

Лабораторна робота №5

Тема: Об'єктно-орієнтоване програмування у мові Python

Мета: набути навичок роботи з класами у мові Python

Теоретична частина

Клас – це складний користувацький тип даних, що складається з полів та методів. *Поля класу* – це змінні, оголошені всередині класу для збереження даних. *Методи класу* – це функції, оголошені для обробки полів класу, взаємодії з основною програмою та іншими даними.

Об'єкти – це окремі екземпляри класу, по суті змінні цього типу. Оголосивши клас із полями та методами, ви можете створювати скільки завгодно об'єктів, кожен з яких міститиме оголошений в класі набір полів та методів.

Класи оголошуються з ключовим словом **class**, ім'я класу за стандартами Python записується наступним чином: всі слова разом, кожне слово з великої літери.

Для створення екземпляру клас викликається як функція з круглими дужками, повертаючи новий об'єкт, який ви можете зберегти у змінній.

Для методів класу можна вказувати атрибути.

Атрибут – це змінна, метод – це функція. Відмінності методу від функції в тому, що у нього є перший параметр – **self**.

Класи, як і модулі, приховують внутрішню будову, залишаючи на поверхні лише зовнішній "інтерфейс" для використання.

Це поєднання даних та функцій всередині однієї сутності, разом із прихованням внутрішньої будови, називається *інкапсуляцією* і є головним принципом ООП.

При оголошенні класу в дужках можуть бути записані (одне або декілька) імена вже існуючих класів – це називається наслідуванням. В такому випадку новий клас (який називається дочірнім) успадковує всі поля та методи класів перелічених в дужках (які називаються батьківськими).

Модель класу:

```
class ім'я_класу:  
    інструкція 1  
    ....  
    інструкція N
```

Створення об'єкта класу:

```
об'єкт_класу = ім'я_класу()
```

Приклад 1. Програма з використанням класу. Обчислення середнього бала студента з трьох предметів.

```
import math  
  
class Student():
```

```

def GPA(self, name, e1, e2, e3) :
    self.name = name
    self.e1 = e1
    self.e2 = e2
    self.e3 = e3
    print(self.name, ' - ', ((self.e1 + self.e2 + self.e3)/3))

s1 = Student()
s2 = Student()
s1.GPA('Dmytro', 5, 3, 4)
s2.GPA('Olena', 5, 4, 5)

def main():

    return 0

if __name__ == '__main__':
    main()

```

Класи збирають в собі набори даних (змінних) разом з наборами функцій, що на них діють. Мета полягає в тому, щоб досягти більш модульного коду за допомогою групування змінних і функцій, в невеликі вузли, що легко модифікувати.

Завдання:

1. Розробити клас "домашня бібліотека". Реалізувати можливість роботи з довільним числом книг, пошуку по книгах за декількома параметрами (за автором, за роком видання, за жанром тощо), додавання книг у бібліотеку, видалення книг з неї, доступу до книги за номером. Написати програму, що буде демонструвати всі розроблені елементи класу.

2. Розробити клас для представлення відомостей про успішність студента. Об'єкт класу має містити поля для збереження імені студента та балів, отриманих ним за виконання лабораторних робіт та індивідуального творчого завдання.

Забезпечити наступні методи класу:

- конструктор, який приймає рядок ім'я_студента та словник, що містить налаштування курсу у наступному форматі:

- 1) максимально можлива кількість балів за здачу індивідуального творчого завдання;
- 2) максимально можлива кількість балів за здачу однієї лабораторної роботи;
- 3) кількість лабораторних робіт в курсі;
- 4) частка балів від максимуму, яку необхідно набрати для отримання екзамену автоматом.

- метод, за допомогою якого вносяться дані про кількість спроб здати лабораторну роботу та оцінка за останню спробу.
- метод, за допомогою якого вносяться дані про кількість спроб здати індивідуальне творче завдання та оцінка за останню спробу.
- метод, який повертає кортеж (tuple), що містить дійсне число (суму балів студента за проходження курсу), та логічне значення True або False в залежності від того, чи достатньо цих балів для отримання оцінки за екзамен автоматом.

3. Розробити клас, який наслідує функціональність стандартного типу str і містить 2 нових методи:

1) метод, який приймає 1 аргумент s та повертає True або False в залежності від того, чи містить рядок повтори послідовностей символів довжиною від 3 символів.

2) метод, який повертає True або False в залежності від того, чи є рядок паліндромом. Регістрами символів нехтувати. Порожній рядок вважати паліндромом.

4. Розробити клас "колода карт", що буде включати закритий масив елементів класу "карта". В карті буде зберігатися масть та номер. При створенні екземпляру класу "колода карт", карти у колоді розташовуються випадковим чином. Забезпечити можливість виведення карти за номером розташування у колоді, виведення всіх карт, перемішування колоди, видачі однієї карти з колоди, видачі 6 карт з колоди. Написати програму, що буде демонструвати всі розроблені елементи класу.

5. Розробити клас "англо-російський словник", забезпечити можливість зберігання декількох варіантів перекладу для кожного слова. Забезпечити можливість виведення всіх варіантів перекладу введеного англійського слова.

6. Створити абстрактний клас "Транспортний засіб". На його основі реалізувати класи "Літак", "Автомобіль" та "Корабель". Класи повинні мати можливість задавати та отримувати координати і параметри засобів пересування (вартість, швидкість, рік випуску тощо) задати за допомогою полів. Для літака повинна бути визначена висота, для літака та корабля – кількість пасажирів, для корабля – порт приписки. Динамічні характеристики задати за допомогою методів.

Контрольні питання:

1. Як створити клас у мові Python?
2. Що таке інкапсуляція?
3. Що таке об'єкт?
4. Що таке атрибут?
5. У чому полягає відмінність методу від функції?