

Ознаки	Види інформації	Приклади
За способом сприйняття	Візуальна, аудіальна, нюхова, смакова, тактильна	Інформація, яку ви сприймаєте, слухаючи вчителя/вчительку, є аудіальною (звуковою)
За формою подання	Текстова, числова, відео, звукова, графічна, комбінована	Інформація, подана у підручнику, є текстовою і графічною
За призначенням	Масова, спеціальна, особиста, розважальна тощо	Дивлячись новини по телебаченню, ви сприймаєте масову інформацію
За корисністю	Корисна, непотрібна, шкідлива	Для вибору, що вдягти, інформація про погоду є корисною

Про правила поведінки в комп'ютерному класі

У комп'ютерному класі потрібно особливо уважно дотримуватися правил безпечної поведінки. Кімната, в якій встановлено багато комп'ютерів, є приміщенням підвищеної небезпеки. Джерелом небезпеки тут може бути розетка з розбитим корпусом, дріт із пошкодженою ізоляцією тощо. Звичайно, ми змалювали неприйнятну ситуацію. Помітивши щось із переліченого, слід обов'язково повідомити старших.

Будь-які правила безпеки та норми поведінки в комп'ютерному класі (рис. 1.6) базуються на таких принципах: не нашкодьте собі; не нашкодьте іншим; не заважайте іншим; не псуйте обладнання.

Комп'ютер сконструйовано так, щоб не допустити випадкового ураження людини електричним струмом. Разом з тим безпечну роботу на комп'ютері гарантує й дисциплінованість усіх користувачів і користувачок (посилання за QR-кодом).



Рис. 1.6

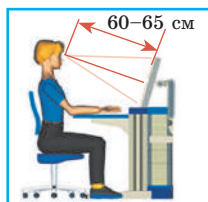


► Правила роботи за комп'ютером

Працюючи за комп'ютером, слід стежити за правильною організацією робочого місця, в перервах робити вправи (див. форзац).

Час неперервної роботи за комп'ютером — не більше 20 хв. Потім варто відпочити або виконати вправи для очей (див. форзац).

Розгляньмо рис. 1.7: спина повинна мати опору, лінія погляду — бути перпендикулярною до екрана, відстань від очей до екрана монітора — на довжину руки.



ПРАВИЛЬНО



НЕПРАВИЛЬНО

Рис. 1.7



Питання для самоперевірки

1. Поясніть, що таке інформація, повідомлення.
2. Як розрізняють інформацію за різними ознаками?
3. Чого не можна робити в комп'ютерному класі?
4. Як правильно сидіти за комп'ютером?
5. Який безпечний час роботи за комп'ютером для дітей?
6. Обговоріть правила безпеки та можливі наслідки з батьками чи старшими.



Вправа 1

- Ознайомитися з робочим місцем, пригадати основи роботи за комп'ютером з дотриманням правил безпеки.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.



- 1) Запустіть клавіатурний тренажер і потренуйтеся набирати пропонований текст українською мовою. Повідомте результат учителю / учительці.
- 2) Запустіть графічний редактор (наприклад, Paint).
- 3) Намалюйте комп'ютер, який стоїть на вашому робочому місці (приклад — на рис. 1.8).
- 4) На малюнку позначте місця, що можуть бути джерелом небезпеки.

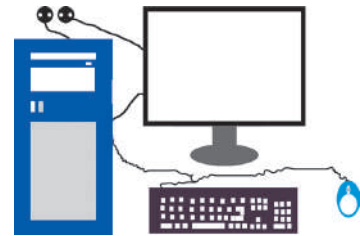


Рис. 1.8



- 5) Намалюйте на екрані «монітора» комп'ютера пейзаж.
- 6) Покажіть малюнок учителю / учительці. Збережіть файл з назвою Вправа 1. Завершіть роботу з графічним редактором.



Комп'ютерне тестування

Виконайте тестове завдання 1 з автоматичною перевіркою результату.



§ 2. Інформаційні процеси. Інформаційні технології

Об'єкти навколишнього світу надзвичайно різноманітні. Кожен об'єкт має певні властивості й може виконувати (або над ним можуть виконуватися) певні дії.

Інформаційні процеси

З точки зору людини інформація є об'єктом, а дії над нею називають інформаційними процесами.



Інформаційні процеси — це дії над інформацією.

До основних інформаційних процесів належить передавання повідомлень каналами зв'язку; зберігання інформації у вигляді повідомлень на носіях інформації; пошук повідомлень із потрібною інформацією; опрацювання інформації для отримання нової інформації або змінення форми її подання.



1 Коли вчитель/вчителька пояснює новий матеріал, то передає інформацію, а ви її отримуєте. Коли ви записуєте щось у зошит — зберігаєте інформацію, а розв'язуючи задачу — опрацюєте (рис. 2.1).



Рис. 2.1

Передавання інформації багатьом приймачам називають її поширенням. Захист інформації полягає або в обмеженні її передавання, або в особливому кодуванні — шифруванні. *Збирання* та *зберігання* великої кількості інформації називають накопиченням.



2 13 січня 2022 року виведено на навколосеземну орбіту український супутник «Січ-2-30». Він збиратиме інформацію про нашу планету, фотографуючи її у видимих і інфрачервоних променях, і передаватиме вченим, які її опрацюватимуть.

Опрацювання інформації полягає у створенні нової для людини інформації завдяки перетворенню початкової інформації.



Наприклад, дівчина складає вірш, перебуваючи під враженням від дивовижної картини карпатської полонини (рис. 2.2).



Рис. 2.2

Існує багато інших інформаційних процесів. Інформацію можна використовувати, оцінювати, ототожнювати, тлумачити, зіставляти, обмірковувати тощо. З інформацією та інформаційними процесами пов'язане не лише життя людей. Безліч інформаційних процесів відбувається в житті тварин, рослин, у неживій природі.

Дані

Щоб повідомлення було придатним для здійснення певного інформаційного процесу, його необхідно подати певним чином. Наприклад, у вигляді звуків, зображень, слів, електричних, світлових та інших сигналів.



Дані — це повідомлення, в яких однотипну інформацію подано у формі, зручній для подальшого опрацювання, зазвичай за допомогою технічних засобів.

Повідомлення, подане у формі, з якою може працювати пристрій, є для нього даними. Дані зберігають, записуючи їх на носії даних (папір, карта пам'яті тощо) (рис. 2.3).



Рис. 2.3

- 4 Дані у вигляді наскельного малюнка в печері можуть зберігатися тисячоліттями. В Україні поблизу Мелітополя є унікальне місце — Кам'яна Могила. На її каменях із прадавніх часів залишилося понад 3 тис. зображень людей, диких і свійських тварин, різноманітних знаків — петрогліфів (рис. 2.4, а). Текстові дані на паперових носіях (рис. 2.4, б) значно легше передавати, хоча зберігання даних на них не таке тривале.



а



б

Рис. 2.4

Завдяки надзвичайно швидкому копіюванню та розповсюдженню інформації сучасними засобами її можна вважати практично незнищеною.

- 5 Тривають розробки щодо збільшення довговічності носіїв даних. 2013 року вчені Інституту проблем реєстрації інформації Національної академії наук України розробили оптичний сапфіровий диск, який може зберігати дані впродовж десятків тисяч років (рис. 2.5).



Рис. 2.5

Інформаційні системи

Слово «система» з грецької — «ціле, що складається із частин».

- Інформаційна система** — це сукупність взаємопов'язаних об'єктів, у якій відбуваються інформаційні процеси.

Об'єкти живої та неживої природи, які складають інформаційну систему, називають *засобами інформаційної системи*. Це можуть бути біологічні клітини, люди, вода, механічні й електронні пристрої тощо.

Способи дії (або взаємодії) цих об'єктів називають *методами інформаційної системи*. Прикладами методів є записування тексту в зошит, танці бджіл задля поширення інформації серед бджолиного рою (рис. 2.6), усне спілкування людей тощо.

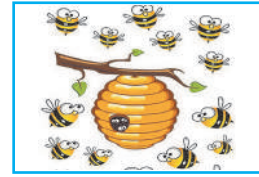


Рис. 2.6



На рис. 2.7 зображено інформаційну систему — учнівський клас на уроці малювання. Засобами системи є вчитель, учні й учениці, приміщення, дошка, крейда (маркери), олівці, аркуші паперу та ін. Методами системи є усне пояснення вчителя / вчительки, демонстрування та обговорення репродукцій картин, відтворення дітьми малюнків олівцями на папері тощо.



Рис. 2.7

Інформаційні системи поділяють на технічні (наприклад, робот), біологічні (людина), соціальні (група людей) та ін.

Роль інформаційних технологій у житті сучасної людини

Інформатика є теоретичною базою для розвитку інформаційних технологій. Слово «технологія» з грецької перекладається як «майстерність», «уміння». На відміну від технології матеріального виробництва, інформаційна зорієнтована на роботу з інформацією.



Інформаційна технологія — це сукупність методів і процесів опрацювання інформації із застосуванням відповідних засобів.

Як відомо, ще перші комп'ютери повністю виправдали свою назву (від англ. *computer* — обчислювач). Адже саме для автоматичного виконання дуже складних обчислень їх і створювали.

- Основоположником обчислювальної техніки в колишньому СРСР був академік Сергій Олексійович Лебедев (рис. 2.8). Під його керівництвом у Київському електротехнічному інституті у 1951 році було створено перший у континентальній Європі комп'ютер «МЕСМ» (рос. *МЭСМ — Малая Электронная Счетная Машина*).



Рис. 2.8

Стрімкий розвиток інформаційних технологій став можливим завдяки поєднанню досягнень у таких галузях, як інформатика, математика, фізика, медицина, філософія, економіка та ін.

Розглянемо кілька пристроїв для опрацювання даних, які набули найбільшого поширення.

Пристрій	Призначення
<p>Комп'ютер</p> 	Здійснення майже всіх інформаційних процесів
<p>Програваач (плеєр)</p> 	Відтворення аудіо- або відеозаписів
<p>Калькулятор</p> 	Виконання операцій над числами
<p>Фото- та відеокамера</p> 	Створення нерухомих або рухомих зображень
<p>Навігатор GPS</p> 	Визначення поточного місця розташування пристрою на Землі



7 Завдяки розвитку сучасних інформаційних технологій світ став значно «прозорішим» і «тіснішим». Так, за кілька секунд ми можемо зв'язатися телефоном з абонентом на іншій півкулі Землі (рис. 2.9); отримати електронного листа, поспілкуватися в режимі відео з будь-якою особою; пограти в комп'ютерну гру, «помандрувати» музеями, дізнатися про новини, прогноз погоди тощо.



Рис. 2.9

Нині комп'ютери постійно вдосконалюються, розширюється сфера їх застосування. Набувають поширення нові види комп'ютерів: роботизовані іграшки, роботи-помічники по господарству тощо (рис. 2.10). Комп'ютери дозволили створити такі цікаві й популярні технології, як «розумний дім» (англ. *smart home*). Домашня автоматика «розумного дому» здатна діяти замість людини, спростити побутові задачі людям похилого віку, людям з інвалідністю (рис. 2.11).



Рис. 2.10



Рис. 2.11

► Мережеві технології

Винайдення комп'ютерних мереж, зокрема інтернету, значно розширило можливості інформаційних технологій. Віднедавна зросла популярність дистанційного навчання. Його основою є інформаційно-комунікаційні технології, які надають можливість учнівській і вчительській спільнотам взаємодіяти на відстані.

Нові можливості для людини відкриває технологія «інтернет речей» (англ. *Internet of Things*, *IoT*) — сукупність з'єднаних з інтернетом пристроїв — «речей», оснащених засобами приймання й передавання інформації. Він здатний прискорити прогрес суспільства, оскільки створить умови для здійснення багатьох процесів без участі людини (рис. 2.12).



Рис. 2.12

8 Завдяки інтернету дорожні знаки, що обмежують швидкість руху, змінюють свій вигляд, зважаючи на погоду; безпілотні автомобілі пересуваються маршрутом, реагуючи на дорожню ситуацію; камери відеоспостереження (рис. 2.13) сповіщають через мобільні пристрої про проникнення; наручні годинники виконують функції крокоміра, тонометра для вимірювання кров'яного тиску, дистанційного пульта керування телевізором (рис. 2.14) тощо.



Рис. 2.13



Рис. 2.14

Це лише окремі приклади сучасних інформаційних технологій, розвиток яких не зупиняється ні на мить.

? Питання для самоперевірки

1. Що таке дані?
2. Назвіть основні інформаційні процеси.
3. Наведіть приклади носіїв для зберігання даних.

4. Якими засобами передають інформацію?
5. Що таке інформаційна технологія?
6. Опишіть роль інформаційних технологій у житті людини. Наведіть приклади.



Вправа 2

■ Ознайомитися з деякими пристроями опрацювання даних. Під час роботи за комп'ютером дотримуйтесь правил безпеки.



1) Запустіть клавіатурний тренажер і потренуйтеся набирати пропонований текст українською та іноземною мовами. Повідомте результат учителю/вчительці.



2) Запустіть браузер і знайдіть в інтернеті зображення комп'ютера та двох пристроїв опрацювання даних.

3) Запустіть графічний редактор (наприклад, Paint).

4) Скопіюйте і вставте знайдені зображення за зразком на рис. 2.15. Підпишіть пристрої.



Рис. 2.15



5) Зробіть колаж із трьох отриманих зображень.

6) Покажіть роботу вчителю/вчительці. Збережіть файл із назвою Вправа 2. Завершіть роботу з графічним редактором.



Комп'ютерне тестування

Виконайте тестове завдання 2 з автоматичною перевіркою результату.



§ 3. Використання комп'ютерів



Як відомо, комп'ютер — універсальний пристрій для опрацювання даних. І хоч дослівно його назва означає «обчислювач», зараз він став справді універсальним, тобто придатним для опрацювання, зберігання й передавання різноманітних даних: числових, графічних, текстових, звукових тощо.

Використання комп'ютерів

Комп'ютери використовують у різних галузях людської діяльності. Так, за їх допомогою можна прогнозувати погоду, керувати автомобілем (автопілот), проводити медичні дослідження (комп'ютерний томограф), навчатися дистанційно в школі або університеті, продавати залізничні й авіаквитки, створювати відео й анімаційні фільми, розробляти ігрові та навчальні програми.

Види комп'ютерів

Комп'ютери поділяють за різними ознаками. Якщо розглядати комп'ютери, призначені для персонального користування — персональні комп'ютери, то їх можна класифікувати, наприклад, таким чином:

Стационарні комп'ютери	
Настільний комп'ютер (десктоп) 	Складається із системного блоку, до якого приєднано монітор, клавіатуру й мишу
Моноблок 	Системний блок і монітор складають єдине ціле

Портативні комп'ютери	
Ноутбук та нетбук	 У корпусі об'єднано дисплей, клавіатуру, тачпад і акумуляторні батареї
Планшетний ноутбук та планшет	 Корпус містить сенсорний екран, є клавіатура, яка або складається, або висувається з ніші під екраном. У планшетів пристроєм уведення є сенсорний екран
Кишеньковий ПК та смартфон	 Мають малі розміри та великий запас часу автономної роботи від акумулятора. Пристрій уведення — сенсорний екран або висувна клавіатура
Мікрокомп'ютер	 Розміщується в мініатюрному корпусі, не має пристрою виведення, тому приєднується до монітора. Миша або клавіатура під'єднується через убудовані USB-порти або Bluetooth

Із розвитком комп'ютерної індустрії пов'язане поняття суперкомп'ютера. Це обчислювальна машина, яка за своїми технічними характеристиками значно перевершує решту комп'ютерів.

Суперкомп'ютери застосовують для проведення широкомасштабних наукових досліджень, передбачення погоди, стихійних явищ та ін.

Японський суперкомп'ютер Фуґафу в Центрі обчислювальної науки RIKEN у місті Кобе виконує понад 500 квадрильйонів операцій на секунду. Для виконання таких розрахунків калькулятором необхідно було б витратити 100 млрд років!

В Україні створено потужні центри суперкомп'ютерних обчислень у Національному технічному університеті «Київський політехнічний університет» та Інституті кібернетики ім. В. М. Глушкова НАН України (див. рисунок).





Питання для самоперевірки

1. Наведіть приклади використання комп'ютерів.
2. Яке призначення персонального комп'ютера?
3. Що таке моноблок?
4. Які характеристики має смартфон?
5. Назвіть види комп'ютерів.
6. Для чого створюють суперкомп'ютери?



Вправа 3

- Розглянути характеристики основних елементів апаратної складової комп'ютера

Під час роботи за комп'ютером дотримуйтесь правил безпеки.



- 1) Запустіть клавіатурний тренажер і потренуйтеся набирати текст, який складається з букв і спеціальних символів. Повідомте результат учителю/вчительці.



- 2) Запустіть браузер. Знайдіть в інтернеті технічні характеристики однієї з моделей стаціонарного комп'ютера.
- 3) Запустіть текстовий процесор, створіть документ. Запишіть характеристики процесора (назва, кількість ядер).



- 4) Запишіть характеристики:
 - а) жорсткого диску (обсяг пам'яті);
 - б) оперативної пам'яті (обсяг пам'яті).



- 5) Знайдіть в інтернеті характеристики кількох моделей комп'ютерів. Скопіюйте в документ характеристики комп'ютера, що є, на вашу думку, найбільш придатним для використання учнями та ученицями 5 класу.

- 6) Збережіть файл із назвою Вправа 3. Завершіть роботу з браузером і текстовим процесором.



Комп'ютерне тестування

Виконайте тестове завдання 3 з автоматичною перевіркою результату.



§ 4. Комп'ютер як інформаційна система. Складові комп'ютера

Як ви знаєте, інформаційна система забезпечує приймання, перетворення, опрацювання, збереження інформації та передавання результатів опрацювання людині, машині або іншій інформаційній системі. Комп'ютер є прикладом технічної інформаційної системи.

Складові комп'ютера та їх призначення

Персональний комп'ютер складається з апаратної частини і програмного забезпечення.

Апаратна частина (від англ. *hardware* — тверда частина) складається зі з'єднаних між собою різноманітних пристроїв, які можна побачити, доторкнутися.

Програмне забезпечення (від англ. *software* — м'яка частина) складається з програм, установлених на комп'ютері та призначених для забезпечення його роботи.



Пристрої апаратної частини комп'ютера не можуть опрацьовувати дані за відсутності програмного забезпечення. Саме взаємодія апаратної та програмної складових лежить в основі роботи комп'ютера.

Апаратна складова комп'ютера

Більшість сучасних комп'ютерів побудовано за принципами, які обґрунтував американський учений Джон фон Нейман (рис. 4.1, а) ще в 1948 році. Відтоді змінився зовнішній вигляд комп'ютерів, з'явилися нові пристрої. Але загальна схема роботи комп'ютера лишилася без змін (рис. 4.1, б).

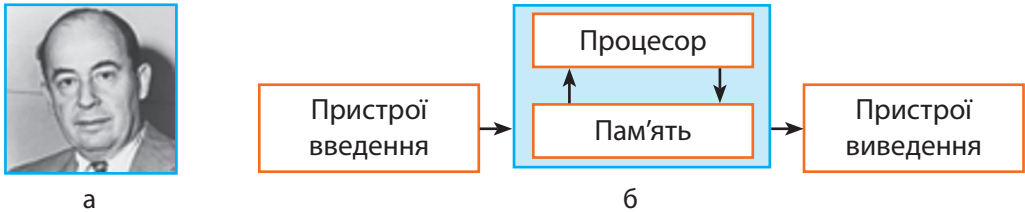


Рис. 4.1

Апаратну складову комп'ютера можна розподілити на пристрої введення, виведення, опрацювання, зберігання.

Пристрої введення інформації (рис. 4.2) призначені для введення даних у комп'ютер. Основні пристрої введення — *клавіатура* і *миша*.



Рис. 4.2



Для введення текстової і графічної інформації також застосовується *електронно-оптичний сканер*, для введення звукової інформації — *мікрофон*, для введення команд під час роботи комп'ютерних ігор і тренажерів — *джойстик* і *кермо*.

Пристрої виведення інформації (рис. 4.3) призначені для виведення результатів опрацювання даних на комп'ютері у вигляді чисел, таблиць, текстів, графіків, малюнків, креслень, анімації, звуків, тобто у вигляді, зручному для сприйняття людиною.



Рис. 4.3

Основний пристрій виведення — *монітор*. Пристрої введення-виведення називають *зовнішніми пристроями* комп'ютера.

Пристроєм опрацювання інформації є *процесор* (рис. 4.4). Він також керує іншими пристроями комп'ютера. Процесор по чергові виконує команди програми. Він складається з арифметико-логічного пристрою, де виконуються всі арифметичні й логічні операції, і пристрою керування, який координує взаємодію всіх пристроїв комп'ютера під час їх роботи.



Рис. 4.4

Пристрої зберігання інформації, пам'ять (рис. 4.5) призначені для тимчасового або постійного зберігання даних та програм, за допомогою яких виконується їх опрацювання.

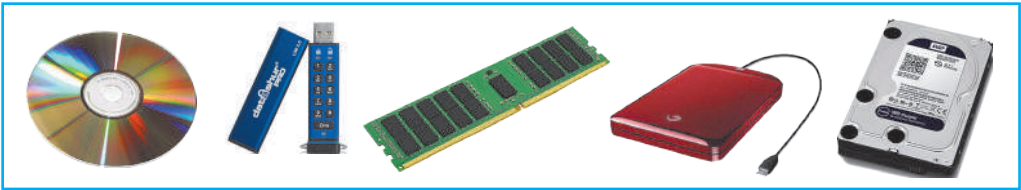


Рис. 4.5

Розрізняють внутрішні та зовнішні пристрої пам'яті.

Зовнішні пристрої — це накопичувачі на жорстких магнітних дисках та накопичувачі на оптичних дисках, флешпам'ять. Пристрій флешпам'яті, на відміну від інших носіїв даних, не містить рухомих деталей, запам'ятовування ґрунтується на електричних зарядах. Пристрій має невелику масу й малі розміри.

Програмне забезпечення комп'ютера

Програмне забезпечення — це набір програм, які керують роботою комп'ютера, забезпечують ведення діалогу з користувачем або користувачкою, за їх допомогою опрацьовують різноманітні дані, створюють нові програми.

Програмне забезпечення комп'ютера поділяють на системне, прикладне та службове (рис. 4.6).

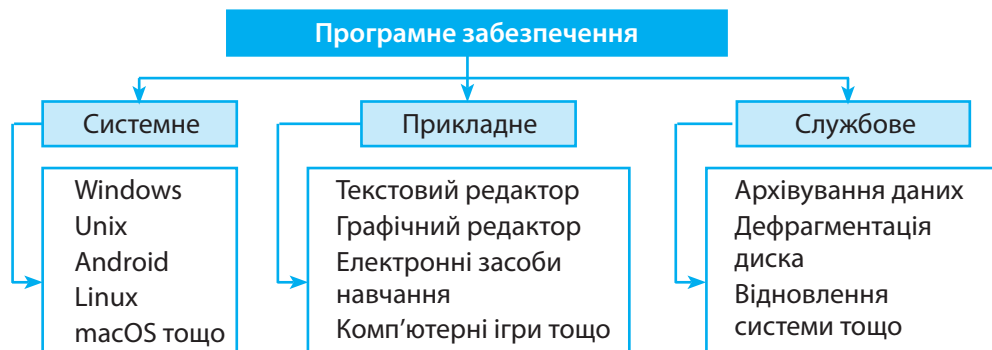


Рис. 4.6

До **системного програмного забезпечення** належать операційна система, системні програми, які виконують функції керування ресурсами комп'ютера. *Операційна система* — це набір програм, що автоматично завантажуються під час вмикання комп'ютера й використовуються для керування обчислювальними процесами, програмними й апаратними ресурсами.

До **прикладного програмного забезпечення** належать програми, що забезпечують виконання необхідних користувачу/користувачці робіт. Розрізняють програми загального (текстові та графічні редактори, програми для роботи в мережі тощо) і спеціального призначення (використовують у лікарнях, поліції, магазинах, банках тощо) та інструментальні програмні засоби (програми для створення інших програм).

До **службового програмного забезпечення** належать так звані утиліти — службові програми для обслуговування дисків, створення та підтримки архівів, боротьби з вірусами тощо.



Питання для самоперевірки

1. Що таке інформаційна система?
2. Які пристрої становлять апаратну частину комп'ютера?
3. Назвіть пристрої введення інформації.
4. Які пристрої призначені для виведення інформації?
5. Що входить до програмного забезпечення комп'ютера?
6. Яке призначення операційної системи?



Вправа 4

- Розглянути елементи апаратної складової комп'ютера.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.



- 1) Запустіть клавіатурний тренажер і потренуйтеся набирати пропонований текст українською та іноземною мовами. Повідомте результат учителю / вчительці.
- 2) Запустіть графічний редактор. Намалюйте схему за наведеним зразком (рис. 4.7).

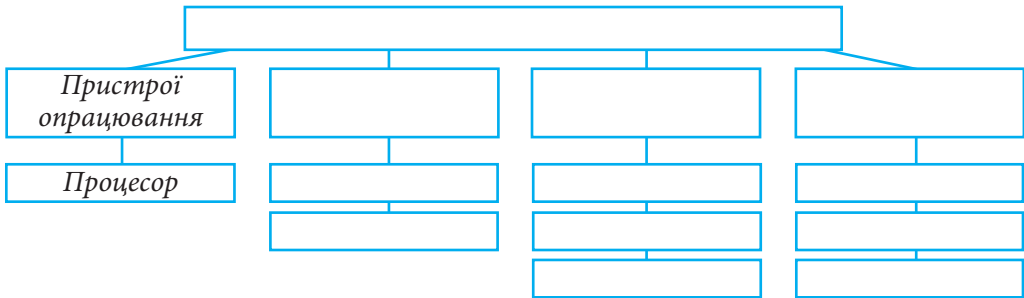


Рис. 4.7

- 3) У верхньому прямокутнику на схемі додайте напис Апаратна складова комп'ютера.
- 4) У другому рядку в прямокутниках допишіть назви видів апаратних складових комп'ютера.
- 5) У третьому і наступних рядках у прямокутниках до кожної апаратної складової комп'ютера запишіть назви кількох відповідних пристроїв.
- 6) Збережіть файл із назвою Вправа 4. Завершіть роботу з графічним редактором.



Комп'ютерне тестування

Виконайте тестове завдання 4 з автоматичною перевіркою результату.



§ 5. Операційна система та її інтерфейс

Зразу після ввімкнення робота комп'ютера розпочинається із завантаження та запуску програм операційної системи. Розглядаючи системне програмне забезпечення, ми вже згадували **операційну систему** — набір програм, призначений для забезпечення взаємодії всіх пристроїв комп'ютера і виконання різних команд користувача або користувачки.

Інтерфейс користувача

Як відбувається діалог комп'ютера і людини?



Інтерфейс користувача — це сукупність засобів для взаємодії між користувачем / користувачкою та комп'ютером.

Перші операційні системи мали *текстовий* інтерфейс, команди керування необхідно було пам'ятати та вводити з клавіатури. *Графічний* інтерфейс більш наочніший: команди вибираються зі списків (меню), додаткові параметри зазначаються в діалогових вікнах, об'єкти зображуються значками — піктограмами із назвами під ними. Під об'єктами розуміють усе, чим керує операційна система: папки, файли, мережа, диски, пристрої.



Значки об'єктів операційної системи часто мають піктограми із зображенням оригіналу: панель керування, телефон і модем, миша, принтери і факси, екран тощо (рис. 5.1).

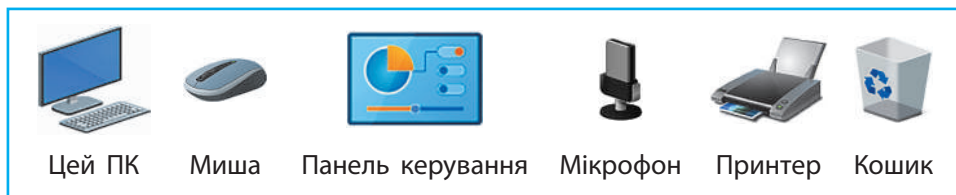


Рис. 5.1

Значки можуть міститися на робочому столі, у вікнах папок.

Кожна програма або папка може мати власну, не схожу на інші піктограму. За виглядом піктограми часто можна визначити призначення об'єкта (рис. 5.2).

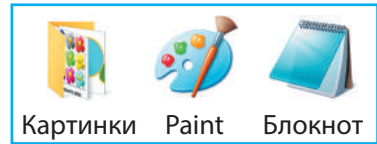


Рис. 5.2

Операції над об'єктами в операційній системі з графічним інтерфейсом зручно виконувати за допомогою миші.

Файли

Всі дані та програми, опрацьовувані за допомогою комп'ютера, зберігаються у файлах. Для комп'ютера вхідна інформація, результати її опрацювання, програми, тексти, малюнки, музика, фільми тощо у зовнішній пам'яті рівнозначні — це файли.



Файл — це іменована сукупність даних, розміщених на носіїві, які в процесі зберігання, передавання та опрацювання вважаються єдиним цілим.

Ім'я файлу складається з назви і розширення, відокремлених крапкою (рис. 5.3).

Ім'я файлу може містити літери (англійські, українські та ін.) і цифри, а також майже всі розділові знаки та спеціальні символи:

`_ $ # & @ ! % () { } ' ~ ^ .`



Рис. 5.3



*В імені файлу не можна використовувати такі символи: `/ \ : * ? " < > -`.*

Здебільшого назву файлу призначає людина, а розширення — програма, якою цей файл створено.

Залежно від розширення операційна система надає файлу піктограму для його позначення й визначає, за допомогою яких програмних засобів цей файл може бути відкритий.

Розглянемо деякі розширення та відповідні їм типи файлів:

.txt — текстові файли	.doc, .docx — файли документів Word
.mp3, .wav — звукові файли	.xls, .xlsx — файли документів Excel
.com, .exe — виконувані програми	.bmp, .png, .jpg — файли зображень

Папки

На дисках комп'ютера одночасно можуть зберігатися сотні тисяч файлів. Для їх групування використовують папки (теки, каталоги). Папка містить дані про розміщення вкладених у неї файлів та інших папок. Кожна папка має назву, розширень в іменах папок не використовують.



2 Розпочинаючи роботу над проектом, який складається з багатьох файлів, варто створити для нього окрему папку, щоб потім було легко відшукати потрібні для роботи файли (рис. 5.4).

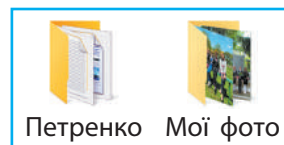


Рис. 5.4

Ярлики

Об'єкт може бути схований у «глибині» папок, і його пошук займатиме тривалий час. Для уникнення незручностей шлях до об'єкта записують у спеціальний файл, під час відкриття якого система відкриває сам об'єкт. Цей файл називають ярликом.

Ярлик об'єкта — це засіб швидкого доступу, який містить лише шлях до відповідного об'єкта. Ярлики для дисків, папок, програм і документів можна створити за допомогою команди контекстного меню.

Піктограма ярлика така сама, як у зв'язаного з ним об'єкта, лише в лівому нижньому куті вона має маленький квадрат, усередині якого зображено стрілку (рис. 5.5).

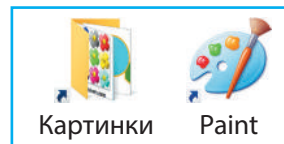


Рис. 5.5

Ярлик відіграє роль «запобіжника» від необережних дій осіб без достатнього досвіду роботи: видалення ярлика не призводить до знищення об'єкта.



Питання для самоперевірки

1. Які функції виконує операційна система?
2. Назвіть об'єкти операційної системи.
3. Що таке інтерфейс користувача?
4. Що таке файл?
5. Як записується ім'я файлу?
6. Поясніть різницю між значками об'єктів і ярликами.



Вправа 5

- Ознайомитися з операційною системою Windows.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.



- 1) Запустіть тренажер редагування тексту та потренуйтеся у виправлянні зайвих символів. Повідомте результат учителю / учительці.



- 2) Запустіть браузер. Знайдіть в інтернеті інформацію про призначення клавіші Print Screen (PrtSc) та програми Ножиці.

- 3) Запустіть текстовий процесор, створіть документ, запишіть (скопіюйте) призначення клавіші Print Screen.

- 4) Запишіть (скопіюйте) до документа призначення програми Ножиці.



- 5) За допомогою клавіші Print Screen або програми Ножиці зробіть скриншот робочого столу, щоб було видно розташовані на ньому значки.

- 6) Уставте в документ отриманий скриншот. Збережіть файл з назвою Вправа 5. Завершіть роботу з текстовим процесором.



Комп'ютерне тестування

Виконайте тестове завдання 5 з автоматичною перевіркою результату.







§ 6. Операції над файлами та папками

Якщо ми розглянемо робочий стіл комп'ютера, то побачимо багато різних значків і ярликів об'єктів. Для успішної роботи слід навчитися виконувати операції над файлами та папками.

Вікно папки «Комп'ютер»

Папка Комп'ютер відображає диски та пристрої, які під'єднані до комп'ютера. Вигляд, зміст і навіть назва вікна Комп'ютер залежить від версії операційної системи, а також від теми оформлення. В операційних системах Windows XP, Windows 7, Windows 10, Windows 13 вони різні. У Windows 10 вікно називається Цей ПК.

Як бачимо на рис. 6.1, вікно Цей ПК має знайомі кнопки, якими його можна:

- згорнути на панель завдань ;
- розгорнути на весь екран ;
- повернути до початкового розміру ;
- закрити .

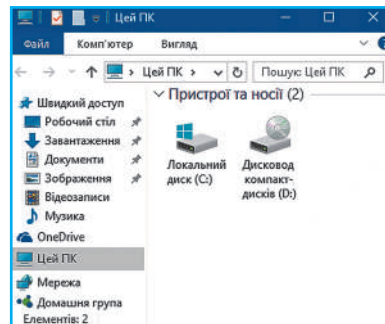


Рис. 6.1

Створення папки

Люди ще досить часто тримають паперові документи в папках, які підписують відповідно до виду документів, що в них зберігаються (ділові листи, накази тощо). Папки розміщують у шафі або в більшій папці.

Розташування документів в іменованих папках полегшує пошук кожного окремого документа (рис. 6.2). Папки на комп'ютері використовують так само: кожна папка має відповідну до її вмісту назву і зберігається на диску.

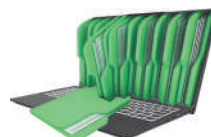


Рис. 6.2



Папка — це елемент файлової системи, контейнер для зберігання файлів та інших папок.

Папку можна створити на робочому столі, усередині іншої папки, на знімному носії даних, накопичувачі на жорсткому магнітному диску.

Щоб створити власну папку на диску D:, потрібно:

- 1) відкрити вікно Комп'ютер і на панелі навігації вибрати Диск D.;
- 2) на вільному місці клацнути правою кнопкою — відкриється контекстне меню;
- 3) установити вказівник на команду Створити — відкриється додаткове меню (рис. 6.3);
- 4) установити вказівник миші на команді Папку і клацнути. У вікні папки з'явиться піктограма нової папки, поряд — поле для введення назви (рис. 6.4);
- 5) увести з клавіатури назву папки й натиснути клавішу Enter — буде створено нову папку з уведеною назвою.

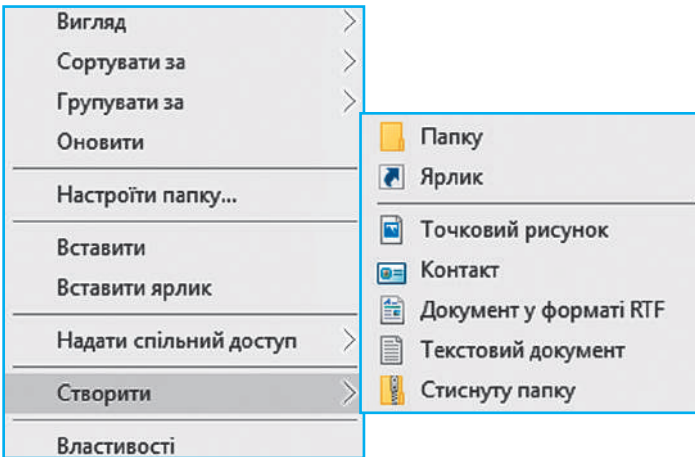


Рис. 6.3

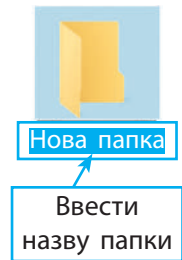


Рис. 6.4



Необхідно запам'ятати, де міститься створена папка, та її назву, щоб надалі її було зручно використовувати.

Створення ярлика

Для швидкого доступу до своїх файлів доцільно на робочому столі комп'ютера створити ярлик власної папки.

Для створення ярлика папки потрібно:

- 1) встановити вказівник миші на значок папки й викликати контекстне меню;
- 2) навести вказівник на команду Надіслати — з'явиться додаткове меню (рис. 6.5);
- 3) вибрати в додатковому меню команду Робочий стіл (створити ярлик) і клацнути;
- 4) закрити вікно — на робочому столі з'явиться ярлик із назвою папки; щоб відкрити власну папку, досить двічі клацнути на її ярлику на робочому столі.

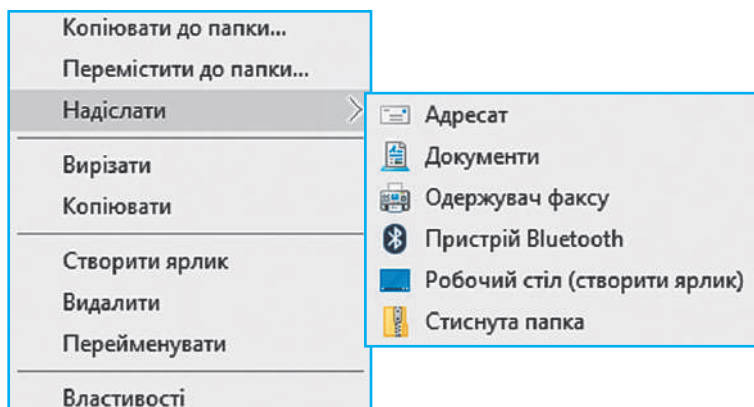


Рис. 6.5

Робота з папками, ярликами та файлами

Папки, ярлики та файли (далі — об'єкти) за необхідності можна перейменовувати, копіювати, переміщувати, видаляти й відновлювати після видалення.

Щоб **перейменувати об'єкт**, слід клацнути його ім'я і, коли зміниться колір фону тексту, клацнути ще раз — з'явиться курсор, і можна буде змінити або ввести нове ім'я.

Щоб скопіювати об'єкт, потрібно:

- 1) на значку об'єкта викликати контекстне меню, в якому вибрати команду Копіювати;
- 2) відкрити папку, де буде розміщена копія об'єкта;
- 3) викликати на вільному місці контекстне меню й вибрати команду Вставити — об'єкт з'явиться на новому місці й такий самий залишиться на старому.

Скопіювати виділений об'єкт в іншу папку можна, натиснувши сполучення клавіш Ctrl+C (копіювати) і Ctrl+V (вставити).

Щоб перемістити об'єкт, потрібно:

- 1) на значку об'єкта викликати контекстне меню, у якому вибрати команду Вирізати;
- 2) перевести вказівник миші в потрібне місце, викликати контекстне меню й вибрати команду Вставити — об'єкт з'явиться на новому місці та зникне зі старого.

Для переміщення виділеного об'єкта можна також використати сполучення клавіш Ctrl+X (вирізати) та Ctrl+V (вставити).

Щоб видалити об'єкт, потрібно на його значку викликати контекстне меню; вибрати команду Видалити — об'єкт зникне й переміститься в Кошик. Також можна скористатися клавішею Delete.

Щоб відновити об'єкт, слід відкрити Кошик, на значку потрібного об'єкта викликати контекстне меню і вибрати Відновити.



Перемістити, скопіювати або видалити об'єкт (тобто перемістити в Кошик) можна перетягуванням значка об'єкта правою або лівою кнопкою миші.

Після перетягування, наприклад, значка файлу в папку правою кнопкою миші, у контекстному меню буде запропоновано вибрати один із варіантів дії: копіювати; перемістити; створити ярлик (рис. 6.6).

Копіювати сюди
Перемістити сюди
Створити ярлики
Скасувати

Рис. 6.6

**Питання для самоперевірки**

1. Які кнопки керування має вікно папки Комп'ютер?
2. Як згорнути вікно на панель завдань?
3. Що таке контекстне меню? Як ним користуватися?
4. Як створити власну папку на диску D:?
5. Як створити й надіслати на робочий стіл ярлик папки?
6. Які дії виконуються над папками, файлами та ярликами?

**Вправа 6**

- Виконати розглянуті операції над файлами та папками.
Під час роботи за комп'ютером дотримуйтесь правил безпеки.



- 1) Запустіть тренажер редагування тексту й потренуйтеся у виправлянні тексту із пропущеними символами. Повідомте результат учителю / вчительці.

- 2) Створіть на робочому столі папку з назвою 5 клас.



- 3) Відкрийте папку 5 клас і створіть у ній три папки з назвами навчальних предметів на свій розсуд (рис. 6.7).

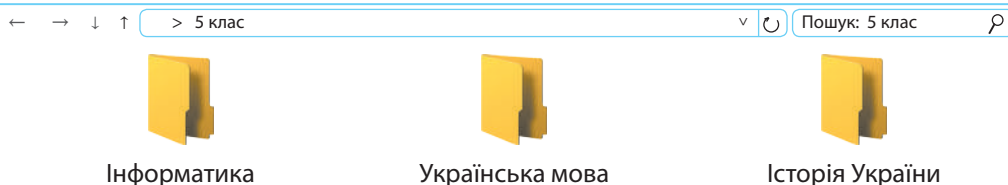


Рис. 6.7

- 4) Закрийте вікно папки. Видаліть папку 5 клас у Кошик.



- 5) Відкрийте папку Кошик. Відновіть папку 5 клас.

- 6) Відкрийте на робочому столі папку 5 клас та переконайтеся, що відновилися й внутрішні папки. Запишіть, яким є призначення папки Кошик.

**Комп'ютерне тестування**

Виконайте тестове завдання 6 з автоматичною перевіркою результату.





Практична робота 1

Виконання операцій над файлами та папками

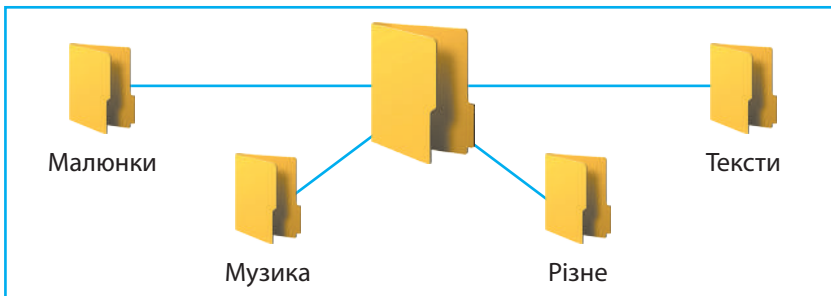
Завдання: виконати операції над файлами та папками засобами операційної системи.

Обладнання: комп'ютер із операційною системою, навушники.

Хід роботи

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Підготуйте комп'ютер до роботи.
2. Створіть у запропонованому вчителем / вчителькою місці папку, назвіть її своїм прізвищем.
3. У власній папці створіть папки з назвами Малюнки, Тексти, Музика, Різне (див. рисунок).



