

Київський національний університет імені Тараса Шевченка

КРЕНЕВИЧ А.П.

Методичні вказівки  
до лабораторних занять із дисципліни

"Об'єктно-орієнтоване програмування"

для студентів механіко-математичного факультету

Київ – 2019

УДК 519.942+550

Рецензенти:  
доктор фіз.-мат. наук, професор  
доктор фіз.-мат. наук,

Рекомендовано до друку вченою радою механіко-математичного  
факультету  
(протокол № \_\_ від \_\_ \_\_\_\_\_ 201\_\_ року)

**Кренивич А.П.**

Методичні вказівки до лабораторних занять із дисципліни "Об'єктно-орієнтоване програмування" для студентів механіко-математичного факультету – К.: ВПЦ "Київський Університет", 2019. – \_\_ с.

Посібник містить перелік завдань для аудиторної та самостійної роботи з дисципліни «Об'єктно орієнтоване програмування», що викладається студентам механіко-математичного факультету Київського національного університету імені Тараса Шевченка. Він містить завдання для засвоєння основних понять цього курсу, таких як клас, об'єкт, інкапсуляція, наслідування, поліморфізм, перевантаження операторів, абстрактні класи тощо.

Для студентів механіко-математичного факультету та викладачів, які проводять заняття з курсу "Об'єктно орієнтоване програмування".

## **ЗМІСТ**

ВСТУП	4
ЛАБОРАТОРНА РОБОТА 1. Інкапсуляція. Конструктор копіювання.	5
ЛАБОРАТОРНА РОБОТА 2. Графічна бібліотека turtle та її застосування до візуалізації об'єктів на екрані комп'ютера.	9
ЛАБОРАТОРНА РОБОТА 3. Наслідування та поліморфізм	10
ЛАБОРАТОРНА РОБОТА 4. Наслідування та поліморфізм – 2. Застосування бібліотеки turtle.	18
ЛАБОРАТОРНА РОБОТА 5. Спеціальні методи. Перевантаження операторів.	20
ЛАБОРАТОРНА РОБОТА 6. Колекції та ітератори.	27
ЛАБОРАТОРНА РОБОТА 7. Генератори та рекурентні співвідношення.	29
ЛАБОРАТОРНА РОБОТА 8. Створення власних класів виключень.	36
ЛАБОРАТОРНА РОБОТА 9. Абстрактні класи, інтерфейси, домішки	42
ЛАБОРАТОРНА РОБОТА 10. Створення графічних інтерфейсів.	39
ЛАБОРАТОРНА РОБОТА 11. Створення анімаційних програм.	39
ЛАБОРАТОРНА РОБОТА 12. Метапрограмування та рефлексія.	43
Список літератури та додаткових джерел	44

## **ВСТУП**

## **ЛАБОРАТОРНА РОБОТА 1.**

### **Інкапсуляція. Конструктор копіювання.**

#### **Контрольні запитання**

- 1.1. Що таке клас?
- 1.2. Що таке об'єкт класу?
- 1.3. Що таке екземпляр класу?
- 1.4. Що таке методи та поля класу?
- 1.5. Які методи/поля класу називаються статичними?
- 1.6. Які методи та поля називаються загальнодоступними? Прихованими?
- 1.7. Що таке інкапсуляція?
- 1.8. Як реалізується інкапсуляція у мові програмування Python?
- 1.9. Чи можна отримати доступ до прихованих методів та полів класу у Python?
- 1.10. Що таке конструктор та деструктор класу?
- 1.11. Що таке конструктор копіювання (у Python конструктор з підтримкою копіювання) та правило його опису?
- 1.12. Для чого застосовується UML?
- 1.13. Які основні зв'язки між класами ви знаєте? Як вони позначаються на діаграмі класів? Наведіть приклад для кожного зі зв'язків.
- 1.14. У чому відмінність між агрегацією і композицією? Наведіть приклад.

#### **Завдання для аудиторної роботи**

- 1.1. Опишіть клас QuadraticEquation (Квадратне рівняння).
  - Продумайте які атрибути та методи повинен мати клас.
  - Реалізуйте конструктор копіювання для екземплярів класу.
  - Реалізуйте метод, що повертає розв'язки цього рівняння.
- 1.2. Опишіть клас VectorNd для моделювання n-вимірних векторів. Реалізуйте методи
  - Введення вектора з клавіатури та виведення на екран.
  - Визначення довжини вектора.
  - Визначення середнього арифметичного його компонент.
  - Визначення найбільшої/найменшої компоненти вектора.

## Варіанти індивідуальних завдань

### Обов'язкові вимоги до виконання завдань

Кожен з класів описаних у цій лабораторній роботі, повинен мати конструктор, причому його реалізація повинна містити можливість копіювання об'єктів (конструктор копіювання). Для всіх класів необхідно реалізувати операції зчитування даних об'єктів з клавіатури та файлу, а також виведення даних екземплярів класів на екран. Кожен клас програми повинен розміщуватися у окремому файлі. Головний файл програми має називатися `main.py`.

Виконання лабораторної роботи передбачає оформлення діаграми класів за допомогою UML, що показує зв'язки між класами.

**1.1.** Опишіть класи для таких геометричних фігур зазначених нижче:

- a) Трикутник;
- b) Прямокутник;
- c) Трапеція;
- d) Паралелограм;
- e) Круг.

У кожному з класів реалізуйте операції знаходження периметра (для кола - довжини кола) та площі. За допомогою цих класів розв'яжіть таку задачу: Задано список фігур вищенаведених класів. Серед заданих фігур знайдіть фігуру, що має найбільшу площу та периметр.

Дані фігур мають зчитуватися з текстового файлу. Продумайте власний універсальний формат зображення даних у такому файлі.

**1.2.** Опишіть клас для геометричних фігур зазначених нижче:

- a) П'ятикутник;
- b) Шестикутник;
- c) Багатокутник (загальний).

У кожному з класів реалізуйте метод перевірки чи є цей багатокутник опуклим. За допомогою цих класів розв'яжіть таку задачу: Серед заданих багатокутників знайдіть всі які є опуклими.

Дані багатокутників мають зчитуватися з текстового файлу. Продумайте власний універсальний формат зображення даних у такому файлі.

**1.3.** Опишіть клас для таких об'ємних геометричних фігур:

- a) Тетраедр;
- b) Чотирикутна піраміда;
- c) Зрізана чотирикутна піраміда;

d) Куля;

У кожному з класів реалізуйте операції знаходження площі поверхні та об'єму. За допомогою цих класів розв'яжіть таку задачу: Задано список фігур вищенаведених класів. Серед заданих фігур знайдіть фігуру, що має найбільшу площу бічної поверхні та об'єм.

Дані фігур мають зчитуватися з текстового файлу. Продумайте власний універсальний формат зображення даних у такому файлі.

#### **1.4. Опишіть класи**

- a) Гість, що містить всю необхідну інформацію про жителя деякого готелю: ім'я, період проживання, кількість грошей виділених на проживання у готелі тощо.
- b) Кімната містить інформацію про кімнату готелю у тому числі вартість проживання за добу її доступність у визначений період.
- c) Готель, що містить список кімнат цього готелю, інформацію про те ким і коли вони зайняті тощо.

Використовуючи вищенаведені класи розв'яжіть задачі:

- a) Про кількість вільних кімнат у готелі на вказаний момент;
- b) Пошуку вільної кімнати у зазначений період;
- c) Поселити жителя на вказаний термін;
- d) Вартості проживання жителя у зазначений період;
- e) Прибутку, який отримає готель за вказаний період;
- f) Пошуку гостя у готелі (у заданий період);

**1.5.** У рядку міститься арифметичний вираз, що складається з чисел, літер латинського алфавіту (параметрів) та символів арифметичних операцій додавання, віднімання, множення та ділення. Опишіть клас «Рядковий калькулятор» для розв'язання цієї задачі. Реалізуйте методи:

- a) Задавання арифметичного виразу;
- b) Визначення значень параметрів, що входять до арифметичного виразу;
- c) Перевірки чи для всіх параметрів арифметичного виразу задано значення;
- d) Обчислення арифметичного значення виразу.

*Вказівка:* Пріоритетом арифметичних операцій можна знехтувати<sup>1</sup>.

---

<sup>1</sup> Для реалізації програми з урахуванням пріоритету арифметичних операцій необхідно здійснити перетворення виразу до оберненої польської нотації (перетворення до постфіксної форми), з якою пропонуємо ознайомитися самостійно.

**1.6.** Опишіть клас `Polynom` (Поліном) та реалізуйте методи:

- a) введення поліному з клавіатури;
- b) введення поліному з файлу;
- c) виведення поліному на екран;
- d) обчислення значення поліному у точці  $x$ ;
- e) обчислення похідної поліному;
- f) суми, різниці та добутку поліномів;

Використати цей клас для розв'язання задачі: задати два поліноми  $P1, P2$  та рядок, який містить вираз, що залежить від 2 поліномів. Наприклад,

$$P1 + P2 * P1 - P2$$

Обчислити поліном, який буде значенням цього виразу. Пріоритетом арифметичних операцій можна знехтувати.<sup>2</sup>

*Вказівка:* поліном зобразити у вигляді словника {ступінь: коефіцієнт}.

**1.7.** Опишіть клас `Студент`, що містить оцінки отримані ним протягом сесії (за 5-ти бальною та 100 бальною шкалами). Складіть програму для обчислення нарахованої студентам стипендії в залежності від результатів сесії:

- a) За старим підходом нарахування стипендії (середній бал за всі іспити має бути не меншим ніж 4 за 5-ти бальною шкалою).
- b) За новим підходом нарахування стипендії (стипендію отримують 40% від загального числа студентів, які є найкращими по рейтингу).

