Спадкування, поліморфізм та інкапсуляція

Спадкування

Спадкування – це передача полів та функцій від батьківських класів дочірнім і, відповідно, їх об'єктам.

Просте спадкування – це передача методів (функцій) батьківського класу до одного підкласу. *Множинне спадкування* відбувається так само, але для кількох підкласів.

```
class Table:
    def __init__(self, l, w, h):
        self.length = l
        self.width = w
        self.height = h


class DeskTable(Table):
    def square(self):
        return self.width * self.length


t1 = Table(1.5, 1.8, 0.75)
t2 = DeskTable(0.8, 0.6, 0.7)
```

```
class Table:
    def __init__(self, l, w, h):
        self.length = l
        self.width = w
        self.height = h


class DeskTable(Table):
    def square(self):
        return self.width * self.length


class ComputerTable(DeskTable):
    def square(self, monitor=0.0):
        return self.width * self.length - monitor
```

При простому та множинному спадкуванні до дочірнього класу переходять як і атрибути, так і функції батьківського. Окрім устаткованих, дочірній клас може мати свої параметри, які працюють у цьому класі та в його дочірних.

Також один клас може мати декілька батьківських. У такому разі підклас буде устатковувати параметри від усіх батьків.

Поліморфізм

Поліморфізм проявляється в перегрузці методів або його перевизначення. Перегрузка метода (функції) – це властивість метода вести себе по-різному в залежності від кількості або типа параметрів.

```
class Car:
  def start(self, a, b=None):
    if b is not None:
      print (a + b)
    else:
      print (a)
```

Перевизначення метода (функції) – це властивість методів з однаковою назвою, що відносяться до батьківського та дочірнього класу. Визначення метода відрізняється в батьківському та дочірньому класах, але назва залишається тією ж. У такому разі у дочірньому класі власний метод перекриває унаслідований від батьків.

```python
class Vehicle:
  def print_details(self):
    print("Це батьківський метод класа класса Vehicle")

class Car(Vehicle):
  def print_details(self):
    print("Це дочірній метод класа Car")

class Cycle(Vehicle):
  def print_details(self):
    print("Це дочірній метод класа Cycle")
```

Інкапсуляція

Інкапсуляція просто означає приховування даних. Як правило, в об'єктно-орієнтованому програмуванні один клас не повинен мати прямого доступу до даних іншого класу. Натомість, доступ повинен контролюватись через методи класу. Для контролю використовуються модифікатори доступу.

Модифікатори доступу – це позначки змінних, які визначають область видимості цієї змінної. Модифікаторів доступу в Python існує 3 вида:

- Публічний (public) – доступ до змінної відкритий з будь-якого місця поза класом.
- Приватний (private) – доступ відкритий тільки у межах класу.
- Захищений (protected) – доступ відкритий тільки у межах одного пакету.

За замовчуванням створюються атрибути з публічним модифікатором доступу. Для створення приватного модифікатора перед назвою змінної ставлять префікс __, для створення захищенного модифікатора – префікс _.

Змінні з однаковою назвою, але з різними модифікаторами є різними змінними.

Доступ до значення змінної з приватним модифікатором доступу можливо отримати тільки через метод класу.

Контрольні питання:

1) Що таке множинне спадкування?
2) Що відбудеться, якщо батьківський і дочірній класи матимуть різні методи, які називаються однаково?
3) Що таке модифікатори доступу? Для чого вони існують? Які вони бувають?