4. Створіть на робочому столі ярлик власної папки. Відкрийте папку за допомогою ярлика.
5. Відкрийте папку Малюнки, згорніть її до кнопки на панель завдань.
6. Розгорніть вікно папки Малюнки на весь екран, закрийте папку.
7. У папку Малюнки з папки, указаної вчителем / вчителькою, скопіюйте графічний файл.

8. У папку Тексти з папки, указаної вчителем /вчителькою, скопіюйте текстовий файл.
9. У папку Музика з папки, указаної вчителем /вчителькою, скопіюйте музичний файл.
10. Відкрийте графічний файл, збережений у папці Малюнки, закрийте файл.
11. Запустіть музичний файл, збережений у папці Музика, закрийте файл.
12. Покажіть роботу вчителю. Закрийте всі вікна, завершіть роботу за комп'ютером.

Зробіть висновок: які засоби для виконання операцій над файлами та папками надає операційна система комп'ютера.

РОЗДІЛ 2

МЕРЕЖЕВІ ТЕХНОЛОГІЇ ТА ІНТЕРНЕТ



§ 7. Комп'ютерні мережі. Локальна комп'ютерна мережа

§ 8. Глобальна комп'ютерна мережа

§ 9. Безпечне використання інтернету. Спілкування в інтернеті

§ 10. Пошук відомостей в інтернеті. Авторське право

Практична робота 2. Пошук інформації в інтернеті

ПОВТОРЮЄМО



Як ви знаєте, *комп'ютерна мережа* — це сукупність комп'ютерів, які можуть надавати один одному доступ до своїх апаратних (дисководи, принтери, сканери тощо) і програмних (програми, дані тощо) ресурсів.

Комп'ютери в мережі відрізняються за функціями, які вони виконують. Комп'ютер, що надає ресурси в мережу, називають *сервером*, а той, який ці ресурси використовує, — *клієнтом* (або *робочою станцією*).

Найбільша глобальна комп'ютерна мережа — *Інтернет*. Вона надає користувачській спільноті усього світу можливості спілкування, пошуку й перегляду інформаційних матеріалів.

Вам відомо, як здійснювати пошук в інтернеті за ключовими словами з використанням пошукових систем.

1. Що таке комп'ютерна мережа?
2. Назвіть апаратні й програмні ресурси мережі.
3. Який комп'ютер у мережі називають сервером?
4. Який комп'ютер у мережі називають клієнтом?
5. Що таке інтернет?
6. Які можливості надає інтернет?



Опрацювавши цей розділ, ви дізнаєтеся про особливості будови й роботи комп'ютерних мереж, навчитеся шукати потрібну інформацію, зберігати її та опрацьовувати.

§ 7. Комп'ютерні мережі. Локальна комп'ютерна мережа

Нагальна потреба в розв'язанні широкого кола завдань підштовхнула людей до ідеї об'єднання ресурсів різних комп'ютерів. Так утворилися комп'ютерні мережі.

Комп'ютерна мережа складається з двох або більше комп'ютерів, з'єднаних каналами зв'язку, якими передаються дані. Залежно від охоплюваної території комп'ютерні мережі бувають двох типів: локальні та глобальні.

Організація роботи в локальній мережі

Локальна мережа (LAN, від англ. *Local Area Network* — локальна мережа) — це мережа, що об'єднує комп'ютери та інші пристрої (принтери, сканери тощо), розташовані близько один від одного — в межах школи, банку, заводу тощо.

Для приєднання до мережі комп'ютер повинен мати спеціальний пристрій — **мережевий адаптер** (рис. 7.1, *а*). Для побудови локальної мережі потрібен також інший пристрій — **мережевий комутатор** (англ. *switch* — перемикач) (рис. 7.1, *б*). До нього кабелями приєднують комп'ютери (рис. 7.1, *в*) й решту пристроїв. Також можна побудувати безпроводну локальну мережу.



Рис. 7.1

► Сервер та робоча станція

Комп'ютер у мережі виконує одну з функцій: або надає мережеві ресурси, або використовує їх.



Комп'ютер, який надає ресурси, називають **сервером**.
Комп'ютер, який використовує ресурси, називають **клієнтом**, або **робочою станцією** (рис. 7.2).

Розглянемо схему на рис. 7.2.

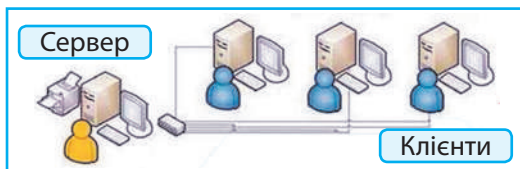


Рис. 7.2

► Адресація в мережі

У мережі обмін даними між пристроями відбувається за певними правилами, які називають **протоколами**. Згідно з ними кожен комп'ютер або інший пристрій у мережі має спеціальний номер (рис. 7.3) — **IP-адресу** (читається «ай-пі-адреса»).



Рис. 7.3

IP-адреса пристрою складається із чотирьох чисел у межах від 0 до 255, відокремлених крапками. У локальних мережах застосовують IP-адреси, що починаються з 10, 172.16 або 192.168.



Сервер локальної мережі шкільного комп'ютерного класу може мати IP-адресу 192.168.0.1, а комп'ютери на робочих місцях учнів і учениць — 192.168.0.2, 192.168.0.3 тощо.



Жодні два пристрої в мережі не можуть мати однакових IP-адрес. Проте один пристрій може належати одночасно до кількох мереж і, відповідно, мати кілька IP-адрес.

► Вхід у локальну мережу

Щоб увійти в локальну мережу, в ОС Windows слід скористатися папкою Мережеве оточення (рис. 7.4). У вікні цієї папки містяться значки інших комп'ютерів і мережевих пристроїв, а також ярлики мережевих ресурсів, які відкривалися раніше.



Мережеве оточення

Рис. 7.4

Щоб знайти певний ресурс у мережі, потрібно:

- 1) на додатковій панелі вибрати команду Показати комп'ютери робочої групи — у робочому полі вікна з'являться значки доступних комп'ютерів робочої групи;
- 2) двічі клацнувши значок потрібного комп'ютера, перейти до роботи з відкритими ресурсами цього комп'ютера — відкрити папки, файли, створювати ярлики тощо.

► Спільне використання ресурсів

Під час колективної роботи в мережі доцільно пам'ятати про таке.

Деякі папки можуть бути доступними лише для читання. У таку папку неможливо ані записати новий файл, ані змінити та зберегти наявний.

Крім того, ресурси іншого комп'ютера стають недоступними після його вимкнення. Тому буває корисно зробити копію файлу на власному комп'ютері навіть тоді, коли немає потреби вносити до нього зміни (рис. 7.5).



Рис. 7.5

Якщо до одного з комп'ютерів локальної мережі приєднано принтер і відкрито доступ через мережу, то користувачі та користувачки інших комп'ютерів можуть друкувати документи (тексти, малюнки тощо).



Питання для самоперевірки

1. Які мережі називають локальними?
2. Поясніть різницю між сервером і робочою станцією.
3. Опишіть принцип адресації комп'ютерів у мережі.
4. У чому полягає спільне використання ресурсів?
5. Що таке протокол?
6. Як знайти певний ресурс у мережі? Опишіть алгоритм пошуку.

**Вправа 7**

- Навчитися працювати з файлами за допомогою комп'ютерної мережі.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.



- 1) Запишіть, як називається робоча група, до якої належить ваш комп'ютер. Яке його ім'я в мережі?
- 2) З'ясуйте і запишіть, яка IP-адреса комп'ютера.
- 3) У мережевій папці, вказаній учителем / учителькою, створіть папку. Назвіть її своїм прізвищем та ім'ям.
- 4) Скопіюйте до створеної папки вказаний учителем / учителькою файл, розміщений на вашому комп'ютері.
- 5) Скопіюйте до створеної папки вказаний учителем / учителькою файл, розміщений на іншому комп'ютері.
- 6) Зробіть скриншот вікна папки зі значками скопійованих файлів. Додайте скриншот у документ і збережіть у цій самій папці з назвою Вправа 7.



Підказка: в ОС Windows натискання сполучення клавіш Alt + PrtScr копіює в буфер обміну скриншот активного вікна. Після цього його можна вставити, наприклад, у документ.

**Комп'ютерне тестування**

Виконайте тестове завдання 7 з автоматичною перевіркою результату.



§ 8. Глобальна комп'ютерна мережа

Глобальна мережа (WAN) — це мережа, що об'єднує комп'ютери та локальні мережі, розташовані на великій відстані один від одного — в різних містах, державах і на різних континентах.

Організація роботи в глобальній мережі

Найбільшою глобальною мережею є інтернет.



Інтернет — глобальна комп'ютерна мережа, що складається з мільярдів комп'ютерів у всьому світі.

Інтернет об'єднує локальні мережі навчальних і наукових закладів, промислових підприємств, державних установ тощо, а також окремі комп'ютери. В інтернеті циркулює різноманітна інформація, доступна майже з будь-якого куточка планети (рис. 8.1).



Рис. 8.1

Користувачі та користувачки інтернету можуть швидко знайти необхідну інформацію, що зберігається на серверах у Канаді чи Японії, надіслати повідомлення з одного комп'ютера на інший чи на мобільний телефон, поспілкуватися з друзями та подругами у відеорежимі знайти партнера/партнерку для гри в шахи тощо.

Глобальна мережа створює можливість дистанційно навчатись, проводити відеоконференції, «мандрувати» світом, стежити за курсами валют, робити покупки в інтернет-магазинах тощо.



У квітні 2021 року необмежений (безлімітний) доступ до інтернету з'явився на українській антарктичній станції «Вернадський» (рис. 8.2).



Рис. 8.2

Провайдери

У повсякденному житті нам часто доводиться чути слово «провайдер» (від англ. *provider* — постачальник).



Провайдер — це організація, що надає послуги, пов'язані з доступом до глобальної мережі.

Серед **функцій провайдерів** найважливішими є забезпечення доступу до інтернету; виділення дискового простору для сайтів (хостинг, від англ. *hosting*); підтримка роботи поштових скриньок; послуги зв'язку з передавання голосової інформації та ін.

Доменне ім'я

Разом із IP-адресами пристроїв для адресації ресурсів в інтернеті використовують доменні імена.



Доменне ім'я — текстова адреса комп'ютера або іншого ресурсу в інтернеті.

Доменне ім'я складається з назв кількох частин доменів (імен доменів), відокремлених крапками. Рівень доменів рахується справа наліво. Останнім у доменному імені зазначають домен *1-го (верхнього) рівня*, який зазвичай вказує тип організації чи державу; назва домену *2-го рівня* найчастіше є іменем сервера даних; найпершим зліва є власне ім'я ресурсу (рис. 8.3).



Рис. 8.3

Отже, до кожного домену 1-го рівня належать багато доменів 2-го рівня; до кожного домену 2-го — багато доменів 3-го рівня.



Кожному доменному імені відповідає певна IP-адреса, але одній IP-адресі може відповідати декілька доменних імен.

Гіпертекст

Документи, розміщені в інтернеті, можуть містити зв'язки, що надають миттєвий доступ до відомостей в інших документах.




Гіпертекст — це текстовий документ, який містить гіперпосилання, клацання на яких дає змогу перейти до іншого місця в документі або до іншого документа.

Термін «гіпертекст» у 1963 році утворив Теодор Нельсон, американський соціолог, філософ, піонер інформаційних технологій (рис. 8.4). Гіпертекстом може бути слово, фраза, графічне зображення.



Рис. 8.4

У разі клацання деяких гіперпосилань виконуються певні дії: відкриття файлу, завантаження поштового клієнта для написання електронного листа тощо.

Гіперпосилання виділяються кольором символів, підкреслюванням, рамкою малюнка тощо. Під час наведення на гіперпосилання вказівник миші набуває вигляду руки: . Документ, пов'язаний із гіперпосиланням, може міститися на будь-якому комп'ютері в мережі (рис. 8.5).

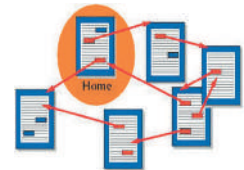


Рис. 8.5

World Wide Web

World Wide Web (WWW) — це сукупність документів, пов'язаних гіперпосиланнями та розміщених на безлічі комп'ютерів, із наданням можливості доступу до них через інтернет.

Інформація подається у вигляді гіпертекстових документів — **вебсторінок**. Кілька пов'язаних гіперпосиланнями вебсторінок, які мають спільну тему та розміщення, називають **вебсайтом** або просто **сайтом**.

Доступ до розміщених на серверах сайтів можливий у будь-який час і з будь-якого приєднаного до інтернету комп'ютера.

Для перегляду вебсторінок потрібна спеціальна програма — **вебголядач** або **браузер** (від англ. *browser* — оглядач, провідник).

Браузер може відображати текстову або графічну інформацію, відтворювати звуки, анімацію та відео, розміщені на сайтах. Завдяки гіперпосиланням користувач та користувачка може переходити на інші вебсторінки. Браузери дають змогу переглядати вебсторінки з віддаленого комп'ютера (режим онлайн) або збережені на комп'ютері користувача або користувачки (режим офлайн).

3 Популярними браузерами є програми Opera, Mozilla Firefox, Google Chrome тощо. З операційними системами Windows постачався Internet Explorer, на зміну якому у Windows 10 прийшов Microsoft Edge (рис. 8.6).



Рис. 8.6

У вікні браузера Google Chrome (його фрагмент на рис. 8.7) є елементи керування, які можна бачити і в інших браузерах.

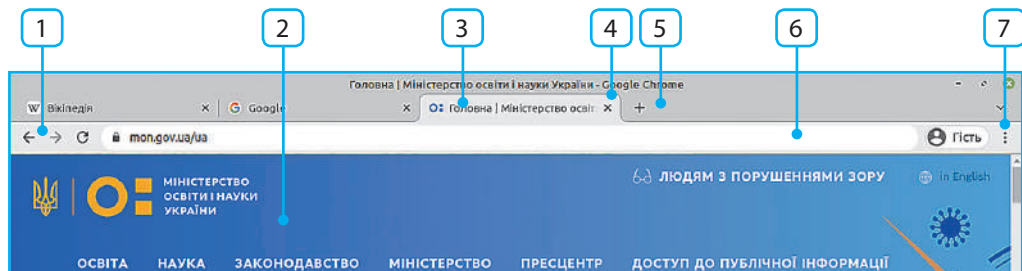


Рис. 8.7

- 1 — кнопки переходу між сторінками;
- 2 — робоче поле;
- 3 — вкладка вебсторінки;
- 4 — кнопка закриття вкладки;
- 5 — кнопка додавання нової вкладки;
- 6 — рядок адреси;
- 7 — кнопка Меню.

● Барбара Лісков — американська дослідниця в галузі інформатики, лауреатка премії Тюрінга 2008 року. У цей час є професоркою у Массачусетському технологічному інституті (рис. 8.8).



Рис. 8.8

Робота із сайтами

Нині приватні особи, організації, підприємства, навчальні заклади розміщують на сайтах багато різноманітної інформації.



4 Sinoptik.ua — прогноз погоди; novaroshta.ua — доставка товарів; megogo.net — розваги та ін.

Для переходу на вебсторінку потрібно ввести її адресу в рядок адреси браузера. Це можна зробити одним із таких способів:

- набрати з клавіатури;
- вибрати адресу вебсторінки зі списку меню Закладки або Історія;
- на відкритій вебсторінці клацнути на гіперпосиланні, що спричинить перехід на нову вебсторінку.

Щоб не втратити зображення поточної сторінки, слід на гіперпосиланні клацнути правою кнопкою й у контекстному меню вибрати Відкрити посилання в новій вкладці.

Після введення адреси вебсторінки автоматично встановлюється зв'язок із сервером і вебсторінка відкривається.



5 Якщо в полі адреси браузера ввести адресу abetka.ukrlife.org, у робочому полі відкриється головна сторінка сайту Весела абетка (рис. 8.9). Тут можна знайти багато цікавих матеріалів (казки, усмішки, загадки, скоромовки тощо).

Якщо клацнути на гіперпосиланнях Абетки, Казки, Читанка, Усмішки, Приказки, Скоромовки тощо, то потрапимо на інші вебсторінки з гіперпосиланнями для подальших переходів.

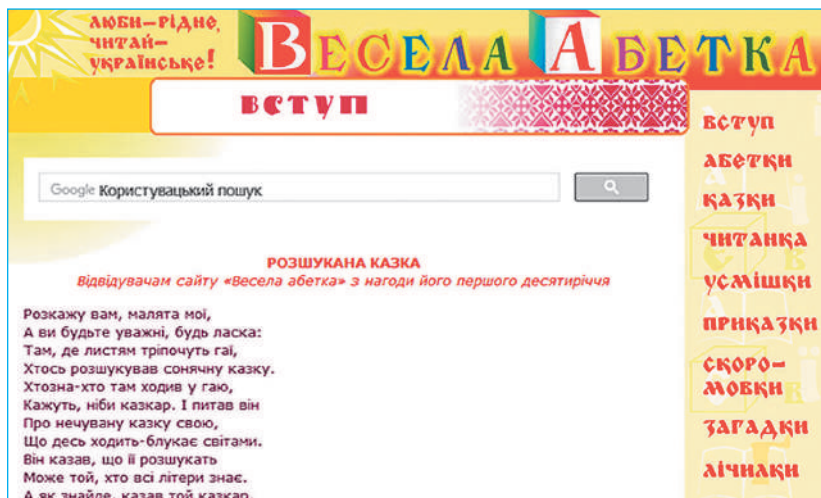


Рис. 8.9



Питання для самоперевірки

1. Розкажіть, що ви знаєте про інтернет.
2. Яку роботу виконує провайдер?
3. Що таке гіпертекст? Наведіть приклади.
4. Що може відбутися після клацання на гіперпосиланні?
5. Для чого призначений браузер?
6. Назвіть поширені браузери.
7. Якими способами можна ввести адресу сайта?
8. Назвіть кілька популярних сайтів.



Вправа 8

- Ввести адресу вебсторінки в браузері, переглядати її та переходити за гіперпосиланнями.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

- 1) Запустіть браузер і текстовий процесор.
- 2) Наберіть в адресному рядку браузера abetka.ukrlife.org і натисніть клавішу Enter. Опишіть, що відбулося після цього. Зробіть скриншот (клавіша Print Screen) і вставте його в текстовий документ.

- 3) Перейдіть в інший розділ сайту (Казки, Усмішки тощо) і додайте в документ скриншот.
- 4) Скопіюйте з адресного рядка браузера адресу сторінки та вставте в документ.
- 5) Перейдіть в інший розділ сайту та додайте в документ скриншот і адресу сторінки.
- 6) Збережіть файл у власній папці (див. вправу 7) з назвою Вправа 8. Заверште роботу з комп'ютером.



Комп'ютерне тестування

Виконайте тестове завдання 8 з автоматичною перевіркою результату.



§ 9. Безпечне використання інтернету. Спілкування в інтернеті

Інтернет — це необмежений ресурс, який можна використовувати для навчання, відпочинку та спілкування з друзями тощо. В інтернеті є багато цікавих і корисних матеріалів, проте він може бути й небезпечним, особливо для дітей.

Небезпеки, пов'язані з використанням інтернету

Кожна особа, яка користується інтернетом, має знати про загрози, що існують у ньому. Наприклад, про комп'ютерні віруси.

Віруси — комп'ютерні програми, здатні самостійно поширюватися з одного комп'ютера на інший. Вони можуть завдавати шкоди: знищувати, пошкоджувати дані або порушувати роботу комп'ютера.

Троянські програми — шкідливі програми, які можуть викрадати дані чи використовувати комп'ютер зі зловмисною метою.

Сайти, які пропагують насилля та поведінку, що несе загрозу життю та здоров'ю.

Хакерство — отримання несанкціонованого доступу до комп'ютерних систем, зазвичай із метою отримання секретної інформації.

Спам — масове розсилання небажаних повідомлень рекламного чи іншого характеру.

Розглянемо програми, що запобігають цим загрозам або навіть унеможливають їх.

Програма	Призначення
Брандмауер	Програмне й апаратне забезпечення, що захищає локальну мережу від небезпек, які є в інтернеті
Антивіруси	Службові програми для знаходження комп'ютерних вірусів та відновлення заражених файлів, а також для запобігання зараженню комп'ютера
Спам-фільтр	Програма для фільтрації електронної пошти та посилань, яка блокує листи від небажаних адресатів

● Веб-серфінг (інтернет-серфінг) — це поверхневе ознайомлення з інтернет-сайтами та сторінками Всесвітньої павутини (рис. 9.1). Надмірне захоплення ним може призвести до бажання постійно накопичувати інформацію, безупинно завантажувати «потрібні» файли тощо.



Рис. 9.1

Правила безпечного користування інтернетом

Інтернет є публічним місцем. Працюючи онлайн, слід дотримуватися правил безпеки так само, як ви, наприклад, дотримуетесь правил дорожнього руху.

Основні **правила безпечної роботи** в інтернеті такі.

- Не надавайте незнайомцям особистих даних рідних.
- Спілкуючись в інтернеті, не вказуйте свої особисті дані, а користуйтеся псевдонімом (ніком).
- Не реєструйтеся на сумнівних сайтах.
- Купуйте в інтернеті товари та послуги тільки під наглядом дорослих.
- Не погоджуйтеся на зустріч із незнайомими людьми.
- Повідомляйте батьків, старших друзів і подруг у разі отримання інформації, що має характер залякування.
- Не заходьте на незнайомі сайти.
- Пам'ятайте, що не вся розміщена в інтернеті інформація є правдивою.



Антивірус українського виробництва Zillya! (рис. 9.2) захищає комп'ютер, на який встановлений, від вірусів, троянських, шпигунських та інших шкідливих програм. Його якість підтверджено сертифікатом Державної служби спеціального зв'язку та захисту інформації України.

Антивірусна програма розробляється від 2009 року для ОС Windows, а від 2016 року — ще й для ОС Android (рис. 9.3) (від 2018 — під назвою Zillya! Internet Security&Scanner for Android). Для користувачів, яким потрібен тільки базовий захист, випускається безплатна версія.

Отримати Zillya!, а також дізнатися більше про різні його версії можна на сайті програми <https://zillya.ua/>.



Рис. 9.2

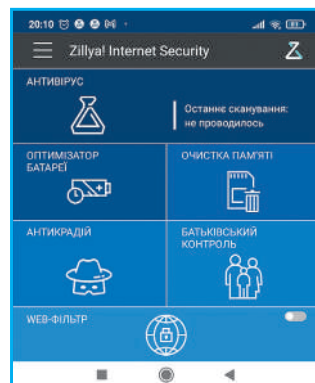


Рис. 9.3

Спілкування в інтернеті

Усе більше людей використовують інтернет для спілкування.

Один з найдавніших засобів спілкування за допомогою комп'ютерів — **електронна пошта** (англ. *E-mail* — Е-пошта) — моделює звичайну пошту: знаючи адресу користувача / користувачки, можна надіслати листа, якого особа прочитає, коли їй буде зручно. Докладніше про електронну пошту ви дізнаєтесь у наступних курсах.

- Існує ціла низка засобів для миттєвого обміну повідомленнями, серед яких Viber, Facebook Messenger (рис. 9.4) та багато інших.



Рис. 9.4

Форум — це засіб спілкування на сайті, де користувач та користувачка може створювати власні теми або ставити запитання для відкритого обговорення. Наприклад, можна написати кулінарний рецепт і запропонувати іншим приготувати за ним страву та поділитись враженнями.

Соціальна мережа — «місце» в інтернеті, де спілкуються одночасно дуже багато людей: створюють групи за інтересами, обговорюють новини, обмінюються різними матеріалами, публікують оголошення.

- Розглянемо логотипи популярних соціальних мереж (рис. 9.5).
2 У найбільшій у світі мережі Facebook спілкуються до 3 млрд людей. Мережа Instagram призначена для обміну фотографіями, а TikTok — для поширення коротких відео. Мережа Twitter призначена для опублікування й обговорення коротких текстових повідомлень (довжиною до 280 символів). Месенджер Telegram — для обміну повідомленнями, схожий на мережу.



Рис. 9.5

Збільшення швидкості передавання даних в інтернеті та зростання кількості людей, які мають доступ до нього, робить електронне спілкування надзвичайно популярним.



Питання для самоперевірки

1. Які небезпеки пов'язані з використанням інтернету?
2. Назвіть основні правила безпечної роботи в інтернеті.
3. Назвіть види електронного спілкування.
4. Які ви знаєте засоби для електронного спілкування?
5. Назвіть декілька соціальних мереж.



Вправа 9

- Зберегти на комп'ютері тексти та зображення, знайдені в інтернеті.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

- 1) Запустіть браузер. Перейдіть на сайт abetka.ukrlife.org.
- 2) Відшукайте на сайті загадку про зиму.
- 3) Запустіть текстовий редактор і скопіюйте загадку в текстовий документ.

Збережіть файл з назвою Загадка у папку Тексти, що міститься у власній папці.

- 4) Відкрийте запропонований учителем / учителькою сайт і відшукайте малюнок на тему «Моя країна».

- 5) Збережіть малюнок у власній папці. Запишіть назву та розмір файлу. Закрийте програми, з якими працювали.

- 6) На сайті uk.wikipedia.org відшукайте сторінку про свій населений пункт.

Скопіюйте і збережіть текст сторінки у власній папці у файлі з назвою Вправа 9.



Комп'ютерне тестування

Виконайте тестове завдання 9 з автоматичною перевіркою результату.



§ 10. Пошук відомостей в інтернеті. Авторське право

Інтернет містить багато різних сайтів, кожен з яких складається з низки вебсторінок. Наприклад, український розділ вікіпедії має більш ніж 1 100 000 статей. Але як у цьому інформаційному «морі» відшукати те, що нас цікавить?

Пошукові системи

Відшукати інформацію можна за допомогою спеціальної програми — **пошукової системи**, яка, за запитами користувачів і користувачок, переглядає мільйони вебсторінок.

Пошукову систему встановлюють на пошуковому сервері. На ньому зберігаються також дані про зміст вебсторінок, серед яких і здійснюється пошук. Спеціальні програми регулярно оновлюють і доповнюють ці дані.

1 До найвідоміших пошукових систем належать DuckDuckGo, Google, Yahoo, Lycos тощо (рис. 10.1).



Рис. 10.1

Отримавши пошуковий запит, програма знаходить сторінки, що містять задані ключові слова, упорядковує їх і передає у вигляді списку з гіперпосиланнями.

Ключове слово — це слово, що є визначальним для пошуку певної інформації.

Для пошуку інформації вбудованими засобами браузера слід завантажити браузер, увести в поле пошуку ключові слова та натиснути клавішу Enter. Після цього буде надіслано запит на одну з пошукових систем і через деякий час у вікні з'явиться список посилань на знайдені вебсторінки.

► Пошукова система Google

Щоб скористатися пошуковою системою Google, потрібно в браузері набрати її адресу google.com.ua і натиснути Enter — після завантаження на екрані з'явиться початкова сторінка (рис. 10.2) Google; далі потрібно ввести ключові слова в текстове поле, натиснути кнопку Пошук Google та дочекатися результатів.

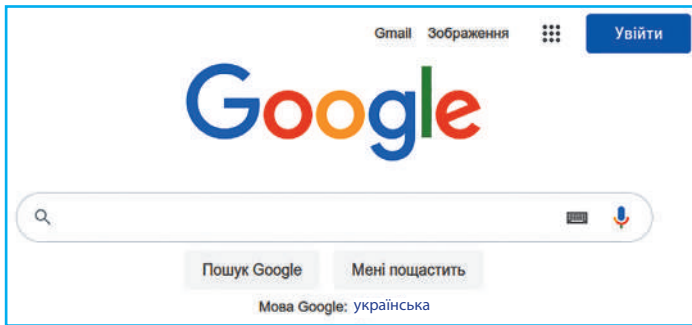


Рис. 10.2

► Стратегія пошуку інформації

Внаслідок пошуку можна отримати занадто багато посилань. Успіх пошуку залежить від набору ключових слів, які варто добирати за такими правилами.

- Шукайте більше ніж за одним словом: краще написати *птахи України*, ніж *птахи*.
- Найважливіші ключові слова пишуть першими.
- Пишіть без помилок.
- Використовуйте синоніми, якщо список знайдених сторінок занадто малий.
- Не починайте слова з великої літери, крім власних назв.
- Якщо хочете знайти точну фразу, візьміть її в лапки: *«Людина нібито не літає, а крила має»*. Лапки означають, що слід шукати слова саме в такій послідовності.
- Дефіс перед ключовим словом означає, що цього слова не повинно бути на шуканій сторінці.





За запитом «Котигорошко» серед перших результатів отримали декілька посилань на український мультфільм «Пригоди Котигорошка та його друзів». Коли ж запит записали так: «Котигорошко — мультфільм», то в перших рядках з'явилися посилання на аудіоказки та інші сторінки, не пов'язані з мультфільмом.

► Завантаження файлів з інтернету й авторське право

Деякі гіперпосилання вказують не на інші вебсторінки, а на файли, які можна завантажити й зберегти на своєму комп'ютері (англ. *download* — завантажити). У разі клацання на такому посиланні відкривається вікно Зберегти як..., у якому слід уточнити папку для збереження й ім'я файлу і, клацнувши Зберегти, розпочати завантаження файлу. Клацнувши Скасувати, можна припинити завантаження файлу.

Усі матеріали, розміщені в інтернеті, мають своїх власників/власниць. Лише вони мають право підписувати матеріали своїм іменем і розповсюджувати їх. Таке право називають **авторським**. Його охороняє Закон України «Про авторське право».

- Плагіатом (від латин. *plagium* — викрадати) називають оприлюднення повністю або частково чужого твору під іменем особи, яка не є його автором / авторкою.

Використовувати матеріали з інтернету для особистих цілей дозволено, але поширювати їх як власні — ні. Наприклад, використавши матеріал у своїй презентації, слід додати до неї посилання на джерело.

► Критичне оцінювання інформації, отриманої з інтернету

Інформація, подана на багатьох сайтах інтернету, не завжди перевірена, а отже, не завжди надійна. Тому, користуючись інтернетом, слід навчитися спостерігати, розпізнавати та критично аналізувати знайдену інформацію (рис. 10.3).



Рис. 10.3

«Фейк» (англ. *fake*) перекладається як «підробка». Спочатку слово «фейк» використовувалося в інтернеті для позначення недостовірної інформації, а потім його стали широко застосовувати в повсякденному житті.

Напротивагу фейку, **факт** — це подія або явище, правдивість якого доведено. Наприклад, фактом є подія, що вже відбулася. **Судження** — це особиста думка людини, яку можна підтвердити або спростувати. Зазвичай судження має емоційне забарвлення.

2 Твердження «Місяць тому вийшла нова серія мультфільму» описує факт, а твердження «Неперевершена графіка сприяла надзвичайній популярності мультфільму» містить судження. У другому випадку слова «неперевершена» і «надзвичайній» передають емоції автора / авторки, що намагається привернути увагу до об'єкта обговорення.

Використання інтернету для навчання

Багато цікавого й корисного можна знайти в інтернеті. Нині набули популярності **онлайн-програми** — програми, з якими можна працювати в інтернеті в режимі реального часу (рис. 10.4).

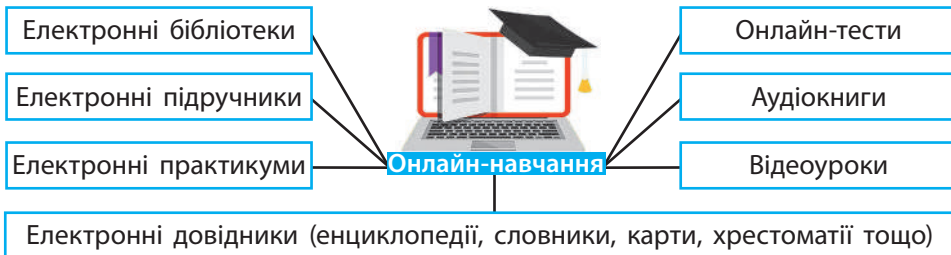


Рис. 10.4

Навчання в глобальній мережі має *певні переваги* порівняно з іншими способами набуття знань: навчальні матеріали доступні в будь-який час, що дозволяє планувати час роботи на власний розсуд; спілкування з іншими дітьми дозволяє швидше зрозуміти складні місця; в інтернеті можна шукати додаткові матеріали.

Проте є й *певні недоліки*: доводиться багато часу проводити перед екраном комп'ютера; не кожна людина може змусити себе працювати, якщо раптом трапиться складне завдання.

Існує багато сайтів, які допомагають учням і ученицям у вивченні навчальних предметів. Не варто засмучуватись, якщо не вдається опанувати якийсь інтернет-курс. Можливо, він розрахований на старше учнівство або невдало побудований. У такому разі слід порадитися зі старшими людьми й вибрати інший навчальний сайт.

Основні варіанти використання інтернет-ресурсів такі.

- **Дистанційна освіта.** Багато курсів для учнів і учениць є на сайтах osvita-online.com.ua, prometheus.org.ua, ed-era.com та ін.
- **Інтелектуальні змагання.** Серед учнів і учениць користуються популярністю інтернет-олімпіади. Майбутній учасник олімпіади має зареєструватися на сайті, щоб виконати завдання.
- **Проектна робота.** Можна працювати над реальним проектом і вчитися, спілкуючись з іншими учасниками.
- **Самостійне навчання.** Наприклад, на сайті code.org вже майже 2 мільйони школярів і школярок з усього світу вивчають основи програмування (рис. 10.5).



Рис. 10.5



Питання для самоперевірки




1. Назвіть засоби пошуку інформації в інтернеті.
2. Опишіть процес пошуку за допомогою засобів пошуку, вбудованих у браузер.
3. Опишіть призначення та дію пошукової системи.

4. Як потрапити на вебсторінку пошукового сервера?
5. У якому вигляді пошукова система подає результати?
6. Назвіть найпопулярніші пошукові системи.
7. Що ви розумієте під авторським правом?
8. Чи поширюється авторське право на матеріали в інтернеті?



Вправа 10

■ Знайти в інтернеті файли і зберегти на комп'ютері.
Під час роботи за комп'ютером дотримуйтесь правил безпеки.

- 1) Запустіть браузер. За допомогою пошукової системи www.google.com або іншої знайдіть пісню-щедрівку у файлі формату MP3.
- 2) Завантажте пісню і збережіть, не змінюючи назви файлу, в папку Музика у власну папку. Запишіть адресу сайту, з якого завантажено файл.
- 3) Прослухайте пісню з папки Музика за допомогою навушників.
- 4)  Перейдіть на сайт дитячого графічного редактора Tux Paint (tuxpaint.org), а потім — на сторінку завантаження програми. Запишіть назви чотирьох операційних систем, для яких розробляється Tux Paint.
- 5)  Завантажте програму для операційної системи, встановленої на вашому комп'ютері (або іншої, зазначеної вчителем / учителькою).
- 6)  Відшукайте в інтернеті текст пісні, яку знайшли раніше (див. п. 1–3), і збережіть у текстовому файлі з назвою Щедрівка.



Комп'ютерне тестування

Виконайте тестове завдання 10 з автоматичною перевіркою результату.





Практична робота 2

Пошук інформації в інтернеті

- Завдання:** навчитися шукати інформацію в інтернеті, засвоїти основні правила безпечної роботи в інтернеті.
- Обладнання:** комп'ютер, приєднаний до інтернету.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть браузер. Перейдіть на сайт пошукової системи. Запишіть назву та адресу сайту.
2. Відшукайте в інтернеті текст української казки (наприклад, «Колобок», «Рукавичка», «Ріпка» тощо).
3. Запустіть текстовий процесор, створіть новий документ і скопіюйте текст казки. Запишіть адресу сайту.
4. Збережіть текстовий документ у власній папці. Запишіть, яка назва файлу підійде найкраще.
5. Відшукайте в інтернеті дві ілюстрації до казки й збережіть у власній папці. Запишіть назви сайтів.
6. Перейдіть на сайт <https://казка.укр/> і завантажте із шуканою казкою (або іншою). Перегляньте цей файл.

Зробіть висновок: як шукати в інтернеті потрібні інформаційні матеріали та зберігати їх на комп'ютері.

РОЗДІЛ 3

КОМП'ЮТЕРНІ ПРЕЗЕНТАЦІЇ



§ 11. Створення та опрацювання комп'ютерних презентацій

§ 12. Оформлення комп'ютерних презентацій

§ 13. Виступи з показом презентації

Практична робота 3. Проектування та розробка власної презентації

ПОВТОРЮЄМО



У попередніх класах ви навчилися працювати з програмою для створення й опрацювання комп'ютерних презентацій — PowerPoint із пакета Microsoft Office (або Impress із пакета Libre Office тощо).

Ви знаєте, що *комп'ютерна презентація* — це електронний документ, який складається з окремих слайдів, призначених для почергового показу на екрані монітора, а за наявності проєктора — на великому екрані.

Як вам відомо, створенню презентації передуює *підготовча робота*: продумування мети та сценарію виступу, обмірковування структури презентації тощо. Ви вмієте додавати слайди, вставляти на них тексти, графічні зображення тощо.

1. Назвіть програми для створення презентацій.
2. Яке призначення комп'ютерної презентації?
3. Із чого складається презентація?
4. Назвіть пристрої для демонстрації презентацій.
5. Що передуює створенню презентації?
6. Об'єкти яких типів може містити слайд?



У цьому розділі ви ознайомитеся з особливостями створення та оформлення комп'ютерних презентацій. А також дізнаєтесь, як правильно підготуватись до виступу, що супроводжується показом слайдової презентації.

§ 11. Створення та опрацювання комп'ютерних презентацій

Чи доводилось вам розповідати друзям і подругам про літній відпочинок, домашнього улюбленця, своє захоплення? Напевно, ви не обмежувалися розповіддю, а показували фотографії, малюнки тощо (рис. 11.1).



Рис. 11.1

Поняття презентації

Згадаємо, що презентація (від латин. *praesento* — представлення, подання) — це сукупність дій (виступів, доповідей тощо) і документів (фотографій, плакатів, схем, аудіо- й відеозаписів, текстів тощо), призначених для донесення до аудиторії певної інформації (про ідею, проєкт, результат роботи, товар, винахід та ін.).

Сьогодні більшість презентацій замість традиційних паперових засобів наочності використовують можливості сучасної комп'ютерної техніки: цифрові фотографії, відео- та аудіодані та ін.



Комп'ютерна презентація — це набір електронних документів, підготовлених для перегляду на екрані монітора або на великому проєкційному чи іншому екрані.

Знайомство із системами опрацювання презентацій

Для створення і представлення комп'ютерних презентацій необхідні певні технічні та програмні засоби.



Прикладне програмне забезпечення для створення й редагування презентацій на комп'ютері називають **системами опрацювання комп'ютерних презентацій**.

Яке апаратне забезпечення потрібне для демонстрації презентації? Для аудиторного показу апаратне (технічне) забезпечення демонстрації презентацій зазвичай складається із комп'ютера, проектора, екрана та акустичних колонок (рис. 11.2).



Рис. 11.2



Кілька осіб можуть переглянути презентацію на моніторі стаціонарного комп'ютера чи ноутбука, навіть у смартфоні (рис. 11.3).



Рис. 11.3



Яке програмне забезпечення потрібне для демонстрації презентації? Для створення презентації на комп'ютері слід встановити відповідні програми. Нині їх перелік дуже великий, і можна вибрати програмні засоби, що дозволяють розкрити задум презентації.


Знайомство з PowerPoint

Відомий вам редактор презентацій Microsoft PowerPoint (рис. 11.4) входить до складу комерційного пакета Microsoft Office для Windows і призначений для роботи зі слайдами. Доступна і безкоштовна вебверсія цієї програми є у складі пакета програм Office 365.



Рис. 11.4

► Запуск PowerPoint

Пригадаємо, що редактор презентацій PowerPoint можна запустити з головного меню Пуск або клацанням значка .

Для запуску PowerPoint в ОС Windows 7 потрібно клацнути: Пуск → Всі програми → Microsoft Office → Microsoft PowerPoint (рис. 11.5). Аналогічно запускається PowerPoint і в ОС Windows 10 (рис. 11.6).

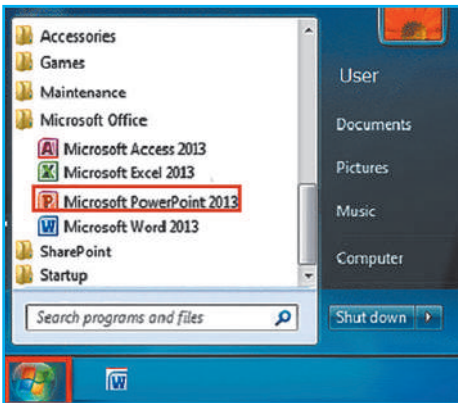


Рис. 11.5

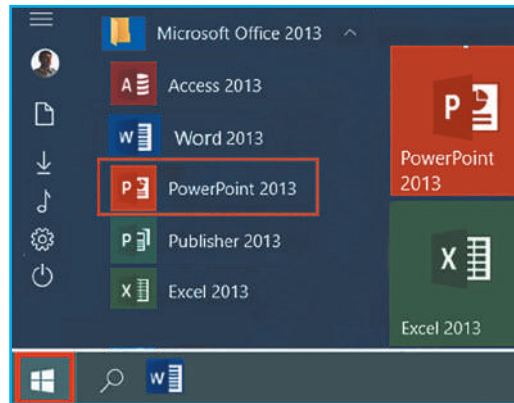


Рис. 11.6

Для швидкого запуску програми PowerPoint її ярлики розміщують безпосередньо в меню Пуск, на панелі завдань, або на робочому столі командами контекстного меню програми.

Команди у PowerPoint від 2010 до 2019 року випуску різняться несуттєво. Навчившись користуватись PowerPoint 2013, ви зорієнтуватиметесь і в решті версій цієї програми.

► Створення презентації

Після запуску програми на екрані з'являється вікно, у якому можна вибрати потрібний шаблон (рис. 11.7) з інтернету, створити нову презентацію або відкрити один з останніх збережених файлів комп'ютерної презентації.

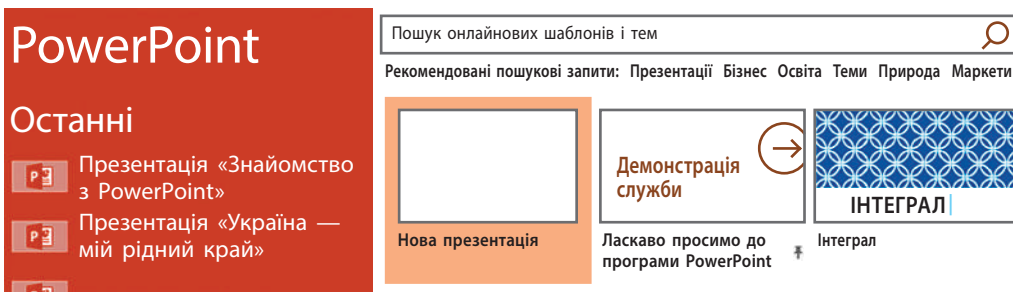


Рис. 11.7

Створення або редагування готової комп'ютерної презентації відбувається в основному вікні програми PowerPoint.