*Вказівка:* інформацію про студентів представити у вигляді списку. Дані зчитувати з (текстового) файлу.

**1.8.** Опишіть клас `Пасажир`, що містить інформацію про «місто відправлення» та «місто прибуття» деякого пасажирів та методи для розрахунку вартості квитка. Використовуючи цей клас, розрахуйте плату за проїзд усіх пасажирів. Усі маршрути зберігаються у списку кортежів (місто1, місто2, відстань), відома плата за 1 км відстані по цьому маршруту, а вартість квитка пропорційна відстані.

*Вказівка:* Дані про всіх пасажирів, маршрути та вартість 1км шляху по цих маршрутах зчитувати з (текстового) файлу.

---

<sup>2</sup> Див. обернена польська нотація.



## **ЛАБОРАТОРНА РОБОТА 2. Графічна бібліотека turtle та її застосування до візуалізації об'єктів на екрані комп'ютера.**

### **Контрольні запитання**

- 2.1. Для чого використовується графічна бібліотека turtle?
- 2.2. Який основний принцип роботи з бібліотекою turtle?
- 2.3. Наведіть приклад програми, що зображує не складну фігуру, наприклад прямокутник.
- 2.4. Наведіть основні операції з бібліотеки turtle?
- 2.5. Якою командою можна зобразити відрізок на екрані?
- 2.6. За допомогою якої команди можна зобразити коло (або його частину) на екрані?
- 2.7. Яким чином здійснюється зміна основного кольору малювання?

### **Завдання для аудиторної роботи**

2.1. Опишіть клас Triangle (Трикутник), для зображення трикутників на екрані за допомогою модуля turtle. Передбачте метод для встановлення кольору зображення трикутника. Використовуючи цей клас, зобразіть на екрані 100 трикутників. Розміри, позиції та кольори трикутників мають вибиратися випадковим чином.

### **Завдання для самостійної роботи**

2.1. Опишіть клас Flower (Квітка) для зображення квітки на екрані за допомогою модуля turtle. Продумайте структуру класу – квітка має бути композицією класів Petal (Пелюстка), Leaf (Листок) та Stem (Стебло). Використовуючи цей клас, зобразіть на екрані букет квітів.

2.2. Опишіть класи для графічного зображення за допомогою бібліотеки turtle таких фігур:

- a) Прямокутник. Сторони прямокутника паралельні осям координат. Для прямокутника задані довжини його сторін, а позиція на екрані визначається позицією лівого нижнього кута.
- b) Трикутник. Основа трикутника паралельна осі OX координат. Для трикутника задані довжини трьох його сторін, а позиція на екрані визначається позицією лівого нижнього кута.
- c) Коло. Для кола задано радіус, а його позиція на екрані визначається позицією центра.

У кожному з класів реалізуйте операції:

- a) Перевірки чи зображена фігура на екрані;
- b) Встановлення кольору малювання фігури;
- c) Зображення фігури на екрані (якщо її не зображено на екрані);
- d) Стирання фігури з екрану – зображення фігури на екрані кольором фону (якщо фігура зображена на екрані);
- e) Переміщення фігури (якщо вона зображена на екрані) у іншу позицію;

Опишіть клас Автомобіль, який буде композицією вищезазначених класів (тобто ці класи зображують частини автомобіля - корпус, колеса, двері, вікна).

Складіть програму зображення автомобіля та переміщення його по екрану

Дані фігур мають зчитуватися з текстового файлу. Продумайте власний універсальний формат зображення даних у такому файлі.

**2.3.** За допомогою модуля turtle, намалуйте аналоговий годинник, стрілки якого будуть оновлюватися кожні  $t$  секунд відповідно до поточного часу.

*Вказівка.* Опишіть класи:

- a) Цифра – призначений для зображення аналогової цифри.
- b) Циферблат – контейнер для об'єктів класу Цифра – призначений для зображення циферблату годинника з усіма цифрами.
- c) Стрілка – призначений для зображення стрілок годинника.

## **ЛАБОРАТОРНА РОБОТА 3.**

### **Наслідкування та поліморфізм**

#### **Контрольні запитання**

- 3.1.** Що таке наслідкування у об'єктно-орієнтованому програмуванні?
- 3.2.** Що таке нащадок класу, підклас або дочірній клас?
- 3.3.** Що таке базовий клас або супер клас?
- 3.4.** Які методи та поля базового класу успадковуються класом-нащадком?
- 3.5.** Що відбудеться, якщо у класі нащадку описати метод з іменем, яке є у батьківському класі?
- 3.6.** Що таке заміщення методу батьківського класу у нащадку?
- 3.7.** Як викликати метод батьківського класу, якщо у класі нащадку його було заміщено?

**3.8.** Що означає термін об'єктно-орієнтованого програмування поліморфізм? Наведіть приклад поліморфізму з реального життя та програмування.

**3.9.** Які методи називаються віртуальними? Чи підтримує Python віртуальні методи?

**3.10.** Що таке множинне наслідування та які основні проблеми пов'язані з ним виникають на практиці? Які способи їхнього розв'язання?

**3.11.** Як на діаграмах класів UML позначається наслідування.

### **Завдання для аудиторної роботи**

**3.1.** Опишіть клас Equation (Рівняння), що моделює лінійне алгебраїчне рівняння виду

$$bx + c = 0.$$

Опишіть у ньому метод solve(), що повертає усі розв'язки цього рівняння, залежно від його коефіцієнтів. Створіть класи QuadraticEquation (Квадратне рівняння) та BiQuadraticEquation (Біквадратне рівняння), що є нащадками класу Equation та моделюють рівняння виду

$$ax^2 + bx + c = 0.$$

та

$$ax^4 + bx^2 + c = 0.$$

відповідно.

Опишіть для них метод solve(), який заміщує відповідний метод батьківського класу та коректно розв'язує кожне з цих рівнянь, повертаючи кортеж його розв'язків. Чи можна задати клас BiQuadraticEquation, як нащадок класу QuadraticEquation, та використати вже описані у ньому методи? Якщо так, то оформіть це саме таким чином.

Використовуючи вищенаведену ієрархію класів розв'яжіть таку задачу: Задано список рівнянь, кожне з яких визначається списком коефіцієнтів починаючи з коефіцієнта при найвищому степені.

Наприклад, лінійне рівняння  $2x + 1 = 0$  задається таким списком [2, 1], квадратне рівняння  $3x^2 + x + 6 = 0$  відповідно списком [3, 1, 6], а біквадратне  $3x^4 + 5x^2 + 6 = 0$  відповідно списком [3, 0, 5, 0, 6].

Усі рівняння зберігається у текстовому файлі – список коефіцієнтів кожного рівняння розміщується в окремому рядку. Коефіцієнти розділені одним або кількома символами пропуску.

Серед заданих рівнянь знайдіть ті з них, що

- не мають розв'язків;
- мають один розв'язок;
- мають два розв'язки;
- мають три розв'язки;
- мають чотири розв'язки;
- мають нескінченну кількість розв'язків.

Крім цього вкажіть рівняння, що має

- найменший розв'язок
- найбільший розв'язок

серед розв'язків усіх рівнянь (що мають скінченну кількість розв'язків).

*Вказівка.* Для виведення рівняння опишіть віртуальний метод `show()`, що виводить рівняння на екран у вигляді зручному для розуміння.

Зобразіть ієрархію класів описаних у програмі за допомогою UML діаграми.

## Варіанти індивідуальних завдань

### Обов'язкові вимоги до виконання завдань

Кожен клас програми повинен розміщуватися у окремому файлі. Головний файл програми має називатися `main.py`.

Виконання лабораторної роботи передбачає оформлення діаграми класів за допомогою UML, що показує зв'язки між класами.

**3.1.** Опишіть клас *Фігура*, що інкапсулює основні геометричні характеристики та методи. Для фігури визначено методи:

- `dimension()` – що повертає вимірність фігури (дво-вимірна чи три-вимірна).
- `perimetr()` – що повертає периметр фігури (для три-вимірних фігур метод має породжувати виключення або повертати `None`).
- `square()` – що повертає площу фігури (для три-вимірних фігур метод має породжувати виключення або повертати `None`).
- `squareSurface()` – що повертає площу бічної фігури (для дво-вимірних фігур метод має породжувати виключення або повертати `None`).
- `squareBase()` – що повертає площу основи фігури (для дво-вимірних фігур метод має породжувати виключення або повертати `None`).
- `height()` – що повертає висоту фігури (для дво-вимірних фігур метод має породжувати виключення або повертати `None`).

- `volume()` –метод, що обчислює міру фігури (для плоскої фігури – площу, для об’ємної – відповідно об’єм).

Від класу Фігура (або його нащадків) наслідуються такі класи

- Трикутник
- Прямокутник
- Трапеція
- Паралелограм
- Круг
- Куля
- Трикутна піраміда (наслідується від класу Трикутник)
- Чотирикутна піраміда (наслідується від класу Прямокутник)
- Прямокутний паралелепіпед (наслідується від класу Прямокутник)
- Конус (наслідується від класу Круг)
- Призма (реалізуйте кілька класів, що є нащадками класів Трикутник, Трапеція, Паралелограм).

