

13. Класи та об'єкти

T13.1. Клас Person описано наступним чином:

```
class Person:                                #Клас Особа
    def __init__(self):
        self.name = None                    #прізвище
        self.byear = None                   #рік народження

    def input(self):                         #ввести особу
        self.name = input('Прізвище: ')
        self.byear = input('Рік народження: ')

    def print(self):                         #вивести особу
        print(self.name, self.byear, end = ' ')
```

Описати клас Знайомий на базі класу Person.

У цьому класі повинно бути як мінімум одне додаткове поле «номер телефону» а також методи введення та виведення інформації про знайомого.

Використати цей клас для побудови телефонного довідника.

Передбачити дії: створення довідника, додавання запису про знайомого, пошуку номера телефону за прізвищем та заміни номера телефону.

Телефонний довідник зберігати у файлі та читати з файлу.

Вказівка: телефонний довідник представити у вигляді списку знайомих.

T13.2. Описати клас Співробітник на базі класу Person (див. завдання T13.1).

У цьому класі повинно бути як мінімум два додаткових поля: «табельний номер» та «заробітна платня», а також методи введення та виведення інформації про співробітника та розрахунку заробітної платні за кількістю відпрацьованих годин та загальною кількістю годин на місяць.

Використати цей клас для розрахунку заробітної платні за місяць, якщо співробітник працює на умовах погодинної оплати. Для розрахунку вводити відпрацьовані години по днях місяця (табель). Вважається, що відома кількість годин, яку повинен відпрацювати співробітник кожного дня, для отримання 100% платні.

Передбачити дії: введення таблицю для співробітника, введення таблиців всіх співробітників, розрахунку заробітної платні всіх співробітників.

Табель зберігати у файлі та читати з файлу.

Вказівка: таблиць представити у вигляді словника співробітників з ключем «табельний номер».

T13.3. Описати клас Гість на базі класу Person (див. завдання T13.1) та клас ТипНомера з полями «назва типу», «плата за добу».

У класі Гість повинно бути як мінімум два додаткових поля: «номер» та «кількість днів проживання», а також методи введення та виведення інформації про гостя та розрахунку плати за проживання.

Використати цей клас для реєстрації гостя у номері та розрахунку плати за проживання усіх гостей.

Номери представити у вигляді списку кортежів (номер, тип номера).

Номери та інформацію про гостей зберігати у файлах та читати з файлів.

Вказівка: інформацію про гостей представити у вигляді списку.

T13.4. Описати клас Пасажир на базі класу Person (див. завдання T13.1).

У класі Пасажир повинно бути як мінімум два додаткових поля: «місто відправлення» та «місто прибуття», а також методи введення та виведення інформації про пасажера та розрахунку плати за білет.

Використати цей клас для розрахунку плати за білети усіх пасажирів. Вважати що маршрути зберігаються у списку кортежів (місто1, місто2, відстань), а також те, що плата за білет пропорційна відстані та відома плата за 1 км відстані.

Вказівка: інформацію про пасажирів представити у вигляді списку.

T13.5. Описати клас Водій на базі класу Person (див. завдання T13.1).

У класі Водій повинно бути як мінімум два додаткових поля: «плата за тоно-кілометр» та «вантажопідйомність автомобіля», а також методи введення та виведення інформації про водія та розрахунку плати за перевезення вантажу на задану відстань.

Маршрутний лист водія за певну дату містить кількість кілометрів перевезень вантажу.

Використати цей клас для розрахунку плати всіх водіїв за період. Вважати що маршрутні листи водіїв зберігаються у списку кортежів (водій, дата, відстань), а також те, що плата за перевезення пропорційна добутку відстані на вагу вантажу.

T13.6. Описати клас Поліном та реалізувати методи: введення поліному, виведення поліному, обчислення значення поліному у точці x , взяття похідної поліному, суми, різниці та добутку поліномів.

Використати цей клас для розв'язання задачі: ввести 2 поліноми P_1 , P_2 та рядок, який містить вираз, що залежить від 2 поліномів. Наприклад, $P_1 + P_2 * P_1 - P_2$

Обчислити поліном, який буде значенням цього виразу.

Вказівка: поліном представити у вигляді словника.

T13.7. Описати клас для роботи з відрізками на числовій осі. Для відрізка передбачити поля:

$(a, b, empty)$

де a, b - границі відрізка, $empty$ - ознака того, що відрізок порожній.

Реалізувати методи:

- 1) зробити відрізок t порожнім;
- 2) чи порожній відрізок t ;
- 3) покласти відрізок t рівним a, b ;
- 4) покласти відрізок t рівним перетину відрізків $t1, t2$.

З використанням класу скласти програму розв'язку системи квадратних нерівностей вигляду $x^2+px+q_i<0$. Пари коефіцієнтів нерівностей p_i, q_i вводяться з пристрою введення.