Пригадаємо основні об'єкти вікна PowerPoint (рис 11.8):

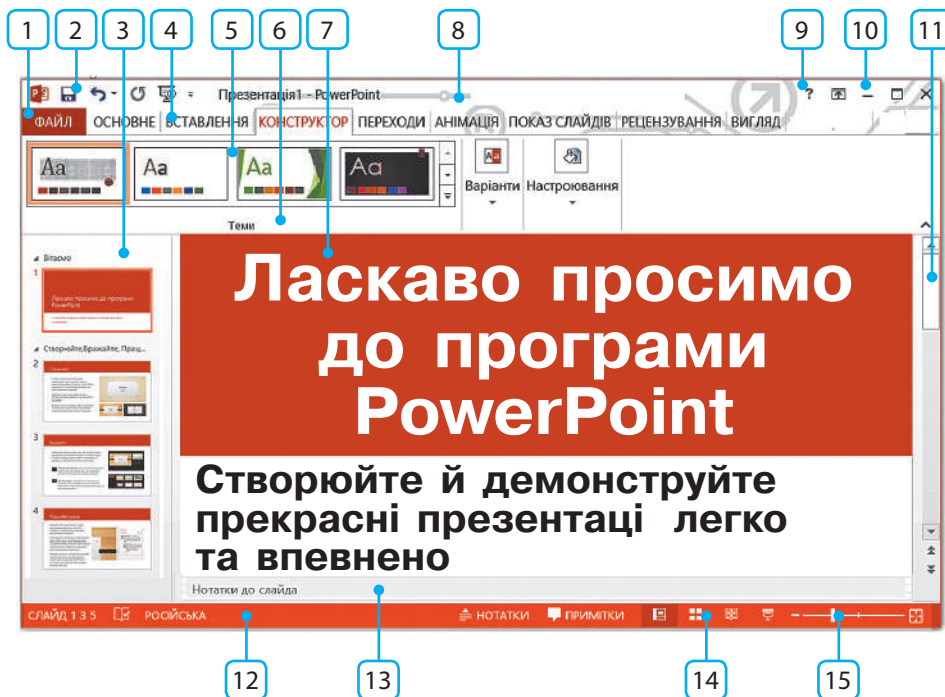


Рис. 11.8

1 — кнопка Файл; 2 — панель швидкого доступу; 3 — панель перегляду слайдів; 4 — меню вкладок; 5 — стрічка інструментів; 6 — групи інструментів; 7 — область відображення слайду;

8 — рядок заголовка; 9 — кнопка довідки; 10 — кнопки керування вікном; 11 — смуга прокручування; 12 — рядок стану; 13 — область нотаток; 14 — кнопки режиму перегляду презентації; 15 — інструменти масштабування.

► Робота зі слайдами

Основним об'єктом презентації є **слайд**. Розрізняють *титульний* і *звичайний* слайди.

Якщо після запуску PowerPoint обрати команду Нова презентація, то в області відображення слайду побачимо порожній титульний слайд з місцем для заголовка і підзаголовка (рис. 11.9).

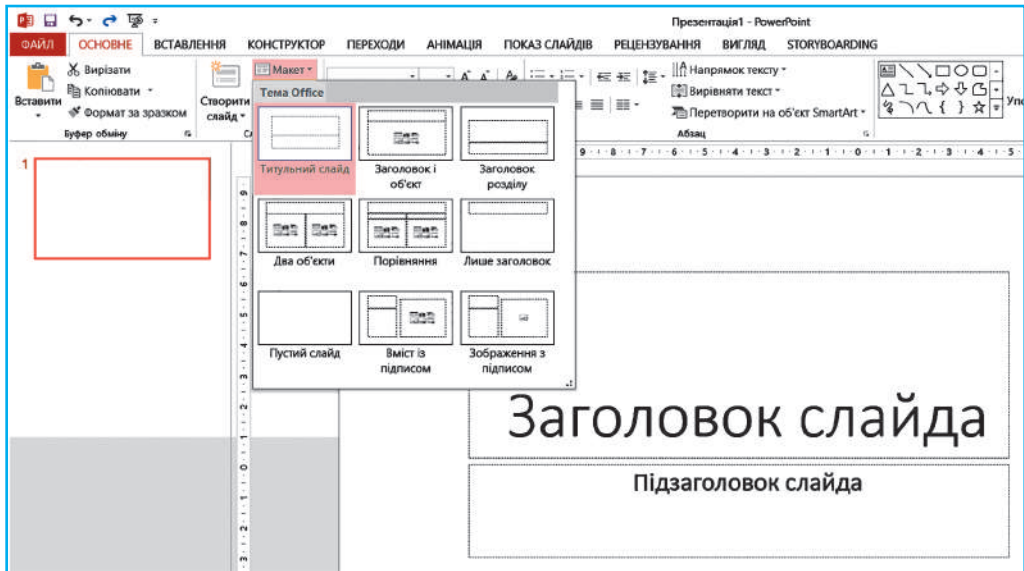


Рис 11.9


Слайди презентації можна виділяти, вставляти, копіювати, вилучати, перетягувати тощо.


Для **додавання слайда до презентації** можна скористатися кнопкою Створити слайд або вибрати потрібний макет слайду у списку Макет (див. рис. 11.7).

Порядок розташування слайдів можна змінити перетягуванням, як окремих, так і групи, на панелі перегляду слайдів.

► Збереження презентації

Як і інші офісні документи, презентацію для подальшого використання слід **зберегти**. Для цього потрібно виконати такі дії:

- 1) на панелі швидкого доступу клацнути значок Зберегти або на вкладці ФАЙЛ вибрати команду Зберегти (або Зберегти як);
- 2) вказати місце збереження в одну з наявних у списку папок, або вибрати іншу папку, скориставшись кнопкою  ;
- 3) ввести назву файлу у поле Ім'я файлу і натиснути кнопку Зберегти (для збереження наявного документа в іншу папку та / або з іншим іменем слід вибрати команду Зберегти як...).

У режим редагування можна повернутися з будь-якого кроку збереження, натиснувши кнопку .

Навчившись створювати власні презентації, ви зможете зацікавити будь-яку аудиторію своїми проектами, подорожами, захопленнями тощо.

- Найбільш професійними вважаються презентації Стіва Джобса (рис. 11.10) — засновника всесвітньо відомої корпорації Apple (США). Його виступи з використанням презентацій захоплювали своєю продуманістю, якістю та простотою.



Рис. 11.10



Питання для самоперевірки


1. Чим звичайна презентація відрізняється від комп'ютерної?
2. Яке апаратне забезпечення необхідне для демонстрації презентації?
3. Назвіть програму для створення і редагування презентації.
4. Як запустити редактор презентацій PowerPoint?
5. Як створити нову презентацію і зберегти її на комп'ютері?
6. Як змінити порядок розташування слайдів у презентації?



Вправа 11

- Виконайте завдання за комп'ютером.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

- 1) Запустіть PowerPoint і створіть нову презентацію. Кнопкою  на вкладці ОСНОВНЕ створіть чотири слайди майбутньої презентації.
- 2) Придумайте назву презентації (наприклад, «Мій котик Мурчик») і її вміст (наприклад, ви бажаєте розповісти про те, як Мурчик розважається і що їсть).
- 3) На *першому* слайді додайте до заголовка назву презентації (наприклад, «Мій котик Мурчик»). У тексті слайда опишіть у підзаголовку, хто є автором / авторкою презентації, наприклад «презентація учениці 5-го класу Сумлінної Марійки».
- 4) На *другому* слайді додайте до заголовка назву слайда (наприклад, «Розваги Мурчика»). У тексті стисло опишіть розваги Мурчика (наприклад, «Мурчик полюбляє гратися із м'ячем та клубком ниток»).
- 5) На *третьому* слайді додайте до заголовка назву слайда, (наприклад, «Раціон Мурчика»). У тексті коротко опишіть, чим полюбляє ласувати Мурчик (наприклад, ковбаса; сметана).
- 6) На *четвертому* слайді вмістіть побажання аудиторії. Наприклад, Заголовок: «Дякуємо за увагу». Текст слайда: «Я і мій кіт Мурчик бажаємо вам гарного дня». Перегляньте презентацію, відкоригуйте її зміст. Збережіть файл у форматі PPTX із назвою Вправа 11.



Комп'ютерне тестування

Виконайте тестове завдання 11 з автоматичною перевіркою результату.



§ 12. Оформлення комп'ютерних презентацій

Пригадаємо, що робота над презентацією складається з кількох етапів: планування, розроблення структури, добирання тексту й малюнків, створення власне презентації, підготовка виступу та виступ.

Слайди презентації можуть містити різні об'єкти (рис. 12.1).



Рис. 12.1

Більшість слайдів містять заголовок слайда і текст слайда. Додавати інші текстові об'єкти до слайдів можна за допомогою інструментів групи Текст вкладки Вставлення. Основні засоби для додавання графічних об'єктів розташовані в групі Зображення вкладки Вставлення.

Об'єкти слайда можна редагувати, тобто виправляти помилки, вносити зміни до тексту, та форматувати — змінювати зовнішній вигляд об'єкта.

Шаблони презентацій

Комп'ютерну презентацію і її слайди кожний автор / авторка оформлюють залежно від теми, змісту, власних уподобань тощо.

Для прискорення роботи можна скористатися готовими проєктами презентацій — так званими шаблонами.



Шаблон презентації — це схема (проект) презентації з одного чи кількох слайдів, які оформлені з дотриманням певного задуму.

Зазвичай шаблони зберігаються у файлах з розширенням .potx. Це дає можливість створювати на основі шаблонів готові презентації і зберігати їх в інших форматах без пошкодження вихідного шаблону для повторного використання. Дуже часто як шаблони використовують презентацію у форматі PPTX.

Потрібний шаблон можна відшукати у вікні PowerPoint. Таку можливість надає програма одразу після її запуску (рис. 12.1).

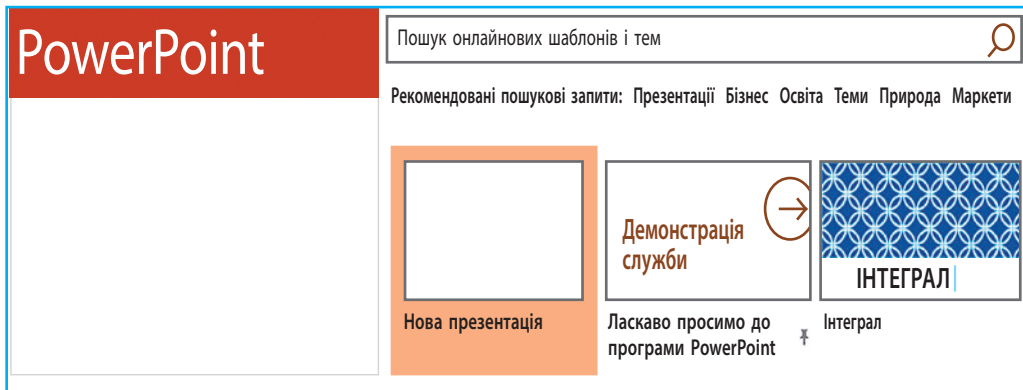


Рис. 12.1

Шаблони і навіть готові презентації різного оформлення та призначення можна безкоштовно завантажити з таких сайтів:

- теми і шаблони Office: templates.office.com;
- безкоштовні шаблони для PowerPoint: free-power-point-templates.com;
- безкоштовні фони для PPT: freerpptbackgrounds.net;
- шаблони PowerPoint: indezine.com;
- шаблони презентацій PPT: powerpointbase.com; powerpointstore.com;
- безкоштовні шаблони PowerPoint: slidehunter.com.



Макет слайда

На слайдах презентації можуть використовуватись різні матеріали. Не завжди вдається швидко дібрати вдалий варіант розміщення. Розв'язати цю проблему допоможе використання готових макетів слайдів, вбудованих у PowerPoint.



Макет — це умовна схема слайда, що визначає спосіб розташування об'єктів на слайді.

Макет має свої властивості: макет, формат тла, оформлення слайда.

Щоб створити новий слайд певного макета, потрібно:

- 1) розгорнути список Створити слайд на вкладці ОСНОВНЕ у групі Слайди;
- 2) клацнути вибраний макет (рис 12.2).

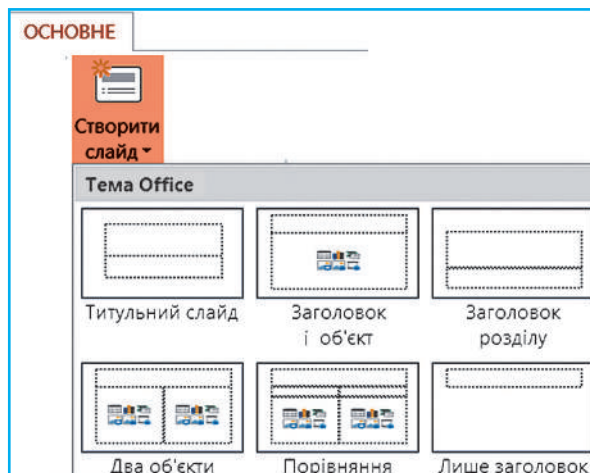


Рис. 12.2

Щоб змінити макет слайдів, потрібно:

- 1) виділити один або декілька слайдів;
- 2) розкрити на вкладці Основне в групі Слайди список Макет;
- 3) клацнути один зі зразків макета.

В області відображення слайда з'являться ділянки, обведені прямокутником, так звані *контейнери* (рис. 12.3).

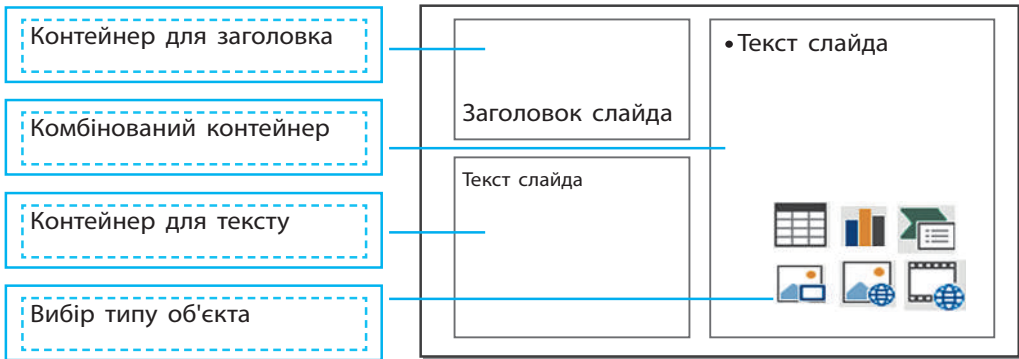
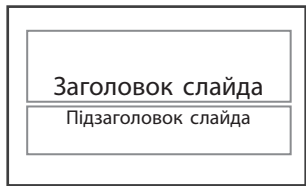


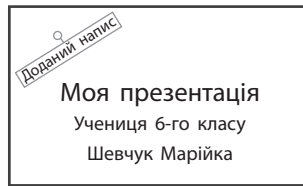
Рис. 12.3

Контейнери різних типів призначені для розміщення в них тих чи інших об'єктів слайда: заголовка, тексту, графічних та інших об'єктів для можливості автоматичного їхнього форматування під час зміни макета та/або оформлення слайда.

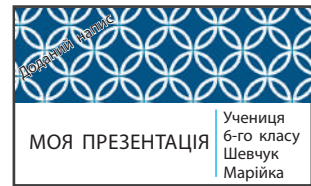
Звичайно, на слайд можна додавати власні об'єкти (написи, малюнки тощо). Пам'ятайте, що об'єкти, розміщені поза контейнерами, не форматовуватимуться під час автоматичного оформлення слайдів засобами PowerPoint (рис. 12.4).



Порожній макет слайда



Заповнений слайд із доданим написом



Застосування автоматичного шаблону. Доданий поза контейнером напис «загубився»

Рис. 12.4



Кожен тип контейнерів має налаштовані параметри автоматичного форматування вбудованих об'єктів, що дозволяє уникнути невдалих варіантів оформлення слайдів.


Для введення тексту в контейнер будь-якого типу потрібно клацнути у вільному місці всередині нього та ввести текст після появи курсора.

Контейнер можна вилучити, клацнувши прямокутну рамку, а потім натиснувши клавішу Delete.

Контейнер для заголовка вилучати небажано, оскільки пізніше, при налаштуванні гіперпосилань (про це далі), заголовок слайда буде присутній у списку слайдів, що значно полегшує роботу.

Комбінований контейнер дозволяє розміщувати дані різних типів. Для введення тексту достатньо клацнути на вільному місці контейнера і розпочати набирати текст.

Піктограми вибору типу об'єкта розташовані в центрі контейнера, клацання по них дозволяє додати об'єкти різних типів:

Піктограма	Назва	Дія
	Додати таблицю	Відкривається вікно для введення числа рядків і стовпчиків
	Вставлення діаграми	Відкривається вікно Вставка діаграми
	Додавання рисунка SmartArt	Додає графічний об'єкт для наочного подання даних
	Рисунки	Відкривається стандартне вікно вибору файлу
	Онлайнові зображення	Відкривається вікно пошуку зображень в інтернеті
	Вставлення відео	Відкривається вікно вибору відео з локального комп'ютера, з YouTube тощо

Щоб додати новий об'єкт до контейнера, з нього необхідно заздалегідь вилучити об'єкт, який був вставлений раніше.

У 1984 році Р. Гаскінс разом із Д. Остіном створив програму Presenter, а Остін і Т. Рудкін створили оригінальну версію програми. Пізніше Гаскінс змінив ім'я програми на PowerPoint.

Теми оформлення слайдів

У Word значну частину роботи допомагають автоматизувати стилі. Подібні засоби є і в програмі PowerPoint. Шаблони презентацій можуть мати різні стилі оформлення слайдів, які називаються темами.



Тема слайда (презентації) — це іменований стиль оформлення об'єктів на слайді: тла, шрифтів, графічних об'єктів тощо.

На вкладці Конструктор доступна стандартна колекція тем оформлення слайдів (рис. 12.5) — наборів властивостей об'єктів.

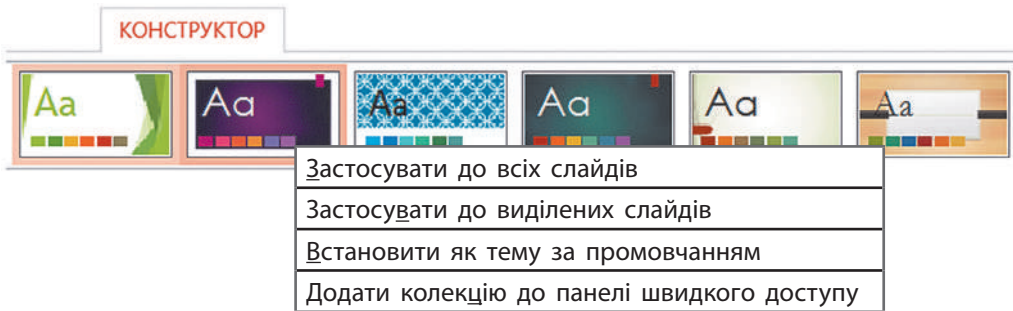


Рис. 12.5

Піктограми в цій колекції ілюструють тло слайда, колірну гаму та стиль тексту кожної з тем оформлення. Достатньо навести вказівник на піктограму, щоб в області відображення слайда одразу побачити відповідне його оформлення.

Якщо ж клацнути одну з піктограм, то всі слайди презентації набудуть обраної теми (стилю) оформлення. Для зміни теми оформлення окремих слайдів слід відкрити контекстне меню обраної теми і вибрати команду Застосувати до виділених слайдів.

► Налаштування тем

Кожна тема презентації чи слайда може бути налаштована добром уподобаних варіантів колірної гами слайда, заповнення тла, шрифту тощо. Для цього потрібно на вкладці КОНСТРУКТОР розгорнути список групи Варіанти та обрати потрібне (рис. 12.6).

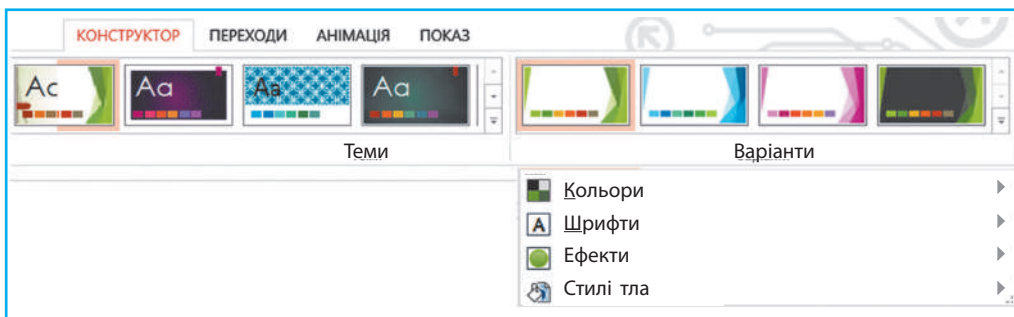


Рис. 12.6

Одна й та сама тема оформлення слайда (наприклад, Ретроспектива) може мати різний вигляд залежно від налаштувань (рис. 12.7).

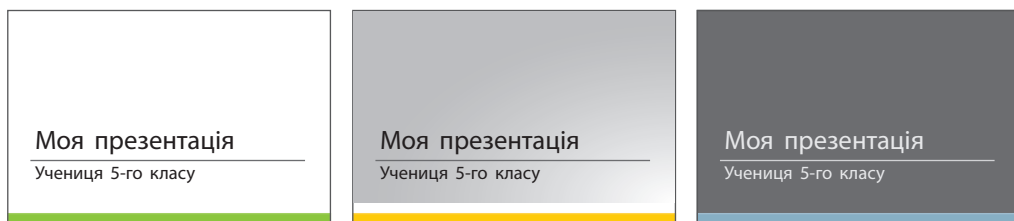


Рис. 12.7



Питання для самоперевірки

1. Що таке шаблон презентації?
2. Де можна знайти і завантажити шаблони презентації?
3. Як обрати готові шаблони презентації?
4. Що визначає макет слайда?
5. Як змінити макет створеного слайда?
6. Яке призначення контейнерів слайда?
7. Чому не варто додавати до слайда об'єкти поза контейнерами?
8. Які об'єкти може містити Комбінований контейнер?
9. Чому небажано вилучати Контейнер для заголовка?
10. Що таке тема слайда?



Вправа 12

- Налаштувати загальний дизайн презентації та додати графічні об'єкти.


Під час роботи за комп'ютером дотримуйтесь правил безпеки.

- 1) Відкрийте файл Вправа 11. Обміркуйте, які зображення варто додати, до яких слайдів і скільки. Наприклад, до *другого* слайда («Розваги Мурчика») доцільно додати фото котика, що грається з м'ячем чи клубком ниток, а до *третього* слайда — зображення двох його улюблених страв тощо (рис. 12.8).



Рис. 12.8

- 2) Змініть макет *другого* слайда на такий, що містить: Заголовок, один Контейнер для тексту і один Комбінований контейнер (назва макета: Вміст із підписом). У *другому* слайді перенесіть текст із Комбінованого контейнера до Контейнера для тексту. Комбінований контейнер має стати порожнім.
- 3) До Комбінованого контейнера *другого* слайда додайте фото чи малюнок котика, скориставшись піктограмами

або . Під час онлайн-пошуку варто ввести до рядка пошуку ключові слова котик, клубок.

- 4) Змініть макет *третього* слайда на такий, що містить Заголовок, два Контейнери для тексту та два Комбіновані контейнери (назва макета: Порівняння).

У *третьому* слайді перенесіть тексти із Комбінованого контейнера до двох Контейнерів для тексту.

Комбіновані контейнери мають стати порожніми.

- 5) До кожного з Комбінованих контейнерів *третього* слайда додайте фото чи малюнок відповідної страви, скориставшись піктограмами



- 6) Застосуйте до *всіх* слайдів презентації одну з тем оформлення, що на вкладці Конструктор.

Задайте однакове налаштування відтінків кольору вибраної теми на *першому* й *останньому* слайдах.

Збережіть файл презентації у форматі PPTX із назвою Вправа 12.



Комп'ютерне тестування

Виконайте тестове завдання 12 з автоматичною перевіркою результату.









§ 13. Виступи з показом презентації

Для демонстрації комп'ютерних презентацій існує кілька режимів показу: звичайний, режим доповідача, довільний показ.

Звичайний режим

Вам уже відомо, що повноекранну демонстрацію слайдів презентації у звичайному режимі здійснюють кнопками із початку (клавіша F5) або із поточного слайда (сполучення клавіш Shift + F5), що містяться на вкладці ПОКАЗ СЛАЙДІВ.

Якщо під час звичайного режиму показу презентації навести вказівник миші на лівий нижній куток слайда, то з'являться приховані елементи керування показом презентації:

Піктограма	Назва
	Повернення до попереднього слайда (ефекту)
	Перехід до наступного слайда (ефекту)
	Використання вказівника, маркера або пера
	Відображення піктограм слайдів для обрання показуваного
	Масштабування показу окремої частини слайда
	Інші елементи: режим доповідача, режим екрана тощо

Деякі доповідачі та доповідачки демонструють свої комп'ютерні презентації в режимі редагування, щоб мати змогу керувати показом, вибираючи потрібні слайди з Області перегляду слайдів. Для аудиторного показу презентацій існує зручний Режим доповідача.

Режим доповідача

Для особи, яка доповідає, важливо не розгубитися під час демонстрування презентації. Тут у пригоді стане можливість додавання прихованих нотаток до слайдів.

► Додавання нотаток

Увімкнення й вимкнення відображення області нотаток у режимі редагування здійснюється кнопкою Нотатки (рис. 13.1). Вона міститься на рожевому Рядку стану внизу вікна PowerPoint.



Рис. 13.1

Для створення нотаток доповідачу достатньо в область нотаток увести потрібний текст.

► Увімкнення режиму доповідача

Під час показу презентації в режимі доповідача Нотатки відображаються на моніторі доповідача / доповідачки, але не відображаються для слухацької аудиторії на проєкційному екрані.



Увімкнення показу презентації в режимі доповідача здійснюється сполученням клавіш **Alt + F5** (з початку) або **Alt + Shift + F5** (з поточного слайда).

Таким чином, можна, не засмічуючи слайди текстами, читати текст доповіді з екрана монітора (рис 13.2).

Зображення на екрані монітора

Зображення на проєкційному екрані



Рис. 13.2

Гіперпосилання

До складних багатослайдових презентацій зручно додавати слайди зі змістом презентації з можливістю автоматичного переходу на потрібний слайд. Організація таких переходів називається посиланням (або гіперпосиланням).



Гіперпосилання (або **посилання**) — це виділений кольором або іншим чином текст, зображення чи кнопка, клацання яких викликає перехід на перегляд інших даних.

Гіперпосилання дають змогу під час перегляду презентації переходити до певних її частин (розділів, слайдів) або на вебсторінки, відкривати інші документи.

Гіперпосилання можна додати як до текстового, так і до графічного об'єкта.

Щоб створити гіперпосилання, потрібно:

- 1) виділити фрагмент тексту або малюнок;
- 2) вибрати команду Гіперпосилання (рис. 13.3), що міститься на вкладці ВСТАВЛЕННЯ в групі Посилання;
- 3) зазначити у вікні, що з'явиться, з чим буде пов'язане гіперпосилання:

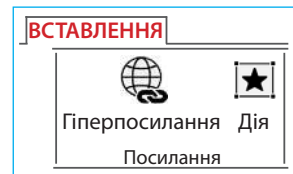


Рис. 13.3

- з файлом або вебсторінкою;
- місцем у документі; новим документом;
- електронною поштою;

4) клацнути кнопку ОК.

Щоб додати посилання на певний слайд, потрібно вибрати варіант місцем у документі. У нашому прикладі у вікні (рис. 13.4) для *першого* пункту слайда ЗМІСТ слід вибрати назву *третього* слайда, який відкриватиметься після клацання посилання.

Якщо потрібно, щоб після перегляду довільного показу презентація продовжувалася зі слайда, на якому клацнули гіперпосилання (у нас — повернутися до слайда ЗМІСТ), необхідно встановити прапорець Показати та повернутися (рис 13.4).

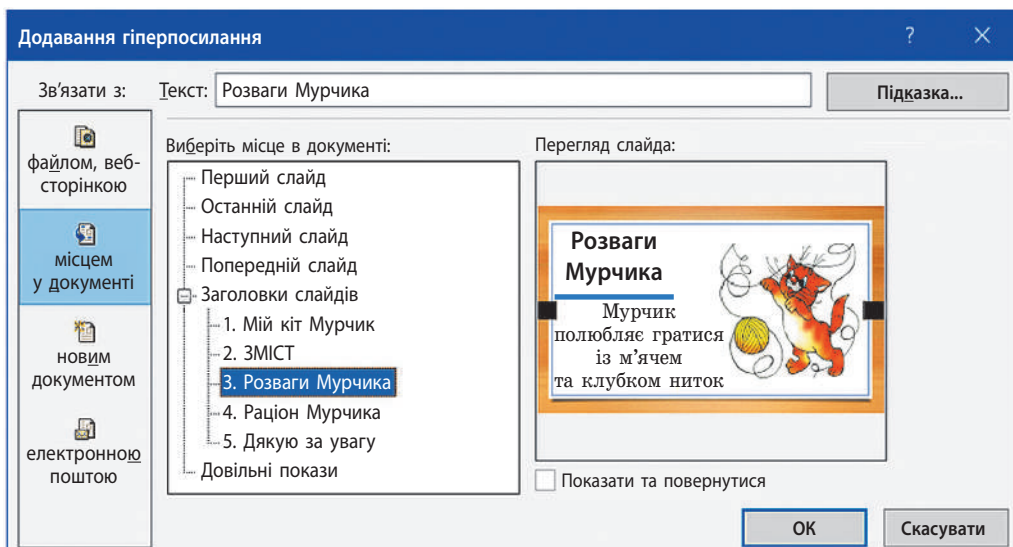


Рис. 13.4

► Кнопки дій

Щоб із будь-якого слайда мати можливість повернутися на слайд змісту, можна на цих слайдах створити кнопку дії за алгоритмом: ВСТАВЛЕННЯ → Фігури → Кнопки дії → Кнопка дії: Додому.



Кнопки дій — це низка фігур, що зображують кнопки, для яких вже призначено одну із часто вживаних дій.

Наприклад, для кнопки дій Додому за замовчуванням призначено перехід до першого слайда після клацання миші. Нам же потрібно до цієї кнопки додати гіперпосилання на слайд ЗМІСТ, який заздалегідь було додано до презентації.



Якщо налаштовану певним чином кнопку (або інший об'єкт) скопіювати (Ctrl + C) і вставити (Ctrl + V) до іншого слайда, то зберігаються її розміри, відносно місце розташування на слайді і гіперпосилання.

Налаштування, притаманні кнопкам дій, можна застосувати до будь-яких об'єктів слайда: фрагментів тексту, графічних об'єктів тощо.

Довільні покази слайдів

Розглянуті приклади показу презентацій мали або лінійну структуру (при перегляді слайди виводилися на екран послідовно один за одним), або показ слайдів проводився вибірково в ручному режимі чи за допомогою гіперпосилань.

Для налаштування попередньо спланованого вибіркового показу слайдів в автоматичному режимі їх слід згрупувати, створюючи довільні покази.



Довільний показ — це іменована послідовність запланованого показу слайдів презентації.

Довільний показ застосовують, якщо потрібно налаштувати показ слайдів не в тій послідовності чи не в тій кількості, яка є у файлі презентації. При цьому зміст і порядок розташування слайдів у самій презентації залишаються незмінними.

Слайд може входити одночасно в кілька *довільних показів*, демонструватися кілька разів в одному *довільному показі* або не демонструватися зовсім.



У презентації щодо захисту проєкту можуть бути створені довільні покази (рис. 13.5) «Тварини», «Дерева», «Хвойні» та ін.

Довільний показ додають вже після створення презентації. Слайди, додані пізніше, не відтворюватимуться автоматично.

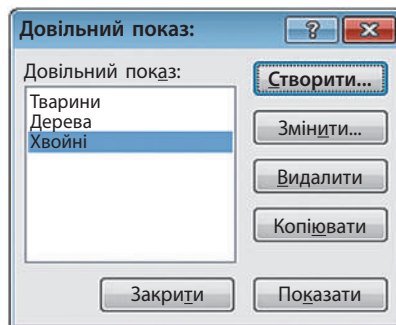


Рис. 13.5

Щоб перейти до створення довільного показу, потрібно:

- 1) на вкладці Показ слайдів у групі Розпочати показ слайдів натиснути кнопку Настроюваний показ слайдів і вибрати команду Довільний показ;
- 2) у діалоговому вікні Довільний показ (див. рис. 13.5) натиснути кнопку Створити;
- 3) зазначити назву показу в наступному вікні (рис. 13.6), а також сформувати список слайдів у тому порядку, в якому вони демонструватимуться під час довільного показу;

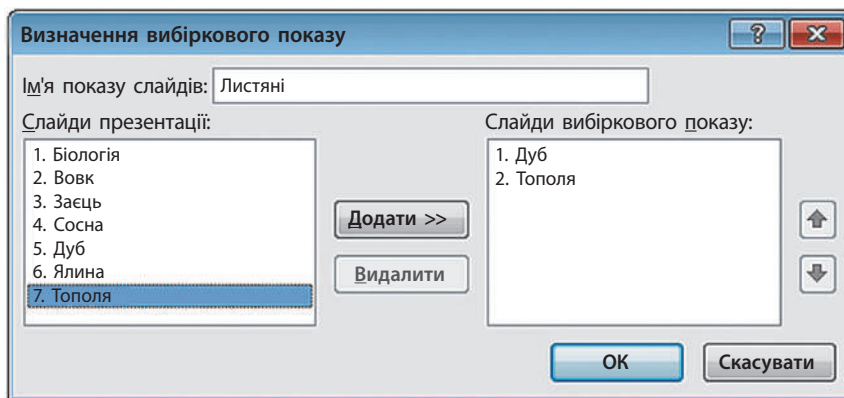


Рис. 13.6

- 4) додати назву слайда до списку вибраних слайдів кнопкою Додати (див. рис. 13.6). Під час натискання кнопки Видалити зі списку вилучається назва слайда.

Кнопки зі стрілками, що містяться в ділянці вікна справа (див. рис. 13.6), дозволяють змінювати порядок слайдів у довільному показі. При цьому порядок слайдів у презентації не змінюється.

Керування показом презентації

Під час планування розробки презентації необхідно урахувати, як буде здійснюватися керування переглядом.

Існують кілька варіантів **керування показом** презентації:

- *автоматичний* (наприклад, на рекламному екрані на вулиці), при якому зміна слайдів відбувається через певний проміжок часу, заданий під час розробки;
- *керований доповідачем* (повноекранний), за якого доповідач / доповідачка змінює слайди, користуючись пультом дистанційного керування, мишею чи режимом доповідача, зображеним на іншому моніторі;
- *керований користувачем* (віконний), за якого презентація виводиться у вікні, користувач / користувачка змінює слайди за допомогою смуги прокрутки.

Ви можете вибрати один із варіантів, уточнити додаткові параметри у вікні, що відкривається командою Показ слайдів → Настроювання показу слайдів.

Друк презентації

Може виникнути потреба в роздрукуванні презентації. Список для вибору режиму друку міститься у вікні Друк, яке відкривається командою Файл → Друк.

Залежно від потреби можна роздрукувати:

- *слайди* — друкується по одному слайду на аркуш;
- *видачі* — друкується один слайд (або більше) на аркуш з датою і номером сторінки;
- *нотатки* — друкується один слайд на аркуш і додані до нього нотатки;
- *структуру* — друкується структура презентації у вигляді багаторівневого списку.

Критерії оцінювання презентації

Якість створеної презентації можна оцінити за певними критеріями (рис. 13.7).

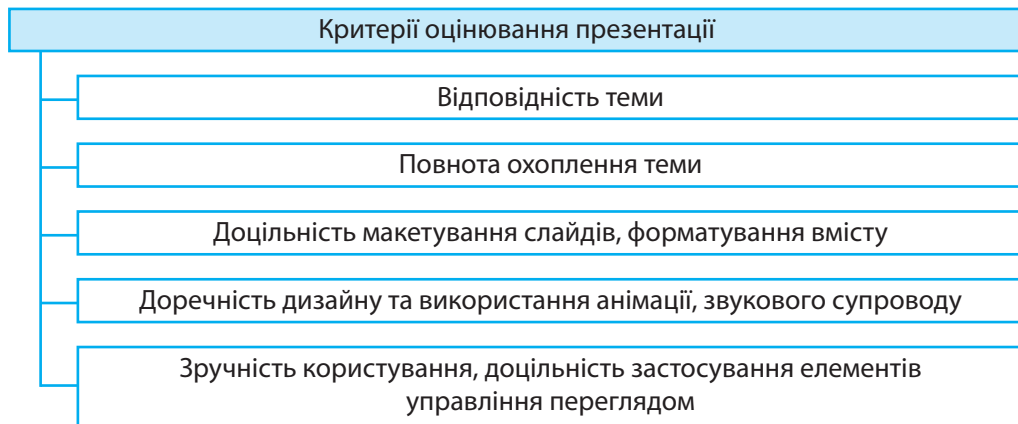


Рис. 13.7



Питання для самоперевірки

1. Як запустити звичайну повноекранну демонстрацію презентації з *першого* слайда; з *поточного* слайда?
2. Як відобразити приховані елементи керування під час показу презентації?
3. Для чого потрібні нотатки? Як їх додати до слайда?
4. Які переваги режиму доповідача та як його увімкнути?
5. Що таке гіперпосилання?
6. Як додати гіперпосилання на слайді до фрагмента тексту або графічного об'єкта?



Вправа 13

- Додати гіперпосилання та налаштувати перегляд презентації.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

- 1) Відкрийте файл презентації Вправа 12. Додайте до презентації другий слайд із макетом: Заголовок і об'єкт.
- 2) Додайте до заголовка *другого* слайда слово Зміст. Додайте два рядки тексту, що є заголовками *третього* та *четвертого* слайдів. Наприклад, «Розваги Мурчика» та «Раціон Мурчика» (рис. 13.7). Додайте до кожного з них гіперпосилання на слайди презентації.

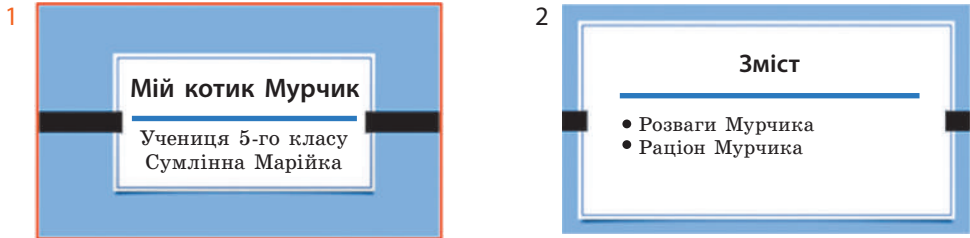


Рис. 13.7

- 3) На *третьому* слайді додайте кнопку дії: Додому. Розташуйте кнопку дії у правому нижньому кутку слайда і налаштуйте її розміри. Додайте до кнопки дію на перехід до слайда Зміст при клацанні мишею. Скопіюйте кнопку дії на третьому слайді («Розваги Мурчика») та додайте її до *четвертого* слайда («Раціон Мурчика»).



- 4) Запишіть нотатки до *третього* та *четвертого* слайдів і перегляньте презентацію в режимі доповідача.
- 5) Створіть Довільний показ презентації, щоб слайди йшли в такому порядку: *перший, четвертий, третій, п'ятий*.
- 6) Перегляньте презентацію в режимі довільного показу. Збережіть файл у форматі PPTX із назвою Вправа 13.



Комп'ютерне тестування

Виконайте тестове завдання 13 з автоматичною перевіркою результату.





Практична робота 3

Проектування та розробка власної презентації

Завдання: спроектувати та розробити презентацію за визначеними критеріями.

Обладнання: комп'ютер із встановленою програмою PowerPoint.

Хід роботи

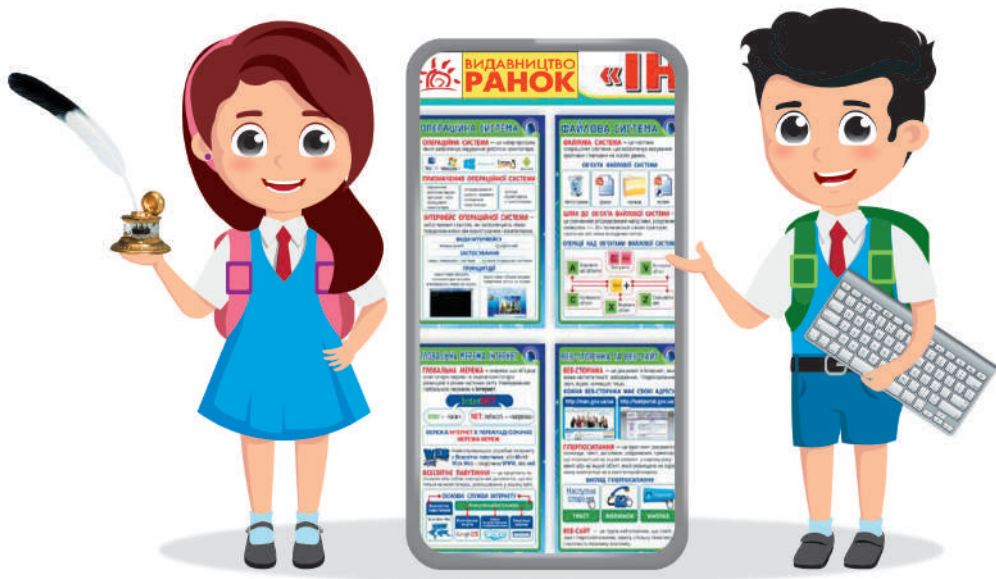
Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запишіть тему презентації про своє захоплення (5 хв).
2. Запишіть текст свого виступу (до 100 слів).
3. Позначте в тексті те, якими елементами презентації супроводжуватиметься ваш виступ.
4. Обміркуйте, яка кількість слайдів вам потрібна для супроводу виступу (не забудьте додати до основних три *традиційні* слайди: Титульний, Слайд змісту, Завершальний).
5. Сплануйте, на яких слайдах і які об'єкти розташовуватимуться, проставте в тексті номери слайдів.
6. Знайдіть, завантажте, збережіть файли із зображеннями.
7. Запустіть PowerPoint та створіть нову порожню презентацію. Створіть заплановану кількість слайдів із відповідними параметрами макета.
8. Додайте до кожного слайда заголовки і тексти.
9. Додайте до слайдів графічні об'єкти.
10. Налаштуйте загальний дизайн презентації.
11. Налаштуйте переходи слайдів.
12. Збережіть файл Практична3.pptx.

Зробіть висновок: для кого призначена створена презентація, в якому режимі її краще демонструвати.

РОЗДІЛ 4

ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ



§ 14. Основні об'єкти текстового документа

§ 15. Робота з фрагментами тексту. Редагування та форматування символів

Практична робота 4. Здійснення форматування символів текстового документа

§ 16. Форматування абзаців

Практична робота 5. Здійснення форматування абзаців текстового документа

Практична робота 6. Створення текстового документа

§ 17. Опрацювання об'єктів текстового документа

ПОВТОРЮЄМО



Ви вже маєте навички роботи з документами в середовищі текстового редактора Блокнот і текстового процесора Word. Ви знаєте, що текстовий документ може містити різні об'єкти, основними з яких є *символ* і *абзац*. Окрім тексту, документ може містити малюнки, таблиці.

Кожен об'єкт має свої властивості. Властивостями символу є шрифт, накреслення, розмір і колір. Абзац має такі властивості, як вирівнювання, міжрядковий інтервал, відступи тощо. Значення властивостей можна змінювати.