*Зауваження. Піраміди та конус вважайте правильними, а призму – прямою.*

1. Побудуйте UML діаграму ієрархії класів.
2. Нехай дано список фігур. Серед заданих фігур, знайдіть фігуру, міра якої є найбільшою.

**3.2.** Змоделюйте роботу деякого автопарку, має у наявності різні типи автомобілів. Опишіть клас `Car`, що має такі атрибути: базова витрата пального, об’єм бензобаку, кількість бензину у баку, пройдена автомобілем відстань та інші атрибути за необхідності.

У класі `Car` опишіть методи

- `go(distance)`, який змінює пройдений кілометраж автомобілем та залишок пального.
- `distanceToRefill()` – відстань яку може проїхати автомобіль до закінчення пального у баку.
- `getEarn()` – розмір прибутку автобазу за весь рейс виконаний автомобілем, якщо автомобіль було використано у комерційних цілях.

Вищенаведені методи можуть залежати від віртуальних методів наведених нижче:

- `fuelPerKm()`, який визначає скільки потрібно пального автомобілю для проїзду одного кілометра;
- `costKm()`, що визначає вартість одного кілометра дороги, якщо автомобіль використовується у комерційних цілях;

та мають різні реалізації залежно від типу автомобіля.

Нехай Personal (легковий автомобіль), Truck (вантажівка), Bus (автобус) – класи, що наслідують клас Car і заміщують методи `fuelPerKm()` та `costKm()`. При цьому, метод `fuelPerKm()` залежить

- для авто класу Personal від кількості пасажирів (+10% на кожного пасажера);
- для Truck від ваги вантажу (+25% на кожен тону вантажу).

Клас Personal містить інформацію про максимальну кількість пасажирів, яку може перевезти автомобіль і величину прибутку, що може бути отриманим за перевезення групи людей на відстань одного кілометра дороги (не залежно від кількості пасажирів), якщо автівка використовується у ролі таксі. Клас Truck містить інформацію про максимальну вантажопідйомність та вартість перевезення однієї тони вантажу на відстань один кілометр. Клас Bus містить інформацію про максимальне число пасажирів, що можуть бути перевезені за один рейс. При цьому заробіток, отриманий за рейс пропорційний кількості пасажирів та не залежить від пройденої автобусом відстані.

Розглядається деякий автопарк, що містить легкові автомобілі для таксі, автобуси для рейсових перевезень та вантажівок для вантажних перевезень. Автопарк необхідно оформити як окремий клас, що агрегує автомобілі і займається їхнім менеджментом.

У деякому файлі міститься список рейсів-запитів до автобази. Кожен рейс містить повну інформацію про його тип (вантажний/пасажирський/рейсовий), довжину рейсу, кількість пасажирів/масу вантажу. Продумайте та створіть власний формат зображення даних у такому файлі.

Використовуючи вищеописані класи потрібно розв'язати таку задачу: до автобази по черзі надходять запити, які вона має виконати. Коли надходить запит, підбирається автомобіль, що може виконати цей запит: вважається, автомобіль може виконати не більше одного запиту.

У результаті потрібно дати відповідь на такі запитання:

1. Чи зможе автобаза виконати всі поставлені запити? Якщо ні, то вивести у інший файл всі запити які автобаза не змогла виконати та причини не виконання.
2. Яку загальну відстань проїдуть всі автомобілі виконуючи всі запити?
3. Яку загальну кількість пасажирів перевезуть легкові автомобілі?
4. Яку середню кількість пасажирів перевезе кожен легковий автомобіль?
5. Яку загальну кількість вантажу перевезуть всі вантажні автомобілі?
6. Яку середню вагу перевезе кожен вантажний автомобіль?
7. Скільки автобаза заробить грошей виконуючи всі поставлені запити?

**3.3.** Задача полягає у обчисленні прибутків до бюджету, отриманих у вигляді податків.

Опишіть таку структуру проекту:

- Клас `Taxable` є суб'єктом оподаткування. Клас має метод `getTax()`, що залежить від віртуального методу `calculateTax()`, що буде реалізований у нащадках.
- Клас `TaxableCat` є нащадком класу `Taxable` – автомобіль що підлягає оподаткуванню. Він замінює метод `calculateTax()` таким чином: розмір податку розраховується за формулою:

*Податок = (Базова ставка) x (К. двигуна) x (К. віку) x (К. екологічний)*

Тут *Базова ставка* становить 200 гривень для бензинових двигунів, 300 грн для дизелів та 0 для електромобілів. Коефіцієнт двигуна – визначається діленням на 1000 робочого об'єму циліндрів в сантиметрах кубічних. Наприклад, якщо двигун має об'єм 1600 см<sup>3</sup>, то К двигуна становитиме 1,6. Коефіцієнт віку – вік авто (повних років) мінус 5. Якщо маємо 10-ти річну машину, для неї цей параметр буде 5. Коефіцієнт екологічний визначатиметься у відповідності до того, якому стандарту «Євро» відповідає автомобіль: Євро 2 і менше – 4; Євро 3 – 3; Євро 4 – 2; Євро 5 – 1,5; Євро 6 – 1.

Крім цього, на автомобілі молодші за 5 років з об'ємом двигуна понад 3 літри для бензинових двигунів та 2.5 літри з дизельним двигуном нараховується податок на розкіш у розмірі 25 тисяч грн.

Зауваження: для зручності рекомендується розділити клас `TaxableCat` на підкласи за типом двигуна – бензиновим/дизельним/електро.

- Клас `TaxableRealty` є нащадком класу `Taxable` – нерухомість що підлягає оподаткуванню. Він замінює метод `calculateTax()` таким чином: розмір податку розраховується як 1.5% від вартості нерухомості, якщо вона не потрапляє у розряд елітної. Елітною нерухомістю будемо вважати у цій задачі нерухомість, площа якої більша за 200 кв.м. У такому разі на таку нерухомість нараховується додатково податок на розкіш у розмірі 3% на площу, що перевищує 200 кв.м. Врахуйте, що нерухомість, що не потрапляє під категорію елітної, власник якої має пільгу звільняється від оподаткування.

Вся інформація про оподатковувані об'єкти зберігається у вигляді файлу. Продумайте та створіть власний формат зображення даних у такому файлі.

Реалізуйте клас `TaxableManager`, що розраховує прибутки до бюджету, отримані у вигляді податків. Використовуючи ці класи напишіть програму, що дає відповіді на такі запитання:

1. Яка загальна сума надходжень до бюджету буде отримана у вигляді податків?
2. Скільки об'єктів розкоші було оподатковано та на яку суму?
3. Кількість об'єктів звільнених від оподаткування.
4. Найвищий податок сплачений до бюджету.

**3.4.** Реалізуйте програму, що моделює роботу деякого магазину. Опишіть базовий клас `Goods` (Товар) та його нащадки, що реалізують конкретні види товарів – продукти харчування, одяг, побутова хімія, тощо. Кожен з видів товарів повинен містити характеристики, що суттєво відрізняють їх від базового класу (наприклад, продукти харчування мають термін придатності, одяг – розмір і так далі). Для зберігання інформації про товари на диску, розробіть власний формат (текстовий). Передбачте, що до магазину можуть надходити різноманітні запити, наприклад на покупка чи повернення товарів, перегляд їхніх характеристик тощо.

**3.5.** Опишіть клас `Matrix`, що моделює роботу з квадратними матрицями. Опишіть методи введення (з клавіатури та з файлу) та виведення (на екран та у файл) матриці. У програмі має бути перевірка, чи задана матриця є квадратною. Створіть класи-нащадки `Matrix2D` та `Matrix3D` класу `Matrix`, що моделюють роботу з квадратними матрицями розмірності 2 та 3 відповідно. Для кожного з цих класів реалізуйте методи відшукування визначника та перевірки чи матриця є виродженою (прототип яких має обов'язково бути визначеним у батьківському класі). Для обчислення визначника матриці розмірності 3 використайте розклад визначника за рядком (чи стовпчиком) та відповідний метод з класу `Matrix2D`.

Опишіть клас `Vector`, що моделює роботу з векторами та його нащадки класи `Vector2D` та `Vector3D` для 2- та 3-вимірних векторів відповідно.

Опишіть клас `Solver`, що використовує вищенаведені класи та розв'язує методом Крамера системи лінійних алгебраїчних рівнянь.

Використовуючи описані структури, напишіть програму, що розв'язує набір заданих алгебраїчних рівнянь.

**3.6.** Задача полягає у написанні програми для обчислення та розподілу стипендій студентів деякої спеціальності.

Розв'язання передбачає опис такої ієрархії класів:



- `Person` – особа – клас, що містить такі атрибути як прізвище та ім'я людини, стать, тощо та методи для роботи з ними.
- `Student` – нащадок класу `Person`. Містить інформацію про студента, таку як номер студентського квитка, спеціальність, результати сесії, тощо та методи роботи з ними.
- `SpecialStudent` – студент-пільговик – студент, що має право на пільгову стипендію.

Повна інформація про студентів (враховуючи результати сесії) міститься у окремому файлі. Продумайте та створіть власний формат зображення даних у такому файлі.

Опишіть клас `StipendManager` - менеджер стипендії, що розраховує та розподіляє стипендію студентів залежно від їхніх результатів сесії.

Стипендія, яку отримує студент буває двох видів – академічна та соціальна. Академічна стипендія нараховується за результатами сесії. Соціальну стипендію можуть отримувати лише студенти-пільговики, якщо вони не отримали академічної та не мають незадовільних оцінок за результатами сесії.

