

Python Базовий

Поліморфізм

Python Базовий

Introduction



Вікторія Бойчук

Python Developer, тренер CBS

 [Вікторія Бойчук](#)



Python Базовий

Тема уроку

Поліморфізм

Python Базовий

План уроку

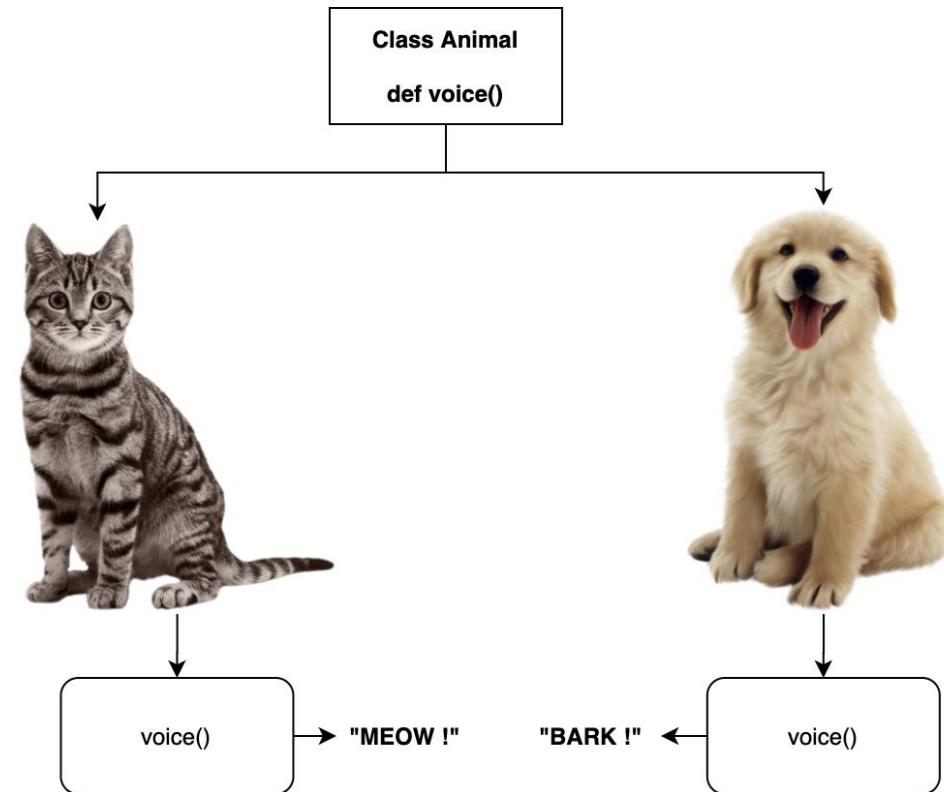
1. Що таке Поліморфізм
2. Як на практиці він застосовується
3. Розв'язання задач

Python Базовий

Поліморфізм (Polymorphism)

Поліморфізм - різна поведінка одного й того методу в різних класах. Поліморфізм досягається шляхом перевантаження чи перевизначення методу.

Таким чином досягається додаткова гнучкість програми. А також, значно скорочується дублювання коду, що дуже важливо для хорошого та підтримуваного проекту.



Python Базовий

Поліморфізм досягається 2 методами

```
1 class Base:
2     def __init__(self, a):
3         self.a = a
4
5     def print_a(self, square=False, multiplier=None):
6         if square and not multiplier:
7             print(self.a ** 2)
8         elif not square and multiplier:
9             print(self.a * multiplier)
10        elif square and multiplier:
11            print((self.a ** 2) * multiplier)
12        else:
13            print(self.a)
14
15
16 base = Base(4)
17 base.print_a(square=True) # 16
18 base.print_a(square=False, multiplier=2) # 8
19 base.print_a(square=True, multiplier=2) # 32
20 base.print_a() # 4
21
22
```

Перевантаження

```
1 class Multiplier:
2     def __init__(self, a):
3         self.a = a
4
5     def print_a(self, x):
6         print(self.a * x)
7
8     class Exponent(Multiplier):
9         def print_a(self, x):
10            print(self.a ** x)
11
12
13 mult = Multiplier(2)
14 mult.print_a(3) # 6
15 expo = Exponent(2)
16 expo.print_a(3) # 8
17
18
```

Перевизначення

Python Базовий

Поліморфізм в Python

```
1 class Car:
2     def __init__(self, name):
3         self.__name_speed_dict = {
4             "Mercedes": 250,
5             "BMW": 300
6         }
7         self._max_speed = self._define_max_speed(name)
8
9     def _define_max_speed(self, name):
10        return self.__name_speed_dict.get(name, 0)
11
12    def distance_time_on_max_speed(self, distance):
13        if not self._max_speed:
14            print("No speed param specified.")
15            return 0
16        return distance / self._max_speed
17
18 car_a = Car(name="BMW")
19 car_b = Car(name="Mercedes")
20 print(car_a.distance_time_on_max_speed(distance=167))
21 print(car_b.distance_time_on_max_speed(distance=167))
22
23
```

```
1 class Animal:
2     def __init__(self, name):
3         self.name = name
4
5     def voice(self):
6         if self.name == "dog":
7             print("Bark !")
8         elif self.name == "cat":
9             print("Meow !")
10        else:
11            print("...")
12
13 cat = Animal(name="cat")
14 dog = Animal(name="dog")
15 veloceraptor = Animal(name="veloceraptor")
16 cat.voice() # Bark !
17 dog.voice() # Meow !
18 veloceraptor.voice() # ...
19
20
```

Python Базовий

Поганий приклад Поліморфізму

```
1  class Messenger:
2      def __init__(self, connection):
3          self._connection = connection
4
5      def send_message(self, text, option=None):
6          self._connection.send(text)
7
8
9      class ExtendedMessenger(Messenger):
10
11         def send_message(self, text, option=None):
12             if option == "message":
13                 self._connection.send(text)
14             elif option == "clean_memory":
15                 print("Cleaning memory")
16             elif option == "math":
17                 print("2+2=4")
18
19
20
```


Python Базовий

Задачі

1. Написати клас `Parallelogram`, який має 2 аргументи `width` і `length` (ширина та довжина) та метод `get_area`, який повертає площу паралелограма з такими сторонами. Успадкувати від нього клас `Square`, перевизначити метод `get_area` як добуток ширини на ширину (або довжини на довжину).
2. Написати функцію, яка приймає 2 параметри `data` та `new_value`. Якщо `data` це список чи множина, то додати до нього елемент `new_value`, якщо рядок, то сконкатенувати (за винятком, якщо `new_value` не список, інакше повернути не змінену `data`), а якщо це число, логічне значення чи словник, то повернути не змінену `data`.

Інформаційний відеосервіс для розробників програмного забезпечення



Перевірка знань

TestProvider.com



Перевірте як Ви засвоїли даний матеріал на [TestProvider.com](https://testprovider.com)

TestProvider – це online сервіс перевірки знань з інформаційних технологій. За його допомогою Ви можете оцінити Ваш рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT спеціаліста.

Успішне проходження фінального тестування дозволить Вам отримати відповідний Сертифікат.

Python Базовий

Дякую за увагу! До нових зустрічей!



Вікторія Бойчук
Python Developer, тренер CBS