Ви умієте виконувати операції над текстовими об'єктами: *редагувати* (видаляти, додавати, переміщувати, копіювати), *форматувати* (встановлювати шрифт, змінювати колір і накреслення символів, розташування абзаців тощо).

1. З яким текстовим процесором ви вже працювали?
2. Назвіть об'єкти текстового документа.
3. Які операції можна виконувати над текстовими документами?
4. Які властивості має символ?
5. Які властивості має абзац?
6. Що таке форматування текстового документа?



Опрацювавши цей розділ, ви навчитесь здійснювати за допомогою текстового процесора редагування та форматування текстового документа, а також додавати в текстовий документ таблиці та рисунки, готувати документ до друку й друкувати його.

§ 14. Основні об'єкти текстового документа

Документ, розроблений за допомогою програм для опрацювання тексту, може містити не лише текст, а й графічні зображення.

Програмне забезпечення для опрацювання текстів

Для підготовки складних текстових документів — листів, статей, рефератів та ін. призначений **текстовий процесор**.

Ви вже знайомі з текстовим процесором Microsoft Word (далі — Word), що входить до складу програмного пакета Microsoft Office. Ознайомимося з ним докладніше.

Існують такі способи запуску Word: за допомогою ярлика на робочому столі, швидкого запуску або відкриття будь-якого документа. У перших двох випадках на екрані з'явиться вікно з порожньою робочою областю, в останньому — в робочій області з'явиться текст відкритого документа.

Розглянемо вікно Word (рис. 14.1).

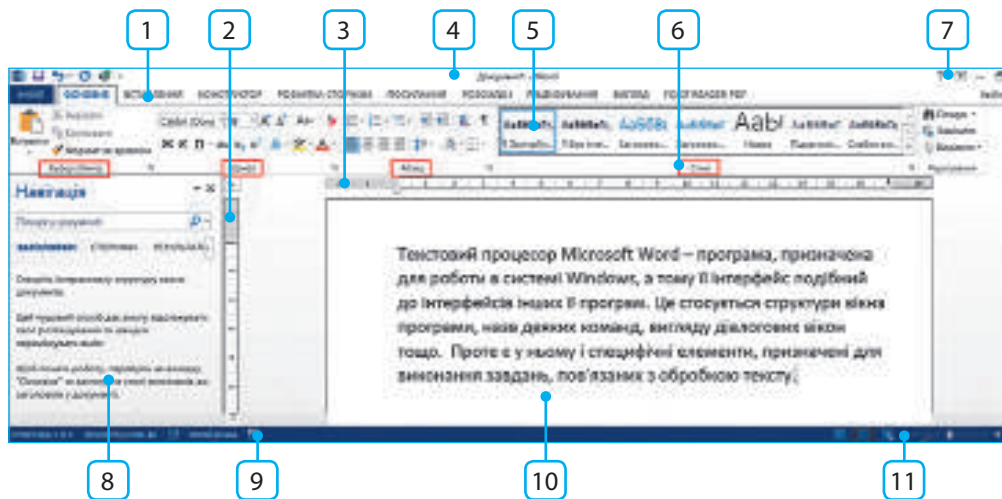


Рис. 14.1

Як бачимо з рис. 14.1, вікно Word містить такі об'єкти: 1 — вкладки; 2, 3 — лінійки; 4 — рядок заголовка; 5 — команди; 6 — групи команд; 7 — кнопки керування вікном; 8 — панель навігації; 9 — рядок стану; 10 — робоча область; 11 — масштаб.

Отже, вікно програми містить рядок заголовка, де зазначено ім'я файлу активного документа (того, з яким у цей час працюють) і назву програми.

Під рядком заголовка розташовані вкладки (ОСНОВНЕ, ВСТАВЛЕННЯ ...), нижче — команди, ще нижче — групи команд (Шрифт, Абзац ...). Більшу частину вікна займає робоча область, у якій міститься текстовий документ.

Ліворуч і вгорі робочої області розміщені лінійки, за допомогою яких можна визначати та встановлювати розміри об'єктів документа, а праворуч і знизу — смуги прокручування для перегляду частин документа, яких не видно в робочій області.

Нижче від робочої області розташовано рядок стану, в якому подається корисна інформація та містяться кнопки для перемикання режимів роботи з програмою.

Об'єкти текстового документа

Розгляньте сторінки свого підручника з інформатики. Ви побачите на них основні об'єкти текстового документа (рис. 14.2).



Рис. 14.2

Символ — літера, цифра, знак пунктуації, знак арифметичної операції чи спеціальний знак.

Слово — послідовність символів, відділена від інших символів пропуском.

Речення — розділена пробілами послідовність слів, яка починається з великої літери та закінчується крапкою, знаком питання, знаком оклику або трьома крапками.

Абзац — частина тексту, набирання якого закінчується натисканням клавіші **Enter**. Абзац може займати один або кілька рядків.

Найменшим об'єктом текстового документа є символ. Групи символів утворюють слова, зі слів складаються речення, які групуються в абзаци (на рис. 14.3 фрагмент з казки про ріпку).

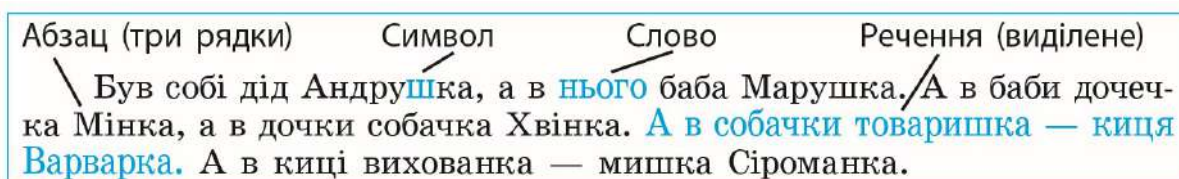


Рис. 14.3

Дії над об'єктами документа

Над текстовим документом та його об'єктами можна виконувати певні дії. Розглянемо таблицю:

Назва дії	Опис дії
Уведення та зберігання	Уведення з клавіатури символів у документ та збереження у вигляді файлу на диску
Редагування	Виправлення помилок, видалення, переміщення, копіювання, вставляння символів, слів, абзацив, рядків тощо
Форматування	Зміна вигляду документа: встановлення кольорів тексту, вирівнювання абзацив, зміна орієнтації сторінки тощо
Вставляння об'єктів	Додавання до документа рисунків, таблиць, формул, діаграм, схем тощо
Макетування	Підготовка документа до друку: оформлення заголовків, розбивка на сторінки, нумерація їх, опрацювання рисунків, створення змісту тощо
Друкування	Виведення на папір усіх або вибраних сторінок створеного документа, а також його фрагментів

Властивості сторінки

Сторінка як об'єкт текстового документа має такі властивості: розміри сторінки, розміри полів, орієнтація сторінки, наявність колонтитулів тощо.

На рис. 14.4:

- 1 — верхнє поле;
- 2 — абзац;
- 3 — ліве поле;
- 4 — сторінка;
- 5 — нижнє поле;
- 6 — верхній колонтитул;
- 7 — праве поле;
- 8 — таблиця;
- 9 — рисунок;
- 10 — нижній колонтитул.

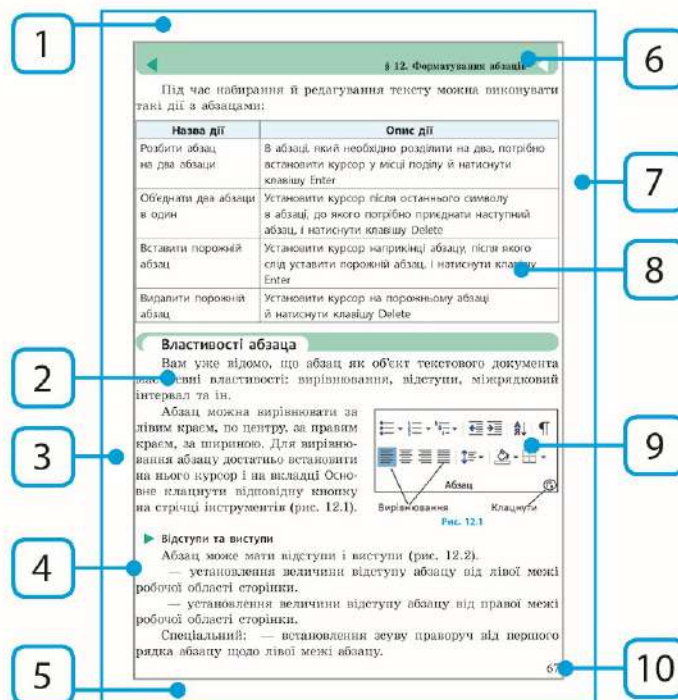


Рис. 14.4

Розміри сторінки — це висота й ширина сторінки документа. Найпоширеніші стандартні розміри сторінки: А4 — 21 × 29,7 см, А5 — 14,8 × 21 см.

Поля — це області сторінки вздовж її країв, у яких не міститься основний текст документа. На сторінці є верхнє, нижнє, ліве та праве поля. Рекомендовані розміри полів такі: ліве — 2,5 см, праве — 1,5 см, верхнє та нижнє поля — по 2 см. Ліве й праве поля частіше залишаються незаповненими, а на верхньому та нижньому полях можуть розміщуватися колонтитули.

Колонтитули — повідомлення, розташовані на полях сторінки. Розрізняють *верхній* та *нижній* колонтитули. Вони можуть містити номери сторінок, назву документа або поточного розділу, прізвище автора, графічні зображення тощо. Інформація, розміщена в колонтитулах, відображається на всіх сторінках документа або частині їх.

Орієнтація сторінки — це спосіб розміщення тексту на сторінці. Розрізняють *книжкову* (вертикальну) й *альбомну* (горизонтальну) орієнтації.

Окрім текстових об'єктів сторінка може містити графічні об'єкти: рисунки, таблиці, діаграми, фотографії, формули тощо.

Щоб **установити розміри полів, орієнтацію та розмір сторінки** (рис. 14.5), слід відкрити вкладку **Розмітка сторінки** (інструменти групи **Параметри сторінки**) і потрібний список, вибрати один з варіантів або встановити потрібні розміри за допомогою лічильників.



Рис. 14.5

Установлення масштабу перегляду

Масштаб відображення документа на екрані монітора встановлюють за допомогою бігунка в правому нижньому куті вікна або за допомогою вкладки **Вигляд**.

Щоб **установити масштаб**, потрібно вибрати команду **Масштаб** — відкриється вікно **Масштаб** (рис. 14.6), де слід зазначити масштаб або встановити потрібне число на лічильнику **Відсоток**.

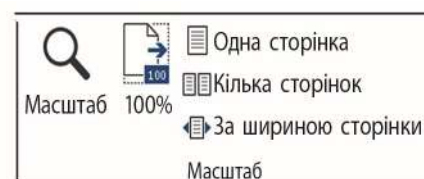


Рис. 14.6

Збереження текстового документа

Після завершення роботи документ потрібно зберегти.

Для збереження текстового документа потрібно:

1) виконати команду **Файл** → **Зберегти** → **Огляд** — відкриється діалогове вікно **Збереження документа** (рис. 14.7);

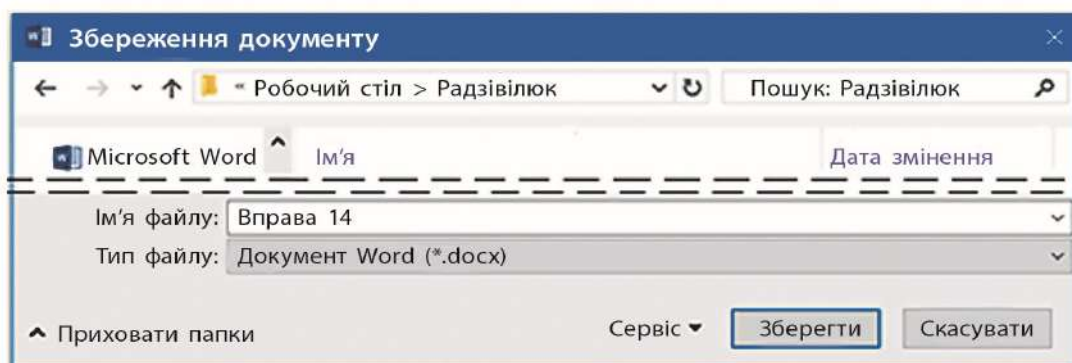


Рис. 14.7

2) відкрити папку, в якій має зберігатися новий документ (тут **Робочий стіл** → **Радзівілюк**);

3) записати ім'я файлу (тут — **Вправа 14**);

4) клацнути кнопку **Зберегти**.

Питання для самоперевірки

1. Назвіть текстові об'єкти сторінки документа.

2. Які поля має сторінка?
3. Яке призначення колонтитулів?
4. Опишіть вікно текстового процесора **Word**.
5. Як установити параметри сторінки?
6. Як установити потрібний масштаб?

Вправа 14

Налаштувати параметри сторінки та зберегти текстовий документ.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Запустіть текстовий процесор і створіть документ.
2. Задайте:
 - розмір паперу — **A5**; полів — **по 2 см**;
 - орієнтацію — **книжкову**.
3. Установіть масштаб — **За шириною сторінки**.
4. Установіть:
 - шрифт — **Times New Roman**;
 - розмір — **14**;
 - накреслення — **звичайний**.
5. Уведіть у текстовий документ свої ім'я та прізвище, клас і назву школи.
6. Збережіть файл з назвою **Вправа 14** у власній папці. Заверште роботу з текстовим процесором.

Комп'ютерне тестування

Виконайте тестове завдання 14 з автоматичною перевіркою результату.

§ 15. РОБОТА З ФРАГМЕНТАМИ ТЕКСТУ. РЕДАГУВАННЯ ТА ФОРМАТУВАННЯ СИМВОЛІВ

Будь-який текст зазвичай потребує подальшого редагування та форматування. Редагування тексту — процес змінення окремих його частин шляхом переміщення, форматування тощо.

Операції над фрагментом тексту

Текстовий процесор дозволяє вирізати, копіювати або переносити на інше місце вибрану частину тексту (символ, слово, речення, кілька речень, абзац, кілька абзаців) — **фрагмент тексту**.

! Перш ніж виконувати дії з фрагментом тексту, його потрібно виділити за допомогою клавіатури або миші.

► Виділення за допомогою клавіатури

Для виділення в документі всього тексту можна скористатися сполученням клавіш **Ctrl + A**.

Щоб **виділити фрагмент тексту**, слід:

- 1) установити текстовий курсор на початок або кінець фрагмента;
- 2) утримуючи клавішу **Shift**, переміщувати курсор клавішами керування курсором (рис. 15.1).

Виділений фрагмент тексту підсвічується кольором (рис. 15.2). Щоб **зняти виділення**, потрібно натиснути будь-яку клавішу керування курсором (див. рис. 15.1).

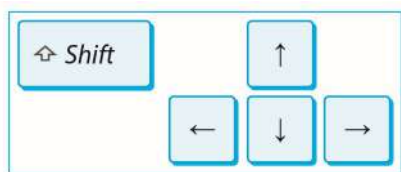


Рис. 15.1

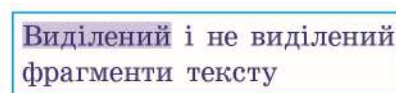


Рис. 15.2

► Виділення за допомогою миші

Для **виділення фрагмента тексту мишею** потрібно встановити її вказівник на початок фрагмента і, утримуючи натиснутою ліву кнопку миші, протягнути вказівник до кінцевого положення.

Для виділення кількох фрагментів тексту вже описані дії мишею повторюють з утриманням натиснутою клавіші **Ctrl**.

Існують швидкі способи виділення мишею:

Дія	Результат
Клацнути на лівому полі документа	Виділиться рядок
Протягнути вказівник по лівому полю	Виділиться декілька рядків
Двічі клацнути на лівому полі	Виділиться абзац
Тричі клацнути на лівому полі	Виділиться весь текст документа
Двічі клацнути на символі	Виділиться слово
Тричі клацнути на символі	Виділиться абзац
Клацнути перед першим символом і, утримуючи клавішу Shift , клацнути за останнім символом фрагмента (рис. 15.3)	Виділиться текстовий фрагмент від першого до останнього символу

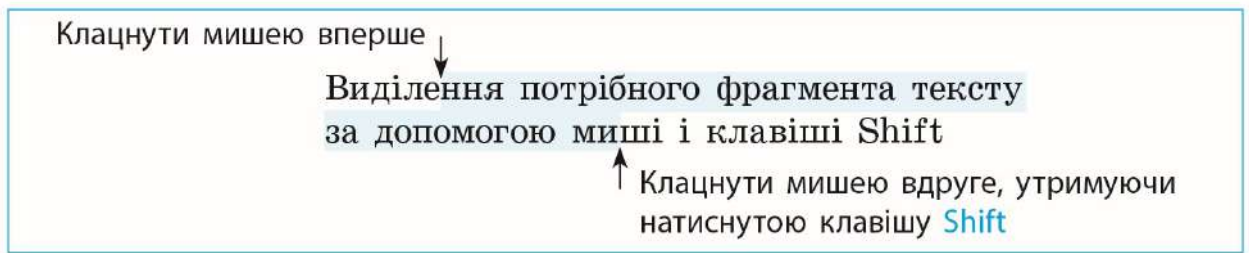


Рис. 15.3

Щоб зняти виділення фрагмента тексту, слід клацнути мишею в будь-якому місці робочого поля (на сторінці документа).

Копіювання та переміщення фрагментів тексту

До виділених фрагментів тексту можна застосовувати низку операцій, які виконуються за допомогою команд контекстного меню або сполучень клавіш. Наприклад, це може бути переміщення або вставляння, що передбачає копіювання в **буфер обміну** — ділянку пам'яті, призначену для тимчасового зберігання даних різних типів.

Таким чином, щоб **перемістити виділений фрагмент** з одного місця документа в інше, потрібно його вирізати (**Ctrl + X**), а потім уставити з буфера обміну (**Ctrl + V**) в інше місце.

Не слід плутати операції вирізання (**Ctrl + X**) і видалення (**Delete**), оскільки вирізаний об'єкт зберігається в буфері обміну, а видалений об'єкт — не зберігається.

На відміну від системного буфера обміну, який може містити лише один об'єкт, буфер обміну **Microsoft Office** може зберігати до 24 об'єктів, скопійованих із будь-яких джерел: із браузера, з різних офісних програм тощо. Якщо об'єктів більше, то скопійовані останніми залишаються в буфері обміну, а скопійовані першими — видаляються.

ПРИКЛАД 1. Існує спосіб переміщення текстового фрагмента без його зберігання в буфері обміну. Для цього достатньо виділити текстовий фрагмент і перетягнути вказівником миші на інше місце в документі (рис. 15.4).



Рис. 15.4

► Редагування символів

У тексті можуть траплятися помилки. Розрізняють помилки трьох основних видів:

- *зайвий символ* — його потрібно видалити;
- *пропущений символ* — його необхідно вставити;
- *неправильний символ* — його слід замінити.

Розглянемо способи виправлення помилок (рис. 15.5):



Рис. 15.5

Розглянемо способи редагування символів:

- Для **видалення зайвого символу**, розміщеного **ПРАВОРУЧ** від курсора, слід натиснути клавішу **Delete**, а розміщеного **ЛІВОРУЧ** від курсора — клавішу **Backspace**.

- Для **вставляння пропущеного символу** слід установити курсор на місце, де він має бути, і ввести з клавіатури потрібний (на рис. 15.5 — буква *В*). Наступні символи зсуваються на одну позицію вправо.

- Для **замінення символу** слід видалити неправильний символ клавішею **Delete** (на рис. 15.5 — буква *У*) і ввести потрібний символ (на рис. 15.5 — буква *О*). На місці неправильного символу з'явиться введений.

Властивості символів

Як об'єкт текстового документа символ має такі властивості, як шрифт, накреслення, розмір, колір.

! Властивості можна встановити для одного або кількох символів, слів, речень, рядків, а також для всього тексту.

► Добір і зміна шрифту

Основні властивості шрифту встановлюють за допомогою:

- відповідних інструментів на вкладці **Основне** (рис. 15.6);
- діалогового вікна **Шрифт**.



Рис. 15.6

Для **викликання діалогового вікна** необхідно:

- 1) клацнути спеціальний значок;
- 2) у вікні **Шрифт** вибрати вид, накреслення, розмір, колір, підкреслення, спецефекти тощо.

ПРИКЛАД 2.

Це приклад шрифту **Times New Roman** розміру 12 пунктів

Це приклад шрифту Arial розміру 12 пунктів

Це приклад шрифту Courier New розміру 11 пунктів

Це приклад шрифту Garamond розміру 14 пунктів

Це приклад шрифту Monotype Corsiva розміру 14 пунктів

► Розмір шрифту

Розмір шрифту та інтервали між символами й рядками вимірюють у типографських пунктах (пт). Один пункт дорівнює 0,376 мм.

► Накреслення символів

У вікні **Шрифт** можна встановити накреслення: звичайний, *курсив*, **напівжирний**, *напівжирний курсив*. Кнопки для вибору накреслення шрифту зазвичай містяться на стрічці інструментів (**Ж** — жирний, **К** — курсив, **П** або **Ч** — підкреслений).

► Спеціальні ефекти

До виділеного тексту можна додати спеціальні ефекти.

ПРИКЛАД 3.

Закреслений;

нарядковий знак $A^2 + B^2 = C^2$;

подвійне підкреслення;

підрядковий знак C_2H_5OH ;

контур тощо.

Для перетворення малих літер на великі та виконання інших додаткових команд форматування на стрічці інструментів можуть бути додаткові кнопки.

► Міжсимвольний інтервал

У разі потреби можна збільшити або зменшити інтервал між символами. Для цього необхідно виділити потрібний фрагмент тексту та у вікні **Шрифт** відкрити вкладку **Додатково** (рис. 15.7):

- для збільшення інтервалу між символами слід збільшити значення лічильника **З кроком:** (тут — 1,3 пт). У полі **Інтервал** автоматично встановиться режим **Розріджений**;

- для зменшення інтервалу між символами необхідно зменшити значення лічильника **З кроком:**. У полі **Інтервал** установається режим **Ущільнений**.

Після виконання дії слід натиснути кнопку **ОК**.



Рис. 15.7

Питання для самоперевірки

1. Наведіть приклади фрагментів тексту.
2. Які є способи виділення фрагмента тексту?
3. Які операції можна виконувати над фрагментом тексту?
4. Які властивості символу як об'єкта текстового документа?
5. Опишіть способи редагування символів.
6. Як вибрати шрифт, накреслення, розмір і колір символів?

Вправа 15

Змінити властивості символів у текстовому документі.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Запустіть текстовий процесор і створіть новий документ. Задайте:
 - розмір паперу — **A4**; полів — **по 1,5 см**; орієнтацію — **альбомну**;
 - шрифт — **Arial**; розмір — **10**; накреслення — **звичайне**;
 - масштаб — **За шириною сторінки**;
 - вирівнювання — **за лівим краєм**.
2. Уведіть у новий документ наведений текст, у кінці кожного абзацу натискаючи клавішу **Enter**.

Україна — найбільша за площею країна Європи.

Територію України омивають Чорне та Азовське моря.

Державними символами України є Гімн, Прапор і малий Герб.

3. Для *першого* абзацу встановіть:

- шрифт — **Arial**; розмір — **12**;
 - накреслення — **курсив**;
 - колір — **червоний**.
4. Для *другого* абзацу встановіть:
- шрифт — **Times New Roman**;
 - розмір — **14**;
 - накреслення — **підкреслений**.
5. Для *третього* абзацу встановіть:
- розмір — **14**;
 - накреслення — **жирний**;
 - інтервал між символами — **ущільнений 0,8 пт.**
6. Збережіть текст у файл з назвою **Вправа 15** у власну папку. Заверште роботу з текстовим процесором.

Комп'ютерне тестування

Виконайте тестове завдання 15 з автоматичною перевіркою результату.

ПРАКТИЧНА РОБОТА 4. Здійснення форматування символів текстового документа

Завдання: виконати форматування тексту за зразком.

Обладнання: комп'ютер зі встановленим текстовим процесором.

Хід роботи

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Запустіть браузер.
2. Знайдіть в інтернеті текст куплету та приспіву Державного Гімну України.
3. Скопіюйте знайдений текст у буфер обміну.
4. Запустіть текстовий процесор і створіть документ.
5. Вставте з буфера обміну в документ текст Державного Гімну України.
6. Розташуйте кожен рядок в окремому абзаці.
7. Під текстом напишіть ініціали, прізвище автора слів.
8. Відформатуйте сторінку за такими параметрами:
 - розмір — **A4**;
 - поля — **по 3 см**;
 - орієнтація — **альбомна**.
9. Відформатуйте шрифт тексту куплету за такими параметрами:
 - тип — **Calibri**;

- розмір — **30**;
- накреслення — **жирний**;
- колір шрифту — **жовтий**;
- колір виділення шрифту — **синій**;
- міжсимвольний інтервал — **звичайний**.

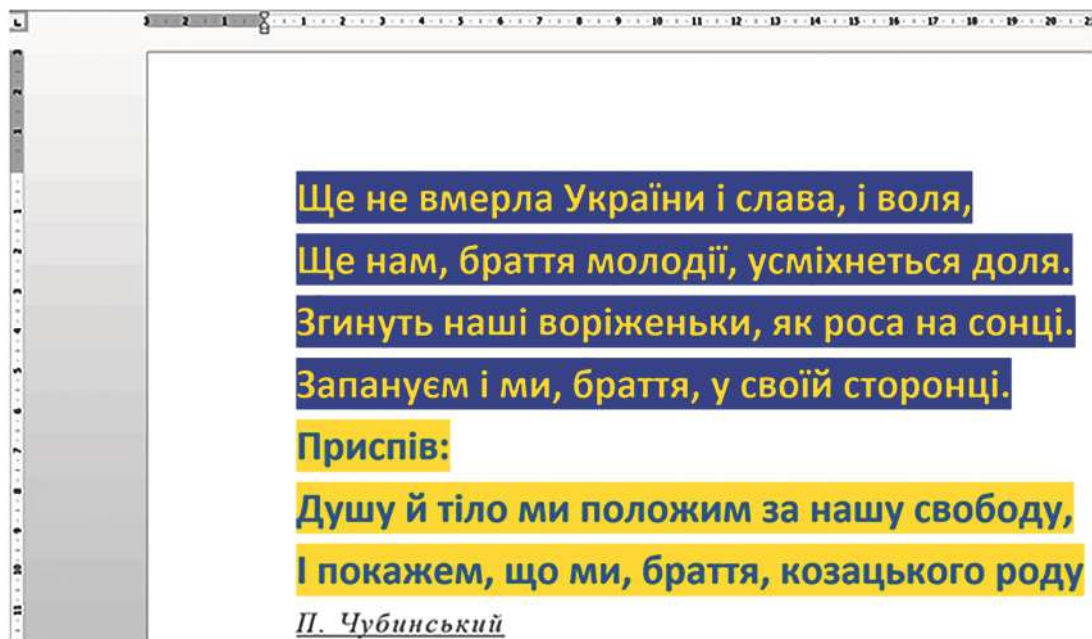
10. Відформатуйте шрифт тексту приспіву за такими параметрами:

- тип — **Calibri**;
- розмір — **30**;
- накреслення — **жирний**;
- колір шрифту — **синій**;
- колір виділення шрифту — **жовтий**;
- міжсимвольний інтервал — **звичайний**.

11. Відформатуйте шрифт прізвища автора слів за такими параметрами:

- тип — **Times New Roman**;
- розмір — **20**;
- накреслення — **курсив і підкреслення**;
- колір шрифту — **чорний**;
- виділення шрифту — **без кольору**;
- міжсимвольний інтервал — **розріджений із кроком 2 пт.**

12. За необхідності виправте помилки написання та форматування тексту відповідно до зразка (див. рисунок).



Збережіть файл із назвою **Практична робота 4** у зазначеному вчителю / учителькою місці (папці). Заверште роботу за комп'ютером.

Зробіть висновок за темою практичної роботи.

§ 16. ФОРМАТУВАННЯ АБЗАЦІВ

Текст зазвичай поділяється на абзаци. Абзац завершується недрукованим символом, який уводиться натисканням клавіші **Enter**. Щоб побачити або приховати його розташування, необхідно на вкладці **Основне** в групі **Абзац** натиснути кнопку ¶.

► Дії з абзацами

З абзацами можна виконувати такі дії:

Назва дії	Опис дії
Розбити абзац на два абзаци	В абзаці, який необхідно розділити на два, потрібно встановити курсор у місці поділу й натиснути клавішу Enter
Об'єднати два абзаци в один	Установити курсор після останнього символу в абзаці, до якого потрібно приєднати наступний абзац, і натиснути клавішу Delete
Вставити порожній абзац	Установити курсор наприкінці абзацу, після якого слід вставити порожній абзац, і натиснути клавішу Enter
Видалити порожній абзац	Установити курсор на порожньому абзаці й натиснути клавішу Delete

Властивості абзацу

Як об'єкт текстового документа **абзац має певні властивості**: вирівнювання, відступи, міжрядковий інтервал та ін.

Абзац можна вирівнювати за лівим краєм, по центру, за правим краєм, за шириною.

Для **вирівнювання абзацу** достатньо встановити на нього курсор і на вкладці **Основне** клацнути відповідну кнопку на стрічці інструментів (рис. 16.1).

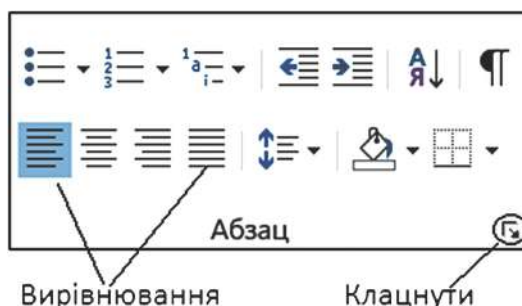


Рис. 16.1

Абзац може мати відступи і виступи (рис. 16.2).

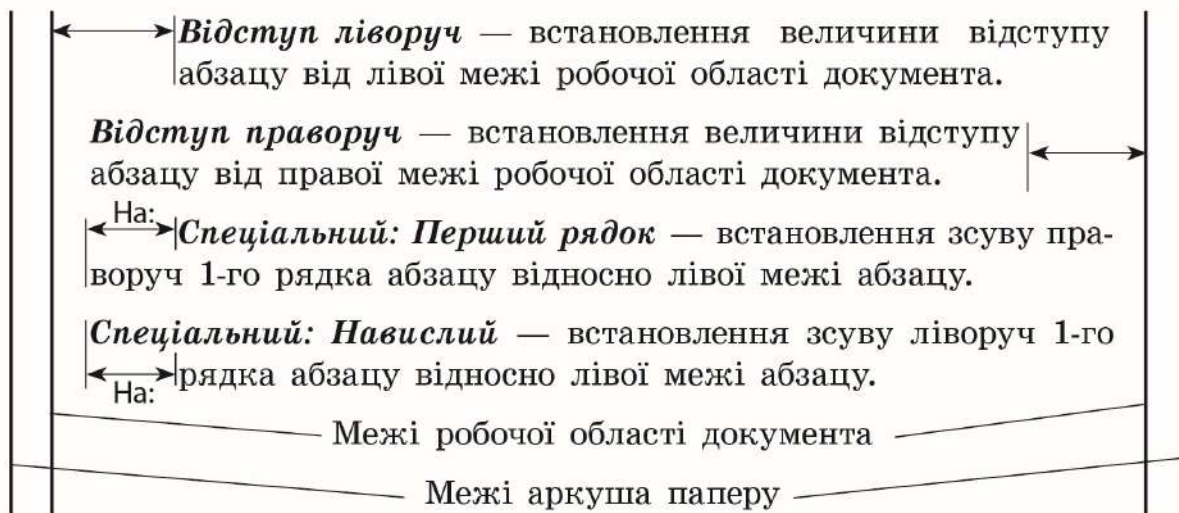


Рис. 16.2

Відступ ліворуч — установлення величини відступу абзацу від лівої межі робочої області сторінки.

Відступ праворуч — установлення величини відступу абзацу від правої межі робочої області сторінки.

Спеціальний: Перший рядок На: — установлення зсуву праворуч від першого рядка абзацу відносно лівої межі абзацу.

Спеціальний: Навислий На: — установлення зсуву ліворуч від першого рядка абзацу відносно лівої межі абзацу.

Швидко встановити відступи абзацу допоможе **горизонтальна лінійка** (рис. 16.3). Щоб змінити величини відступів і виступів, достатньо перетягти на нове місце відповідний маркер.



Рис. 16.3

Відступи та виступи абзацу можна встановити відповідними інструментами вкладки Основне або діалогового вікна Абзац, для виклику якого слід клацнути спеціальний значок (рис. 16.4).



Рис. 16.4

Діалогове вікно **Абзац** можна викликати командою **Абзац...** із контекстного меню, клацнувши правою кнопкою миші на потрібному абзаці.

Для **встановлення відступів і виступів** необхідно:

- 1) установити курсор на абзац і викликати діалогове вікно **Абзац**;

2) у діалоговому вікні **Абзац** (на рис. 16.5 подано його фрагмент) за допомогою лічильників **Ліворуч** (тут 0,5 см) і **Праворуч** (тут 0,7 см) установити потрібні лівий і правий відступи.



Рис. 16.5

Відступ або виступ першого рядка встановлюється в списку **Спеціальний** за допомогою лічильника **На:**.

► Міжабзацний і міжрядковий інтервали

У текстовому документі можна встановлювати та змінювати інтервали між абзацами й рядками.

Міжабзацний інтервал встановлюють за допомогою лічильників **Перед:** і **Після:** (на рис. 12.5 **Перед:** — 12 пт, **Після:** — 6 пт).

Для встановлення міжрядкового інтервалу необхідно:

1) на вибраному абзаці або групі виділених абзаців установити вказівник;

2) відкрити список **Міжрядковий інтервал** у діалоговому вікні **Абзац**;

3) скористатися одним зі способів:

- задати один зі стандартних інтервалів: одинарний, **1,5 рядка**, **подвійний** або інше (рис. 16.6);

- вибравши **Точно**, установити на лічильнику **На:** числове значення інтервалу в пунктах менше 12 (стиснутий) або більше 12 (розтягнутий);

- вибравши **Множинний**, установити на лічильнику **На:** значення менше 1 — міжрядковий інтервал стиснутий або більше 1 — розтягнутий (на рис. 12.6 встановлено 1,08);

4) натиснути кнопку **ОК**.

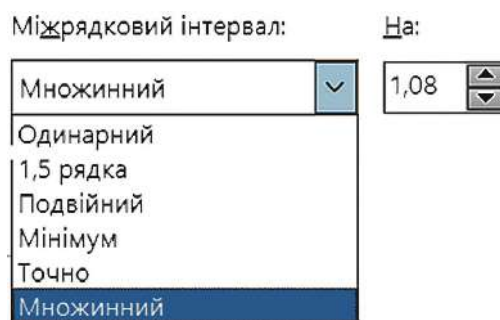



Рис. 16.6

Правила набору тексту

Вміння швидко набирати текст на комп'ютері дуже важливе. Разом із тим текст потрібно набирати грамотно, а для цього необхідно дотримуватися певних правил.

Розглянемо деякі з правил набору тексту:

Правило	Приклади	
	правильно	неправильно
Під час друку між словами ставте лише один пропуск (проміжок)	Правила набору текстів	Правила набору текстів
Розділовий знак (крапку, кому, двокрапку тощо) не відокремлюйте пропуском від попереднього слова	Привіт, друже!	Привіт, друже!
Після кожного розділового знака ставте пропуск	З повагою, Марійка.	З повагою,Марійка.
Текст, який береться в лапки або дужки, пропуском від них не відділяйте	(правила набору) «цифровий»	(правила набору) «цифровий»
Біля дефіса не робіть пропуски	Людина-робот	Людина-робот
Тире відділяйте пропусками	Книга — джерело знань	Книга-джерело знань
Не використовуйте клавішу Enter для переходу на новий рядок у межах одного абзацу (перехід відбувається автоматично)	Не використовуйте клавішу Enter для переходу на новий рядок в межах одного абзацу¶	Не використовуйте¶ клавішу Enter для¶ переходу на новий¶ рядок в межах одного¶ абзацу¶
Не ставте в кінці рядка знак переносу (це можна налаштувати)		Не ставте знаків переносу

Правило	Приклади	
	правильно	неправильно
Не вставляйте порожні рядки (для цього існують міжабзацні та міжрядкові інтервали)		Не вставляйте рядки.¶ ¶ Існують інтервали.¶
Не нумеруйте вручну списки та сторінки (для цього є спеціальні методи)		1. Січень¶ 2. Лютий¶ 3. Березень¶

Тут маємо такі позначення: ¶ — кінець абзацу (натискання **Enter**); • — пропуск; — знак табуляції (під час автонумерації).

Отже, набір тексту, виконаний за всіма правилами, сприяє правильній автоматичній перевірці правопису, полегшує читання та сприймання тексту, спрощує подальшу роботу над документом.

► «Гарячі» клавіші Word

Для підвищення швидкості введення та опрацювання текстів багато команд у **Word** можна викликати за допомогою клавіш. Такі клавіші та їхнє сполучення називають «гарячими».

Питання для самоперевірки

1. Які операції виконують над абзацами?
2. Опишіть порядок дій для розбиття одного абзацу на два.
3. Наведіть послідовність дій для об'єднання двох абзаців.
4. Як можна вирівняти абзац?
5. Які відступи може мати абзац; виступи?
6. Як установити інтервали перед абзацем і після нього?

Вправа 16

Налаштувати властивості абзаців у текстовому документі.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Запустіть текстовий процесор і створіть новий документ. Задайте:
 - розмір паперу — **A5**; поля — **по 2 см**;
 - орієнтацію — **книжкову**;
 - шрифт — **Arial**; розмір — **14**;

- вирівнювання — за лівим краєм;
- інтервали **Перед:** і **Після:** — 0 пт;
- міжрядковий інтервал — **одинарний**;
- масштаб — **За шириною сторінки**.

2. Уведіть у документ наведений текст, у кінці кожного абзацу натискаючи клавішу **Enter**.

Правила набирання тексту

Робити «пропуск» між словами. Після розділового знаку робити «пропуск».

До і після дефісу «пропуск» не робити.

До і після тире робити «пропуск».

3. *Другий* абзац розбийте на два. Після *третього* абзацу вставте порожній абзац, після нового *п'ятого* — два порожні абзаци.

4. Установіть відступи для першого абзацу:

- зліва — **1 см**; справа — **2 см**;
- для першого рядка — **1 см**.

5. Установіть відступи для інших абзаців:

- зліва — **0 см**; справа — **3 см**;
- виступ навислий для першого рядка — **1 см**.

6*. Допишіть імена п'ятох своїх друзів і подруг. Для всього тексту встановіть міжрядковий інтервал із множителем **1,1**.

Збережіть файл з назвою **Вправа 16** у власну папку. Заверште роботу з текстовим процесором.

Комп'ютерне тестування

Виконайте тестове завдання 16 з автоматичною перевіркою результату.

ПРАКТИЧНА РОБОТА 5. Здійснення форматування абзаців текстового документа

Завдання: виконати форматування текстових абзаців за зразком.

Обладнання: комп'ютер зі встановленим текстовим процесором.

Хід роботи

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Відкрийте документ **Практична робота 4**.
2. Об'єднайте рядки тексту Державного Гімну України у два абзаци: *перший* — куплет, *другий* — приспів.
3. Відформатуйте *перший* абзац за параметрами:

- вирівнювання — **За шириною**;
- відступ: ліворуч — **0 см**, праворуч — **0 см**, спеціальний — **немає**;
- інтервал: **Перед:** — **0 пт**, **Після:** — **0 пт**;
- міжрядковий інтервал — **0,8**.

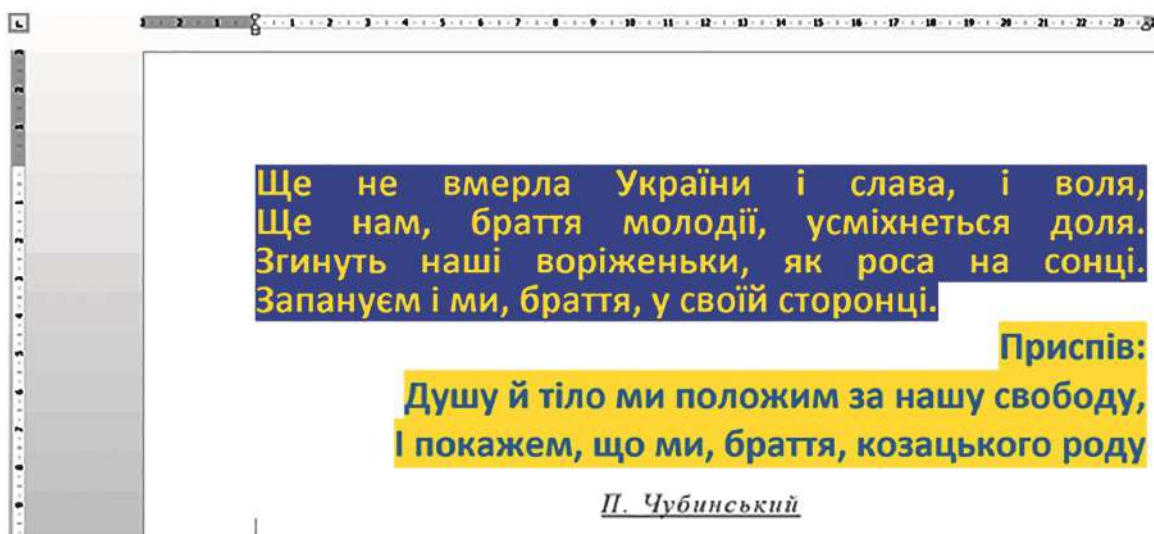
4. Відформатуйте *другий* абзац (приспів) за такими параметрами:

- вирівнювання — **справа**;
- відступ: ліворуч — **0 см**, праворуч — **0 см**, спеціальний — **немає**;
- інтервал: **Перед:** — **0 пт**, **Після:** — **0 пт**;
- міжрядковий інтервал — **одинарний**.

5. Відформатуйте *третьий* абзац (прізвище автора чи авторки) за такими параметрами:

- вирівнювання — **По центру**;
- відступ: ліворуч — **0 см**, праворуч — **0 см**, спеціальний — **немає**;
- інтервал: **Перед:** — **15 пт**, **Після:** — **0 пт**;
- міжрядковий інтервал — **одинарний**.

6. За необхідності виправте помилки форматування відповідно до зразка (див. рисунок).



Збережіть файл із назвою **Практична робота 5** у зазначеному вчителем / учителькою місці (папці). Заверште роботу за комп'ютером.

Зробіть висновок: за результатами практичної роботи.

ПРАКТИЧНА РОБОТА 6. Створення текстового документа

Завдання: створити простий текстовий документ і здійснити його форматування.

Обладнання: комп'ютер зі встановленим текстовим процесором.

Хід роботи

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Запустіть текстовий процесор і створіть новий документ. Задайте:

- розмір паперу — **A5** (14,8 × 21 см);
- полів — **по 1,5 см**;
- орієнтацію — **книжкову**;
- шрифт — **Arial** (або інший, зазначений вчителем / учителькою);
- розмір — **12**; накреслення — **звичайний**;
- масштаб — **За шириною сторінки**;
- вирівнювання — **За лівим краєм**;
- інтервали абзацу **Перед: і Після: — 0 пт**;
- міжрядковий інтервал — **1,5 рядка**.