Клас `StipendManager` розподіляє стипендію, використовуючи старий підхід: студент отримує академічну стипендію якщо його середній бал (без врахування заліків) є не нижчим ніж 75 балів. Якщо студент має всі оцінки за іспити не нижчі за 90 балів – такий студент отримує підвищену стипендію, що на 20% вища за звичайну академічну стипендію.

Опишіть клас `ModernStipendManager`, що є нащадком класу `StipendManager` і розподіляє стипендію за новим підходом: стипендію отримують 45% студентів спеціальності з найкращими результатами відповідно до рейтингового списку.

Використовуючи вищенаведену структуру класів, реалізуйте програму, що має можливість відповісти на такі запитання:

1. Скільки студентів отримає академічну стипендію?
2. Скільки студентів отримає соціальну стипендію?
3. Яка загальна сума, буде виплачена студентам?
4. Який середній бал студентів що отримують академічну стипендію?
5. Який середній бал студентів, що отримують соціальну стипендію?
6. Який підхід передбачає більші витрати грошей по кожній спеціальності та в цілому?

**3.7.** Опишіть клас `Employee`, що моделює працівника деякої ІТ компанії. Цей клас має містити інформацію, що може стосуватися усіх працівників – прізвище та ім'я, початок роботи у компанії, освіта, базовий оклад, тощо.

Проте, компанія має робітників різних типів – працівники відділу кадрів, інженери-програмісти, тестувальники, тощо – які є нащадками класу `Employee`. Заробітна плата ( $S$ ) працівників рахується за різними формулами, наведеними нижче, та залежить від типу працівника, базового окладу ( $B$ ), що визначається залежно від типу працівника, стажу працівника у компанії ( $Y$  – кількість повних років роботи у компанії) тощо.

- Інженер-програміст:

$$S = G \times B \times (1 + Y/10)$$

де  $G$  – коефіцієнт, що відповідає рівню кваліфікації працівника: 1 – junior, 2 – middle, 3 – senior.

- Працівник відділу кадрів:

$$S = B \times (1 + 0.1 \times [Y/3])$$

- Тестувальник:

$$S = B \times (1 + Y/5)$$

У кожному класі опишіть метод `calculateSalary()`, що повертає заробітну плату працівника та заміщує задекларований у базовому класі відповідний метод.

Напишіть програму, що підраховує

- загальну суму витратків по заробітній платі;
- загальну суму податків, що має сплатити компанія державі, якщо податок на одного працівника визначається як 20% від його заробітної плати.
- загальну суму виплат працівникам компанії.

## ЛАБОРАТОРНА РОБОТА 4.

### Наслідування та поліморфізм – 2. Застосування бібліотеки `turtle`.

#### Завдання для аудиторної роботи

**4.1.** Переробіть задачу 2.1 із застосуванням наслідування та поліморфізму. Реалізуйте базовий клас `Figure` (Фігура), що містить усі спільні методи, які не залежать від форми фігури. Опис методів, що залежать від форми фігури реалізуйте у нащадках класу `Figure`. На базі класу `Figure` опишіть класи для таких фігур – Коло, Квадрат (одна з основ паралельна осі

OX), Рівносторонній трикутник (одна з основ паралельна осі OX), Рівнобічна трапеція (нижня основа паралельна осі координат OX).

Опис базового класу та заготовки для його нащадків містяться у модулі за посиланням <https://github.com/krenevych/oop/blob/master/labs/lab04/t1.py>. Завантажте вказаний файл, реалізуйте код, що відмічений секціями TODO.

Для перевірки правильності роботи реалізованих алгоритмів запустіть реалізовану програму на виконання. Ви маєте побачити процес зображення та переміщення чотирьох фігур – по одному екземпляру описаних класів.

Використовуючи описані класи, зобразіть на екрані 100 різних фігур. Розміри, позиції та кольори фігур мають вибиратися випадковим чином.

*Вказівка.* Використайте модуль `random` генерації випадкових чисел.

### Варіанти індивідуальних завдань

**4.1.** Задача полягає у реалізації гри «Хрестики-нулики». Для цього Реалізуйте клас `Figure` (Фігура), що має методи:

- `setPosition(x,y)` – встановити положення фігури на екрані у позицію  $(x, y)$ .
- `show()` – зобразити фігуру на екрані у позиції  $(x, y)$ .
- `hide()` – приховати фігуру з екрану.

останні два з яких залежать від віртуального методу `draw(color)`, що зображує фігуру на екрані заданим кольором та буде реалізований у нащадках класу `Figure`. Реалізуйте класи для зображення хрестиків, нуликів та ігрового поля, що є нащадками класу `Figure` з заміщеною реалізацією методу `draw(color)`.

*Вказівка.* Для відслідковування подій мишки скористайтеся методом `listen()` з бібліотеки `turtle`.

**4.2.** Розширте функціональність задачі 2.3 додавши можливість відображення годинника у цифровому форматі. При цьому створіть базовий клас `Watch` (годинник) та його нащадки `AnalogWatch` та `DigitalWatch`, що реалізують функціонал аналогового та цифрового годинника, відповідно. Крім цього необхідно перебачити розширення функціоналу цифрового годинника можливістю показувати час у 12- та 24 годинному стандарті.

**4.3.** Задача полягає у реалізації гри «Шашки». Для цього реалізуйте клас `Figure` (Фігура), що має методи:

- `setPosition(x,y)` – встановити положення фігури на екрані у позицію  $(x, y)$ .
- `setBorderColor(color)` – встановити колір рамки фігури.
- `setFillColor(color)` – встановити колір заливки фігури.

- `show()` – зобразити фігуру на екрані у позиції  $(x, y)$ .
- `hide()` – приховати фігуру з екрану.

останні два з яких залежать від віртуальних методів `draw(color)` та `drawBorder(color)`, що зображують на екрані заданим кольором фігуру та її рамку та реалізовані у нащадках класу `Figure`.

Реалізуйте класи для зображення шашок, клітин ігрового поля та, власне, ігрового поля, що є нащадками класу `Figure` з заміщеними реалізаціями методів `draw(color)` та `drawBorder(color)`.

У програмі потрібно реалізувати перевірку правильності черговості ходу та позиції шашки. Вибір шашки реалізуйте за допомогою миші – для обраної шашки має змінюватися колір рамки (наприклад на зелений).

*Вказівка.* Для відслідковування подій мишки скористайтеся методом `listen()` з бібліотеки `turtle`.

## ЛАБОРАТОРНА РОБОТА 5.

### Спеціальні методи. Перевантаження операторів.

#### Контрольні запитання

- 5.1. Які методи класу називають «спеціальними»? Чи можна за іменем методу визначити його приналежність до спеціальних методів?
- 5.2. Яке правило виклику спеціальних методів?
- 5.3. Назвіть кілька спеціальних методів та поясніть їхнє призначення?

#### Завдання для аудиторної роботи

5.1. Опишіть клас `ProtectedDictInt`, що є словником у якому, ключами можуть бути лише цілі числа та заборонена операція зміни значення за ключем (крім випадку додавання нової пари ключ-значення). Перевантажте для цього класу оператор `[]` (квадратні дужки) у такий спосіб:

- для запису – додавання нової пари ключ-значення. Операція має працювати лише для додавання нової пари – у випадку спроби зміни значення за існуючим ключем має породжуватися помилка;
- для читання – повертає значення зі словника.

Також для цього класу перевантажте оператори арифметичних операцій у такий спосіб:

- операція `+` означає додавання до словника пару ключ-значення. Правий операнд може бути іншим словником `ProtectedDictInt` або кортежем з двох елементів, що містить пару ключ-значення. У

випадку, якщо правий операнд – словник, що містить принаймні один ключ, що міститься у словнику, операція має породжувати помилку, оскільки операція зміни значень заборонена.

- операція – означає видалення пари ключ значення. Правий операнд у цій операції – ключ, який потрібно видалити зі словника.

Також для цього класу перевантажте

- оператор `in`, що перевіряє чи входить заданий ключ до словника.
- функцію `len()` – визначення кількості елементів у словнику (пара ключ-значення це один елемент).
- функцію `str()` – перетворення словника у рядок, для використання у функції `print()`.

## Варіанти індивідуальних завдань

### Обов'язкові вимоги до виконання завдань

У кожному з класів, описаних для розв'язання задачі має бути перевантажений спеціальний метод `__str__(self)`, який буде повертати рядок, що містить докладну інформацію про конкретний об'єкт та може використовуватися як аргумент функції `print()`. Конструктор кожного з класів повинен бути з підтримкою копіювання. Вхідні дані для кожної задачі мають зберігатися у текстових файлах у форматі зручному для сприйняття. У випадку необхідності, рекомендується описати інші функції та методи, крім зазначених в умові.

**5.1.** Опишіть клас `Rational`, що призначений для моделювання роботи з раціональними числами. Кожне раціональне число визначається як пара цілих чисел

$$(n, d)$$

таких, що  $n/d$  – нескоротний дріб. У цьому класі

- опишіть конструктор (з можливістю копіювання), що приймає два цілих аргументи та проводить, за необхідності, операцію скорочення.
- перевантажте оператори арифметичних операцій: передбачте, що правим операндом відповідної арифметичної операції може бути не лише екземпляр класу `Rational` але й ціле число (якщо відповідна операція не має сенсу, то має породжуватися повідомлення про помилку);

- оператор «круглі дужки», що повертає число у вигляді десяткового дробу;
- оператор «квадратні дужки», що за ключами "n" (скорочення від «numerator» – чисельник) та "d" (скорочення від «denominator» – знаменник) повертає для читання/запису чисельник та знаменник цього числа відповідно.