T13.8. Описати клас для реалізації мультимножини цілих чисел на базі словника. Мультимножина - це множина в якій для кожного елемента запам'ятовується не лише його входження, але й кількість входжень.

Кількість входжень елемента k ($0 \leq k \leq n$) у мультимножину - це значення елемента словника з ключем k .

Реалізувати методи:

- 1) зробити мультимножину порожньою;
- 2) чи є мультимножина порожньою;
- 3) додати елемент до мультимножини;
- 4) забрати елемент з мультимножини (кількість входжень елемента зменшується на 1, якщо елемент не входить - відмова);
- 5) кількість входжень елемента у мультимножину;
- 6) об'єднання двох мультимножин (в результаті об'єднання кількість входжень елемента визначається як максимальна з двох мультимножин);
- 7) перетин двох мультимножин (в результаті кількість входжень елемента визначається як мінімальна з двох мультимножин);

З використанням класу розв'язати задачі:

- а) знайти символ, який входить у рядок S максимальну кількість разів (див. завдання 7.111);
- б) перевірити, чи складаються рядки $S1, S2$ з одних і тих же символів, які входять у ці рядки однаково кількість разів;
- в) перевірити, чи вірно, що всі символи рядка $S1$, входять також у рядок $S2$, причому не меншу кількість разів, ніж у $S1$.

T13.9. Описати класи для роботи з точками та відрізками на площині. Для точки та відрізка передбачити такі поля:

Точка – (x, y) ,

Відрізок - (a, b) ,

де a, b - точки

Реалізувати методи:

- 1) взяти точку t ;
- 2) покласти точку t рівною (x, y) ;
- 3) показати точку t .

Реалізувати методи:

- 1) взяти відрізок s ;
- 2) показати відрізок s ;
- 3) покласти відрізок s рівним a, b ;
- 4) довжина відрізка s ;
- 5) чи лежить точка t на одній прямій з відрізком s ;
- 6) чи лежить точка t всередині відрізка s ;
- 7) площа трикутника, утвореного точкою t та відрізком s .

У файлі записано послідовність точок. З використанням класів роботи з точками та відрізками на площині знайти:

- а) трикутник з найбільшою площею, утворений точками послідовності;
- б) коло найменшого радіуса, всередині якого лежать всі точки послідовності;
- в) відрізок, на якому лежить найбільша кількість точок послідовності;
- г) коло, на якому лежить найбільша кількість точок послідовності.

T13.10. Клас Point (точка екрану) описано наступним чином:

```
import turtle

class Point:
    '''Точка екрану'''
    ...
    _count = 0

    def __init__(self, x, y):
        self._x = x           # _x - координата x точки
        self._y = y           # _y - координата y точки
        self._visible = False # _visible - чи є точка видимою на екрані
        Point._count += 1

    def getx(self):
        '''Повертає координату x точки'''
        ...
        return self._x

    def gety(self):
        '''Повертає координату y точки'''
        ...
        return self._y

    def onscreen(self):
        '''Перевіряє, чи є точка видимою на екрані'''
        ...
        return self._visible

    def switchon(self):
        '''Робить точку видимою на екрані'''
        ...
        if not self._visible:
            self._visible = True
            turtle.up()
            turtle.setpos(self._x, self._y)
            turtle.down()
            turtle.dot()
```

```

def switchoff(self):
    '''Робить точку невидимою на екрані'''
    if self._visible:
        self._visible = False
        turtle.up()
        turtle.setpos(self._x, self._y)
        turtle.down()
        turtle.dot(turtle.bgcolor())

def move(self, dx, dy):
    '''Пересуває точку на екрані на dx, dy позицій'''
    vis = self._visible
    if vis:
        self.switchoff()
    self._x += dx
    self._y += dy
    if vis:
        self.switchon()

def printcount():
    print('Кількість точок:', Point._count)

printcount = staticmethod(printcount)

```

Описати клас Прямокутник на базі класу Point. Сторони прямокутника паралельні осям координат. Для прямокутника задані лівий верхній кут та довжини сторін. Описати методи отримання довжини кожної з сторін, стерти прямокутник з екрану та зробити прямокутник видимим на екрані. Скласти програму створення заданої кількості прямокутників та переміщення їх по екрану.

Вказівка: використати графічну бібліотеку turtle.

T13.11. Клас Point (точка екрану) описано наступним чином (див. T13.10.) Описати клас Трикутник на базі класу Point. Основа трикутника паралельна осі x координат. Для трикутника задані лівий нижній кут (координати) та довжини сторін. Описати методи отримання довжини кожної з сторін, стерти трикутник з екрану та зробити трикутник видимим на екрані. Скласти програму створення заданої кількості трикутників та переміщення їх по екрану.

Вказівка: використати графічну бібліотеку turtle.