2. Наберіть текст заяви запропонованого змісту на ім'я директора / директорки школи від імені свого татка (мами).

Директорці ЗОШ № 9

Марчук Майї Іванівні

Сумлінної Катерини Михайлівни

Заява

Оскільки моя дочка учениця 5-А класу Сумлінна Ліна під час навчальних занять брала участь у змаганнях з шахів, прошу надати їй можливість підвищити результати семестрового оцінювання з інформатики.

24.12.2022 _____ Сумлінна К. М.

3. Для перших 3 рядків установіть:

- накреслення — **курсив**;
- відступ зліва — **7 см**.

4. Для абзацу зі словом «заява» встановіть:

- шрифт — **Arial**; розмір — **14**;
- накреслення — **жирний**;
- вирівнювання — **По центру**;
- інтервали: **Перед: — 24 пт, Після: — 6 пт**;
- розріджений на **5 пт**.

5. Для змістової частини заяви встановіть:

- вирівнювання — **За шириною**;
- відступ першого рядка — **1 см**.

6. Для прізвища заявника/заявниці встановіть:

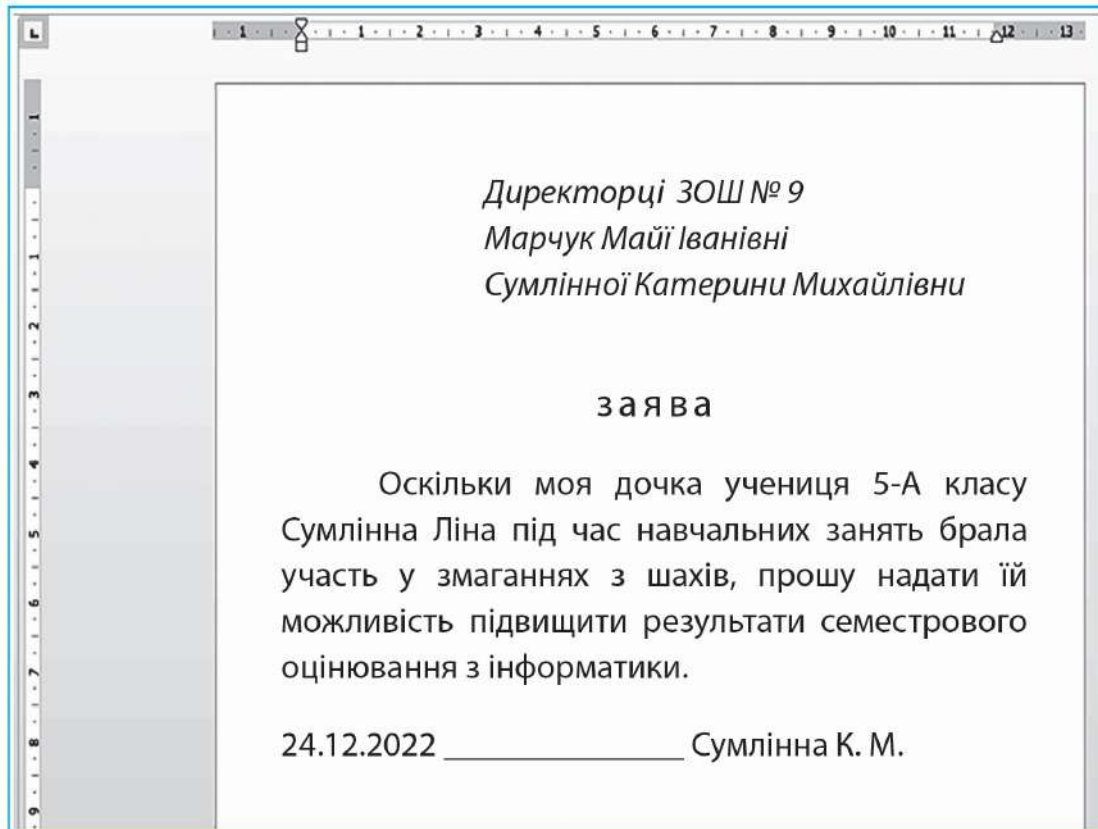
- вирівнювання — **За правим краєм**;

- інтервал **Перед:** — 18 пт.

Збережіть документ у файлі з назвою **Практична робота 6** у власній папці.

Порівняйте результат з наведеним на рисунку зразком.

Заверште роботу за комп'ютером.



Зробіть висновок: які засоби для створення й форматування простого текстового документа надає текстовий процесор.

§ 17. ОПРАЦЮВАННЯ ОБ'ЄКТІВ ТЕКСТОВОГО ДОКУМЕНТА

Знайти в тексті потрібні фрагменти (наприклад, повторювані) вручну буває важко, особливо у великих за обсягом документах.

Автоматичний пошук і заміна фрагментів тексту

Текстовий процесор оснащений засобом для автоматичного пошуку об'єктів документа (фрагментів, рисунків, формул та ін.).

► Пошук текстових фрагментів

Для пошуку фрагмента тексту потрібно:

1) на вкладці **Основне** в групі інструментів **Редагування** клацнути команду **Пошук** (рис. 17.1) або натиснути сполучення клавіш **Ctrl + F** — відкриється бокова вкладка **Навігація**;

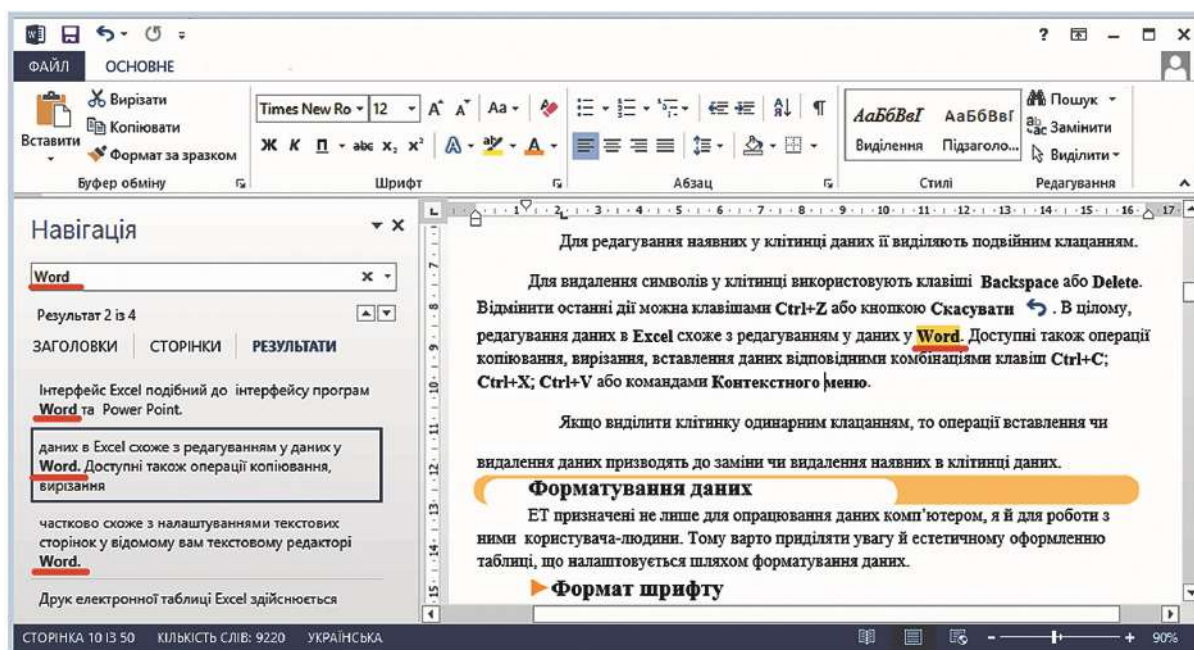


Рис. 17.1

2) у текстове поле пошуку ввести текст, який потрібно відшукати, — на вкладці **Навігація** з'являться уривки текстового документа зі знайденим фрагментом тексту;

3) для переходу до відповідного місця в документі клацнути вибраний уривок, який виділено жовтим тлом (див. рис. 17.1).

! Якщо текстовий фрагмент перед пошуком виділити та скопіювати в буфер обміну (**Ctrl + C**), то пошук цього фрагмента можна здійснити за допомогою «гарячих» клавіш **Ctrl + F**.

▶ Заміна текстових фрагментів

Досить часто буває потрібно не просто знайти фрагмент тексту в документі, а й замінити його на інший.

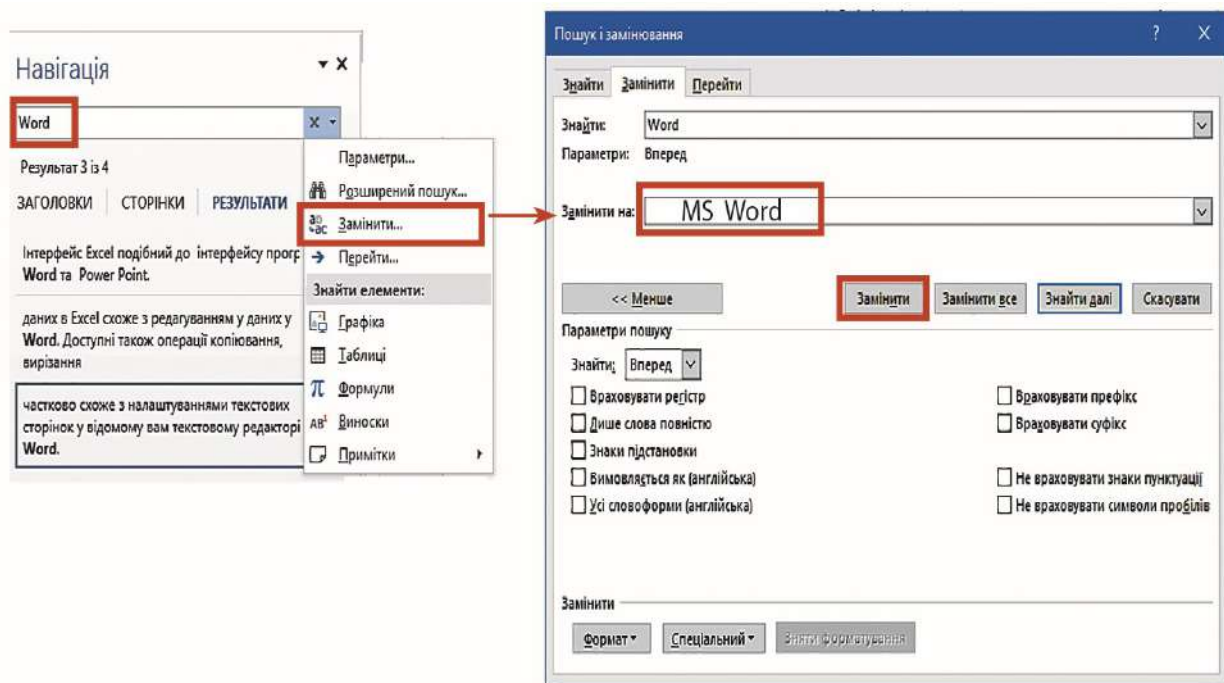
Щоб **замінити фрагмент**, слід виконати такі дії:

1) на боковій вкладці **Навігація** у списку, що розгортається, вибрати команду **Замінити** (рис. 17.2, а);

2) у вікно **Пошук і замінювання**, що відкриється, у полі **Замінити на** ввести текст, яким потрібно замінити знайдений текстовий фрагмент, наприклад **MS Word** (рис. 17.2, б);

3) натиснути кнопку **Замінити** (або **Замінити все**) (рис. 17.2, б).

За необхідності можна налаштувати додаткові параметри пошуку, які з'являються після розгортання вікна кнопкою **Більше**.



а

б

Рис. 17.2

Діалогове вікно **Пошук і замінювання** (див. рис. 17.2, б) можна відкрити також сполученням клавіш **Ctrl + H**.

! У документі можна здійснювати пошук тексту не лише за змістом, а й такого, що має певне форматування.

Для **пошуку інших об'єктів** (графічних зображень, таблиць, формул тощо) слід вибрати відповідну команду зі списку, що розгортається біля текстового поля пошуку на вкладці **Навігація** (див. рис. 17.2, а).

Перевірка правопису

Текстовий процесор оснащений засобом, який дозволяє перевірити й виправити написання слів у тексті. Якщо слово підкреслюється червоною хвилястою лінією, це зазвичай свідчить про можливу помилку.

▶ Автоматична перевірка правопису

Щоб увімкнути автоматичну перевірку правопису, потрібно вибрати **Файл** → **Параметри** → **Правопис** → **Автоматична перевірка правопису**.

Виклик контекстного меню для слова з помилкою (клацання по ньому правою кнопкою миші) дає змогу вибрати один із варіантів автоматичного виправлення помилки, або знехтувати нею (**Пропустити все**), або зберегти як правильний варіант для подальших перевірок (**Додати до словника**) (рис. 17.3).

помилка :

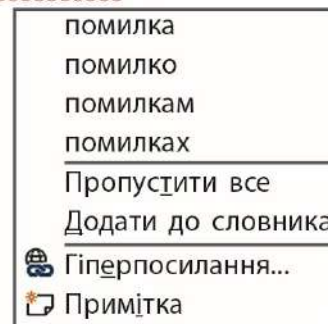


Рис. 17.3

! Перевірка правопису вимикається автоматично, якщо кількість помилок у текстовому документі перевищуватиме дві тисячі.

▶ Вибір мови перевірки правопису

Мову перевірки правопису можна змінити. Для цього в рядку стану необхідно клацнути команду **Мова**, а в діалоговому вікні вибрати потрібну мову.

Питання для самоперевірки

1. Як виділити в документі слово, рядок, речення, абзац?
2. Як скопіювати, вирізати, видалити, перемістити в текстовому документі виділений об'єкт?
3. Чим відрізняються операції **Вирізати** і **Видалити**?
4. Що таке буфер обміну і як переглянути об'єкти, збережені в буфері обміну **Microsoft Office**?
5. Поясніть, як здійснити автоматичний пошук текстового фрагмента в документі. Опишіть, як можна виконати автоматичний пошук малюнків, таблиць тощо.
6. Як виконати автоматичну заміну в документі фрагмента?

Вправа 17

Ознайомитись з особливостями копіювання, вставлення, пошуку й заміни фрагментів тексту.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Запустіть текстовий процесор і створіть документ.
2. Знайдіть в інтернеті опис однієї з визначних пам'яток історії або природи області, де ви мешкаєте.
3. Скопіюйте в буфер обміну декілька абзаців опису і вставте в документ.
- 4*. За потреби відредагуйте та відформатуйте текст.
- 5*. Виберіть у тексті одне з повторюваних слів або словосполучень і замініть його в документі синонімом. За потреби налаштуйте параметри розширеного пошуку.
6. Збережіть документ із назвою **Вправа 17**.

Комп'ютерне тестування

Виконайте тестове завдання 17 з автоматичною перевіркою результату.

Розділ 5

АЛГОРИТМИ ТА ПРОГРАМИ



§ 18. Алгоритм і його властивості

§ 19. Виконавець алгоритмів і система його команд

§ 20. Способи опису алгоритму. Алгоритмічні структури

§ 21. Середовище опису та виконання алгоритмів

§ 22. Основні поняття мови програмування **Python**

§ 23. Лінійні алгоритми та програми

§ 24. Математичні оператори мови **Python**

Практична робота 7. Складання та виконання лінійних алгоритмів

§ 25. «Черепашача» графіка

§ 26. Алгоритми створення зображень

Практична робота 8. Створення зображень за алгоритмами

§ 27. Логічні вирази

§ 28. Алгоритми і програми з розгалуженнями. Оператор **if**

§ 29. Алгоритми і програми з розгалуженнями. Оператор **if ... else**

Практична робота 9. Складання та виконання алгоритмів із розгалуженнями

§ 30. Алгоритми з повтореннями. Цикл із лічильником

§ 31. Алгоритми і програми з циклами з лічильником

Практична робота 10. Складання та виконання алгоритмів з повтореннями

ПОВТОРЮЄМО

На уроках інформатики в молодшій школі ви ознайомилися з *поняттям алгоритму* й навчилися *складати алгоритми* для розв'язування різних задач. Ви вже знаєте, що алгоритм можна подавати *словесним і графічним способами*, а також у *вигляді програм*, записаних певною мовою програмування.

Мова програмування — це штучна мова, яка є системою позначень і правил для запису алгоритмів у формі, придатній для їх виконання на комп'ютері. Для зручної розробки програм існують спеціальні засоби їх створення — *системи програмування*. Усе, що ми робимо за допомогою комп'ютера — від пошуку інформації в інтернеті до комп'ютерної гри, — можливо завдяки програмному коду, написаному програмістами. Ви вже ознайомилися з основними поняттями програмування, працюючи в програмному середовищі **Scratch**.

1. Що таке алгоритм?
2. Що таке програмування?
3. Що таке мова програмування?
4. Які способи опису алгоритму ви знаєте?
5. З якими мовами програмування ви вже знайомі?

Опрацювавши цей розділ, ви ознайомитеся з популярною сучасною мовою програмування **Python**, якою можна створювати програми для розв'язування різних задач, таких як розробка ігор, створення вебсайтів, виконання обчислень тощо.

§ 18. АЛГОРИТМ І ЙОГО ВЛАСТИВОСТІ

Поняття алгоритму

Пригадаймо поняття алгоритму, знайоме вам із курсу інформатики 2–4 класів.



Алгоритм — це скінченна послідовність указівок на виконання дій, спрямованих на розв’язування певного завдання.

ЦІКАВІ ФАКТИ

Слово «алгоритм» походить від імені арабського математика Аль-Хорезмі (800–847 рр.). Абу Абдулла Абу Джафар Мухаммад ібн Муса аль-Хорезмі (рис. 18.1) сформулював правила чотирьох арифметичних дій над багатозначними числами. Латиною ім’я автора європейці писали як *Algorithmi*, і спочатку алгоритмами називали саме ці чотири правила виконання арифметичних дій. Із часом алгоритмами почали називати способи розв’язування найрізноманітніших задач.



Рис. 18.1

У повсякденному житті людина стикається з алгоритмами, що визначають послідовність дій різної природи. Із курсу математики вам відомі алгоритми виконання арифметичних операцій над багатоцифровими числами, алгоритми знаходження коренів лінійних рівнянь та ін.

ПРИКЛАД 1

Алгоритмами є рецепти приготування страв, порядок підготовки автоматичного пристрою до використання, розпорядок дня (рис. 18.2) тощо.



Рис. 18.2

Для багатьох ігор існують алгоритми виграшу, якщо результат гри залежить не від випадкового збігу обставин, а від кмітливості гравця й попередніх розрахунків.

ПРИКЛАД 2

Гра Баше. Є 11 предметів. За один хід можна взяти 1, 2, 3 предмети. Програє особа, якій дістанеться останній предмет.

Алгоритм виграшу для першого гравця/гравчині:

1-й хід: узяти два предмети.

2-й хід і далі: брати стільки предметів, щоб кількість предметів, узятих разом із суперником/суперницею за черговий хід, у сумі складала 4.

Уміння складати алгоритми є дуже важливим для людини будь-якої професії.

Властивості алгоритму

Алгоритм складається з окремих кроків, які потрібно виконати в певному порядку. Якщо порушити порядок виконання кроків або пропустити якийсь крок, то алгоритм не буде виконаний до кінця або призведе до неправильного результату.

ПРИКЛАД 3

Розглянемо алгоритм відмикання дверей.

1. Дістань ключ.
2. Встав ключ у замкову шпарину (рис. 18.3).
3. Двічі поверни ключ за годинниковою стрілкою.
4. Вийми ключ.

Чи відкриємо ми двері, якщо поміняти місцями вказівки 2 і 3?



Рис. 18.3

Алгоритм має певні *властивості*. Щоб алгоритм виконав своє призначення, його необхідно будувати за певними правилами.

Дискретність означає, що алгоритм повинен складатися з окремих кроків, кожний із яких має завершуватися.

Визначеність означає однозначність тлумачення правил виконання кроків і порядку їх виконання. Алгоритм не повинен містити команди, які можуть сприйматися виконавцем неоднозначно.

Виконуваність означає, що алгоритм, призначений для певного виконавця, може містити лише ті команди, які виконавець здатний виконати.

Так, алгоритм для виконавця **Першокласник** не може містити команду «Побудуй бісектрису даного кута», хоча така команда може бути в алгоритмі, призначеному для виконавця **Восьмикласник**.

Скінченність — обов'язкова виконуваність алгоритму. Алгоритм має складатися зі скінченної кількості кроків, кожен з яких потребує для свого виконання скінченного проміжку часу.

ПРИКЛАД 4

Наведена послідовність команд є нескінченною.

1. Візьміть число 2.
2. Помножте задане число на 10.
3. Додайте до результату 5.
4. Якщо одержано додатне число, то перейдіть до команди 3, якщо ні, то припиніть виконання алгоритму.

Масовість означає можливість виконання алгоритму для різних вхідних даних. Наприклад, ви вивчали алгоритм знаходження коренів рівняння виду $ax + b = c$ для розв'язування рівнянь $5x + 3 = 8$, $17x + 6 = 40$ та інших.

Результативність означає, що після виконання послідовності вказівок алгоритму має бути отримано конкретний результат. Наприклад, послідовність вказівок «налий води; увімкни плитку; вимкни плитку» не є результативною, якщо потрібно було нагріти воду.

Формальність означає, що будь-який виконавець, здатний сприймати й виконувати вказівки алгоритму (навіть не розуміючи його змісту), може виконати завдання за заданим алгоритмом. Як відомо, автоматизовані пристрої правильно розв'язують багато задач за заданими їм алгоритмами, хоча змісту задач вони, безумовно, розуміти не можуть.

Питання для самоперевірки

1. Назвіть основні властивості алгоритмів і поясніть суть кожної з них.
2. Розгляньте заданий алгоритм.
 1. Прочитайте число **a1**.
 2. Прочитайте число **a2**.
 3. Поділіть число **a1** на число **a2**.
 4. Запишіть результат.

Чи має цей алгоритм властивості масовості та визначеності?

3. *Задача.* Задане число, більше за 1, зменшити до 1 шляхом ділення на 2.

Алгоритм розв'язування задачі:

1. Поділіть задане число на 2.
2. Якщо результат не дорівнює 1, то виконайте команду 1, інакше — припиніть виконання алгоритму.

Чи має цей алгоритм властивість скінченності?

4. Чи можна скласти алгоритми розв'язування таких задач:
 - а) знайти корінь рівняння $ax + b = c$;
 - б) відвідати театр;
 - в) вивести новий сорт пшениці;
 - г) сконструювати машину для виконання домашніх завдань?
5. Ознайомтеся з наведеним алгоритмом отримання окропу.
 1. Налийте воду в чайник.
 2. Відкрийте кран газової конфорки.
 3. Поставте чайник на плиту.
 4. Почекайте, поки вода закипить.
 5. Піднесіть запалений сірник до конфорки.
 6. Вимкніть газ.





Визначте правильну послідовність дій, яка дозволить запобігти нещасному випадку.

6*. *Задача.* Перевізнику потрібно човном переправити через річку вовка, козу й капусту по одному. Опишіть алгоритм дій перевізника, виходячи з того, що небезпечно залишати разом без нагляду козу й капусту, вовка й козу.

Вправа 18






Завдання: скласти алгоритм для розв'язування задачі.

1. Як, маючи два відра ємністю 2 і 5 л, набрати з водопровідного крана 1 л води? Складіть алгоритм розв'язування задачі. За рисунком визначте об'єм води, яку потрібно набрати або яка залишатиметься у відрах на кожному кроці алгоритму.




Крок 1	Крок 2	Крок 3	Крок 4
			

2. Як із річки принести 6 л води, якщо є лише два відра: ємністю 4 і 9 л? Складіть алгоритм розв'язування задачі.

За рисунком визначте об'єм води, яку слід наливати або яка залишатиметься у відрах на кожному кроці алгоритму.

Крок 1	Крок 2	Крок 3	Крок 4	Крок 5
				

3. Марійка фарбує паркан. Вона має 12 л фарби. Є дві посудини ємністю 8 і 5 л. Яким чином налити 6 л фарби в посудину ємністю 8 л? Складіть алгоритм розв'язування задачі. Складіть таблицю для об'єму води, наявного на кожному кроці алгоритму.

Об'єм	Крок 1	Крок 2	Крок 3	Крок 4	Крок 5	Крок 6
						
						
						

Комп'ютерне тестування

Виконайте тестове завдання 18 з автоматичною перевіркою результату.

§ 19. ВИКОНАВЕЦЬ АЛГОРИТМІВ І СИСТЕМА ЙОГО КОМАНД

Кожен алгоритм орієнтовано на певного виконавця. Виконавцем алгоритму може бути людина, тварина, комп'ютер, система *людина — машина*, система *верстат — автомат* тощо, яких «навчено» виконувати вказівки алгоритму. Якщо виконавцем є деякий пристрій, то вираз «виконавця навчено виконувати вказівку» означає, що пристрій може виконати задану вказівку автоматично, без зовнішнього втручання.



Виконавець — об'єкт, здатний діяти за заданим алгоритмом.

Щоб скласти алгоритм, орієнтований на конкретного виконавця, необхідно знати характеристики цього виконавця.

Характеристики виконавця

Кожен виконавець існує в певних умовах — середовищі — і може виконувати певний набір дій (рис. 19.1). Перш ніж складати алгоритм розв'язування задачі, необхідно дізнатися, у якому середовищі які дії може виконувати виконавець і за допомогою яких команд.

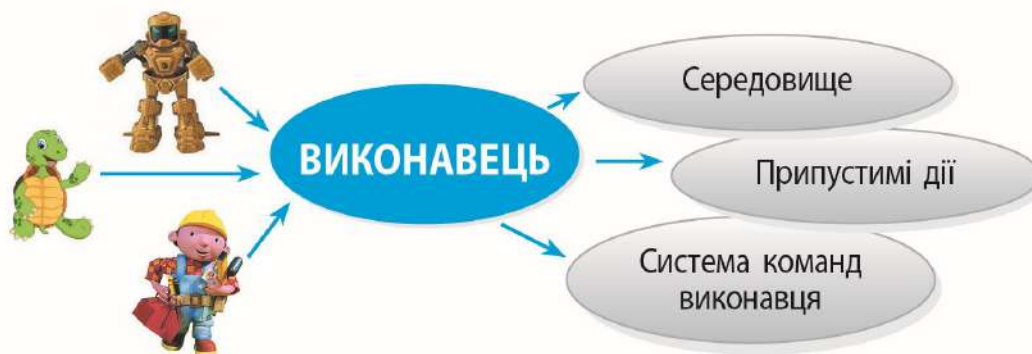


Рис. 19.1

Середовище — «місце існування» виконавця.

Припустимі дії — обмежений набір дій, що вміє виконувати певний виконавець.

Описати виконавця означає вказати його припустимі дії. Досяжні цілі — результати, які виконавець може отримати при виконанні припустимих дій.

Система команд виконавця — повний перелік команд, за якими виконавець може виконати одну або серію припустимих дій. Виконавця можна уявити як пристрій із кнопковим керуванням. Настикання кнопки означає виклик однієї команди.

Відмова — подія, що виникає в разі виклику команди в неприпустимому для цієї команди стані середовища.

Розглянемо виконавця **Кресляр**, який призначений для побудови малюнків на полі розміром 5×5 клітинок (рис. 19.2) і вміє виконувати три команди: підніми перо, опусти перо, перейди до точки (x, y) .

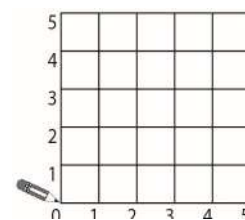


Рис. 19.2

На початку роботи **Кресляр** розташовується в точці $(0, 0)$ і тримає перо піднятим. Якщо перо опущене, під час пересування **Кресляра** за ним залишається слід.

Припустимим діям виконавця відповідає система команд:

Система команд виконавця	Дії за командами
Підніми перо	Підготуватися до переміщення без сліду
Опусти перо	Підготуватися до переміщення зі слідом
Перейди до точки (x, y)	Перейти до точки з координатами (x, y)

Відмова **Кресляра** виникає, якщо він отримує команду перейти до точки, яка розміщується за межами поля.

ПРИКЛАД. Складемо для виконавця **Кресляр** алгоритм побудови даху будиночка (рис. 19.3).

1. Перейди до точки $(0, 3)$.
2. Опустити перо.
3. Перейди до точки $(2, 5)$.
4. Перейди до точки $(3, 5)$.
5. Перейди до точки $(5, 3)$.
6. Перейди до точки $(0, 3)$.
7. Підніми перо.

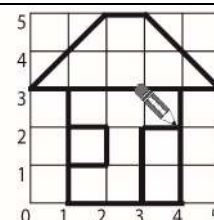


Рис. 19.3

Виконуючи алгоритм, виконавець може не розуміти смисл того, що він робить, і тим не менше отримувати потрібний результат, тобто виконавець діє формально (рис. 19.4).

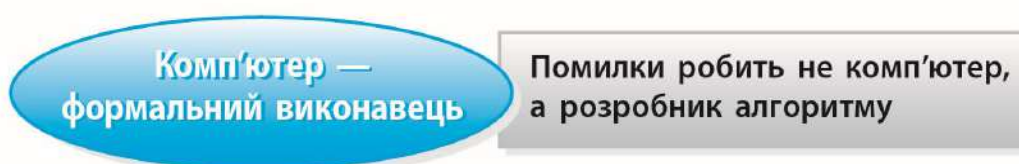


Рис. 19.4

ЦІКАВІ ФАКТИ. Аду Лавлейс вважають першою у світі програмісткою. Вона склала опис та програму для програмованої обчислювальної машини, проєкт якої у 1833 році розробив її вчитель, англійський математик Чарлз Беббідж.



Питання для самоперевірки

1. Поясніть поняття «виконавець алгоритму». Перелічіть характеристики виконавця.

2. Опишіть систему команд виконавця, щоб реалізувати алгоритм обчислення за формулою $p = \frac{1+x}{1-x}$.

3. Якими припустимими діями ви оснастили б автомат, що заміняє:

- а) касирку;
- б) вахтерку;
- в) двірника?

4. Виконавець уміє виконувати команди: помнож число на 2; знайди суму двох чисел. Складіть для виконавця алгоритм обчислення виразу $y = 2a + 3b + 4c$ для заданих a , b і c .

5*. Виконавець уміє виконувати команди: помнож число на 2; збільш число на 1. Складіть для виконавця алгоритм отримання:

- а) числа 4 з 1;
- б) числа 5 з 1;
- в) числа 100 з 1.

6*. Фірма «Електронні прилади» випустила автоматизовану ванну, пульт керування якої має дві кнопки: **Долити 5 л** і **Злити 3 л**. Складіть алгоритм доливання у ванну 4 л води за якомога меншу кількість команд.

Вправа 19

Завдання: скласти алгоритм, використовуючи систему команд виконавця.


1. Ознайомтесь, які команди може виконувати виконавець **Кресляр**:

Система команд виконавця	Дії за командами
Зроби крок Повернися	Пройти 1 см, залишаючи слід Повернутися на 90° ліворуч


2. Складіть для виконавця **Кресляр** алгоритм малювання. До початку малювання **Кресляр** спрямований вправо (→).

	1.	4.
	2.	5.
	3.	6.

3. Складіть для виконавця **Кресляр** алгоритм малювання.

	1.	6.
	2.	7.
	3.	8.
	4.	9.
	5.	10.

4. Складіть для виконавця **Кресляр** алгоритм малювання.

	1.	8.
	2.	9.
	3.	10.
	4.	11.
	5.	12.
	6.	13.
	7.	14.

6. Запропонуйте однокласникам і однокласницям виконати складені алгоритми. Чи правильно відтворюються малюнки за вашими алгоритмами?

Комп'ютерне тестування

Виконайте тестове завдання 19 з автоматичною перевіркою результату.

§ 20. СПОСОБИ ОПИСУ АЛГОРИТМУ. АЛГОРИТМІЧНІ СТРУКТУРИ

Форми подання алгоритмів

Існують *різні форми подання алгоритмів* — словесна, словесно-формульна, графічна, у вигляді програмного коду та інші — залежно від того, на якого виконавця орієнтований алгоритм.

У повсякденному житті найчастіше застосовується **словесна форма**. Алгоритм подається як послідовність окремих пронумерованих пунктів, кожен з яких містить команду на виконання певної дії. Команди виконують одну за одною в порядку зростання їх номерів, якщо немає спеціальної вказівки на перехід до виконання команди за іншим номером в алгоритмі.

Словесна форма подання алгоритму є найприйнятнішою для опису інструкцій побутового характеру, поведінки в разі виникнення певної надзвичайної ситуації, кулінарних рецептів тощо.

ПРИКЛАД 1. Подамо алгоритм приготування «швидкої» піци (рис. 20.1) у словесній формі.

1. Змішайте сметану з томатною пастою, додайте спеції.
2. Отриманим соусом намастіть основу для піци.
3. Зверху викладіть шматочки помідорів і болгарського перцю, половинки маслин.
4. Посипте натертим сиром.
5. Поставте піцу в духову шафу на 5 хвилин для запікання.



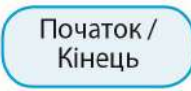

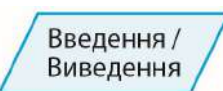

Рис. 20.1

Алгоритми обчислень зручно подавати у **вигляді формул**. Так, алгоритм обчислення площі прямокутного трикутника для виконавця **Учень 5 класу** можна подати у вигляді $s = \frac{a \cdot b}{2}$, де a, b — катети трикутника.

Записуючи алгоритми, часто комбінують словесне та формульне подання запису команд. Для **графічного подання** алгоритмів використовують *блок-схеми*.

При цьому команди подаються в окремих блоках, а послідовність виконання команд позначаються стрілками.

Кожна команда міститься в блоці певного вигляду:

Назва блоку	Опис
	Позначає початок або кінець алгоритму
	Позначає дію, яку потрібно виконати. Може бути позначена як указівка виконати окрему дію (додати два числа, накреслити лінію), так і послідовність логічно об'єднаних у блок дій (виконати розрахунки за заданими формулами, зробити малюнок), тобто певний процес
	Позначає введення вхідних даних або виведення вихідних даних
	Позначає перевірку деякої умови. Умова може набувати одного з двох значень — TRUE (істина) або FALSE (хибність)

Розглянемо подання алгоритму в графічному вигляді.

ПРИКЛАД 2. На рис. 20.2 зображено блок-схему алгоритму знаходження середнього арифметичного трьох чисел, де:

a, b, c — вхідні дані;

S — проміжний результат;

Sr — вихідні дані (результат виконання алгоритму).

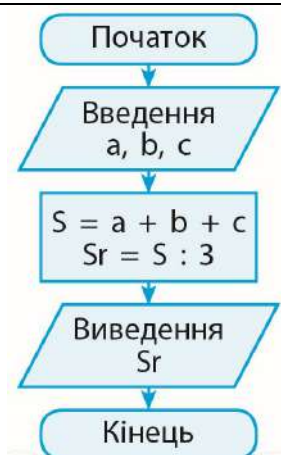


Рис. 20.2

Базові алгоритмічні структури

Під час конструювання алгоритмів використовують три базові алгоритмічні структури: *слідування*, *розгалуження*, *повторення*. Згадаймо означення цих алгоритмічних структур.



Слідування — це така форма організації вказівок в алгоритмі, за якої дії виконуються послідовно одна за одною, без пропусків або повторень (рис. 20.3).



Рис. 20.3

Алгоритм створення комп'ютерної програми

1. Складіть алгоритм.
2. Напишіть програму.
3. Відлагодьте програму на комп'ютері.
4. Отримайте результат розв'язування задачі.



Розгалуження — це така форма організації дій, коли залежно

від виконання або невиконання певної умови виконується одна з двох вказівок (рис. 20.4).

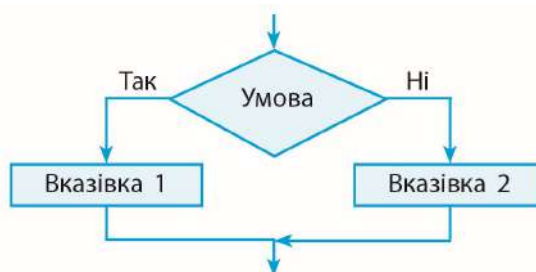


Рис. 20.4

Умову подають у вигляді запитання, сформульованого так, щоб допускалася лише одна з двох відповідей: «так» або «ні». Перевірка умови має бути допустимою дією виконавця.

Якщо умова істинна, то виконується **Вказівка 1** (гілка **Так**); якщо умова хибна, то **Вказівка 2** (гілка **Ні**).

Після виконання однієї з указівок виконавець переходить до наступної після розгалуження команди.

ПРИКЛАД 3. Складемо блок-схему алгоритму обирання розваги: якщо в касі є квитки, то придбати квиток і переглянути фільм, інакше — піти на прогулянку до парку (рис. 20.5).

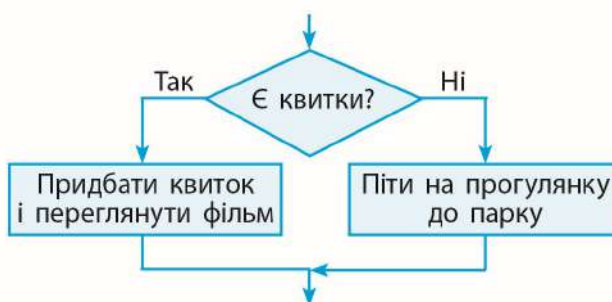


Рис. 20.5



Повторення (цикл) — це така форма організації дій, за якої

одна й та сама послідовність дій виконується кілька разів залежно від певної умови.

Сукупність вказівок, що повторюється під час кожного проходження (ітерації) циклу, називається **тілом циклу**.

Наприклад, на рис. 20.6 подано циклічний алгоритм, у якому певні вказівки повторюються доти, поки задана умова є істинною.

Під час виконання такого алгоритму спочатку перевіряється умова. Якщо умова *істинна*, то тіло циклу виконується й відбувається повернення на перевірку умови.

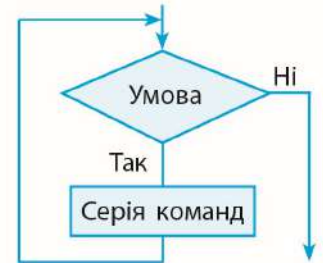


Рис. 20.6

Якщо умова є *хибною*, то виконання циклу припиняється.

! Якщо умова в команді повторення виявиться хибною за першої перевірки, то тіло циклу не виконається жодного разу.

! Якщо під час повторення циклу умова незмінно залишається істинною, то цикл може повторюватися нескінченно (кажуть, що програма «зациклена»).

ПРИКЛАД 4. Блок-схему алгоритму збирання яблук подано на рис. 20.7.

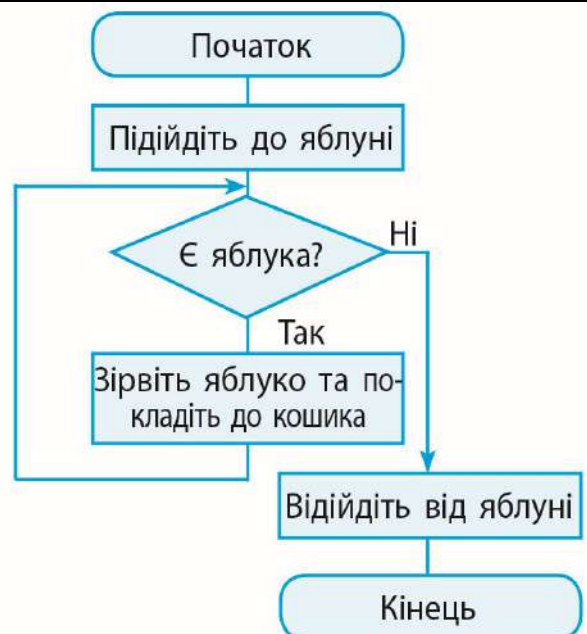


Рис. 20.7

Під час складання алгоритму розв'язування задачі необхідно дотримуватися чіткого плану дій.

1. Уважно прочитайте умову задачі.
2. З'ясуйте:
 - а) що дано — *вхідні дані* (аргументи);
 - б) що потрібно знайти — *вихідні дані* (результати).
3. Визначте спосіб розв'язування задачі та виявіть необхідні проміжні величини.
4. Складіть блок-схему алгоритму.
5. Перевірте правильність складання алгоритму для різних значень аргументів.

Програма та мова програмування

Програма — це алгоритм розв'язування певного завдання, записаний мовою програмування.



Мова програмування — це система позначень, яка використовується для запису алгоритмів для їх виконання за допомогою комп'ютера.

Існує багато мов програмування. Ви вже знайомі з візуальною мовою програмування **Scratch** і основними поняттями, які будуть корисними під час вивчення інших мов програмування.

Питання для самоперевірки

1. Назвіть основні способи опису алгоритмів.
2. Як подаються базові алгоритмічні структури на блок-схемі?
3. Наведіть приклади умов, які можуть бути використані для організації розгалуження.
4. Поясніть переваги подання алгоритму у вигляді блок-схеми.
5. Наведіть приклад алгоритму з розгалуженням, поданого у словесному вигляді.
6. Наведіть приклад алгоритму з повторенням, поданого у словесному вигляді.

Вправа 20

Завдання: виконати алгоритм, поданий у вигляді блок-схеми.

1. На рис. 20.8 подано блок-схему алгоритму обчислення площі прямокутного трикутника за формулою $s = \frac{a \cdot b}{2}$. Які написи потрібно вписати?

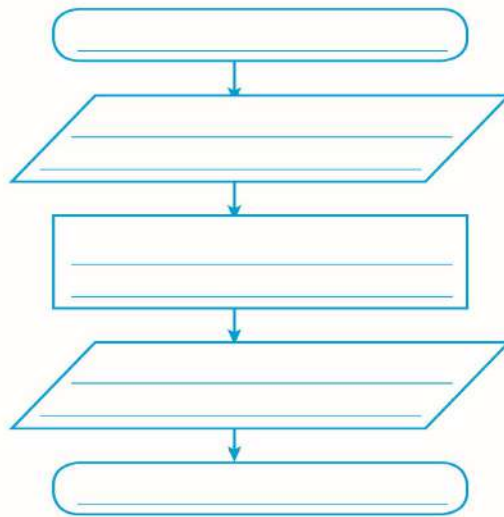


Рис. 20.8

2. Визначте істинність умови $a > 5$ за різних значень a . Заповніть таблицю.

a	8	5	2
$a > 5$			

3. Знайдіть значення величини c після виконання команди розгалуження для різних початкових значень b (рис. 20.9). Заповніть таблицю.

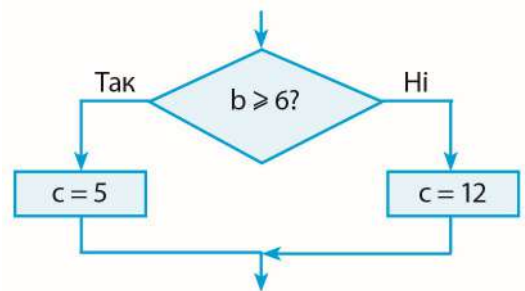


Рис. 20.9

b	4	6	10
c			

4. Впишіть потрібні написи в блок-схему команди розгалуження (рис. 20.10): якщо ціна c книжки не перевищує n гривень, то купити цю книжку і сувенір, в іншому випадку — купити тільки книжку.