Використовуючи вищенаведений клас та перевантажені оператори обчисліть значення заданого арифметичного виразу.

**5.2.** Опишіть клас `Segment` для роботи з відрізками/інтервалами на числовій осі. Відрізок/інтервал містить такі атрибути:

- ознака чи порожній відрізок/інтервал;
- початок та кінець відрізка/інтервалу;
- ознаки чи включений початок/кінець у відрізок/інтервал (чи є об'єкт інтервалом, лівим/правим напівінтервалом чи відрізком);

Опишіть клас `SegmentSet`, що є множиною, що утворена об'єднанням кількох інтервалів та підтримує операції роботи з множинами (перетин, об'єднання, різницю). Для цього класу перевантажте оператори арифметичних операцій: передбачте, що правим операндом відповідної арифметичної операції може бути не лише екземпляр класу `SegmentSet`, але й екземпляр класу `Segment` або дійсне число. При цьому будемо вважати, що

- операція  $*$  означає перетин множин;
- операція  $+$  означає об'єднання множин;
- операція  $-$  означає різницю множин;
- операція  $/$  означає симетричну різницю множин;

Використовуючи вищеописані класи, напишіть програму, що розв'язує систему квадратних нерівностей вигляду

$$a_i x^2 + b_i x + c_i \geq 0, \quad i = 1, \dots, n$$

причому у деяких нерівностях вище може бути строга нерівність.

**5.3.** Опишіть клас `CustomSet`, що моделює дискретну множину об'єктів. Для цього класу перевантажте оператори арифметичних операцій: передбачте, що правим операндом відповідної арифметичної операції може бути не лише екземпляр класу `CustomSet`, але й екземпляр іншого класу (у цьому випадку правий операнд інтерпретується як одно-елементна множина). При цьому будемо вважати, що

- операція  $*$  означає перетин множин;
- операція  $+$  означає об'єднання множин;
- операція  $-$  означає різницю множин;

- операція / означає симетричну різницю множин;

Також для цього класу перевантажте функцію `len()` визначення кількості елементів множини.

Використовуючи вищеописаний клас, напишіть програму, що для заданого набору текстових файлів знаходить перелік та кількість різних слів, що

- містяться у всіх файлах;
- що міститься принаймні у одному з файлів;
- перелік слів, що містяться лише у першому файлі і не містяться у інших файлах.

**5.4.** Опишіть клас `MutableString` (Змінюваний рядок), що моделює роботу з рядком як списком символів. Для цього класу перевантажте функцію `len()` – визначення кількості символів рядка. Перевантажте для цього класу оператор `[]` (квадратні дужки), що повертає за індексом для читання/запису відповідний символ рядка. Також для цього класу перевантажте оператори арифметичних операцій у такий спосіб:

- операція `+` означає конкатенацію рядків;
- операція `*` означає дублювання рядків (правим операндом має бути ціле число).

Крім цього перевантажте оператор `in`, що перевіряє чи входить заданий символ до рядка.

Використовуючи цей клас розв'яжіть таку задачу: Текстовий файл містить багаторядковий текст українською мовою. Проте деякі українські літери у цьому тексті замінені на схожі за нарисом англійські літери (i, p, c, тощо). Потрібно виправити заданий текст, записавши результат у той же файл. У результаті своєї роботи, програма має повертати загальну довжину відкоригованого тексту.

**5.5.** Для класу `Matrix`, (у тому числі `Matrix2D` та `Matrix3D`) визначених у задачі 3.5 перевантажте

- оператори арифметичних операцій: передбачте, що правим операндом відповідної арифметичної операції може бути не лише екземпляр класу `Matrix2D` та `Matrix3D`, але й екземпляр класу `Vector2D` або `Vector3D` (якщо відповідна операція не має сенсу, то має породжуватися повідомлення про помилку);
- оператор «круглі дужки», що повертає значення визначника цих матриць;

- оператор «квадратні дужки», що повертає для читання/запису відповідний рядок матриці як екземпляр класу Vector2D або Vector3D.

Для класу Vector (у тому числі Vector2D та Vector3D) перевантажте

- оператори арифметичних операцій: передбачте, що правим операндом відповідної арифметичної операції може бути не лише екземпляр класу Vector2D або Vector3D, але й екземпляр класу Matrix2D, Matrix3D, а також дійсні числа, для множення вектора на скаляр (якщо відповідна операція не має сенсу, то має породжуватися повідомлення про помилку);
- оператор «квадратні дужки», що повертає для читання/запису відповідний елемент вектора.

Використовуючи вищенаведені класи та їхні перевантажені оператори, реалізуйте алгоритми:

- що розв'язує систему лінійних алгебраїчних рівнянь методом Гауса;
- що знаходить, обернену матрицю методом Гауса;

Перевірте правильність реалізованих алгоритмів використовуючи метод Крамера, реалізований раніше, під час виконання задачі 3.5.

**5.6.** Розширте клас Polynom, описаний для розв'язання задачі 1.6, перевантаживши

- оператори арифметичних операцій: передбачте, що правим операндом відповідної арифметичної операції може бути не лише екземпляр класу Polynom, але й дійсні числа (якщо відповідна операція не має сенсу, то має породжуватися повідомлення про помилку);
- оператор «круглі дужки», що повертає значення полінома для заданого значення аргументу;
- оператор «квадратні дужки», що повертає для читання/запису коефіцієнт при відповідному степені полінома.

Використовуючи вищенаведений клас та його перевантажені оператори, розв'яжіть задачу 1.6.

**5.7.** Опишіть клас Complex (Комплексне число), що моделює роботу з комплексними числами. Для цього класу перевантажте оператори арифметичних операцій (+, -, \*, /): передбачте, що правим операндом відповідної арифметичної операції може бути не лише екземпляр класу Complex, але й дійсне/ціле число. Крім цього, для класу Complex, перевизначте операцію \*\* правим операндом якої може бути не лише ціле



число, але й дійсне (у випадку останнього використайте тригонометричну чи показникову форму комплексного числа та виберіть головну гілку кореня). Використовуючи описаний клас, реалізуйте програму, що розв'язує квадратне рівняння з дійсними та комплексними коефіцієнтами.

**5.8.** Визначте клас `Sentence` (Речення), як список слів. Для цього класу перевантажте функцію `len()` – визначення кількості слів речення. Перевантажте для цього класу оператор `[]` (квадратні дужки), що повертає за індексом для читання/запису відповідне слово речення. Також для цього класу перевантажте оператори арифметичних операцій у такий спосіб:

- операція `+` означає злиття двох речень, якщо правим операндом є екземпляр класу `Sentence` або додавання слова до речення, якщо правим операндом є рядковий літерал;
- операція `-` означає вилучення всіх слів з речення, що входять до правого операнду (якщо правим операндом є екземпляр класу `Sentence`) або вилучення слова, якщо правим операндом є рядковий літерал.

Крім цього перевантажте оператор `in`, що перевіряє чи входить задане слово до речення.

Використовуючи цей клас розв'яжіть таку задачу: Текстовий файл містить багаторядковий текст українською мовою. Необхідно у цьому тексті

- замінити всі задані слова іншими заданими словами;
- видалити задані слова;

У результаті своєї роботи, програма має повертати загальну кількість слів відкоригованого тексту.

**5.9.** Опишіть клас `CustoomDict` (Словник користувача), екземпляри якого можуть містити лише пари (ключ : значення), у яких ключами можуть бути лише рядки, що містять послідовності літер українського (англійського) алфавіту, а значеннями елементи чисельних типів (`int`, `float`). У цьому класі перевантажте

- оператор «квадратні дужки» для читання/зміни/запису відповідної пари `{key:val}`;
- функцію `len()` – визначення кількості пар у словнику;
- оператор `in`, що перевіряє чи міститься заданий ключ у словнику;
- оператор `+=`, що додає у поточний словник усі пари зі словника, що є правим операндом (при цьому, якщо у словниках є значення з однаковими ключами, то їхні значення додаються).

Використовуючи цей клас, розв'яжіть задачу: Текстовий файл містить багаторядкову послідовність слів українською (англійською) мовою. Використовуючи клас `CustoomDict`

- перевірте чи входить задане слово до рядка та яку кількість разів;
- знайдіть слово, що зустрічається у тексті найбільшу кількість разів;
- знайдіть слово, що зустрічається у тексті задану кількість разів;
- знайдіть найдовше та найкоротше слово тексту;
- знайдіть кількість різних слів у тексті.

**5.10.** Опишіть клас `StringDict` (Рядковий словник), екземпляри якого можуть містити лише пари (ключ : значення), у яких ключами можуть бути лише рядки, що містять послідовності літер українського та/або англійського алфавіту, а значеннями – списки рядків з літер українського та/або англійського алфавіту.

У цьому класі перевантажте

- оператор «квадратні дужки» для читання/зміни/запису відповідної пари `{key:val}`;
- функцію `len()` – визначення кількості пар у словнику;
- оператор `in`, що перевіряє чи міститься заданий ключ у словнику;
- оператор `+=`, що додає у поточний словник усі пари зі словника, що є правим операндом (при цьому, якщо у словниках є значення з однаковими ключами, то їхні значення конкатенуються без повторень).

Використовуючи цей клас, розв'яжіть задачу: Дано два текстових файли. Перший файл містить послідовність англійських слів, а другий – послідовність українських слів, що є перекладами слів першого файлу. При цьому кожному англійському слову з першого файлу відповідає рівно одне слово з другого файлу (якщо для англійського слова існує кілька перекладів, то це слово повинно траплятися у першому файлі рівно стільки разів, скільки для нього зазначено перекладів). Порядок пар може також бути довільним.

