

Лабораторна робота №11

Тема: Основи об'єктно-орієнтованого програмування (ООП).

Мета роботи: ознайомитися з основними поняттями ООП, навчитися розрізняти класи, об'єкти та їх властивості.

1. Теоретичні відомості

Об'єктно орієнтоване програмування (ООП) – це парадигма програмування, де різні компоненти комп'ютерної програми моделюються на основі реальних об'єктів.

Об'єкт – це одиниця в програмуванні, що має якісь характеристики і те, що може виконати будь-яку функцію.

Клас в об'єктно-орієнтованом програмуванні виступає в ролі креслення для об'єкта. Клас має опис властивостей, методів та функцій об'єкта, але сам їм не являється.

Об'єкт та клас – це реально існуючі одиниці в коді, а не лише поняття. Об'єкти народжуються від своїх класів. В мові програмування Python такі об'єкти зазвичай називають екземплярами.

ООП має 3 основних принципи: спадкування, поліморфізм та інкапсуляція.

Спадкування – це умовне поняття, яке передбачає собою передавання властивостей від батьків до дочірніх елементів. У данному випадку – від класу до екземпляру або від старшого класу до молодшого. Властивість, яку має батьківський клас, буде мати і дочірній, і їх екземпляри.

Поліморфізм – це безліч форм. Тобто, екземпляри різних класів з різною внутрішньою складовою можуть мати однакові інтерфейси. Наприклад, інтерфейс “+” має як і клас чисел, так і клас рядків, але, якщо підклас int з підкласом float буде складувати числа, то клас str конкатенує свої об'єкти (рядки).

Інкапсуляція – це приховування даних, тобто відсутність можливості отримати їх напрямку. Для отримання існують інтерфейси. У Python інкапсуляція відсутня, але, при необхідності, її можна імітувати.

В мові програмування Python класи створюються дуже легко: пишемо class, його назву та вписуємо його властивості, які будуть передаватися об'єктам.

```
class A():  
    властивості
```

В мові програмування Python класи створюються дуже легко: пишемо class, його назву та вписуємо його властивості, які будуть передаватися об'єктам.

```
class A():  
    властивості
```

Для додавання функції класу пишемо функцію у тілі класу. При написанні функції обов'язковим першим параметром є змінна, яка позначає об'єкт.

```
class A():  
    m = 20  
    l = 10  
    def dodavannya(self):  
        return self.m+3
```

2. Хід роботи

- 1) Виконати завдання на мові Python згідно з варіантом. Написати програму та перевірити її виконання.
- 2) Скласти звіт, який має мати:
 - найменування та мету роботи
 - завдання за варіантом
 - код програми
 - результат виконання програми
 - стислі відповіді на контрольні питання

3. Завдання

Намалювати графічну схему (на Ваш погляд) за умовою, згідно з варіантом, пояснити її (як позначені елементи) та повторити її у вигляді коду на мові Python. Виконати будь-які дії між об'єктами (*примітка*: для цього потрібно реалізувати методи).

Варіант 1

Створено чотири об'єкти від двох класів. Від першого класу успадковано 2 властивості (поля), від другого – 1. Об'єкти першого класу мають 1 інтерфейс у бік об'єктів другого класу.

Варіант 2

Створено шість об'єктів від двох класів. Усі об'єкти мають інтерфейс між собою у обидві сторони. Класи повністю описують свої об'єкти, окрім однієї властивості (поля).

Варіант 3

Створено об'єкти від трьох класів. Класи повністю описують свої об'єкти, окрім однієї властивості (поля). Об'єкти від двох класів мають інтерфейси до об'єктів третього класу.

Варіант 4

Створено три об'єкти від одного класу. Вони мають 4 властивості (поля), 3 з яких переходить від класу. У одному з об'єктів змінена одна з властивостей класу.

Варіант 5

Створено три об'єкта, кожен від свого класу. Об'єкти повністю описуються від свого класу та мають інтерфейси з усіма іншими об'єктами.

4. Контрольні питання

- 1) Назвати основні елементи ООП.
- 2) Перелічити основні принципи ООП та кратко їх описати.
- 3) Чи може об'єкт належати декільком класам одночасно? Чи може клас мати кілька об'єктів одночасно?