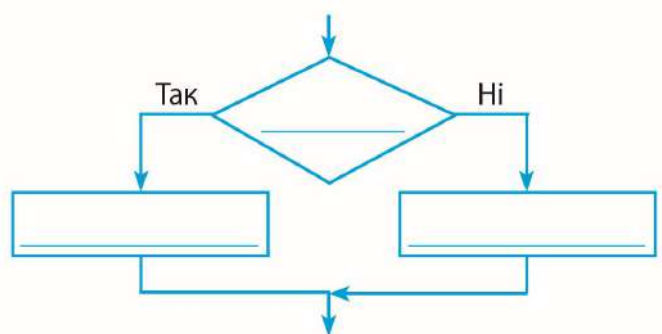


Рис. 20.10

5. Знайдіть значення величини x після виконання команди повторення для різних початкових значень x (рис. 20.11). Заповніть таблицю.

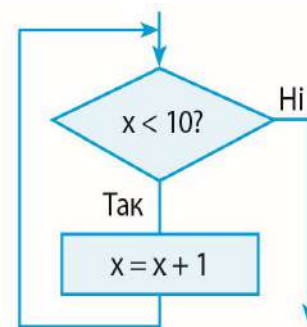


Рис. 20.11

Початкове значення x	4	6	10
Значення x після виконання циклу			

6*. Гра Баше.

Є 11 предметів. За один хід можна взяти 1, 2, 3 предмети. Програє та особа, якій дістанеться останній предмет.

Існує алгоритм виграшу для особи, яка робить перший хід:

1-й хід: узяти два предмети.

2-й хід і далі: брати стільки предметів, щоб їх кількість, узятих разом із суперником за черговий хід, у сумі становила 4.

Впишіть потрібні написи в блок-схему виграшного алгоритму для першого гравця/гравчині (рис. 20.12).

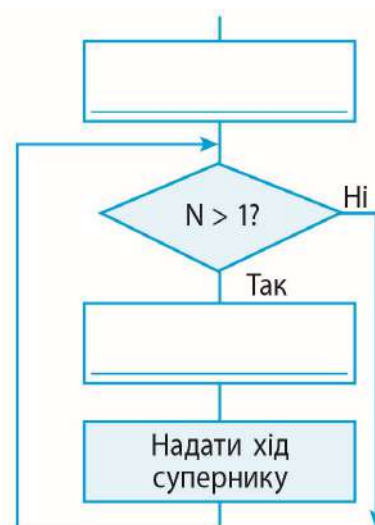


Рис. 20.12

Комп'ютерне тестування

Виконайте тестове завдання 20 з автоматичною перевіркою результату.

§ 21. СЕРЕДОВИЩЕ ОПИСУ ТА ВИКОНАННЯ АЛГОРИТМІВ

Починаємо знайомство з популярною сучасною мовою програмування **Python**, яка застосовується для розв'язування різних задач: написання прикладних програм, створення ігор, розробки вебсайтів.

ЦІКАВІ ФАКТИ

Мова програмування **Python** була створена в 1991 році нідерландським програмістом *Гвідо ван Россумом* (рис. 21.1) і названа ним на честь скетч-серіалу «Літаючий цирк Монті Пайтона» (англ. *Monty Python's Flying Circus*).



Рис. 21.1

Мова **Python** підтримується всіма операційними системами. Існують версії для **Linux**, **Windows**. Наразі використовують дві версії **Python**: **2.x** і **3.x**. У нашому підручнику розглядатимемо версію **Python 3**.

Установлення програмного середовища Python

Перш ніж почати програмувати на **Python**, середовище програмування потрібно встановити на комп'ютер. Завантажити файл для інсталяції можна із сайту **python.org** (рис. 21.2).



Рис. 21.2

Разом із **Python 3** на комп'ютер буде встановлено програму **IDLE** — орієнтоване на початківців середовище програмування, у якому є засоби для написання та налагодження програм мовою **Python**.

Алгоритм установлення програмного середовища Python

1. Зайдіть на сайт <http://www.python.org>
2. Клацніть **Downloads**, щоб відкрити сторінку завантаження (рис. 21.3).



Рис. 21.3

3. Клацніть кнопку з версією **Python 3.6.1**.

4. Після завантаження інсталяційного файлу двічі клацніть на ньому, щоб установити **Python** на своєму комп'ютері.

5. Запустіть **IDLE**. Для цього виберіть команди **Пуск** → **Всі програми** → **Python** → **IDLE**. Відкриється вікно **IDLE** (рис. 21.4).

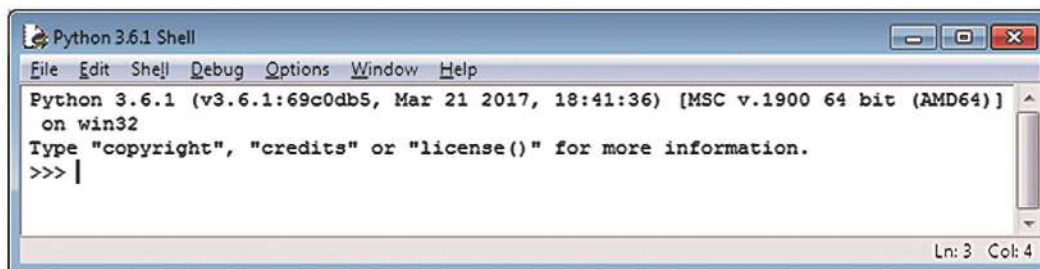


Рис. 21.4

Ми можемо починати програмувати, записуючи команди після позначки `>>>`.

Рядок `>>>` називається **запрошенням**, і його наявність означає, що комп'ютер готовий прийняти вашу першу команду.

У **Python** існують два види вікон: *вікно програми* і *вікно консолі (IDLE)*. У *вікні програми* можна вводити й зберігати програмний код для подальшого виконання, а у *вікні консолі* — вводити команди й одразу отримувати результат виконання.

Знайомство з IDLE

Вікно **IDLE** є вікном консолі. У ньому відображаються результати виконання програми й повідомлення про помилки в програмному коді.

На рис. 21.5 проілюстровано виконання команд у вікні **IDLE**.

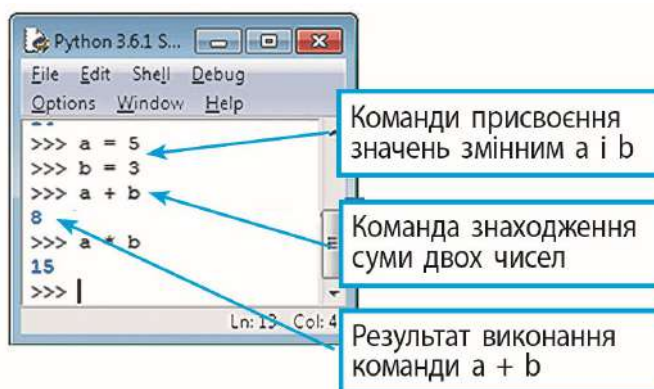


Рис. 21.5

Команди, які введено у вікні консолі, виконуються після натискання клавіші **Enter**, і результат одразу виводиться у вікні **IDLE**.

ПРИКЛАД 1. Щоб зрозуміти, як працює **IDLE**, після позначки `>>>` запишемо команду `print('Hello, World!')` і натиснемо клавішу **Enter**. У вікні **IDLE** буде виведено привітання (рис. 21.6).

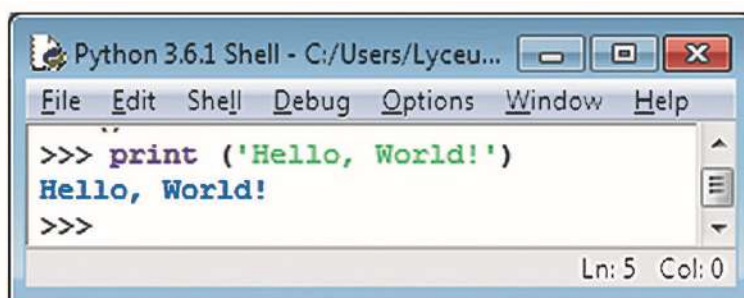


Рис. 21.6

Вікно **IDLE** зручно використовувати для того, щоб зрозуміти, що виконує та чи інша команда. Але для створення великих програм, які необхідно зберігати й редагувати, використовують вікно програми.

Вікно програми

Вікно програми призначене для введення та редагування тексту програми.

Щоб відкрити вікно програми, в **IDLE** виберіть команду **File** → **New File**. Відкриється окреме вікно програми, яка до першого збереження має ім'я **Untitled** (рис. 21.7).

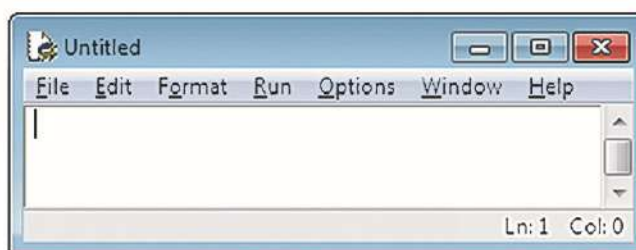


Рис. 21.7

На рис. 21.8 наведено алгоритм роботи з програмою в середовищі **Python** у вікні програми.



Рис. 21.8

ПРИКЛАД 2. Створимо програму розв’язування задачі за поданим раніше алгоритмом роботи з програмою.

Задача. Знайти суму і добуток двох чисел.

1. Уведемо код у вікні програми (рис. 21.9).
2. Виберемо команду **File** → **Save As**. Уведемо ім’я файлу **Перша програма** і натиснемо **Save**. Файл збережено з розширенням **.py**.

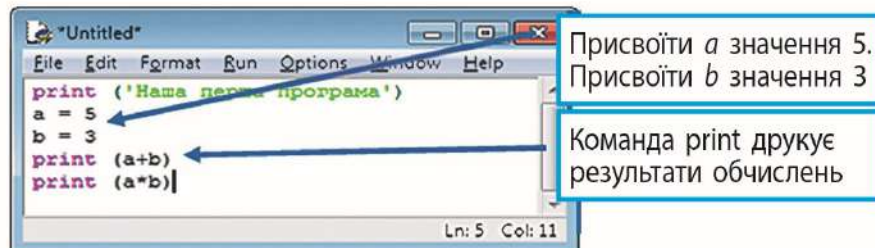


Рис. 21.9

3. Виберемо команду **Run** → **Run Module** (або натиснемо **F5**).
4. Переглянемо результат роботи програми у вікні консолі **IDLE** (рис. 21.10).

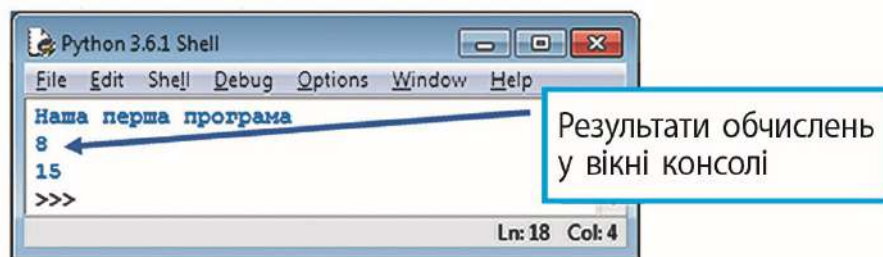


Рис. 21.10

Програму не можна запустити на виконання, поки вона не збережена. У разі спроби виконати програму без збереження з’явиться попередження. Зверніть увагу, що попередження виділено червоним кольором.

У **Python** різні складові програмного коду виділяються певними кольорами:

	Команди Python , наприклад, print , — фіолетового кольору
	Рядки в лапках виділені зеленим кольором. Якщо дужки навколо текстового рядка теж виділені зеленим, десь не вистачає лапок
	Службові слова мови Python (if , while ті інші) — помаранчевого кольору
	Помилки у вікні програми та повідомлення про помилки у вікні консолі виділяються червоним

Кольорові підказки допомагають уникати помилок під час уведення тексту програми.

Питання для самоперевірки

1. Які види вікон є у програмному середовищі **Python**?
2. Які повідомлення відображаються у вікні консолі?
3. Як відкрити вікно програми?
4. Для чого призначене вікно програми?
5. Опишіть алгоритм роботи з програмою в середовищі **Python**.
6. Поясніть, чому складові програмного коду виділяються різними кольорами.

Вправа 21

Завдання: виконати програму у вікні програми.

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Завантажте програму **Python** → **IDLE**.

Виберіть команду **File** → **New File** (**Файл** → **Новий файл**), щоб відкрити вікно програми.

Наберіть текст програми додавання двох цілих чисел:

```
a = 5
```

```
b = 10
```

```
c = a+b
```

```
print('c = ', c)
```

2. Для збереження файлу виберіть команду **File** → **Save As** (**Файл** →

Зберегти як).

У вікні збереження файлу виберіть власну папку, наберіть назву файлу **Vprava21** та натисніть **Save** (**Зберегти**). Файл збережено з розширенням **.py**. Запишіть повну адресу файлу.

3. Для запуску програми на виконання виберіть команду **Run** → **Run Module** (**Виконати** → **Виконати модуль**) або натисніть клавішу **F5**. Програма почне виконуватися.

Якщо з'явиться повідомлення про помилку, перевірте, чи правильно набрано текст програми.

Перегляньте результат виконання програми у вікні консолі. Закрийте вікно **IDLE**.

4. Повторно виконайте програму. Для цього ввійдіть у власну папку, установіть курсор на позначку програми й відкрийте контекстне меню, у якому виберіть команду **Edit with IDLE** (**Змінити в IDLE**).

5*. У вікні програми змініть значення змінних *a* і *b*:

a = 10

b = 17

6. Збережіть файл і запустіть програму на виконання. Запишіть результат виконання програми. Закрийте вікно **IDLE**.

Комп'ютерне тестування

Виконайте тестове завдання 21 з автоматичною перевіркою результату.

§ 22. ОСНОВНІ ПОНЯТТЯ МОВИ ПРОГРАМУВАННЯ PYTHON

Для написання програми певною мовою програмування необхідно знати алфавіт мови, правила запису команд та правила їх використання в програмі.

Алфавіт мови Python

Як і будь-яка мова, **Python** має *алфавіт, синтаксис, семантику*.



Алфавіт мови — набір символів, що може використовуватись у програмному коді.

Під час створення програм мовою **Python** можна використовувати такі символи:

- літери латинського алфавіту: **A...Z, a...z**;
- цифри: **0...9**;
- знаки арифметичних операцій, спеціальні символи: **+ — / \ ^ = < > () . , : ; ' # _**.

У мові **Python** використовують також комбінації символів і службові слова, що мають фіксований зміст:

- комбінації символів: **<=, >=, <>, ==, !=, ****;
- службові слова: **and, elif, if, as, else, import** тощо.



Синтаксис мови — сукупність правил побудови команд мовою програмування.

Якщо ви неправильно напишете ключове слово, що позначає команду, **Python** не зможе зрозуміти введену вами команду і виведе у відповідь повідомлення про синтаксичну помилку **SyntaxError**. Місце помилки у вікні консолі помічається червоним кольором. Після команди, що містить помилку, виводиться повідомлення про помилку. Уважно читайте такі повідомлення — це допоможе зрозуміти, у чому помилка, і виправити її.

ПРИКЛАД 1. Розглянемо програмний код:

```
>>> (x+5))
```

```
SyntaxError: invalid syntax
```

У виразі допущено помилку: надруковано зайву дужку — у консолі з'являється повідомлення про синтаксичну помилку.



Семантика мови — сукупність правил виконання

комп'ютером команд, записаних мовою програмування. Із семантикою мови **Python** ми будемо знайомитись у міру вивчення команд.

Величини в мові Python

Окремий інформаційний об'єкт (число, символ, рядок тощо) називають величиною. Кожна величина характеризується *розміром виділеної пам'яті для зберігання, назвою (ідентифікатором), типом і значенням*.

Вид величини визначає, чи можна змінювати значення цієї величини в програмі. Величина може бути *константою* (тобто *постійною*) або *змінною*.



Константи — це величини, значення яких не можуть

змінюватися в ході виконання програми.

Прикладом константи може бути число (5, 1.23) або рядок («Це рядок!»).



Змінні — це величини, значення яких можуть змінюватися в

ході виконання програми.

Змінні потрібні для зберігання даних. Змінним дають імена (ідентифікатори).

З'ясуємо правила, яких необхідно дотримуватися під час **надання імен змінним**.

- Першим символом імені має бути літера чи знак нижнього підкреслювання «_».

- Решта імені може складатися з літер, чисел або знаків нижнього підкреслювання.

- Не можна використовувати спеціальні символи, такі як /, # або @. Не можна використовувати пробіли.

- Імена змінних чутливі до регістру. Наприклад, **myname** і **myName** — це різні змінні.

- Не можна надавати змінним імена команд, наприклад **print**.

ПРИКЛАД 2.

Правильні імена: `i`, `__my_name`, `name_23`, `a1`, `b2`.

Неправильні імена: `2things`, `this is spaced out`, `my-name`.

Типи величин у мові Python

Тип величини визначається *обсягом пам'яті*, необхідним для її збереження, *множиною припустимих значень* величини та *операціями*, які можна над нею виконувати.

Основними типами величин є *числа* й *рядки*.

У **Python** є два типи числових даних: *цілі числа* (**int**) і *дійсні* (**float**) — дробові числа.



Цілими числами називають натуральні числа (1, 2, 3...), їм протилежні числа (-1, -2, -3...) і число 0.

Кількість учнів та учениць у класі, кількість предметів ми зазвичай вказуємо за допомогою цілих чисел. Дійсні числа — це десяткові дроби. Вони потрібні, коли ми хочемо вказати частину чого-небудь, наприклад, 3.5 м, 1.25 грн. Як роздільник між цілою та дробовою частиною дійсного числа використовується крапка.



Рядок — це взята в одинарні або подвійні лапки послідовність будь-яких символів: цифр, літер, розділових знаків та ін. Рядки належать до типу **str**.

У змінних рядкового типу зберігають фрагменти тексту.

Щоб дізнатися, до якого типу належить змінна або константа, можна скористатися командою **type** (рис. 22.1).

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
>>> a=12 # ціле число
>>> b=3.14 # дробове число
>>> c='Hello!' # рядок
>>> print (a, b, c)
12 3.14 Hello!

Python 3.4.4 Shell
File Edit Shell Debug Options Window Help
>>> type (25)
<class 'int'>
>>> type (3.14)
<class 'float'>
>>> type ('a book')
<class 'str'>
>>> type (True)
<class 'bool'>
>>> |
Ln: 37 Col: 4
```

Рис. 22.1

Указівка присвоєння



Значення — характеристика величини, яка може багаторазово

змінюватися в процесі опрацювання інформації.

Щоб створити змінну в **Python**, необхідно дати їй назву й присвоїти значення (рис. 22.2). Загальний вигляд команди присвоєння: $A = B$, де **A** — назва змінної, **B** — константа, змінна або вираз.



Рис. 22.2

Змінну часто порівнюють зі скринькою, в якій зберігається значення величини.

ПРИКЛАД 3. Створимо змінну *a*:

```
a = 7
```

Вираз присвоювання наказує комп'ютеру запам'ятати число 7 у змінній *a*.

Схема виконання вказівки присвоювання: спочатку обчислюється значення виразу в правій частині вказівки присвоювання, потім це значення надається змінній, ім'я якої записане в лівій частині (рис. 22.3).

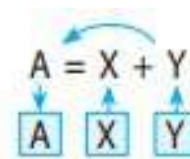


Рис. 22.3

Під час виконання команди присвоювання ділянка пам'яті, відведена для змінної, заповнюється новим значенням, одночасно знищуючи старе. Тому в правій частині виразу може бути вказане тільки ім'я змінної.

ПРИКЛАД 4. Наступні команди присвоюють змінній **rabbits** значення 5, потім те саме значення присвоюють змінній **hats**:

```
>>> rabbits = 5
```

```
>>> hats = rabbits
```

У виразах можна використовувати змінні. Якщо в правій частині оператора присвоювання записати вираз, то змінна в лівій частині набуває значення виразу.

ПРИКЛАД 5. Нехай $a = 10$. Тоді після виконання вказівки присвоювання $a = a + 5$ змінна *a* набуде значення 15.

При спробі виконати оператор $a + 2 = b$ буде виведено повідомлення про синтаксичну помилку (рис. 22.4).

```
>>> a=5
>>> a+2=b
SyntaxError: can't assign to operator
>>> |
```

Рис. 22.4

Питання для самоперевірки

1. Назвіть основні характеристики величини.
2. Які послідовності символів неприпустимо використовувати як імена (ідентифікатори) й чому?
 - а) **suma**;
 - б) **w1**;
 - в) **primer 1**;
 - г) **(sum)**;
 - г) **a-4**;
 - д) **if**;
 - е) **8a**.
3. До яких типів належать величини:
 - а) кількість учнів і учениць у класі;
 - б) вага кошика з яблуками в кг;
 - в) прізвище учня/учениці.
4. Чому дорівнює значення x після виконання послідовності присвоєвань:
 - а) $y = 2; x = y$;
 - б) $x = 8; x = x + 2$;
 - в) $x = 5; x = -x$;
 - г) $x = 10; x = x + 3$?
5. Після виконання оператора $x = y + x$ нові значення змінних $x = 10$, $y = 3$. Чому дорівнювали x і y до виконання оператора присвоєвання?
6. Після виконання оператора $x = y + x$ нові значення змінних $x = 3$, $y = 10$. Чому дорівнювали x і y до виконання оператора присвоєвання?

Вправа 22

Виконати дії над величинами в середовищі **Python**.

1. Завантажте програму **Python** → **IDLE**.

Після запрошення `>>>` запишіть команду створення змінної a :

```
>>>a = 10
```

і натисніть клавішу **Enter**. Інтерпретатор створить змінну a і надасть їй значення 10.

Щоб перевірити значення змінної a , після запрошення `>>>` запишіть назву змінної і натисніть **Enter**.

2. Дізнайтеся тип змінної a . Для цього виконайте команду:

```
>>>type(a)
```

3. Виконайте команду **Збільшити значення змінної a на 3**:

```
>>>a = a+3
```

Виведіть значення змінної a :

```
>>>a
```

4. Виконайте команду **Збільшити значення змінної a вдвічі** і виведіть значення змінної a .

5*. Створіть змінну c рядкового типу:

```
>>>c = "Вітаю!"
```

Дізнайтеся тип змінної c .

Виконайте команду **Продублювати значення змінної c** :

```
>>>c = c*2
```

6*. Виведіть значення змінної c . Поясніть результат виконання команд.

Комп'ютерне тестування

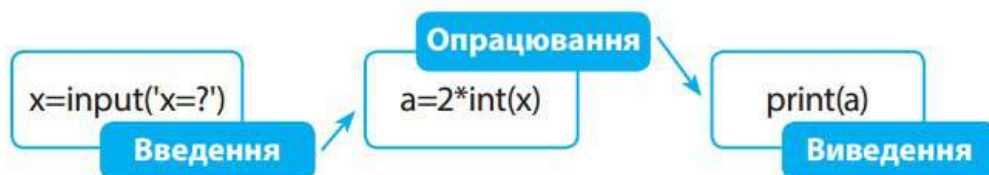
Виконайте тестове завдання 22 з автоматичною перевіркою результату.

§ 23. ЛІНІЙНІ АЛГОРИТМИ ТА ПРОГРАМИ



Алгоритм, у якому використовується тільки структура слідування, називається **лінійним**.

Будь-яка програма має прийняти вхідні дані (введення), опрацювати їх і повернути результат (виведення). У програмах, що реалізують лінійні алгоритми, використовуються команди введення даних, присвоювання та виведення. На рисунку наведено логічну структуру програми, у якій реалізовано лінійний алгоритм.



Введення даних

Команда **input()** призначена для введення даних із клавіатури.

Коли програма зустрічає команду `input()`, вона припиняє роботу й очікує, поки користувач/користувачка введе дані й натисне **Enter**. У дужках може бути записана підказка, що саме потрібно ввести. Ця підказка виводиться на екран.

ПРИКЛАД 1. Випробуємо у вікні консолі, як працює команда введення `input()`:

```
>>> name = input('Як тебе звати?')
Як тебе звати?Петро
>>> name
'Петро'
```

Значення, отримане від команди `input()`, **Python** сприймає як рядок (послідовність літер), навіть якщо ми ввели число.

ПРИКЛАД 2. У разі спроби додати до значення змінної *a* число 3 виникне помилка, оскільки **Python** не знає, як додати число до рядка.

```
>>> a = input('a = ?')
a =? 5
>>> a+3
TypeError: must be str, not int
```

Необхідно виконати перетворення введеного значення на число за допомогою функції `int()`. Функція `int(s)` перетворює рядок *s* на ціле число.

ПРИКЛАД 3. Тепер помилки немає:

```
>>> a = input('a = ?')
a =? 3
>>> int(a)+5
8
```

При введенні числових значень зручно застосовувати функції перетворення типів до значення, яке повертає функція `input()`:

```
a = int(input('a = ?'))
```

Під час запису дійсного числа у формі десяткового дробу для розділення цілої та дробової частин як десятковий роздільник використовується крапка «.». Функція `float(s)` перетворює рядок *s* на дробове число.

ПРИКЛАД 4. Введення з клавіатури дійсного значення:

```
>>> x = float (input('x = ?'))
x = ?5.25
>>> x+3
8.25
```

Виведення значень змінних

Команда **print()** виводить текст у вікно консолі:

print(<список виведення>)

ПРИКЛАД 5. За допомогою команди **print** можна дізнатися значення змінної.

```
>>> print(rabbits)
5
```

У списку виведення можуть бути константи, змінні, вирази.

ПРИКЛАД 6. Вивести значення змінної *x*:

```
print('x = ', x')
```

Якщо потрібно вивести значення кількох змінних або виразів, їх необхідно перелічити через кому:

```
>>> x = 4
>>> print (x, 2*x, 3*x)
4 8 12
```

За допомогою команд **input()** і **print()** можна організувати діалог користувача з програмою:

```
>>> name = input('Як тебе звати?')
Як тебе звати? Петро
>>> print('Привіт, ', name)
Привіт, Петро
```

Коментарі в програмі

Коментар — це текст, призначений для читання людиною, а не комп'ютером. Коментар — це підказка, яку програмісти/програмістки записують у своїй програмі. Щоб комп'ютер відрізняв команди від коментарів, у мові **Python** перед текстом коментаря ставиться знак **#**. Редактор **IDLE** виділяє коментарі червоним кольором, нагадуючи про те, що ці фрагменти коду будуть проігноровані.

ПРИКЛАД 7. Коментар пояснює призначення наступної команди:

```
# Запит імені користувача  
s = input('Як тебе звати?')
```

Питання для самоперевірки

1. Чому виникає помилка при спробі виконання коду:

```
a = input('Уведіть значення а:')  
b = a+10
```

2. Для чого призначена команда **print()**?

3. Назвіть константи та змінні у списку виведення:

```
print('a = ', a, 5, '3*b', 3*b)
```

4. Що буде виведено у вікно консолі в результаті виконання коду:

а) a = 5

в) a = 5

```
print("a = ", a)
```

```
print("a**2 = ", a**2)
```

б) a = 5

г) a = 5

```
b = 7
```

```
b = 7
```

```
print("a+b = ", a+b)
```

```
print("Площа дорівнює", a*b, "кв. м")
```

5. Поясніть помилки в записі операторів:

а) `a = int(input('a = ?'))`

б) `print("s = ", "v*t", "км")`

6. Відновіть можливий варіант тексту програми, якщо після роботи програми вікно консолі має вигляд:

а) a=? 3

б) Швидкість? 15

b=? 10

Час? 2

a+b = 13

Пройдений шлях 30 км

Вправа 23

Скласти програму-гру «Нісенітниця».

Алгоритм гри такий. Програма просить увести відповіді на запитання «Хто?», «Де?», «Коли?» і зберігає введені дані в трьох різних змінних. Зі створених змінних із додаванням жартівливої кінцівки формується речення та виводиться на екран.

1) Відкрийте вікно **IDLE** і створіть нове вікно програми.

2) Уведіть текст програми:

```
c1 = input('Хто?')
```

```
c2 = input('Де?')
```

```
c3 = input('Коли?')
```

```
phrase = c1+' '+c2+' '+c3+' співає пісню'
```

```
print(phrase)
```


3) Збережіть файл у власній папці з іменем **Вправа23**. Запустіть програму на виконання, дайте відповіді на питання програми, проаналізуйте результат виконання програми.

4*) Додайте до програмного коду оператор для введення відповіді на питання «Що робив?». Збережіть відповідь у змінній **c4**.

5*) Додайте значення змінної **c4** до рядка **phrase**.

6) Збережіть програму й запустіть її на виконання.

Комп'ютерне тестування

Виконайте тестове завдання 23 з автоматичною перевіркою результату.

§ 24. МАТЕМАТИЧНІ ОПЕРАТОРИ МОВИ PYTHON

Майже в усіх програмах вам доведеться використовувати математичні операції, наприклад для підрахунку балів в ігровій програмі, розв'язування рівнянь, руху об'єктів у вікні програми.

Обчислення здійснюються за допомогою арифметичних виразів. Арифметичні вирази будуються з імен змінних, констант, знаків арифметичних операцій і круглих дужок так само, як у математиці. З арифметичними діями додавання і множення, доступними в **Python**, ми познайомилися в попередніх параграфах.

Символи **+**, **-**, ***** тощо, що використовуються для позначення арифметичних дій (операцій), прийнято називати **операторами**.

Об'єкти (змінні або константи), над якими виконуються операції, називають **операндами** (рис. 24.1).



Рис. 24.1

Математичні оператори Python

Розглянемо математичні операції та відповідні їм символи операторів:

Операція	Символ оператора	Приклад	Результат
Додавання	+	a = 15+3	a = 18
Віднімання	-	b = 15-3	b = 12
Множення	*	c = 8*2	c = 16
Ділення	/	d = 7/2	d = 3.5

Обчислення неповної частки від ділення	//	e = 7//2	e = 3
Обчислення остачі від ділення	%	f = 7%2	f = 1
Піднесення до степеня	**	j = 7**2	j = 49

При виконанні декількох операцій в одному виразі, необхідно враховувати їх пріоритет (порядок виконання):

- 1) піднесення до степеня (**);
- 2) множення (*) і ділення (/), цілочисельне ділення (//), одержання остачі від цілочисельного ділення (%);
- 3) додавання (+) і віднімання (-).

Якщо операції мають однаковий пріоритет, то вони виконуються зліва направо по черзі. Для того, щоб змінити порядок виконання, можна користуватися дужками.

ПРИКЛАД 1. Обчислити значення змінних **a** і **b**:

$$a = 10 + 2 * 3 ** 2 \quad \# a = 28$$

$$b = (10 + 2) * 3 ** 2 \quad \# b = 108$$

Під час запису в тексті програми арифметичних виразів потрібно дотримуватися правил лінійного запису (рис. 24.2):

- вираз має бути записаний у вигляді лінійного ланцюжка символів;
- не можна опускати знак операції множення;
- порядок виконання операцій одного пріоритету регулюється дужками.

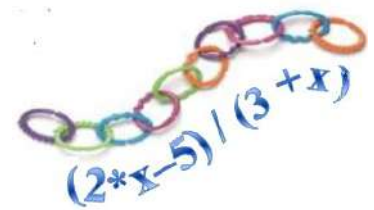


Рис. 24.2

ПРИКЛАД 2. Змінній *a* присвоїти значення виразу $\frac{2x-5}{3+x}$:

$$a = (2*x-5)/(3+x)$$

Приклад 3. Розв'язати задачу: відомо, що деяка подія відбулась *x* секунд тому. Виразити цей часовий період у годинах, хвилинах і секундах.

```
x = int(input('Кількість секунд?'))
h = x//3600
print(h, 'год')
m = (x%3600)//60
print(m, 'хв')
s = (x%3600)%60
print(s, 'сек')
```

```
Кількість секунд?10000
2 год.
46 хв.
40 сек.
```

Випадкові числа

При створенні ігрових або тестових програм іноді потрібно вибрати число з певного проміжку випадковим чином. У програмуванні випадкові числа отримують за допомогою спеціальних пристроїв — *генераторів випадкових чисел*. Щоб отримати випадкове число, необхідно за допомогою команди **import** завантажити модуль **random**:

```
from random import*
```

Функція **randint(x1, x2)** вибирає ціле випадкове число в діапазоні від **x1** до **x2**.

ПРИКЛАД 4. Отримаємо випадкове число в діапазоні від 1 до 10.

```
>>> from random import*  
>>> randint(1, 10)  
8  
>>> randint(1, 10)  
6
```

Приклад 5. Для навчальної програми з вивчення таблиці множення згенерувати приклад із випадковими значеннями множників.

```
from random import*  
x = randint(2, 9)  
y = randint(2, 9)  
print(x, '*', y, ' = ?')  
vidp = int(input('Добуток: '))
```

```
2 * 3 =?  
Добуток: 6  
>>> |
```

Питання для самоперевірки

1. Обчисліть значення виразів:

- а) $7/2$; г) $123//100$;
- б) $7//2$; г) $123\%10$;
- в) $7\%2$; д) $(123//10)\%10$.

2. Обчисліть значення, якого набуває змінна **a** в результаті виконання оператора присвоювання, якщо **b = 4**:

- а) $a = 37 - b**2$; в) $a = (b+12)/2$;
- б) $a = 100 - 2*b**2$; г) $a = b+12/2$.

3. Запишіть вирази за правилами лінійного запису:

- а) $\frac{x-5}{2x}$; г) $\frac{x}{2+x}$;
- б) x^5 ; г) $\frac{35x}{x+y} - \frac{8x+5}{4}$.

в) $2x^2 - 3x + 5$;

4. Як отримати випадкове число в діапазоні від 1 до 100? Як отримати випадкові числа? Наведіть приклади задач, для яких потрібні випадкові числа.

5. Запишіть оператори присвоювання, які реалізують такі дії:

а) змінній s присвоїти значення суми змінних a і b ;

б) обчислити значення добутку змінних a і b й результат присвоїти змінній c ;

в) подвоїти значення змінної a ;

г) змінну a збільшити на 10;

г) змінній n присвоїти значення неповної частки від ділення 100 на 7;

д) змінній k присвоїти значення остачі від ділення значення змінної a на 7.

6. Напишіть програму для обчислення площі та периметра прямокутної кімнати, ширина якої a м, довжина b м. Виконайте програму для $a = 3,9$ м, $b = 7,2$ м.

Вправа 24

Розв'яжіть задачу в середовищі **Python**.

Задача. Обчислити суму цифр двоцифрового числа a .

1) Завантажте програму **Python** → **IDLE**. В **IDLE** виберіть команду **File** → **New File**, щоб відкрити вікно програми.

2) У вікні програми наберіть текст програми:

```
a = 25
```

```
a1 = a//10
```

```
a2 = a%10
```

```
sum = a1+a2
```

```
print(sum)
```

3) Для збереження файлу виберіть команду **File** → **Save As**. У вікні збереження файлу виберіть власну папку, введіть ім'я файлу **Вправа 24** і натисніть **Save**.

4) Для запуску програми на виконання виберіть команду **Run** → **Run Module** або натисніть **F5**. Перегляньте результат виконання програми. Запустіть програму кілька разів, змінюючи значення a .

5*) Змініть програму таким чином, щоб значення змінної a вводилося з клавіатури. Запустіть програму і введіть значення $a = 85$.

6*) Змініть програму таким чином, щоб обчислювалася сума цифр трицифрового числа. Запустіть програму і введіть значення $a = 825$.

Збережіть програму, закривши вікно **IDLE**.

Комп'ютерне тестування

Виконайте тестове завдання 24 з автоматичною перевіркою результату.

ПРАКТИЧНА РОБОТА 7

Складання та виконання лінійних алгоритмів

Завдання: створити програму «Калькулятор піци».

Обладнання: комп'ютер зі встановленим середовищем програмування Python.

Програма має виконати такі дії: запитати, скільки піц бажає замовити покупець; запитати ціну піци, зазначену в меню; обчислити підсумкову вартість покупки, ураховуючи знижку 10 % на честь ювілею піцерії; вивести ціну покупки.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Відкрийте вікно **IDLE** і створіть нове вікно програми.

2. Заповніть пропуски в записі оператора для введення кількості піц:

number = int(_____)

Запишіть оператор у вікні програми.

3. Заповніть пропуски в записі оператора для введення ціни однієї піци:

cost = int(_____)

Запишіть оператор у вікні програми.

4. Заповніть пропуски в записі операторів для обчислення ціни покупки з урахуванням знижки:

total = _____

discount = total*0.1

total = _____

Запишіть оператор у вікні програми.

5. Додайте до програмного коду оператори виведення значень змінних **discount** і **total**.

6. Збережіть файл у власній папці з ім'ям **Pizza.py**. Запустіть програму на виконання, перевірте її для різних початкових даних. Запишіть початкові дані та отримані результати:

Кількість піц _____

Ціна 1 піци _____

Ціна без знижки _____

Ціна зі знижкою _____

Закінчить роботу, закривши вікно **IDLE**.

Зробіть висновок: як складати та виконувати лінійні алгоритми для розв'язування задач.

§ 25. «ЧЕРЕПАШАЧА» ГРАФІКА

У середовищі **Python** Черепашкою зветься уявний робот — пристрій, який переміщується по екрану й повертається в заданих напрямках, водночас залишаючи (або, за вибором, не залишаючи) за собою намальований слід заданого кольору й ширини.

Положення та напрямок руху Черепашки відображає невелика чорна стрілочка, яка повільно пересувається по екрану. Це дає змогу відстежити рух Черепашки та зрозуміти, яким чином кожен рядок коду впливає на траєкторію її руху.



Черепашка допоможе нам вивчити основи комп'ютерної графіки, і ми будемо малювати за її допомогою цікаві рисунки.

Система координат

Результат виконання Черепашкою команд відображається у графічному вікні **Python Turtle Graphics** (рис. 25.1).

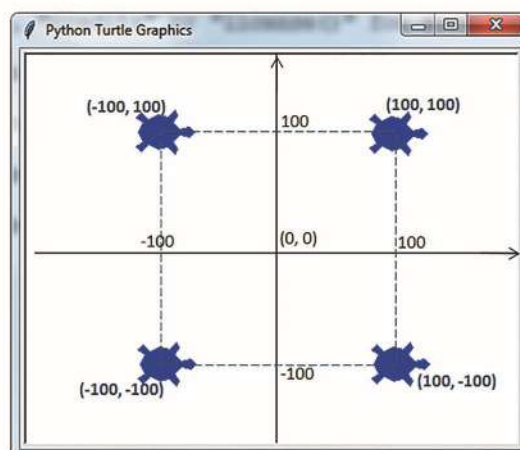


Рис. 25.1

Для визначення місцезнаходження Черепашки використовують координати. Ви вже знаєте, що таке координатна пряма, і вмiєте визначати положення точки на прямій.

Але для малювання нам доведеться користуватися орієнтирами не тільки вздовж прямої, а й на площині. Будь-яка точка у вікні **Python Turtle Graphics** може бути задана парою чисел (**X, Y**).

Координатні осі — це дві координатні прямі, які перетинаються під прямим кутом. Центр вікна **Python Turtle Graphics** — точка перетину невидимих координатних осей — точка з координатами (0, 0). Вертикальна координата **Y** зростає знизу догори, а горизонтальна **X** — зліва направо.

На уроках математики ви працювали з числами, розташованими на координатній прямій праворуч від 0. Але горизонтальну числову пряму можна продовжити вліво, а вертикальну — вниз від 0, а на променях ліворуч і знизу від 0 розташовані числа зі знаком мінус (–).

Повернемося до рис. 25.1. На ньому зображені черепашки і вказано координати їх розташування.

Команди Черепашки

Для завантаження команд для роботи з Черепашкою потрібна команда:
from turtle import*

Після введення цієї команди ви можете подавати Черепашці команди малювання.

Черепашка на початку малювання знаходиться у точці з координатами (0, 0). У процесі малювання ви можете дізнатися поточні координати Черепашки за допомогою функції **position()**.

Приклад 1. Вивести в консоль координати місцезнаходження Черепашки після виконання команд малювання. Результат виконання програми наведено на рис. 25.2.

```
from turtle import*  
forward(50)  
left(90)  
forward(50)  
right(90)  
forward(50)  
print(position())
```

```
=== RESTART: C:/Use  
(100.00,50.00)  
>>> |
```

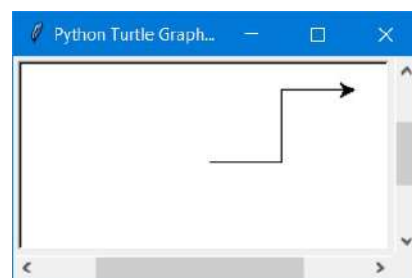


Рис. 25.2

За допомогою команди **speed(n)** ви можете змінювати швидкість руху Черепашки. Значення **n** може бути від 1 (повільно) до 10 (швидко); якщо **n = 0**, малюнок буде створено миттєво.

За замовчуванням «перо» Черепашки опущене на поверхню вікна, і Черепашка залишає слід під час пересування. Для керування пером призначені команди:

- **down()** — опустити перо;
- **up()** — підняти перо.

Розглянемо команди малювання та їх призначення:

Команда	Призначення
width(n)	Установити ширину сліду Черепашки в n пікселів
forward(n)	Проповзти вперед n кроків (пікселів)
left(angle)	Повернути ліворуч на angle градусів
right(angle)	Повернути праворуч на angle градусів
circle(r)	Намалювати коло радіуса r
circle(r, angle)	Намалювати дугу радіуса r з градусною мірою angle
dot(size, color)	Намалювати точку діаметра size кольору color . Параметр color необов'язковий
clear()	Очистити область малювання

Розглянемо приклади застосування команд.

Приклад 2. Намалюємо трикутник, використовуючи команди малювання.

Уведіть ці команди у вікні консолі:

```
>>> from turtle import *
>>> forward(100)
>>> right(120)
>>> forward(100)
>>> right(120)
>>> forward(100)
```

Черепашка рухається, залишаючи за собою слід. Результат виконання команд малювання наведено на рис. 25.3.

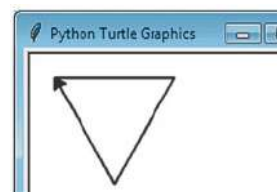


Рис. 25.3

Приклад 3. Намалювати коло радіуса 50 з центром у точці (0; 0) і товщиною лінії 3.

Команда **circle(50)** малює коло радіуса 50, яке повністю знаходиться зліва від положення черепашки. Оскільки черепашка на початку дивиться вправо, то точка **C** (точка початку малювання) буде найнижчою точкою кола.

Знайдемо координати точки **C**: оскільки радіус кола за умовою дорівнює 50 пікселів, координати точки **O(0, 0)**, то **OC = 50**. Точка **C** знаходиться на 50 пікселів нижче осі **OX**, її координати **C(0, -50)** (рис. 25.4).

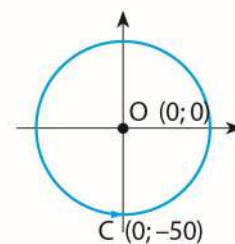


Рис. 25.4

Алгоритм малювання буде таким:

```
from turtle import*
```

```
width(3)
```

```
up() # Підняти перо, щоб при пересуванні в т. С не залишати сліду
```

```
goto(0,-50) # Пересунути черепашку в точку С
```

```
down() # Опустити перо, щоб почати залишати слід
```

```
circle(50) # Намалювати коло
```

Приклад 3. Намалювати смайлик (рис. 25.5).