Англійською	Українською
mystery mystery cat mystery cat apple	таємниця загадка кіт таїнство кішка яблуко

Використовуючи клас `StringDict`

- побудуйте англо-український словник;
- побудуйте україно-англійський словник;
- знайдіть усі переклади для заданого слова (якщо вхідне слово задається українською мовою, то переклади мають бути англійською і навпаки);

- знайдіть слово, що має найбільшу кількість перекладів;
- знайдіть усі англійські слова, що мають рівно один переклад;
- знайдіть кількість різних англійських слів у словнику;
- знайдіть кількість різних українських слів у словнику.

## **ЛАБОРАТОРНА РОБОТА 6.**

### **Колекції та ітератори.**

#### **Контрольні запитання**

- 6.1.** Що у програмуванні означає термін колекція (контейнер)?
- 6.2.** Що таке ітератор?
- 6.3.** Що означає: клас підтримує ітераційний протокол?
- 6.4.** Наведіть приклади базових типів мови програмування Python, що підтримують ітераційний протокол?
- 6.5.** Наведіть шаблон реалізації ітераційного протоколу для класу? Особливості реалізації?
- 6.6.** Наведіть правило використання екземпляру класу, що підтримує ітераційний протокол? За допомогою циклу `for`? За допомогою циклу `while`?

#### **Завдання для аудиторної роботи**

- 6.1.** Реалізуйте для класу `ProtectedDictInt` з задачі 5.1, підтримку ітераційного протоколу, таким чином, щоб ітератор перебирав усі ключі словника у порядку зростання.

#### **Варіанти індивідуальних завдань**

- 6.1.** Опишіть клас `CustomList`, що є списком який може містити лише цілі числа. Для описаного класу опишіть ітератор, що перебирає всі елементи списку за таким правилом:

- спочатку непарні елементи списку у порядку зростання;
- далі парні елементи списку у порядку спадання.

- 6.2.** Опишіть ітератор для ітерації всіх елементів об'єкту класу `SegmentSet` описаного в задачі 5.2, у порядку відповідно до їхнього розташування на дійсній осі. Виведіть результат задачі 5.2 застосовуючи описаний ітератор.

- 6.3.** Опишіть ітератор для ітерації всіх елементів екземпляру класу `CustomSet`, описаного в задачі 5.3. Виведіть результат задачі 5.3 застосовуючи описаний ітератор.

**6.4.** Опишіть ітератор для ітерації всіх елементів екземпляру класу `MutableString`, описаного в задачі 5.4. Розв'яжіть задачу 5.4 застосовуючи описаний ітератор.

**6.5.** Опишіть ітератор для ітерації всіх елементів матриці, що є екземпляром класу `Matrix`, описаного в задачі 5.5. Перебір елементів має здійснюватися по рядках. Використовуючи описаний ітератор знайдіть:

- найбільший та найменший елементи матриці;
- суму та середнє арифметичне елементів матриці.

**6.6.** Опишіть ітератор для перебору всіх коефіцієнтів поліному, що є екземпляром класу `Polynom`, описаного в задачі 5.6. Перебір елементів має здійснюватися у порядку зростання степенів змінної. Використовуючи описаний ітератор

- виведіть на екран зображення поліному;
- знайдіть найбільший та найменший коефіцієнти заданого поліному;
- знайдіть найбільше відношення коефіцієнту полінома до степеня, що йому відповідає.

**6.7.** Опишіть клас `CustomSet`, що є множиною чисел та рядків. Для описаного класу опишіть ітератор, що перебирає всі елементи множини за таким принципом:

- спочатку цілі елементи множини у порядку зростання;
- далі дійсні елементи множини у порядку спадання;
- в кінці рядкові елементи множини у лексикографічному порядку.

**6.8.** Опишіть ітератор для перебору всіх слів речення, що є екземпляром класу `Sentence`, описаного в задачі 5.8. Перебір слів має здійснюватися у лексикографічному порядку. Використовуючи описаний ітератор, виведіть всі слова заданого речення.

**6.9.** Опишіть ітератор для перебору всіх ключів словника, що є екземпляром класу `CustomDict`, описаного в задачі 5.9. Перебір ключів має здійснюватися у лексикографічному порядку. Використовуючи описаний ітератор, виведіть всі слова заданого словника.

**6.10.** Опишіть ітератор для перебору всіх ключів словника, що є екземпляром класу `StringDict`, описаного в задачі 5.10. Перебір ключів має здійснюватися у лексикографічному порядку. Використовуючи описаний ітератор, виведіть всі слова заданого словника разом з їхніми перекладами.

## ЛАБОРАТОРНА РОБОТА 7.

### Генератори та рекурентні співвідношення.

#### Контрольні запитання

7.1.

7.2.

#### Варіанти індивідуальних завдань

#### Обов'язкові вимоги до виконання завдань

В усіх задачах цієї лабораторної роботи потрібно

- побудувати рекурентне співвідношення (за необхідності, якщо воно явно не задане в умові задачі);
- описати генератор-функцію для генерування членів заданої послідовності;
- описати програму обчислення необхідного члена послідовності використовуючи цикл з лічильником або цикл з умовою.

7.1.

- а) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{x^k}{k} \quad (k \geq 1)$$

- б) Створіть програму для обчислення добутку для заданого значення параметра  $n$

$$P_n = \prod_{i=1}^n \frac{1}{i+1!};$$

- в) Створіть програму для обчислення визначника порядку  $n$

$$\begin{vmatrix} 2 & 3 & 0 & \dots & 0 \\ 1 & 2 & 3 & \dots & 0 \\ 0 & 1 & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 2 \end{vmatrix};$$

- г) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \sum_{k=1}^n 2^k a_k, \quad \text{де } a_1 = 0, a_2 = 1, a_k = a_{k-1} + k a_{k-2}, k \geq 3;$$

- е) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots;$$

### 7.2.

- a) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{(-1)^k x^k}{k} \quad (k \geq 1)$$

- b) Створіть програму для обчислення добутку для заданого значення параметра  $n$

$$P_n = \prod_{i=1}^n \frac{i+1}{i+2};$$

- c) Створіть програму для обчислення визначника порядку  $n$  для заданого значення параметра  $x$ .

$$\begin{vmatrix} 1+x^2 & x & 0 & 0 & \dots & 0 & 0 \\ x & 1+x^2 & x & 0 & \dots & 0 & 0 \\ 0 & x & 1+x^2 & x & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & x & 1+x^2 \end{vmatrix}.$$

- d) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \sum_{k=1}^n \frac{3^k}{a_k}, \quad \text{де } a_1 = 1, a_2 = 1, a_k = \frac{a_{k-1}}{k} + a_{k-2}, \quad k \geq 3;$$

- e) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = \operatorname{ch} x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots;$$

### 7.3.

- a) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{x^k}{(k+1)! k!} \quad (k \geq 0)$$

- b) Створіть програму для обчислення добутку для заданого значення параметра  $n$

$$P_n = \prod_{i=1}^n \left(2 + \frac{1}{i!}\right)$$

- с) Створіть програму для обчислення визначника порядку  $n$  для заданого значення параметра  $a$

$$\begin{vmatrix} a & 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & a & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & a & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 & a \end{vmatrix};$$

- д) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \sum_{k=1}^n \frac{k!}{a_k}, \quad \text{де } a_1 = 1, a_2 = 1, a_k = a_{k-1} + \frac{a_{k-2}}{2^k}, \quad k \geq 3;$$

- е) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots; \quad (|x| < 1);$$

#### 7.4.

- а) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{(-1)^k x^k}{(k^2 + k)!}, \quad k \geq 0;$$

- б) Створіть програму для обчислення добутку для заданого значення параметра  $n$

$$P_n = \prod_{i=2}^n \left(1 - \frac{1}{i^2}\right) = \left(1 - \frac{1}{2^2}\right) \left(1 - \frac{1}{3^2}\right) \dots \left(1 - \frac{1}{n^2}\right);$$

- с) Створіть програму для обчислення визначника порядку  $n$

$$\begin{vmatrix} 1 & 2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 3 & 4 & 3 & 0 & 0 & \dots & 0 & 0 \\ 0 & 2 & 5 & 3 & 0 & \dots & 0 & 0 \\ 0 & 0 & 2 & 5 & 3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 5 & 3 \\ 0 & 0 & 0 & 0 & 0 & \dots & 2 & 5 \end{vmatrix};$$

- д) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \sum_{k=1}^n k! a_k, \quad \text{де } a_1 = 0, a_2 = 1, a_k = a_{k-1} + \frac{a_{k-2}}{(k-1)!}, \quad k \geq 3;$$

- е) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = \frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots; (|x| < 1);$$

### 7.5.

- а) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{x^{2k}}{(2k)!} \quad (k \geq 0)$$

- б) Створіть програму для обчислення добутку для заданого значення параметра  $n$

$$P_n = \left(1 + \frac{1}{1^2}\right) \left(1 + \frac{1}{2^2}\right) \dots \left(1 + \frac{1}{n^2}\right)$$

- в) Створіть програму для обчислення визначника порядку  $n$  для заданих значень параметрів  $a$  та  $b$

$$\begin{vmatrix} a+b & ab & 0 & 0 & \dots & 0 \\ 1 & a+b & ab & 0 & \dots & 0 \\ 0 & 1 & a+b & ab & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & a+b \end{vmatrix};$$