```
from turtle import*
```

```
circle(50) # Малювання обличчя
```

```
up()
```

```
goto(-20, 60)
```

```
down()
```

```
dot(20) # Малювання лівого ока
```

```
up()
```

```
goto(20, 60)
```

```
down()
```

```
dot(20) # Малювання правого ока
```

```
up()
```

```
goto(20, 40)
```

```
left(90)
```

```
down()
```

```
circle(20, -160) # Посмішка
```



Рис. 25.5

Питання для самоперевірки

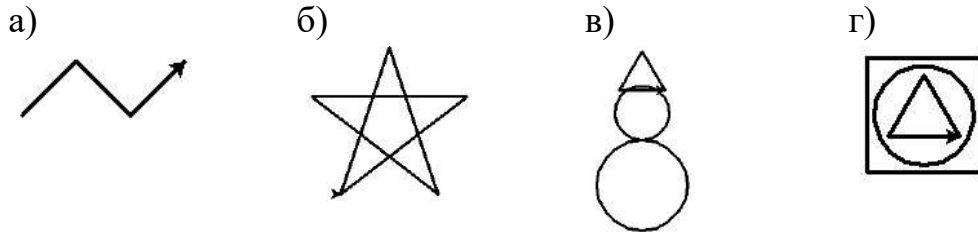
1. Яку команду потрібно застосувати для завантаження команд роботи з Черепашкою?
2. Запишіть оператор для виведення поточних координат Черепашки.
3. Поясніть призначення команд **down()**, **up()**.
4. Запишіть команди для Черепашки:
 - 1) проповзти вперед 20 кроків;
 - 2) намалювати коло радіуса 30 пікселів;
 - 3) підняти перо;

4) повернутися на 30° проти годинникової стрілки.

5. Черепашка знаходиться у точці (100; 100). Які координати матиме Черепашка після виконання команд малювання:

- | | | | |
|-----------------------|-----------------------|--------------------|-----------------------|
| а) forward(50) | б) forward(50) | в) left(90) | г) forward(50) |
| right(90) | left(90) | forward(50) | forward(50) |
| forward(50) | forward(50) | forward(50) | left(90) |
| | | | forward(50) |

6. Напишіть програми для створення зображень:



Вправа 25

Скласти програму малювання човника (рис. 25.6).

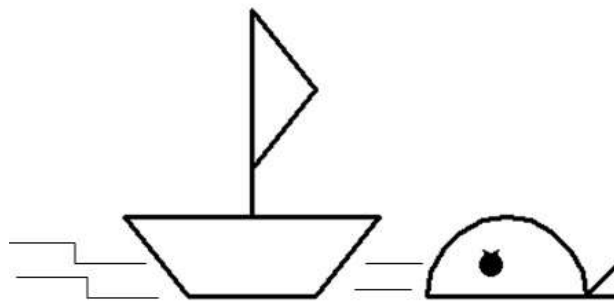


Рис. 25.6

1) Відкрийте вікно **IDLE** і створіть нове вікно програми. Завантажте модуль **turtle**. Установіть ширину сліду Черепашки 3 пікселі. Збережіть програму у файлі **Вправа 25**.

2) Визначте координати кінців відрізків, які утворюють зображення човника. Наприклад, якщо почати малювання з точки **O(0; 0)**, точка **A** може мати координати **(-40; 50)**, а точка **B(120; 50)** (рис. 25.7).

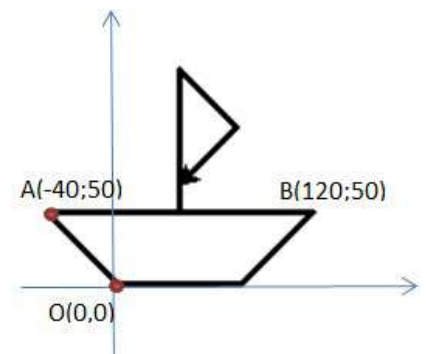


Рис. 25.7

3) Запрограмуйте пересування Черепашки по визначених точках:

goto(-40, 50) # Побудова відрізка OA

goto(120, 50) # Побудова відрізка AB

4*) Додайте зображення кита. Для цього слід перейти в точку (250, 0), повернути Черепашку вліво на 90° (щоб дуга була зліва від Черепашки), і намалюємо дугу з градусною мірою 180° .

```

up()
goto(250, 0)
down()
left(90)
circle(50, 180)    # Малювання спини кита
goto(270, 0)
goto(270, 20)     # Малювання хвоста кита
goto(250, 0)

```

5*) За допомогою команди **dot()** намалюйте око кита.

6) Додайте кілька ліній, що зображують хвилі.

Комп'ютерне тестування

Виконайте тестове завдання 25 з автоматичною перевіркою результату.

§ 26. АЛГОРИТМИ СТВОРЕННЯ ЗОБРАЖЕНЬ

У попередньому параграфі ви ознайомились із системою команд Черепашки та навчилися визначати її координати у графічному вікні, створювати малюнки пером чорного кольору. Модуль **turtle** надає можливості створення яскравих різнокольорових зображень, ми також можемо змінювати форму, розмір та колір самої Черепашки.

Створення кольорового малюнка

Черепашка може залишати не лише чорний, а й кольоровий слід. Ви можете використовувати велику кількість різноманітних кольорів. Назва кольору (англійською мовою) береться в одинарні лапки, наприклад **'red'**, **'yellow'**, **'green'** тощо. Відвідайте сайт www.tcl.tk/man/tcl8.4/TkCmd/colors.htm, щоб переглянути повний список кольорів.

Команди для встановлення кольору сліду:

Команда	Призначення
color(s)	Установити s — колір лінії, яку малює Черепашка
color(s1, s2)	Установити s1 — колір сліду Черепашки, s2 — колір заливки замкненої фігури

Команди для зафарбовування замкнених фігур:

Команда	Призначення
fillcolor (s)	Установити s — колір заповнення.

begin_fill()	Почати стежити за Черепашкою для заповнення області
end_fill()	Заповнити кольором s2 область, пройдену Черепашкою, починаючи з begin_fill()

Приклад 1. У вікні консолі введемо команди для малювання червоного кола, зафарбованого синім кольором.

```
>>> color('red', 'blue')
>>> begin_fill()
>>> circle(50)
>>> end_fill()
```

Результат виконання команд малювання кола наведено на рис. 26.1.

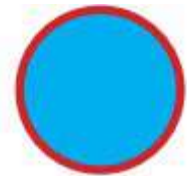


Рис. 26.1

Зміна вигляду Черепашки

Щоб змінити форму Черепашки, використовуйте команду **shape(<форма>)**, де **<форма>** — назва форми, узята в лапки. У модулі **turtle** існують такі форми черепашки, як **"arrow"**, **"circle"**, **"square"**, **"triangle"**, **"turtle"**, **"classic"**. Наприклад, якщо ви бажаєте створити Черепашку у формі квадрата, введіть команду **shape("square")**.

Розмір черепашки можна змінити з допомогою команди **shapsize(n)**, де **n** — коефіцієнт збільшення розміру черепашки. Наприклад, команда **shapsize(2)** збільшить Черепашку вдвічі.

Черепашка може залишати свої відбитки за допомогою команди **stamp()**. Після виконання цієї команди у вікні для графіки в місці, де була Черепашка, залишиться малюнок цієї Черепашки. Якщо потрібно, щоб після завершення малювання Черепашка не з'являлася на екрані, використовуйте команду **hideturtle()**. Щоб Черепашка знову показала, використовуйте команду **showturtle()**.

Приклад 2. Наказати Черепащі креслити лінію синього кольору, залишаючи сліди (рис. 26.2).

```
from turtle import*
shape('turtle')
color('blue' )
stamp()
forward(50)
shapsize(2)
stamp()
forward(100)
```

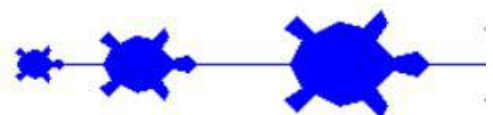


Рис. 26.2

```
shapesize(3)
stamp()
forward(100)
hideturtle()
```

Повернути Черепашку на початок координат (у точку з координатами та $y = 0$) можна командою **home()**. Команда використовується без аргументів.

Приклад 3. Повернути Черепашку в точку (0, 0) після малювання квадрата (рис. 26.3).

```
home()
print(position())
```



Рис. 26.3

Змінювати можна не тільки вигляд Черепашки, але й деякі властивості графічного вікна.

Команда	Призначення
bgcolor(<колір>)	Задати колір фону графічного вікна
setup(n, m)	Установити ширину вікна n пікселів, висоту m пікселів
reset()	Очистити вікно й перемістити Черепашку до центру вікна

Якщо ви хочете, щоб графічне вікно закривалося за клацанням миші, завершить програму командою **exitonclick()**.

Приклад 4. Задати розміри вікна 200 × 100 пікселів, колір фону вікна — помаранчевий (рис. 26.4).

```
from turtle import*
setup(200, 100)
bgcolor('orange')
exitonclick()
```

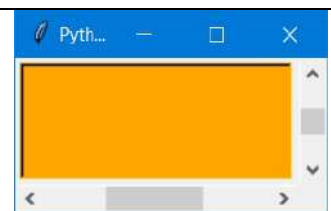


Рис. 26.4

Додавання тексту до графічного вікна

У вікні для графіки модуля **turtle** можна малювати текст. Для відображення тексту використовується команда **write(<текст>)** — вивести текст:

```
write(text, move, align, font = (fontname, fontsize, fontstyle)),
```

де **text** — текст (у лапках), який буде відображатися у вікні для графіки **turtle Python**;

move — параметр, що відповідає появі анімації Черепашки після відображення тексту. В анімації Черепашка підкреслює написаний текст. Параметр набуває лише логічних значень (**True**, **False**);

align — параметр, що відповідає за розташування тексту щодо Черепашки; набуває значення "**left**", "**right**", "**center**". Зверніть увагу: якщо наявний параметр **align**, параметр **move** не спрацьовує;

font — параметри шрифту:

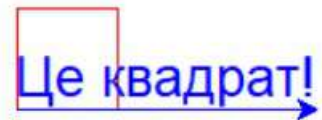
fontname — назва шрифту (в лапках);

fontsize — розмір шрифту;

fontstyle — стиль тексту ("**normal**", "**bold**", "**italic**").

ПРИКЛАД 5. Додати текст у графічне вікно, застосовуючи різні значення параметрів тексту.

```
write('Це квадрат!', move = True, font = 'Arial  
20')
```



```
write('Це квадрат!', align = 'right', font = ('Arial',  
16, 'italic'))
```



Питання для самоперевірки

1. Запишіть команди для Черепашки:

1) установити зелений колір сліду Черепашки;

2) установити жовтий колір фону вікна;

3) установити синій колір заливки замкненої фігури.

2. Як виконати заливання графічного об'єкта заданим кольором?

Опишіть алгоритм дій.

3. Намалюйте прапори України, Франції, Польщі.



4. Побудуйте і зафарбуйте зеленим кольором коло радіуса 100, центр якого співпадає з центром вікна.

5. Запишіть фрагмент програми для побудови трикутника з вершинами в точках (100, 100), (150, 100), (80, 70); колір фону — сірий; колір ліній — червоний.

6. Запишіть фрагмент програми для побудови прямокутника з вершинами (80, 80), (170, 80), (80, 150). Зафарбувати його жовтим кольором. Колір фону — синій.

Вправа 26

Розфарбувати малюнок (рис. 26.5).

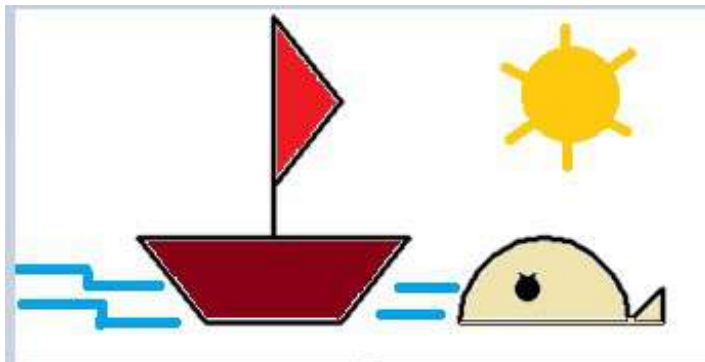


Рис. 26.5

1) Запустити **Python IDLE** і відкрити файл **Вправа 25**.

2) Додайте заливку до корпусу човника:

```
fillcolor('brown')
```

```
begin_fill()
```

```
goto(-40, 50)
```

```
goto(120, 50)
```

```
goto(80, 0)
```

```
goto(0, 0)
```

```
end_fill()
```

3) Додайте заливку до вітрила човника.

4) Змініть колір хвиль на синій.

5) Зафарбуйте кита.

6*) Намалуйте сонце:

```
color('orange', 'orange')
```

```
up()
```

```
goto(200, 200)
```

```
down()
```

```
begin_fill()
```

```
circle(30)
```

```
end_fill()
```

Додайте сонячні промені.

Комп'ютерне тестування

Виконайте тестове завдання 26 з автоматичною перевіркою результату.

ПРАКТИЧНА РОБОТА 8

Створення зображень за алгоритмами

Завдання: скласти програму малювання олімпійського символу.

Обладнання: комп'ютер зі встановленим середовищем програмування

Python.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Відкрийте вікно **IDLE** і створіть нове вікно програми.
2. Збережіть файл у власній папці з іменем **Практична робота 8**.
3. Для малювання кожного кільця виконайте такий алгоритм: задати колір, яким будемо малювати; підняти перо, щоб під час переходу не малювати лінії; перейти на точку початку малювання кільця; опустити перо; малювати коло радіуса 45.
4. Завантажте модуль **turtle** і встановіть ширину лінії 3 пікселі.
5. Запишіть програму малювання синього кола:
`color('blue')` # Встановлюємо синій колір
`up()` # Піднімаємо перо
`goto(-110, -25)` # Переходимо в потрібну точку
`down()` # Опускаємо перо
`circle(45)` # Малюємо коло радіуса 45
6. Підніміть перо і перейдіть у точку з координатами (0, -25).
7. Запишіть код для малювання кола чорного кольору.
8. Додайте код для малювання інших кілець за такими даними: третє — **red**, (110, -25); четверте — **yellow**, (-55, -75); п'яте — **green**, (55, -75).
9. Перейдіть у точку з координатами (0, 80) і встановіть червоний колір малювання.
10. Над олімпійським символом червоним кольором запишіть власне прізвище та ім'я.
11. Запустіть програму на виконання, проаналізуйте результат виконання програми у вікні **Python Turtle Graphics**. На рисунку наведено результат малювання олімпійського символу.
12. Закінчіть роботу, закривши вікно **IDLE**.

Зробіть висновок: як складати та виконувати лінійні алгоритми для створення зображень.



§ 27. ЛОГІЧНІ ВИРАЗИ

Окрім уже відомих нам числового та рядкового типів даних, у мові **Python** є логічний тип **bool**. Змінна типу **bool** може набувати одного з двох значень — **True** (*Істина*) або **False** (*Хибність*).

Якщо змінній надати значення **True**, це буде змінна типу **bool**:

```
>>> type(True)
<class 'bool'>
>>> a = True
>>> print(a)
True
```

ЦІКАВІ ФАКТИ.

Логічний тип даних отримав свою назву на честь англійського математика Джорджа Буля (рис. 27.1), засновника математичної логіки — розділу математики, що побудований на застосуванні математичних методів для розв'язування логічних задач. Сьогодні ідеї Буля використовуються у всіх сучасних цифрових пристроях.



Рис. 27.1

Логічні операції



Логічними є вирази, результатом яких є **True** або **False**.

Умови в програмі записуються у вигляді логічних виразів.

Простий логічний вираз (проста умова) утворюється за допомогою логічних операцій:

Знак операції	Значення	Логічний вираз	Результат
==	Дорівнює	8 == 9	False
>	Більше	8 > 9	False
<	Менше	8 < 9	True
>=	Більше або дорівнює	5 >= 5	True
<=	Менше або дорівнює	5 <= 2	False
!=	Не дорівнює	2 != 5	True

ПРИКЛАД 1. Проаналізуємо результати обчислення логічних виразів у вікні консолі:

```
>>> books = 10
```

```
>>> books == 5    # Перевірка, чи дорівнює books 5
False
>>> books < 10    # Перевірка, чи є books меншим за 10
False
>>> books >= 5    # Перевірка, чи books більше або дорівнює 5
True
>>> books != 10   # Перевірка, чи не дорівнює books 10
False
```

Значення логічного виразу можна зберегти у змінній.

ПРИКЛАД 2. Обчислити значення логічних виразів:

```
x = 5
y = 2
a = x > y          # a = True
a = x < y          # a = False
a = x - 4.5 < y * 2 # a = True
```

Бувають ситуації, коли одночасно необхідно перевірити виконання кількох умов.



Складена умова — це кілька простих умов, з'єднаних логічними операціями **and** (логічне **I**, інакше — логічний добуток), **or** (логічне **АБО**, інакше — логічна сума), **not** (логічне заперечення).

Складена умова **A and B = True**, тільки якщо й **A**, і **B** істинні.

Складена умова **A or B = False**, якщо й **A**, і **B** хибні.

ПРИКЛАД 3. Розглянемо приклади складених умов.

not a <= 3 — рівнозначне виразу $a > 3$;

age >= 10 and age <= 18 — істинне тоді й тільки тоді, коли значення **age** розташовується в проміжку від 10 до 18 включно;

age < 10 or age > 18 — істинне для всіх значень **age**, які не належать проміжку від 10 до 18.

ПРИКЛАД 4. Проаналізуємо результати обчислення логічних виразів у вікні консолі.

```
>>> books = 10
>>> books == 10 or books == 5 # Перевірка, чи дорівнює books 10 або 5
```

True

```
>>> books == 10 and books == 5 # Перевірка, чи дорівнює books  
одночасно 10 і 5
```

False

ПРИКЛАД 5. Визначити, чи належить точка з координатою x відрізка $[-5; 5]$ (рис. 27.2).

Точка належить відрізку, якщо справджується нерівність $-5 \leq x \leq 5$.
У програмуванні таку подвійну нерівність записують як складену умову:

```
 $x \geq -5$  and  $x \leq 5$ 
```



Рис. 27.2

ПРИКЛАД 6. Обчислити значення логічних виразів при $x = 1; y = 2; z = 3$.

```
 $a = (x < y)$  and  $(y < z)$  #  $a = \text{True}$ 
```

```
 $a = (x > y)$  or  $(y > z)$  #  $a = \text{False}$ 
```

```
 $a = \text{not}(x > y)$  #  $a = \text{True}$ 
```

Питання для самоперевірки

1. Яких значень може набувати змінна логічного типу?
2. Які операції можна виконувати над змінними логічного типу?
3. Запишіть мовою програмування прості умови:
 - а) x більше 10;
 - б) x не більше y ;
 - в) x — парне число.
4. Запишіть мовою програмування складені умови:
 - а) $2 < x < 10$;
 - б) x не належить проміжку $(2, 10)$.
5. Запишіть у вигляді логічних виразів подані умови:
 - а) $3 \leq x \leq 10$;
 - б) $-5 \leq x \leq 5$ за умови, що $x \neq 0$.
6. Обчисліть значення логічних виразів:
 - а) $(A \leq B)$ and $(A = B-1)$, якщо $A = 2, B = 4$;
 - б) $(A \leq B)$ or $(A = B-1)$, якщо $A = 2, B = 4$.

Вправа 30

Обчислити логічні вирази.

Написати програму, у якій залежно від значень змінних A та B обчислюються значення логічних виразів, а результат обчислення виводиться у вікно консолі.

1) Запустити **Python IDLE** і створити файл **Логічні операції**. Завантажити модуль **random** для генерації випадкових чисел.

2) Надайте змінній **a** випадкове значення з проміжку від 1 до 20, виведіть значення **a** у вікно консолі:

```
a = randint(1, 20)
print("a = ", a)
```

Надайте змінній **b** випадкове значення з проміжку від 1 до 20, виведіть значення **b** у вікно консолі.

3) Змінній **x** присвойте значення логічного виразу $a > b$, виведіть значення **x**:

```
x = a > b
print("a > b = ", x)
```

4) Змінній **y** присвойте значення логічного виразу $b \leq 10$, виведіть значення **y**:

```
y = b <= 10
print("b <= 10 = ", y)
```

5) Виведіть у консоль значення складеної умови **x and y**:

```
print("x and y = ", x and y)
```

6*) Виведіть у консоль значення логічних виразів **x or y**, **not x and y**, **not(x and y)**. Приклад виконання програми наведено на рис. 27.3. Виконайте програму кілька разів, проаналізуйте значення логічних виразів.

```
a = 18
b = 12
a > b = True
b <= 10 = False
x and y = False
x or y = True
not x and y = False
not(x and y) = True
>>>
```

Рис. 27.3

Комп'ютерне тестування

Виконайте тестове завдання 27 з автоматичною перевіркою результату.

§ 28. АЛГОРИТМИ І ПРОГРАМИ З РОЗГАЛУЖЕННЯМИ. ОПЕРАТОР IF

Програма має вміти вибирати різні дії в залежності від уведених даних. Ви ознайомились із алгоритмічною структурою **Розгалуження** і розглянули приклади алгоритмів з розгалуженнями. Для перевірки істинності умов і організації розгалуження у програмах мовою **Python** призначені умовні оператори **if** та **if-else**.



Оператори — це команди програми.

Умовний оператор if

Оператор **if** призначено для виконання деякої послідовності дій у тому випадку, якщо істинною є зазначена умова. Цей умовний оператор відповідає алгоритмічній конструкції **неповне розгалуження** (рис. 28.1).

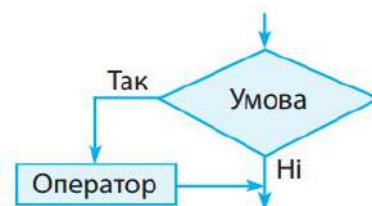


Рис. 28.1

Синтаксис умовного оператора в неповній формі:

if <Умова>:

 <Оператор>

де — обов'язковий відступ від лівого краю.

Після запису умови слід поставити двокрапку, яка показує, що далі має бути розташований блок дій. Блок дій записується з обов'язковим однаковим відступом від лівого краю.

Оператор **if** перевіряє істинність зазначеної умови. Якщо умова набуває значення **True (Істина)**, програма виконає дію, зазначену в частині <Оператор>. Якщо ж умова приймає значення **False (Хибність)**, то блок <Оператор> пропускається, і керування передається оператору, що йде після оператора **if**.

Приклади використання неповного розгалуження у програмах

Неповне розгалуження використовують тоді, коли деяку послідовність команд слід виконати лише за умови істинності висловлювання. Якщо ж записане в умові висловлювання хибне, то жодна з команд не виконується.

Приклад 1. Після виконання цієї програми у вікні консолі отримаємо результат **Правильно**.

```
books = 10
if books == 10:
    print('Правильно!')
```

Приклад 2. Вітання **Вітаю!** виводиться, якщо користувач уводить літеру «у».

```
answer = input('Сьогодні твій день народження? (y/n)')
if answer == 'y':
    print('Вітаю!')
print('Гарного дня!')
```

Приклад виконання команди наведено на рис. 28.2.

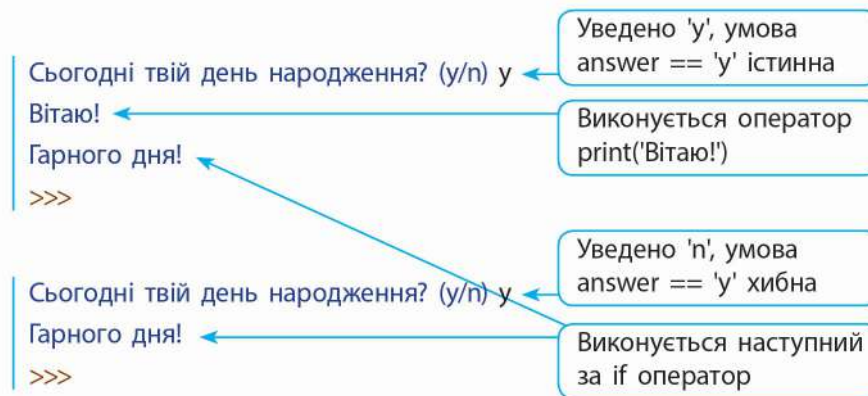


Рис. 28.2

Приклад 3. Нехай потрібно збільшити значення змінної a на одиницю, якщо її поточне значення менше 5. Оператор розгалуження має вигляд:

```
if a < 5:
    a = a + 1
```

Оператор $a = a + 1$ виконується тільки в тому випадку, коли істинна умова $a < 5$:

Початкове значення a	Значення умови	Оператор $a = a + 1$	Значення a після виконання оператора if
$a = 1$	True	Виконується	2
$a = 5$	False	Пропускається	5
$a = 10$	False	Пропускається	10

Приклад 4. Визначити, чи є число n кратним 2, 3 або 5.

```
n = int(input("n =? "))
if n % 2 == 0:
    print(n, "парне число")
if n % 3 == 0:
    print(n, "кратне 3")
if n % 5 == 0:
    print(n, "кратне 5")
```

```
n=?27
27 кратне 3
```

У програмі використовуються послідовні розгалуження, тобто розгалуження, що йдуть одне за одним. Програма по черзі перевіряє умову кожного розгалуження, і, якщо умова істинна, виконує блок **<оператор>** і після цього переходить до перевірки умови наступного розгалуження та подальших операцій. Перевірка наступної умови розгалуження не залежить від результату попередньої.

Питання для самоперевірки

1. Як записується й виконується умовний оператор у неповній формі?
2. Яких значень набудуть змінні a і b після виконання умовних операторів, наведених нижче, для початкових значень $a = 3; b = 10$?

а) `if a%2 == 0:` б) `if a != b:` в) `if a < b:` г) `if b >= 10:`
 $a = a//2$ $a = b$ $b = b-a$ $a = b-a$
 $b = 0$

3. Виберіть оператори, виконання яких не викличе повідомлення про помилку:

а) `if a > b:` б) `if a % 3:` в) `if a < 10` г) `if b > 10:`
`print("a > b")` $a = a//3$ $a = a+5$ $b = b-10$

4. Запишіть оператори розгалуження, які реалізують такі дії:

- а) якщо a більше b , замінити a нулем;
- б) якщо ціле число a парне, поділити його на 2;
- в) якщо числа a і b не рівні, змінній a надати значення b .

5. Скласти програму, яка запитує значення числа n , $1 < n < 10000$, і повідомляє кількість цифр у числі n .

6. Скласти програму, яка запитує значення температури тіла людини і визначає, здорова людина чи хвора. Якщо температура менше 36°C , виводиться порада зігритися. Якщо температура від 36 до 37 градусів, людина здорова. Якщо температура вище 37°C , виводиться порада звернутися до лікаря.

Вправа 28

Скласти програму, яка запитує значення n кількості кутів багатокутника і креслить фігуру, що має n кутів.

1) Запустити **Python IDLE** і створити файл **Вправа 28**. Завантажити модуль для роботи з «черепашачою» графікою.

2) Запишіть оператор для введення значення n .

3) Якщо $n = 3$, програма креслить трикутник. Запишіть оператор неповного розгалуження, який перевіряє умову $n == 3$. Блок дій, який

виконується, якщо умова істинна, має містити оператори малювання трикутника.

```
if n == 3:  
    forward(50)  
    right(120)  
    forward(50)  
    right(120)  
    forward(50)  
    right(120)
```

4) Якщо $n = 4$, програма має накреслити квадрат. Запишіть оператор неповного розгалуження, який перевіряє умову $n == 4$, і, якщо умова істинна, креслить квадрат.

5) Якщо $n = 5$, програма креслить п'ятикутник. Запишіть оператор неповного розгалуження, який перевіряє умову $n == 5$, і, якщо умова істинна, креслить п'ятикутник.

6*) Запишіть оператор розгалуження, який виконує таку дію: якщо $n < 3$ або $n > 5$, вивести повідомлення **Такої фігури не передбачено**. Перевірте роботу програми.

Комп'ютерне тестування

Виконайте тестове завдання 28 з автоматичною перевіркою результату.

§ 29. АЛГОРИТМИ І ПРОГРАМИ З РОЗГАЛУЖЕННЯМИ. УМОВНИЙ ОПЕРАТОР IF ... ELSE

Ви вже знаєте, що у випадках, коли нам потрібно, щоб наша програма виконувала одні дії, якщо деяка умова істинна, та інші дії, якщо ця умова хибна, використовується алгоритмічна конструкція **повне розгалуження**. Цій конструкції відповідає умовний оператор **if ... else** (рис. 29.1).



Рис. 29.1

Синтаксис умовного оператора в повній формі:

```
if <умова> :  
    <оператор 1>  
else:  
    <оператор 2>
```


Якщо умова істинна (**True**), програма виконає блок дій <Оператор 1>. Якщо умова хибна (**False**), виконується блок дій <Оператор 2>, який міститься після службового слова **else**.

ПРИКЛАД 1. Якщо користувач уводить літеру «у», виводиться вітання Вітаю!, інакше текст Щастя тобі!. Після виконання одної з гілок оператора **if ... else** програма переходить до виконання наступного оператора.

```
answer = input('Сьогодні твій день народження? (y/n)')
if answer == 'y':
    print('Вітаю!')
else:
    print('Щастя тобі!')
print('Гарного дня!')
```

ПРИКЛАД 2. Оператор **if ... else** реалізує таку дію: якщо $a > b$, то змінні міняються значеннями так, щоб виявилось $a < b$. Результат виконання програми наведено на рис. 29.2.

```
if a>b:
    c = a
    a = b
    b = c
    print('a і b помінялися
значеннями')
else:
    print('обміну значеннями не
потрібно')
print ('a = ', a, 'b = ', b)
```

```
a = 5 b = 2
a і b помінялися значеннями
a = 2 b = 5
>>>
a = 3 b = 8
обміну значеннями не потрібно
a = 3 b = 8
>>>
```

Рис. 29.2

Зверніть увагу на відступи від лівого краю в прикладі 2: команди, вкладені в гілки оператора **if**, записані на одній вертикалі.

Приклад 3. Оператор **if ... else** перевіряє, чи існує трикутник зі сторонами a , b , c . Правило трикутника: сума двох будь-яких сторін повинна бути більша за третю.

```
if(a<b+c) and (b<a+c) and (c<a+b):
    print('Трикутник з такими сторонами існує')
else: print('Трикутника з такими сторонами не існує')
```

У програмі можуть послідовно виконуватись умовні оператори повного і неповного розгалуження.

ПРИКЛАД 4. Скласти програму пошуку найбільшого з трьох чисел a , b , c . Блок-схему алгоритму пошуку найбільшого з трьох чисел наведено на рис. 29.3.

```
a = int(input("a = ?"))
b = int(input("b = ?"))
c = int(input("c = ?"))
if a > b: m = a
else: m = b
if c > m: m = c
print('m = ', m)
```

```
a=?3
b=?12
c=?8
m= 12
>>>
```

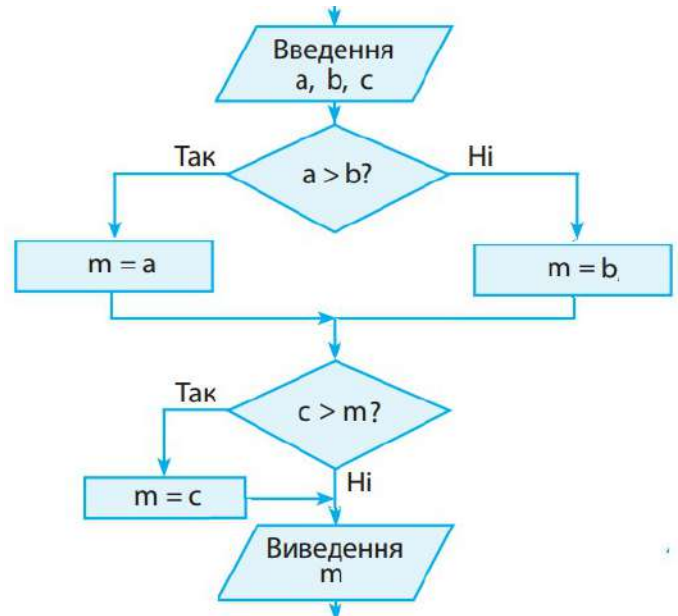


Рис. 29.3

Отже, умовні оператори **if** та **if ... else** є важливими інструментами програмування, які стануть вам у пригоді під час створення тестових та ігрових програм.

Питання для самоперевірки

1. Як записується й виконується умовний оператор у повній формі?
2. Як виконуються логічні операції **and**, **or**, **not**?
3. Яких значень набудуть змінні a і b після виконання умовних операторів, наведених нижче, для початкових значень $a = 3$; $b = 5$?
4. Яких значень набудуть змінні a і b після виконання умовних операторів, наведених у таблиці, для початкових значень $a = 3$; $b = 5$?

а) if a > b:	б) if a != b:	в) if a % 2 == 0:	г) if a > b :	ґ) if a < b :
a = 0	a = b	a = a // 2	a = a + 10	a = 2 * a
else:	else:	else:	else:	else:
b = 0	a = 0	a = a + 1	b = b + 10	b = b * a

5. Запишіть оператори розгалуження, які реалізують такі дії:

а) змінній m присвоїти значення меншого з чисел a і b ;

б) якщо ціле число a парне, поділити його на 2, інакше збільшити a на 1;

в) значення більшого з чисел a і b замінити нулем.

6. Скласти програму, у якій перевіряється уведене з клавіатури число. Якщо число менше від 40, то виводиться повідомлення **ВЛУЧИВ**, якщо число більше за 40 — повідомлення **ПЕРЕЛІТ**.

Вправа 29

Скласти програму, яка визначає, чи достатньо користувачеві років, щоб керувати автомобілем.

Задача. Потрібно ввести вік користувача та зберегти значення в змінній **age**; якщо **age** ≥ 18 , то вивести повідомлення **Ваш вік дозволяє керувати авто**, інакше — змінній **r** присвоїти значення виразу **(18–age)**, вивести повідомлення, через скільки років користувач зможе отримати посвідчення.

1) Відкрийте вікно **IDLE** і створіть нове вікно програми.

2) Уведіть текст програми, наприклад такий:

```
age = int(input('Скільки вам років?'))
if age >= 18: print('Ваш вік дозволяє керувати авто')
else:
```

```
    r = 18–age
```

```
    print('Доведеться почекати', r, ' років')
```

3) Збережіть файл у власній папці з іменем **Вправа 32**.

4) Запустіть програму на виконання, проаналізуйте результат виконання програми у вікні консолі. Випробуйте програму для різних значень змінної **age**.

5*) Змініть програмний код так, щоб у випадку **age** ≥ 18 виводилось повідомлення, скільки років людина вже могла б керувати автомобілем.

б) Закінчіть роботу, заклавши вікно **IDLE**.

Комп'ютерне тестування

Виконайте тестове завдання 29 з автоматичною перевіркою результату.

ПРАКТИЧНА РОБОТА 9. Складання та виконання алгоритмів із розгалуженнями

Завдання: скласти програму для обчислення коренів рівняння $ax + b = c$.

Обладнання: комп'ютер зі встановленим середовищем програмування **Python**.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1) Проаналізуйте словесний алгоритм розв'язування рівняння:

якщо $a = 0$, $b = c$, то коренем рівняння є будь-яке число;

якщо $a = 0$, $b \neq c$, то коренів немає;

якщо $a \neq 0$, то $x = (c - b) / a$.

2) Відкрийте вікно **IDLE** і створіть нове вікно програми. Збережіть файл у власній папці з іменем **Практична 10**.

3) Складіть програму на основі блок-схеми (рис. 1).

Програма розв'язування рівняння

```
a = int(input('a = ?'))
```

```
b = int(input('b = ?'))
```

```
c = int(input('c = ?'))
```

```
if a != 0:
```

```
    x = (c-b)/a
```

```
    print('x = ', x)
```

```
if a == 0 and b == c: print('безліч коренів')
```

```
if a == 0 and b != c: print('корені відсутні')
```

4) Запустіть програму на виконання. Перевірте роботу програми для тестового набору значень (рис. 2).

5) Випробуйте програму для таких наборів коефіцієнтів:

а) $a = 5$; $b = 3$; $c = 3$;

б) $a = 0$; $b = 2$; $c = 17$;

в) $a = 0$; $b = 4$; $c = 4$.

6) Закінчіть роботу, закривши вікно **IDLE**.

Зробіть висновок: як складати та виконувати алгоритми з розгалуженням для розв'язування задач.

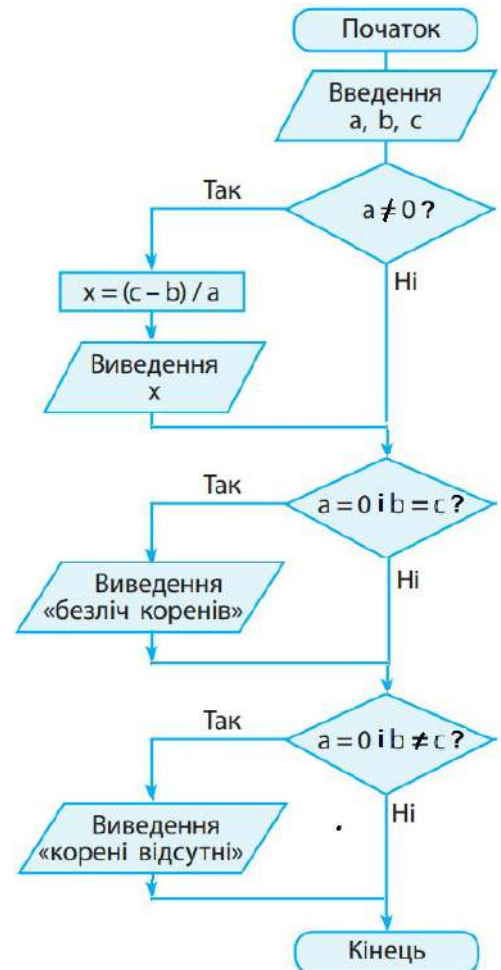


Рис. 1

```
=== RESTART:
a=? 5
b=? 2
c=? 17
x= 3.0
```

Рис. 2

§ 30. АЛГОРИТМИ З ПОВТОРЕННЯМИ. ЦИКЛ ІЗ ЛІЧИЛЬНИКОМ



Повторення (цикл) — це алгоритмічна структура, за

допомогою якої та сама послідовність дій виконується кілька разів.

Розглянемо правила запису алгоритмів із повторенням (циклів) мовою **Python** з використанням оператора циклу **for** — циклу із лічильником.

Повторення команд

Цикл **for** (цикл із параметром) повторює блок команд (тіло циклу) задану кількість разів, позбавляючи необхідності кількаразово писати одні й ті самі команди.

Синтаксис оператора циклу for:

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

```
    <тіло циклу>
```

Насамперед ми вказуємо ключове слово **for**, потім — змінну **i**, яка буде лічильником циклу. Ключове слово **in** наказує по чергово надати змінній **i** усі цілі значення з послідовності чисел від 0 до **n-1**. Не забувайте, що лічильник починає рахувати з 0, а не з 1.

Серію команд, що повторюється під час виконання циклу, називають **тілом циклу**. Кожне виконання тіла циклу називають **ітерацією**.

ПРИКЛАД 1. Проаналізуйте зміну значень лічильника **i** у циклі **for**. Функція **range(4)** задає список значень [0, 1, 2, 3]:

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

Вбудована функція **range()** повертає зростаючу послідовність цілих чисел, які можна використовувати у виразах всередині циклу. Щоб дати зрозуміти комп'ютеру, які команди слід повторити, використовуються відступи. Можна зробити відступ для кожної повторюваної в тілі циклу команди, натискаючи для цього клавішу **Tab**.

ПРИКЛАД 2. У прикладі 1 (§ 25) для малювання трикутника ми тричі, для кожної сторони трикутника, давали Черепащі команди **forward(100)** і **right(120)**. Замість цього можна повторити ці команди в циклі (рис. 30.1). Результат виконання циклу представлений на рис. 30.2, *a*.

Щоб оператор циклу намалював квадрат (рис. 30.2, *б*), змінимо кількість повторень циклу і кут повороту:

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

```
forward(100)
right(90)
```



Рис. 30.1



Рис. 30.2

Лічильник циклу for

Лічильник циклу (змінна циклу) по чергово набуває всіх значень з указаної послідовності. Цикл виконає стільки повторень, скільки значень задано у цій послідовності.

ПРИКЛАД 3. Лічильник по чергово набуває значення літер, з яких складається зазначений рядок (рис. 30.3)

```
>>> a = 'Рядок'
>>> for x in a:
>>>     print ('->', x)

-> Р
-> я
-> д
-> о
-> к
>>>
```

Рис. 30.3

Послідовність значень може бути задана у вигляді списку.



Список — це набір значень, розділених комами, взятий у квадратні дужки.

У списках можна зберігати значення будь-якого типу — числа або рядки. Можна сказати, що список — це контейнер для збереження набору значень.

ПРИКЛАД 4. Лічильник по чергово набуває заданих значень зі списку (рис. 30.4).

```
>>> b = [2, 4, 6, 8, 10]
>>> for x in b:
>>>     print ('->', x)

-> 2
-> 4
-> 6
-> 8
-> 10
```

Рис. 30.4

Якщо список значень лічильника достатньо великий і числа в списку змінюються з певним кроком, можна застосувати функцію **range()** із такими параметрами:

range(від, до, крок)

Відлік починається зі значення **від** і завершується на 1 раніше за значення **до**.

ПРИКЛАД 5. Функція range(2, 11, 3) надає змінній x значення від 2 до (11-1) із кроком 3, тому, хоча $8 + 3 = 11$, значення 11 уже до послідовності значень не входить (рис. 30.5).	<pre>>>> for x in range(2, 11, 3) : print ('->', x) -> 2 -> 5 -> 8</pre>
--	--

Рис. 30.5

! Якщо значення **від** більше, ніж значення **до**, і значення **крок** ≥ 0 , цикл не виконається жодного разу.

ПРИКЛАД 6. Якщо потрібно вести зворотний відлік (у бік зменшення значень), укажіть значення від більшим, ніж значення до , а крок має бути від'ємним (рис. 30.6).	<pre>>>> for x in range(5, 1, -1) : print ('->', x) -> 5 -> 4 -> 3 -> 2</pre>
--	--

Рис. 30.6

Приклади розв'язування задач за допомогою циклу for

ПРИКЛАД 7. Вивести квадрати непарних чисел з проміжку від 11 до 19 (рис. 30.7). <pre>for i in range(11, 20, 2): print(i, "**2 =", i*i)</pre>	<pre>11 **2 = 121 13 **2 = 169 15 **2 = 225 17 **2 = 289 19 **2 = 361</pre>
--	---

Рис. 30.7

ПРИКЛАД 8. Увести з клавіатури 5 цілих чисел і визначити, скільки серед уведених чисел було парних. Створимо змінну k , у якій зберігатимемо кількість парних чисел. Змінна a при кожній ітерації циклу буде набувати введеного з клавіатури значення (рис. 30.8).	<pre>0? 10 1? 3 2? 12 3? 7 4? 25 k = 2</pre>
--	--

Рис. 30.8

```

k = 0          # Початкове значення лічильника парних чисел
for i in range(5):
    a = int(input(str(i)+"?"))          # Уведення чергового числа
    if a%2 == 0: k = k+1                # Якщо число парне, лічильник k
збільшується на 1
    print ("k = ", k)

```

ПРИКЛАД 9. Написати програму, яка здійснює зворотний відлік.