- г) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \sum_{k=1}^n \frac{a_k}{2^k}, \quad \text{де } a_1 = a_2 = a_3 = 1, \quad a_k = a_{k-1} + a_{k-3}, \quad k \geq 4;$$

- е) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = \ln \frac{1+x}{1-x} = 2 \cdot \left[ \frac{x}{1} + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right]; (|x| < 1);$$

### 7.6.

- а) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{(-1)^k x^{2k+1}}{(2k+1)!} \quad (k \geq 0)$$

- б) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \frac{1}{1!!} + \frac{2}{3!!} + \frac{4}{5!!} + \dots + \frac{2^n}{(2n+1)!!};$$

- в) Створіть програму для обчислення визначника порядку  $n$



$$\begin{vmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{vmatrix};$$

- d) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$\sum_{k=1}^n 2^k a_k, \quad \text{де } a_1 = 1, \quad a_k = k a_{k-1} + \frac{1}{k}, \quad k \geq 2.$$

- e) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = \frac{1}{(1+x)^3} = 1 - \frac{2 \cdot 3}{2} \cdot x + \frac{3 \cdot 4}{2} \cdot x^2 - \frac{4 \cdot 5}{2} \cdot x^3 + \dots; \quad (|x| < 1);$$

### 7.7.

- a) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{x^{2k+1}}{(2k+1)!} \quad (k \geq 0)$$

- b) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \frac{1}{2} - \frac{2}{3} + \frac{3}{4} - \dots + \frac{(-1)^n (n-1)}{n};$$

- c) Створіть програму для обчислення визначника порядку  $n$

$$\begin{vmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 \end{vmatrix};$$

- d) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \sum_{k=1}^n \frac{2^k}{a_k + b_k},$$

$$\text{де } \begin{cases} a_1 = 0, a_2 = 1, \\ a_k = \frac{a_{k-1}}{k} + a_{k-2} b_k \end{cases} \quad \begin{cases} b_1 = 1, b_2 = 1, \\ b_k = b_{k-1} + a_{k-1} \end{cases} \quad k \geq 3;$$

- e) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = \sqrt{1+x} = 1 + \frac{1}{2} \cdot x - \frac{1}{2 \cdot 4} \cdot x^2 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 6} \cdot x^3 - \dots, \quad (|x| < 1);$$

**7.8.**

- a) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{x^{2k+1}}{(2k-1)!(2k+1)!} \quad (k \geq 0)$$

- b) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = 1 - \frac{2}{1!} + \frac{4}{2!} + \dots + \frac{(-2)^n}{n!};$$

- c) Створіть програму для обчислення визначника порядку  $n$

$$\begin{vmatrix} 7 & 5 & 0 & \dots & 0 \\ 2 & 7 & 5 & \dots & 0 \\ 0 & 2 & 7 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 7 \end{vmatrix};$$

- d) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \sum_{k=1}^n \frac{a_k b_k}{(k+1)!}$$

$$\text{де } \begin{cases} a_1 = u, \\ a_k = 2b_{k-1} + a_{k-1} \end{cases} \quad \begin{cases} b_1 = v, \\ b_k = 2a_{k-1}^2 + b_{k-1} \end{cases} \quad k \geq 2;$$

- e) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = \frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^3 + \dots; \quad (|x| < 1);$$

**7.9.**

- a) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{(-1)^k x^{2k}}{(2k)!} \quad (k \geq 0)$$

- b) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \dots + \frac{1}{(n-1) \cdot n}$$

- c) Створіть програму для обчислення визначника порядку  $n$

$$\begin{vmatrix} 3 & 2 & 0 & \dots & 0 \\ 1 & 3 & 2 & \dots & 0 \\ 0 & 1 & 3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 3 \end{vmatrix};$$

- d) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$\sum_{k=1}^n \frac{2^k}{(1+a_k^2+b_k^2)k!},$$

$$\text{де } \begin{cases} a_1 = 1, \\ a_k = 3b_{k-1} + 2a_{k-1} \end{cases} \quad \begin{cases} b_1 = 1, \\ b_k = 2a_{k-1} + b_{k-1} \end{cases} \quad d, \quad k \geq 2;$$

- e) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = shx = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots;$$

### 7.10.

- a) Створіть програму для обчислення елементів послідовності для заданих  $x$  та  $k$

$$x_k = \frac{(-1)^k x^{2k+1}}{(2k-1)!(2k+1)!} \quad (k \geq 0)$$

- b) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$S_n = 1 - \frac{1}{2} + \frac{1}{3} - \dots + (-1)^{n-1} \frac{1}{n}$$

- c) Створіть програму для обчислення визначника порядку  $n$

$$\begin{vmatrix} 2 & 1 & 0 & \dots & 0 \\ 1 & 2 & 1 & \dots & 0 \\ 0 & 1 & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 2 \end{vmatrix};$$

- d) Створіть програму для обчислення суми для заданого значення параметра  $n$

$$\sum_{k=1}^n \left( \frac{a_k}{b_k} \right)^k$$

$$\text{де } \begin{cases} a_0 = 1, a_1 = 2, \\ a_k = a_{k-2} + \frac{b_k}{2} \end{cases} \quad \begin{cases} b_0 = 5, b_1 = 5, \\ b^k = b_{k-2}^2 - a_{k-1} \end{cases} \quad k \geq 2;$$

- е) За допомогою розкладу функції в ряд Тейлора обчисліть з точністю  $\varepsilon > 0$  її значення для заданого значення  $x$ . Порівняйте отримані результати із значеннями відповідних функцій з бібліотеки `math`

$$y = \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots;$$

## ЛАБОРАТОРНА РОБОТА 8.

### Створення власних класів виключень.

#### Контрольні запитання

- 8.1.** Наведіть алгоритм опису власного класу виключення.
- 8.2.** Чи можна використовувати як батьківський клас для створення власного класу виключення клас `BaseException`?
- 8.3.** Чи відрізняється використання власного класу виключення у програмі від використання вбудованих класів виключень?
- 8.4.** У чому полягає важливість оператора `raise` при створенні власних класів виключень.

#### Завдання для аудиторної роботи

**8.1.** Опишіть клас `ProtectedDictInt`, (див задачу 5.1) що є словником у якому, ключами можуть бути лише цілі числа та заборонена операція зміни значення за ключем (крім випадку додавання нової пари ключ-значення). Для обробки заборонених ситуацій створіть клас виключення `ProtectedDictIntError`, що

- є нащадком класу `KeyError`;
- ініціюється, якщо здійснюється спроба використання ключа, що не є цілим числом;
- ініціюється якщо відбувається спроба змінити у словнику значення за заданим ключем;
- генерує різні повідомлення у кожній із заборонених ситуацій.

#### Варіанти індивідуальних завдань

**8.1.** Опишіть клас виключення `RationalError`, що є нащадком класу `ZeroDivisionError` та ініціюється у випадку якщо при створенні екземпляру класу `Rational` описаного у задачі 5.1 знаменник дорівнює нулю.

**8.2.** Опишіть клас `SegmentError`, який ініціюється у випадку, якщо задається об'єкт класу `Segment` з задачі 5.2, такий, що початок відрізка, більший за його кінець.

**8.3.** Будемо вважати, що у задачі 5.3 елементами множини `CustomSet` можуть бути лише об'єкти одного типу. При цьому тип об'єктів визначається типом першого доданого до множини об'єкта. Тип цього об'єкта будемо вважати типом екземпляру множини `CustomSet`. Опишіть клас `CustomSetError`, що:

- ініціюється під час спроби додавання до множини об'єкту не відповідного типу;
- ініціюється при застосуванні бінарних операцій перетину, об'єднання або різниці, описаних у задачі 5.3 використовуються різнотипні екземпляри класу `CustomSet`;
- генерує різні повідомлення у кожній зазначений вище ситуації.

**8.4.** Опишіть клас виключення `IndexPositiveError` як нащадок класу `IndexError`. Виключення цього типу можуть ініціюватися при зверненні до символу рядка за індексом, причому не лише у випадку, якщо звернення відбувається за індексом, якого не існує у рядку, але й якщо індекс від'ємний. Модифікуйте клас `MutableString`, що був описаний у задачі 5.4 таким чином, щоб спеціальні методи `__setitem__` та `__getitem__` породжували виключення типу `IndexPositiveError` при виникненні згаданих вище ситуацій. При цьому повідомлення виключення має однозначно описувати ситуацію, що має місце.

**8.5.** Для задачі 5.5 реалізуйте клас виключення `MatrixOperationError`, який є нащадком класу `ArithmeticError`, ініціюється при застосуванні неприпустимих арифметичними операціями для матриць та векторів (наприклад під час додавання чи множення матриць/векторів різної розмірності). Під час ініціювання виключення, екземпляр виключення має містити детальну інформацію про тип виключної ситуації, що трапилася.

**8.6.** Опишіть клас `DictIntFloat`, що є словником у якому, ключами можуть бути лише цілі (`int`) числа, а значеннями – дійсні (`float` або `int`). Опишіть клас виключення `IndexValueError` для обробки заборонених ситуацій, що

- ініціюється, якщо додається або змінюється неприпустима пара ключ-значення.
- генерує різні повідомлення у кожній із неприпустимих ситуацій.

Реалізуйте клас `Polinom` (див задачі 1.6 та 5.6), що є нащадком класу `DictIntFloat` та моделює роботу з поліномами.