```

import time          # Модуль для роботи з часом
print("Готуємось до зльоту!")
for i in range(10, 0, -1): # Зворотний відлік
    print(i)
    time.sleep(1)      # Затримка виведення на 1 секунду

```

Отже, цикл із лічильником є потужним інструментом програмування, він позбавляє програміста необхідності писати однакові блоки коду кілька разів, дозволяє в компактному коді виконувати велику кількість операцій.

ЦІКАВІ ФАКТИ. Віктор Михайлович Глушков — засновник Інституту кібернетики Національної академії наук України, активний учасник світового процесу становлення та розвитку нових напрямів науки ХХ і ХХІ ст. — обчислювальної техніки, інформатики, кібернетики. Під його керівництвом був розроблений перший в Україні та колишньому СРСР персональний комп'ютер «МИР-1».



Питання для самоперевірки

1. Поясніть правила виконання циклу **for**.
2. Яких значень набуває змінна **x** у ході виконання циклу:
for x in range(5)?
3. Яких значень набуває змінна **x** у ході виконання циклу:
for x in range(0, 10, 2)?
4. Що буде надруковано в ході виконання циклу:

а) s = 'цикл' for x in s: print(x)	в) for x in range(5): print(x*x)
б) b = [1, 3, 5, 7, 9] for x in b: print(x)	г) for x in range(2, 10, 2): print(x*2)

5. Чому дорівнює s після виконання циклу:

- | | |
|---|---|
| а) $s = 0$
for a in range(5, 7, 1)
$s = s+1$ | в) $s = 0$
for a in range(10, 5, 1)
$s = s+1$ |
| б) $s = 0$
for a in range(5, 7, 1)
$s = s+a$ | г) $s = 0$
for a in range(10, 5, -1)
$s = s+a$ |

Вправа 30

Створити рисунки з використанням циклу **for**.

1) Відкрийте вікно **IDLE** і створіть нове вікно програми. Збережіть файл з іменем **Вправа 30**.

2) Запишіть програму для побудови поданого малюнка (рис. 30.9).

```
from turtle import*  
color('blue')  
for i in range(1, 100, 2):  
up()  
goto(i*2, 0)  
down()  
circle(i)
```



Рис. 30.9

3) Запустіть програму на виконання. Скільки кіл побудовано?

4) Змініть параметри функції **range()** таким чином: **range(1, 100, 6)**. Запустіть програму на виконання. Скільки кіл побудовано?

5) Поставте на початку рядка **goto(i*2, 0)** символ **#**. Запустіть програму на виконання. Як змінився рисунок?

6*) Змініть команду **goto(i*2, 0)**, щоб малюнок мав такий вигляд, як на рис. 30.10.

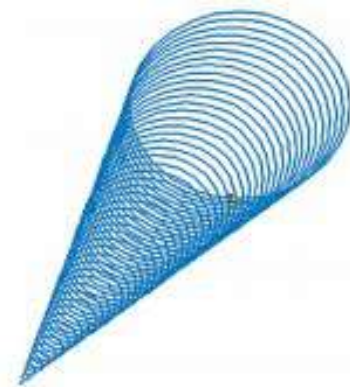


Рис. 30.10

Комп'ютерне тестування

Виконайте тестове завдання 30 з автоматичною перевіркою результату.

§ 31. АЛГОРИТМИ І ПРОГРАМИ З ЦИКЛАМИ З ЛІЧИЛЬНИКОМ

Багаторазове повторення команд малювання з використанням циклу **for** допоможе вам створювати цікаві різнокольорові малюнки.

Алгоритми створення зображень

Запис алгоритму малювання багатокутників із використанням оператора циклу робить код компактним і універсальним.

ПРИКЛАД 1. Скласти програму малювання n -кутників. Приклад виконання програми наведено на рис. 31.1

```
n = int(input("Скільки кутів? "))
x = int(input("Довжина сторони? "))
for i in range(n):
    forward(x)
    left(360/n)
```

```
Скільки кутів?7
Довжина сторони?50
>>>
```

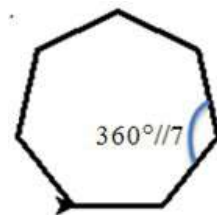


Рис. 31.1

Розглянемо алгоритми малювання фігур, наведених на рис. 31.2.

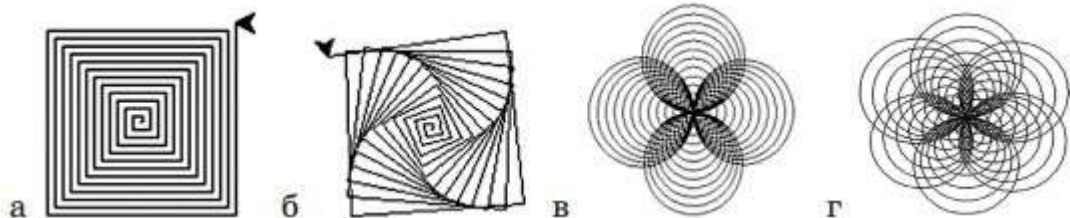


Рис. 31.2

Проаналізуємо виконання програми:

```
from turtle import*
for x in range(1, 100, 2):
    forward(x)
    left(90)
```

Під час кожного проходу циклу Черепашка малює лінію довжиною x пікселів і повертається вліво на 90° (рис. 31.2, *a*). Як ви знаєте, повне обертання навколо точки — це 360° . Отже, за 4 ітерації циклу Черепашка повертається до початкового напрямку ($4 \cdot 90^\circ = 360^\circ$). Програма малює спіраль, оскільки кожен відрізок на 2 пікселі довший за попередній: перший відрізок має довжину 1 піксель, другий — 3, останній (п'ятдесятий) — 99.

Поворот на 90° створює квадратну спіраль. Змінимо кут повороту на 2° : `left(92)`. Зміна лише одного значення викликає зміни у вигляді спіралі (рис. 31.2, б).

Змінимо команду `forward(x)` на `circle(x)`, яка малює коло радіуса x із поточної позиції:

```
for x in range(1, 100, 2):
    circle(x)
    left(90)
```

Отримаємо 4 набори кіл, оскільки після малювання кожного кола Черепашка повертається вліво на 90° (рис. 31.2, в). Щоб намалювати n кіл, потрібно повертати Черепашку на $360/n$ градусів. Унесемо такі зміни до програми, щоб вона малювала 6 наборів кіл ($360/6 = 60$) (рис. 31.2, г):

```
for x in range(1, 100, 2):
    circle(x)
    left(60)
```

ЦІКАВІ ФАКТИ.

Біля витоків української школи програмування стояла Катерина Ющенко, докторка фізико-математичних наук, авторка однієї з перших у світі мов програмування високого рівня, в якій застосовуються операції над адресами об'єктів у пам'яті комп'ютера.



Зміна кольорів у циклі

Як зробити спіралі різнокольоровими? Для цього нам потрібен список кольорів, а не один колір.

Створимо змінну `colors` і помістимо у список назви чотирьох кольорів:
`colors = ['red', 'yellow', 'green', 'blue']`

Зверніть увагу, що звернутися до значення зі списку з номером x можна як до `colors[x]`. Наприклад, `colors[0] = 'red'`, `colors[3] = 'blue'`.

Складне питання: як змусити Черепашку під час кожної ітерації циклу, коли x змінюється від 1 до 99, обирати одне з 4 значень списку? Використаємо операцію `%` (остача від ділення). Під час обчислення виразу `x%4` ми можемо отримати 4 значення остачі (0, 1, 2, 3), що відповідає номерам кольорів у списку `colors`.

Повернемося до програми малювання квадратної спіралі, але крок циклу зробимо рівним 1 (рис. 31.3):

```

from turtle import*
colors = ['red', 'yellow', 'green', 'blue']
for x in range(100):
color(colors[x% 4])
forward(x)
left(90)

```

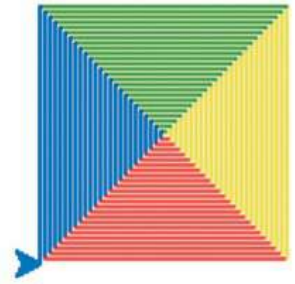


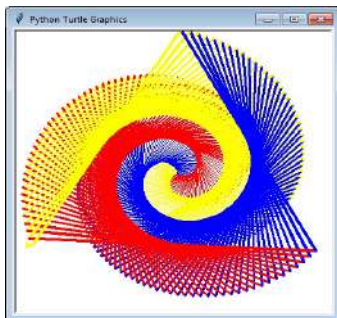
Рис. 31.3

ПРИКЛАД 2. Намалювати різнокольорову спіраль, збільшивши кут повороту на 1° і збільшуючи товщину лінії в залежності від змінної циклу. Результати виконання програми для $n = 3$, $n = 8$ наведені на рис. 31.4.

```

from turtle import*
n = int(input('n = ? '))
colors = ['red', 'blue', 'yellow', 'green', 'violet', 'orange', 'purple',
'white']
for x in range(360):
color(colors[x%n])
forward(x)
left (360/n+1) # Щоб «розтягнути» спіраль, кут повороту
збільшено на 1°
width(x*n/300) # Товщина лінії залежить від змінної циклу x

```



$n = 3$
а



$n = 8$
б

Рис. 31.4

Отже, вносячи нескладні зміни до операторів малювання в тілі циклу **for**, ви можете створювати цікаві яскраві зображення.

ЦІКАВІ ФАКТИ. Адель Голдберг — американська вчена в галузі інформатики, відома своєю роботою у галузі об'єктно-орієнтованого програмування і графічних інтерфейсів та розробкою мови програмування Smalltalk.



Питання для самоперевірки

- 1) Обговоріть алгоритм зміни кольору лінії при кожній ітерації циклу.
- 2) Яку відстань пройде Черепашка в ході виконання циклу:
 - а) `for x in range(5):`
`forward(x)`
 - б) `for x in range(2, 10, 2):`
`forward(10)`
- 3) Визначте градусну міру кута, на який повернеться Черепашка в ході виконання циклу:
 - а) `for x in range(4):`
`left(20)`
 - б) `for x in range(5, 9):`
`left(10)`
- 4) Створіть список `clrs`, елементами якого будуть назви червоного, жовтого, зеленого кольорів.
- 5) Обговоріть алгоритм встановлення Черепашки у точку з випадковими координатами.
- 6) Напишіть програму малювання грона калини (рис. 31.5). Ягоди калини зобразіть за допомогою кіл з випадковими координатами початку малювання:

```
color("red")
for i in range(1, 20):
    up()
    goto(randint(0, 100), randint(1, 70))
    down()
    circle(10)
```

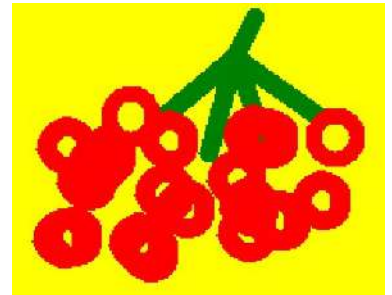


Рис. 31.5

Вправа 31

Створити програму «Зоряне небо».

Завдання: заповнити область вікна зображеннями зірок із випадковими координатами.

- 1) Відкрийте вікно **IDLE** і створіть нове вікно програми. Збережіть файл з іменем **Вправа 34**. Завантажте модулі, які потрібні для роботи з Черепашкою та випадковими числами.

```
from turtle import*
from random import*
```

- 2) Задайте темно-синій колір фону вікна:

```
bgcolor("darkblue")
```

- 3) Створіть список `colors`, елементами якого будуть назви білого, жовтого, помаранчевого кольорів.

```
colors = ['white', 'yellow', 'orange']
```

4) Запишіть оператор циклу **for** для малювання 100 точок, координати і радіуси яких задаються випадковим чином, а кольори по чергово вибираються зі списку **colors** (рис. 31.6).

```
for i in range(1, 100):  
    color(colors[i%3])  
    up()  
    goto(randint(-200, 200), randint(-200, 200))  
    down()  
    dot(randint(1, 8))
```

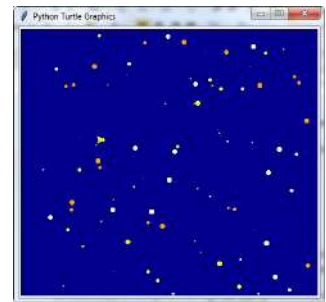


Рис. 31.6

5*) Додайте до списку **colors** ще 4 назви кольорів. Змініть програмний код так, щоб були намальовані 100 кіл, малювання яких починається з точок із випадковими координатами в діапазоні від -200 до 200 по обох осях, а радіуси є випадковими числами в діапазоні від 5 до 50 (рис. 31.7).

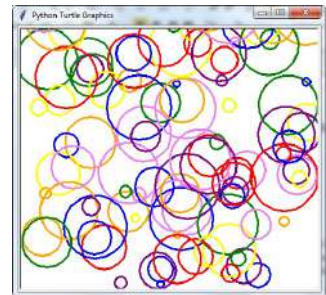


Рис. 31.7

6*) Додайте до програми команди, потрібні для того, щоб кола були зафарбовані.

Перевірте роботу програми. Закінчіть роботу, закривши вікно **IDLE**.

Комп'ютерне тестування

Виконайте тестове завдання 31 з автоматичною перевіркою результату.

ПРАКТИЧНА РОБОТА 10. Складання та виконання алгоритмів із повтореннями

Завдання: скласти програму для перевірки знання таблиці множення. Скласти програму, яка пропонує користувачеві розв'язати 10 виразів і підраховує кількість правильних відповідей.

Обладнання: комп'ютер зі встановленим середовищем програмування **Python**.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1) Відкрийте вікно **IDLE** і створіть нове вікно програми. Збережіть файл з іменем **Практична 11**. Завантажте модулі, які потрібні для роботи з Черепашкою та випадковими числами.

2) Створіть змінну **k** — лічильник правильних відповідей. Задайте початкове значення **k = 0**.

3) Запишіть оператор **for**, лічильник якого набуває значення з послідовності 0...11.

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

4) Запишіть оператори тіла циклу:

- присвойте змінним **a** і **b** випадкові значення з діапазону 1...9:

```
a = randint(1, 10)
```

```
b = randint(1, 10)
```

- напишіть оператор виведення, який виводить у консоль черговий вираз:

```
print (a, "*", b, "=", end = ' ')
```

Зверніть увагу: **end = ' '** у списку виведення означає, що після виведення виразу курсор не перейде на наступний рядок;

- напишіть оператор введення, який запитує у користувача відповідь на питання:

```
c = int(input("?"))
```

- напишіть оператор розгалуження, який перевіряє, чи правильна відповідь на питання:

```
if a*b == c:
```

```
    print("Так!")
```

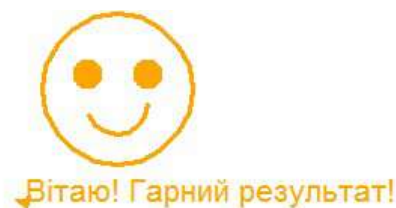
```
    k = k+1
```

```
else: print("Hi!")
```

5) Напишіть оператор виведення, який виводить у консоль значення лічильника правильних відповідей:

```
print("Правильних відповідей – ", k)
```

6) Напишіть код створення малюнка-привітання, який виводиться, якщо користувач надав більше 9 правильних відповідей. Малюнок може бути таким, як на рисунку.

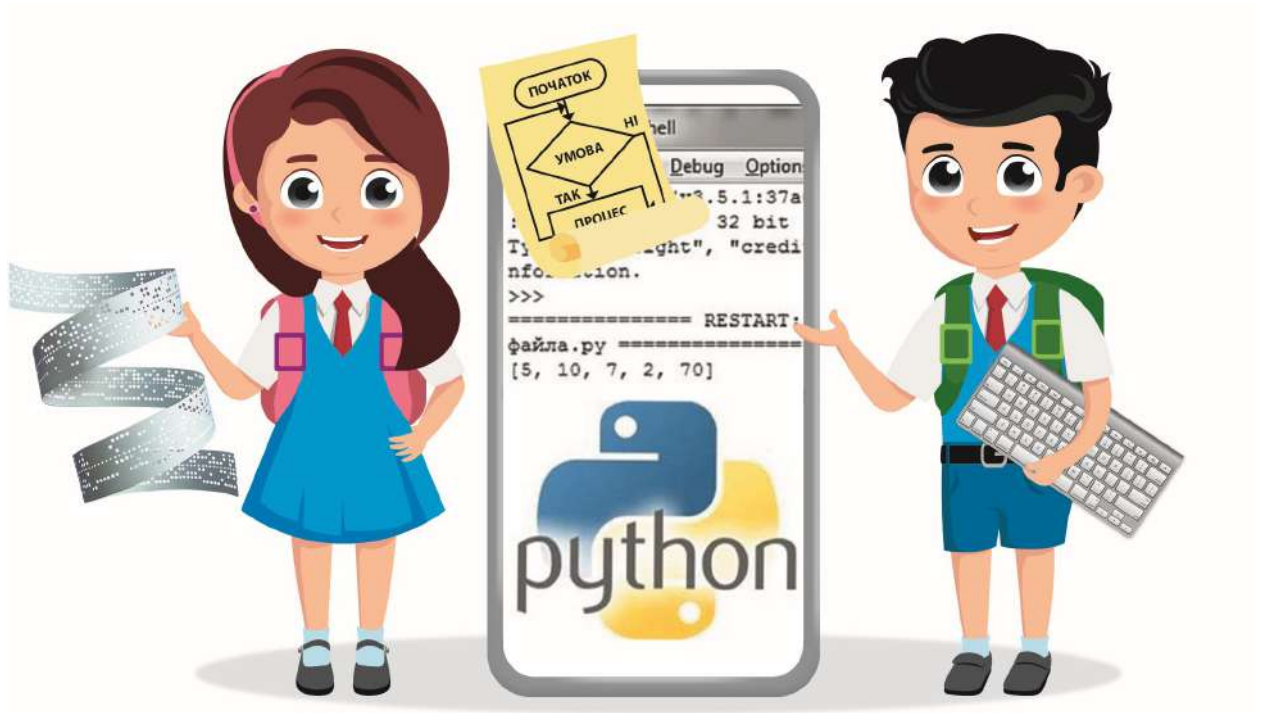


Перевірте роботу програми.

Зробіть висновок: як складати та виконувати циклічні алгоритми для розв'язування задач.

РОЗДІЛ 6

ПРАКТИКУМ З ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Практична робота 11. Створення словника комп'ютерних термінів

Практична робота 12. Створення презентації «Передавання інформації за допомогою графічних символів»

Практична робота 13. Створення зображень засобами «черепашчої» графіки

Практична робота 14. Складання та виконання алгоритмів із розгалуженнями та повтореннями

Практична робота 15. Складання та виконання алгоритмів із повтореннями

ПОВТОРЮЄМО

На уроках інформатики ви вже здобули певні знання та навички. А саме:

— ознайомилися з будовою та роботою комп'ютера; навчилися безпечно працювати з *операційною системою* — комплексом програм для керування комп'ютером, із засобами операційної системи для здійснення операцій над файлами та папками;

— дізналися про особливості *будови та роботи комп'ютерних мереж*, навчилися шукати потрібну інформацію, зберігати її та опрацьовувати;

— навчилися *редагувати та формувати текстовий документ* за допомогою текстового процесора; створювати презентацію: дізнались, як розробляти дизайн презентації та добирати стильове оформлення слайдів, додавати до слайдів анімаційні ефекти, звуковий супровід;

— ознайомилися з *прийомами роботи в програмному середовищі Python* і основами записування алгоритмів мовою **Python**, навчилися працювати в інтерактивному режимі, створювати й зберігати програмний код для подальшого використання; вмiєте створювати малюнки за допомогою «черепашачої» графіки.

1. Наведіть приклади опрацювання інформації.
2. Опишіть роль інформаційних технологій у житті людини.
3. Опишіть найпопулярніші служби інтернету.
4. Опишіть основні правила безпечної роботи в інтернеті.
5. Опишіть інструменти для форматування текстового документа.
6. Які об'єкти може містити комп'ютерна презентація?
7. Які види вікон існують у середовищі **Python**?
8. Для чого призначена функція **input()**; функція **print()**?
9. Який модуль містить команди «черепашачої» графіки?

У цьому розділі ви створите інформаційні продукти, користуючись інструментами програмних середовищ, з якими ознайомились у 5 класі.

ПРАКТИЧНА РОБОТА 11

Створення словника комп'ютерних термінів

Завдання: створити словничок комп'ютерних термінів, які стосуються апаратної частини комп'ютера.

Обладнання: комп'ютер зі встановленим текстовим процесором.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Завантажте текстовий процесор і створіть документ. Задайте:

- розмір паперу — **A4**;
- поля — **по 2 см**;
- орієнтацію — **книжкова**;
- шрифт — **Arial**;
- розмір — **36**;
- накреслення — **жирний**;
- масштаб — **одна сторінка**;
- вирівнювання — **по центру**;
- інтервал **Перед:** — **6 пт**,
- міжрядковий інтервал — **1,5 рядка**.

2. Оформіть *першу (титульну)* сторінку за зразком (рис. 1). Відформатуйте текст і оформіть титульну сторінку на свій розсуд.



Рис. 1

3. Виберіть у комп'ютерному словнику підручника терміни, що стосуються апаратної частини комп'ютера. Наберіть 6 означень.

4. Відформатуйте шрифт тексту *другої* сторінки за такими параметрами: розмір шрифту — **16 пт**; шрифт — **Times New Roman**; накреслення — **жирний**; колір шрифту термінів — **темно-синій**.

5. Установіть формат абзаців *другої* сторінки за такими параметрами (рис. 2): вирівнювання — **за шириною**; перший рядок — **навислий, 2 см**, інтервал: **Перед:, Після:** — по **6 пт**, міжрядковий — **1,5 рядка**.

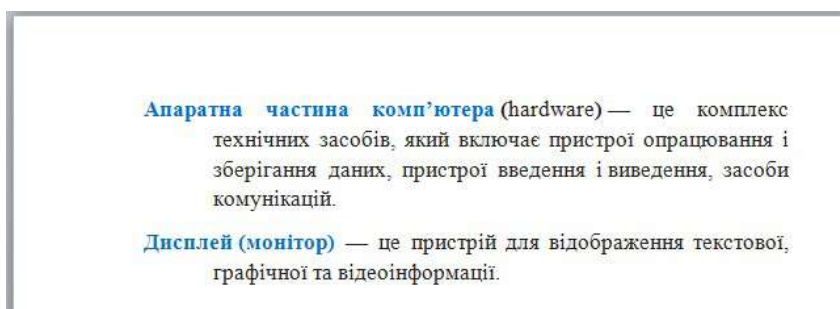


Рис. 2

6. Знайдіть в інтернеті зображення складових комп'ютера, скопіюйте та додайте на *титульну* сторінку.

7. Збережіть документ у файлі з назвою **Комп'ютерний словник** у власній папці. Закрийте всі вікна. Закінчіть роботу за комп'ютером.

Зробіть висновок: які засоби для створення та форматування текстового документа надає текстовий процесор.

ПРАКТИЧНА РОБОТА 12

Створення презентації «Передавання інформації за допомогою графічних символів»

Завдання: створити презентацію засобами **MS PowerPoint**, у якій дати пояснення, що символізують певні зображення.


Примітка. Мова є основним каналом інформації. Проте інформацію можна передавати й іншими шляхами (за допомогою зображень, піктограм, графічних символів). Використання знаково-символічних засобів дозволяє в стислому вигляді передавати значний обсяг інформації.

Обладнання: комп'ютер зі встановленою програмою **MS PowerPoint**.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Завантажте програму **PowerPoint** і створіть нову презентацію.

Кнопкою  на вкладці **ОСНОВНЕ** створіть 4 слайди майбутньої презентації. Змініть макет слайдів на такий, що містить **Заголовок**, два **Контейнери для тексту** та два **Комбіновані контейнери**. Застосуйте до всіх слайдів одну з тем оформлення, що на вкладці **Конструктор**.

2. На *першому* слайді додайте до заголовка назву презентації: **Передавання інформації за допомогою графічних символів**. У тексті слайда опишіть у підзаголовку, хто є автором презентації, наприклад **Презентація учениці 5-Б класу Сумлінної Поліни**.

3. Знайдіть в інтернеті інформацію за ключовими словами «Національний стандарт України», «графічні символи, кольори та знаки безпеки» і з'ясуйте, що означають символи безпеки, наведені на рис. 1.

Додайте до **Комбінованих контейнерів** зображення 2 знаків безпеки, до **Контейнерів тексту** уведіть пояснення використання цих знаків.



Рис. 1

4. Знайдіть в інтернеті інформацію за ключовими словами «символіка рослин». Додайте до заголовку *четвертого* слайда текст «Що символізують рослини?», до **Комбінованих контейнерів** — зображення рослин, які мають символічне значення.

Наприклад, за віруваннями українців верба — це священне дерево життя. Освяченою гілочкою верби вперше виганяли худобу на випас, з верби робили колиски. «Без верби і калини нема України», — говорить прислів'я.

Додайте написи з поясненням символіки цих зображень. Приклад оформлення слайда наведено на рис. 2.



Рис. 2

5. Знайдіть в інтернеті інформацію за ключовими словами «символіка тварин». Додайте до заголовку *четвертого* слайду текст «Що символізують рослини?», до **Комбінованих контейнерів** — зображення тварин, які символізують людські якості: мудрість, хоробрість тощо.

Наприклад, лев — символ відваги, безстрашності, мужності, а голуб — символ миру і добра, любові, дружби.

Додайте написи з поясненням символіки цих зображень.

6. Збережіть файл у форматі **PPTX** із назвою **Символіка** у власній папці.

Зробіть висновок: як коригувати розмір, колір та розташування текстів і зображень на слайдах презентації.

ПРАКТИЧНА РОБОТА 13

Створення зображень засобами «черепашачої» графіки

Завдання: скласти програму малювання символу виду спорту.

Обладнання: комп'ютер зі встановленим середовищем програмування **Python**.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Відкрийте вікно **IDLE** і створіть нове вікно програми. Збережіть файл у власній папці з назвою **Спорт**.

Основна вимога до графічних символів громадського призначення — бути зрозумілими без словесних пояснень. Розробіть на папері у клітинку

схематичне зображення спортсмена чи спортсменки, яке символізує певний вид спорту (рис. 1).

Визначте координати кінців відрізків (на рис. 1 сторона клітинки дорівнює 5 пікселів).

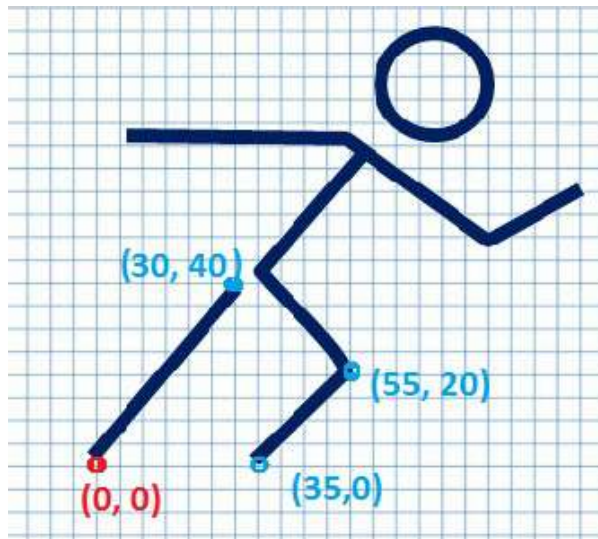


Рис. 1

2. Завантажте модуль **turtle** і встановіть ширину лінії 5 пікселів, колір ліній — синій.

```
from turtle import*
```

```
width(5)
```

```
color('blue')
```

3. Запишіть програму малювання символу, наведеного на рис. 1.

```
goto(30, 40)
```

```
up() # Піднімаємо перо
```

```
goto(35, 0) # Переходимо в потрібну точку
```

```
down() # Опускаємо перо
```

```
goto(55, 20)
```

```
goto(35, 45)
```

```
goto(55, 70)
```

```
up()
```

```
goto(5, 75)
```

```
down()
```

```
goto(50, 75)
```

```
goto(75, 50)
```

```
goto(105, 60)
```

```
up()
```

```
goto(70, 75)
```

```
down()
```

```
circle(15) # Малюємо коло радіуса 15
```

4. Додайте у графічне вікно назву виду спорту:

```
up()
```

```
goto(0, -25)
```

```
down()
```

```
write('Легка атлетика', font = 'Arial 16')
```

5. Запустіть програму на виконання, проаналізуйте результат виконання програми у вікні **Python Turtle Graphics**.

На рис. 2 наведено результат малювання спортивного символу.



Рис. 2

6. Розробіть програму малювання одного з символів видів спорту, поданих на рис. 3.



Рис. 3

Зробіть висновок: як передати інформацію мовою символів; яку роль відіграють символи на мапах, вказівниках, у путівниках тощо.

ПРАКТИЧНА РОБОТА 14

Складання та виконання алгоритмів із розгалуженнями та повтореннями

Завдання: відшукати базові алгоритмічні структури в прислів'ях та приказках, скласти програму, яка реалізує алгоритм прислів'я.

Обладнання: комп'ютер зі встановленим середовищем програмування Python.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правила безпеки.

1. Проаналізуйте наведені прислів'я та приказки. Визначте тип алгоритму, описаного в них.

- По бурі приходить гарна погода.
- Після дощу і сонце засяє.
- Посієш вітер — пожнеш бурю.
- Як випадуть у маю три дощі добрих, то дадуть хліба на три роки.
- Коси, коса, поки роса. Роса спала — робота пропала.
- Змазуй підводу, Клим, якщо зібрався в млин.
- Сім раз відміряй — один відріж.

2. Впишіть потрібні написи в блок-схему команди розгалуження (рис. 1) відповідно до прислів'я «В чужому оці й порошок видно, а у своєму і колоди не побачиш».

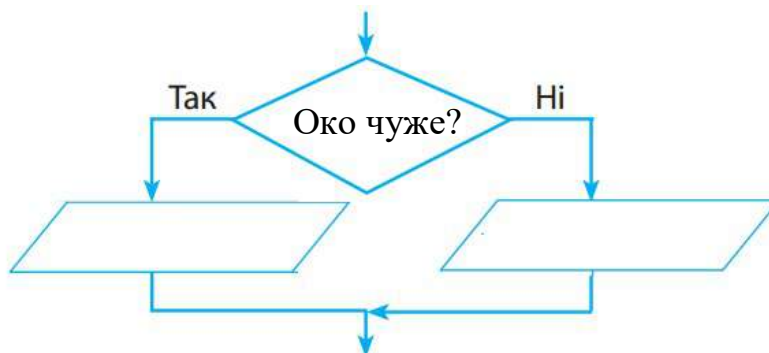


Рис. 1

3. Відкрийте вікно **IDLE** і створіть нове вікно програми. Наберіть код, що реалізує алгоритм прислів'я.

```
vidp = input('Око чуже? (Так/Ні)')  
if vidp == 'Так':  
    print('Видно порошок')  
else:  
    print('Не бачу колоду')
```

4. Впишіть потрібні написи в блок-схему команди повторення (рис. 2) відповідно до приказки «Раз, два, три, чотири, кицю грамоти учили».

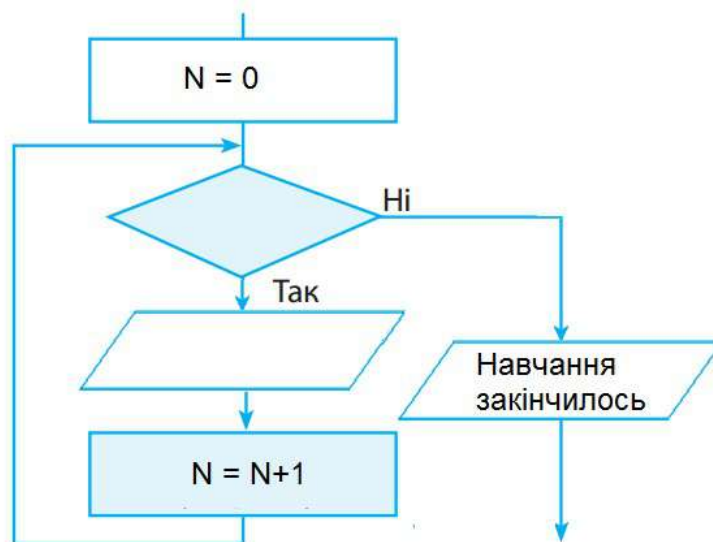


Рис. 2

5. Створіть нове вікно програми. Наберіть код, що реалізує алгоритм прислів'я.

```

for N in range(4):
    print('Кицю грамоти учили')
    print('Навчання закінчилось')
  
```

6. Побудуйте блок-схему команди розгалуження відповідно до приказки «Немазаний віз скрипить, а мазаний сам біжить» і напишіть програму для реалізації алгоритму.

Зробіть висновок: які базові алгоритмічні структури ви знайшли у прислів'ях та приказках; як створити і реалізувати алгоритми з розгалуженнями та повтореннями.

ПРАКТИЧНА РОБОТА 15

Складання та виконання алгоритмів із повтореннями

Завдання: скласти програму для перевірки уміння виконувати операції цілочисельного ділення (`//`) і знаходження остачі від ділення (`%`).

Обладнання: комп'ютер зі встановленим середовищем програмування **Python**.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Відкрийте вікно **IDLE** і створіть нове вікно програми. Збережіть файл із назвою **Тест**. Завантажте модуль для роботи з випадковими числами.

2. Програма пропонує користувачеві/користувачці розв'язати 6 виразів і підраховує кількість правильних відповідей. Створіть змінну **k** — лічильник правильних відповідей. Задайте початкове значення **k = 0**.

3. Запишіть оператор **for**, лічильник якого набуває значення з послідовності **0...2**.

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

4. Запишіть оператори тіла циклу:

1) присвойте змінним **a** і **b** випадкові значення з діапазону **1...9**:

```
a = randint(1, 20)
```

```
b = randint(1, 10)
```

2) напишіть оператор виведення, який виводить у консоль вираз цілочисельного ділення:

```
print(a, "/", b, "=", end = ' ')
```

Зверніть увагу! **end = ' '** у списку виведення означає, що після виведення виразу курсор не перейде на наступний рядок.

3) напишіть оператор введення, який запитує у користувача відповідь на питання:

```
c1 = int(input("?"))
```

4) напишіть оператор розгалуження, який перевіряє, чи правильна відповідь на питання:

```
if a//b == c1:
```

```
    print("Так!")
```

```
    k = k+1
```

```
else: print("Hi!")
```

5. Додайте до тіла циклу блок для перевірки вміння обчислювати остачу від ділення:

```
print(a, "%", b, "=", end = ' ')
```

```
c2 = int(input("?"))
```

```
if a%b == c2:
```

```
    print("Так!")
```

```
    k = k+1
```

```
else: print("Hi!")
```

6. Напишіть оператор виведення, який виводить у консоль значення лічильника правильних відповідей:

```
print("Правильних відповідей – ", k)
```

Перевірте роботу програми.

Зробіть висновок: як складати тестові програми для перевірки знань та умінь.

КОМП'ЮТЕРНИЙ СЛОВНИК

Абзац — частина тексту, набирання якого закінчується натисканням клавіші **Enter**.

Алгоритм — це скінченна послідовність указівок на виконання дій, спрямованих на розв'язування задачі.

Алфавіт мови — набір символів, що може використовуватись у програмному коді.

Блок-схема алгоритму — графічне подання алгоритму у вигляді блоків, сполучених лініями.

Виконавець — людина, тварина чи пристрій, здатні діяти за заданим алгоритмом.

Віруси — комп'ютерні програми, здатні самостійно поширюватися з одного комп'ютера на інший.

Гіперпосилання (посилання) — це виділений кольором або іншим чином текст, зображення чи кнопка, клацання яких викликає перехід на перегляд інших даних.

Гіпертекст — це текстовий документ, який містить гіперпосилання (слова, фрази, графічні об'єкти), клацання на яких дає змогу перейти до іншого місця в документі або до іншого документа.

Глобальна мережа — це мережа, що об'єднує комп'ютери та локальні мережі, розташовані в різних містах, державах і на різних континентах.

Дані — це інформація, подана в придатній для виконання інформаційних процесів формі.

Довільний показ — це іменована послідовність запланованого показу слайдів презентації.

Доменне ім'я — текстова адреса комп'ютера або іншого ресурсу в інтернеті.

Засоби інформаційної системи — це об'єкти живої та неживої природи.

Змінні — це величини, значення яких можуть змінюватися в ході виконання програми.

Значення — характеристика величини, яка може багаторазово змінюватися в процесі опрацювання інформації.

Інтернет — глобальна комп'ютерна мережа, що складається з мільярдів комп'ютерів у всьому світі.

Інтерфейс користувача — це сукупність засобів для взаємодії між користувачем і комп'ютером.

Інформаційна система — це сукупність взаємопов'язаних засобів і методів здійснення інформаційних процесів.

Інформаційна технологія — сукупність методів і процесів опрацювання інформації із застосуванням комплексу відповідних засобів.

Інформаційні процеси — це дії над інформацією.

Інформація — це відомості про об'єкти Всесвіту.

Кнопки дій — це низка фігур, що зображують кнопки, для яких вже призначено одну з часто вживаних дій.

Колонтитули — повідомлення, розташовані на полях сторінки.

Комп'ютерна мережа — це два або більше комп'ютерів, з'єднаних каналами зв'язку, якими передаються дані.

Комп'ютерна презентація — це набір електронних документів, підготовлених для перегляду на екрані монітора або на великому проєкційному чи іншому екрані.

Константи — це величини, значення яких не можуть змінюватися в ході виконання програми.

Локальна мережа — це мережа, що об'єднує комп'ютери та інші пристрої (принтери сканери тощо), розташовані близько один від одного — в межах школи, банку, заводу тощо.

Макет слайда — це умовна схема слайда, що визначає спосіб розташування об'єктів на слайді.

Методи інформаційної системи — це способи дії (взаємодії) об'єктів.

Мова програмування — це система позначень, яка використовується для запису алгоритмів для їх виконання за допомогою комп'ютера.

Онлайн-програми — програми, з якими можна працювати в інтернеті в режимі реального часу.

Операційна система — це набір програм, що автоматично завантажуються під час вмикання комп'ютера й використовуються для керування обчислювальними процесами, програмними й апаратними ресурсами.

Орієнтація сторінки — це спосіб розміщення тексту на сторінці.

Папка — це елемент файлової системи, контейнер для зберігання файлів та інших папок.

Повідомлення — це форма подання інформації. Без повідомлень не можна сприйняти інформацію, тому кажуть, що інформація передається за допомогою повідомлень.

Повторення (цикл) — це така форма організації дій, за якої одна й та сама послідовність дій виконується кілька разів залежно від певної умови.

Поля — це області сторінки вздовж її країв, у яких не міститься основний текст документа.

Припустимі дії — обмежений набір дій, що вміє виконувати певний виконавець.

Провайдер (від англ. *provider* — постачальник) — це організація, що надає послуги, пов'язані з доступом до глобальної мережі.

Програмне забезпечення — це набір різноманітних програм, які керують роботою комп'ютера, забезпечують ведення діалогу з користувачем, за їх допомогою опрацьовують різноманітні дані, створюють нові програми.

Речення — розділена пробілами послідовність слів, яка починається з великої літери та закінчується крапкою, знаком питання, знаком оклику або трьома крапками.

Розгалуження — це така форма організації дій, коли залежно від виконання або невиконання певної умови виконується одна з двох вказівок.

Рядок — це взята в одинарні або подвійні лапки послідовність будь-яких символів: цифр, літер, розділових знаків та ін.

Семантика мови — сукупність правил виконання комп'ютером команд, записаних мовою програмування.

Сервер — комп'ютер, який надає ресурси.

Символ — літера, цифра, знак пунктуації, арифметичної операції чи спеціальний знак.

Синтаксис мови — сукупність правил побудови команд мовою програмування.

Система команд виконавця — повний перелік команд, за якими виконавець може виконати одну або серію припустимих дій.

Системи опрацювання комп'ютерних презентацій — прикладне програмне забезпечення для створення й редагування презентацій на комп'ютері.

Слідування — це така форма організації вказівок в алгоритмі, за якої дії виконуються послідовно одна за одною, без пропусків або повторень.

Слово — послідовність символів, відділена від інших символів пропуском.

Соціальна мережа — «місце» в інтернеті, де спілкуються одночасно дуже багато людей: створюють групи за інтересами, обговорюють новини, обмінюються різними матеріалами, публікують оголошення.

Спам — масове розсилання небажаних повідомлень рекламного чи іншого характеру.

Тема слайда (презентації) — це іменованій стиль оформлення об'єктів на слайді: тла, шрифтів, графічних об'єктів тощо.

Тіло циклу — сукупність вказівок, що повторюється під час кожного проходження (ітерації) циклу.

Троянські програми — шкідливі програми, які можуть викрадати дані чи використовувати комп'ютер зі зловмисною метою.

Файл — це іменована сукупність даних, розміщених на носіїві, які в процесі зберігання, передавання та опрацювання вважаються єдиним цілим.

Форум — це засіб спілкування на сайті, де користувач може створювати власні теми або ставити запитання для відкритого обговорення.

Хакерство — отримання несанкціонованого доступу до комп'ютерних систем, як правило, із метою отримання секретної інформації.

Шаблон презентації PowerPoint — це схема (проєкт) презентації з одного чи кількох слайдів, які оформлені з дотриманням певного задуму.

Ярлик — це засіб швидкого доступу, який містить лише шлях до відповідного об'єкта.

World Wide Web (WWW) — це сукупність пов'язаних гіперпосиланнями документів, розміщених на безлічі комп'ютерів, із наданням можливості доступу до них через інтернет.

АЛФАВІТНИЙ ПОКАЖЧИК

А

Алгоритм 123

Алфавіт мови 144

Антивірус 54

Б

Блок-схема алгоритму 132

В

Виконавець алгоритму 128

Властивості алгоритму 124

Властивості сторінки 98

Г

Гіперпосилання (посилання) 85

Гіпертекст 49

Глобальна комп'ютерна мережа 47

Д

Дані 15

Довільний показ 87

Доменне ім'я 48

З

Засоби інформаційної системи 17

Змінна 145

Значення 147

І

Інтернет 47

Інтерфейс користувача 30

Інформатика 9

Інформаційна система 16

Інформаційна технологія 17

Інформаційні процеси 14

Інформація 9

К

Клієнт (робоча станція) 44

Ключове слово 58

Кнопки дій 86

Колонтитул 98

Коментар 151

Комп'ютерна презентація 67

Комп'ютерний вірус 53

Контейнер 77

Константа 145

Л

Лінійний алгоритм 149

Логічний вираз 167

Локальна комп'ютерна мережа 43

М

Методи інформаційної системи 17

Мова програмування 136

О

Об'єкт текстового документа 96

Оператор 173

П

Папка 32

Повідомлення 10

Повторення (цикл) 135

Пошукова система 58

Припустимі дії 109

Провайдер 48

Програмне забезпечення 25

Р

Розгалуження 134

Розміри сторінки 98

Рядок 127

С

Семантика мови 145

Сервер 44

Середовище виконавця 128

Синтаксис мови 144

Система команд виконавця 128

Системи опрацювання комп'ютерних презентацій 67

Складена умова 170

Слідування 133

Соціальна мережа 56

Спам 64

Список 182

Т

Тема оформлення слайда 79

Троянські програми 53

Ф

Файл 31

Форум 56

Х

Хакерство 53

Ш

Шаблон презентації 75

Я

Ярлик 32