**8.7.** Для використання у задачі 6.7 опишіть клас `CustomSetError`, що ініціюється під час спроби додавання до множини об'єкту не відповідного типу та генерує повідомлення у виключній ситуації.

**8.8.** Для використання у задачі 5.8 опишіть клас виключення `SentenceError`, що

- ініціюється, якщо у реченні змінюється слово не рядковим літералом;
- ініціюється, якщо у операціях `+` або `-` операндом правої частини є екземпляр не припустимого класу;
- генерує різні повідомлення у кожній із неприпустимих ситуацій.

**8.9.** Опишіть клас `DictStrInt`, що є словником у якому, ключами можуть бути лише рядки, а значеннями – цілі числа. Для обробки заборонених ситуацій створіть клас виключення `DictStrIntError`, що

- ініціюється, якщо здійснюється спроба використання ключа, що не є рядком;
- ініціюється, якщо здійснюється спроба використання значення, що не є цілим числом;
- генерує різні повідомлення у кожній із заборонених ситуацій.

**8.10.** Для використання у задачі 5.10 опишіть клас виключення `StringDictError`, що

- ініціюється, якщо здійснюється спроба використання ключа, що не є рядком;
- ініціюється, якщо здійснюється спроба одночасного використання українських та англійських літер у ключі словника;
- ініціюється, якщо здійснюється спроба одночасного використання українських та англійських літер у значенні словника;
- ініціюється, як ключ словника складається з англійських літер, а значення містить літери, що не є входять до українського алфавіту і навпаки.
- генерує різні повідомлення у кожній із заборонених ситуацій.

## **ЛАБОРАТОРНА РОБОТА 9.**

### **Створення графічних інтерфейсів.**

#### **Контрольні запитання**

**9.1.**

#### **Завдання для аудиторної роботи**

**9.1.** Напишіть програму для перевірки, чи є заданий рядок паліндромом. Створіть графічний інтерфейс, для введення рядка та виведення результату

**9.2.** Напишіть програму, що виводить усі різні слова заданого рядка. Створіть графічний інтерфейс для введення рядка та виведення результату. Слова виводити у віджет список.

#### **Варіанти індивідуальних завдань**

**9.1.** Створіть графічний інтерфейс для програми, що розв'язує задачу 3.1.

**9.2.** Створіть графічний інтерфейс для програми, що розв'язує задачу 3.2.

**9.3.** Створіть графічний інтерфейс для програми, що розв'язує задачу 3.3.

**9.4.** Створіть графічний інтерфейс для програми, що розв'язує задачу 3.4.

**9.5.** Створіть графічний інтерфейс для програми, що розв'язує задачу 3.5.

**9.6.** Створіть графічний інтерфейс для програми, що розв'язує задачу 3.6.

**9.7.** Створіть графічний інтерфейс для програми, що розв'язує задачу 3.7.

## **ЛАБОРАТОРНА РОБОТА 10.**

### **Створення анімаційних програм.**

#### **Варіанти індивідуальних завдань**

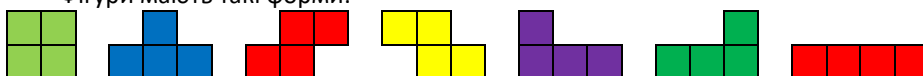
**10.1.** Скласти програму з графічним інтерфейсом для показу піщаного годинника. Показати корпус годинника та область, заповнену піском. Показати анімацію переміщення піску з верхньої до нижньої частини годинника. Дати можливість встановити час вичерпання піску а також перегорнути годинник після вичерпання часу для нового обліку.

**10.2.** Скласти програму з графічним інтерфейсом для гри у «Тетрис». Гра полягає у наступному: є «склянка» розміром 20x10, у яку згори з центру падають з постійною швидкістю випадкові фігури різної форми з 4 клітинок.

Гравець керує грою за допомогою клавіш-стрілок. Стрілки вліво та вправо відповідно переміщують фігуру. Стрілка вгору повертає фігуру на 90 градусів. Стрілка вниз «скидає» фігуру до низу.

Якщо деякий ряд поля повністю заповнений клітинками фігур, такий ряд знімається та гравцю нараховуються бали. Гра закінчується, коли чергова фігура відразу після появи не може рухатись до низу. Задача – набрати максимальну кількість балів до закінчення гри.

Фігури мають такі форми:



**10.3.** Скласти програму з графічним інтерфейсом для реалізації гри у «Морський бій». Гра полягає у наступному: гравець та комп'ютер на полі 10x10 розміщують «кораблі». Кожен корабель – це одна, дві, три або чотири сусідніх клітинки по горизонталі або вертикалі. На полі розміщують 4 кораблі по 1 клітинці, 3 – по 2 клітинки, 2 – по 3 клітинки та 1 – по 4 клітинки. Два кораблі не можуть бути розміщені поруч по горизонталі, вертикалі та діагоналі.

Гравець та комп'ютер по черзі роблять свої ходи. Кожний хід – це вказання клітинки, наприклад A1. Якщо у вказаній клітинці є корабель, то корабель вважається «пораним» (або «вбитим», якщо поранено усі клітинки) та гравець (або комп'ютер) отримує бонусний хід. Виграє той, хто першим уразив усі кораблі опонента.

Графічний інтерфейс повинен показувати ігрове поле гравця та комп'ютера, надавати можливість гравцю розміщувати кораблі на полі з дотриманням правил а також показувати хід гри (зроблені постріли та підбиті кораблі).



	А	Б	В	Г	Д	Е	Ж	З	И	К		А	Б	В	Г	Д	Е	Ж	З	И	К	
1	×	×	•			•	×	×	×	×		1										
2	•	•	•				•	•	•	•		2										
3			•									3		•								
4												4		•	•	•	•	•	•	•	•	•
5												5		•	×	×	•	×	×	×	×	×
6												6		•	•	•	•	•	•	•	•	•
7												7										
8												8							•			
9												9										
10												10										

**10.4.** Скласти програму з графічним інтерфейсом для реалізації простого графічного редактора. Надати можливість зображувати лінії, прямокутники, еліпси заданого кольору та з заданою товщиною та кольором границі.

Зберігати параметри зображення у файлі та відновлювати з файлу.

**10.5.** Скласти програму з графічним інтерфейсом для реалізації розставлення меблів у кімнаті. Задавати розміри кімнати. Кожен елемент меблів зображувати прямокутником заданих розмірів та назвою. Меблі можна пересувати та повертати на заданий кут. Меблі не повинні перетинатися під час переміщення. Зберігати параметри зображення кімнати та меблів у файлі та відновлювати з файлу.

**10.6.** Скласти програму з графічним інтерфейсом для реалізації гри «Посадка на Місяць». Зобразити ламаною лінією рельєф поверхні Місяця. На поверхні повинно бути 2 майданчика для посадки (рівна поверхня шириною не менше ширини корабля). Корабель починає рух до поверхні із заданою вертикальною швидкістю. Швидкість та позиція корабля керуються стрілками вниз (увімкнення двигуна та зменшення швидкості) а також праворуч та ліворуч. Вертикальна швидкість при посадці не повинна перевищувати заданої границі. Зобразити корабель та полум'я двигуна при увімкненні у вигляді зображень (gif).

**10.7.** Скласти програму з графічним інтерфейсом для реалізації гри «Зенітка». Зенітка повинна уражати літаки, що рухаються із постійною швидкістю з правого до лівого краю полотна. Час виникнення наступного літака – випадковий. Зенітка може стріляти снарядами з визначеною швидкострільністю та з визначеною швидкістю вильоту снарядів. Постріл здійснюється натисненням клавіші «пропуск». Є також можливість

встановити кут нахилу зенітки. Літак уражається, якщо зустрічається зі снарядом. За уражений літак гравцю додаються бали. Завдання – уразити якомога більше літаків.

**10.8.** Скласти програму з графічним інтерфейсом для реалізації гри тренування пам'яті.



На екрані знаходиться таблиця з прихованими зображеннями кількох типів. Зображень кожного типу є рівно два і розташовані вони у довільних позиціях. Гравець по черзі відкриває будь-яких два зображення – якщо зображення однакові, то вони залишаються відкритими і перестають бути активними. Якщо відкриті зображення різні, то вони повертаються у вихідне положення. Гра відбувається до тих пір, поки не будуть відкриті всі зображення.

**10.9.** Скласти програму з графічним інтерфейсом для гри у «Шашки» між двома гравцями.

**10.10.** Скласти програму з графічним інтерфейсом для реалізації гри «Танки». На полі знаходяться два ворожих танки. Кожен з яких по черзі робить постріл по противнику. Виграє той гравець, що першим вражає танк противника. Для кожного танку можна керувати силою пострілу та кутом гармати танка.

## **ЛАБОРАТОРНА РОБОТА 11.**

### **Абстрактні класи, інтерфейси, домішки**

#### **Контрольні запитання**

**11.1.** Який клас називається абстрактним?

**11.2.** Яке основне призначення абстрактних класів?

**11.3.** Як оголосити абстрактний клас у Python?

**11.4.** Що таке інтерфейс? Який зв'язок між інтерфейсом та абстрактним класом?

**11.5.** Що таке клас-домішок? Для чого їх використовують?

### **Варіанти індивідуальних завдань**

**11.1.**

## **ЛАБОРАТОРНА РОБОТА 12.**

### **Метапрограмування та рефлексія.**

#### **Контрольні запитання**

**12.1.**

#### **Варіанти індивідуальних завдань**

**12.1.**

## **Список літератури та додаткових джерел**

1.

