РЕФЕРАТ

Тема: Поліморфізм

Поліморфізм є ще одним базовим аспектом об'єктно-орієнтованого програмування та передбачає здатність до зміни функціоналу, успадкованого від базового класу.

Наприклад, нехай у нас буде наступна ієрархія класів:

class Person:

def \_\_init\_\_(self, name, age):

self.\_\_name = name

self.\_\_age = age

@property

def name(self):

return self.\_\_name

@property

def age(self):

return self.\_\_age

@age.setter

def age(self, age):

if age in range(1, 100):

self.\_\_age = age

else:

print("Недопустимый возраст")

def display\_info(self):

print("Имя:", self.\_\_name, "\tВозраст:", self.\_\_age)

class Employee(Person):

def \_\_init\_\_(self, name, age, company):

Person.\_\_init\_\_(self, name, age)

self.company = company

def display\_info(self):

Person.display\_info(self)

print("Компания:", self.company)

class Student(Person):

# определение конструктора

def \_\_init\_\_(self, name, age, university):

Person.\_\_init\_\_(self, name, age)

self.university = university

def display\_info(self):

print("Студент", self.name, "учится в университете", self.university)

people = [Person("Tom", 23), Student("Bob", 19, "Harvard"), Employee("Sam", 35, "Google")]

for person in people:

person.display\_info()

print()

У похідному класі Employee, який представляє службовця, визначається власний конструктор. Тому що нам потрібно встановлювати при створенні об'єкта ще й компанію, де працює працівник. Для цього конструктор приймає чотири параметри: стандартний параметр self, параметри name та age та параметр company.

У конструкторі Employee викликається конструктор базового класу Person. Звернення до методів базового класу має наступний синтаксис:

суперкласс.назва\_метода(self [, параметри])

Тому в конструктор базового класу передаються ім'я та вік. Сам клас Employee додає до функціоналу класу Person ще один атрибут - self.company.

Крім того, клас Employee перевизначає метод display\_info() класу Person, оскільки, крім імені та віку, необхідно виводити ще й компанію, в якій працює службовець. І щоб повторно не писати код виведення імені та віку, тут також відбувається звернення до методу базового класу - методу get\_info: Person.display\_info(self).

Так само визначено клас Student, який представляє студента. Він також перевизначає конструктор і метод display\_info за тим винятком, що замість методу display\_info не викликається версія цього методу з базового класу.

В основній частині програми створюється список із трьох об'єктів Person, у якому два об'єкти також представляють класи Employee та Student. І в циклі цей перелік перебирається, і для кожного об'єкта в переліку викликається спосіб display\_info. На етапі виконання програми Python враховує ієрархію успадкування та вибирає потрібну версію методу display\_info() для кожного об'єкта.

При роботі з об'єктами буває необхідно в залежності від їх типу виконати ті чи інші операції. І за допомогою вбудованої функції isinstance() ми можемо перевірити тип об'єкта. Ця функція приймає два параметри:

isinstance(object, type)

Перший параметр представляє об'єкт, а другий - тип, на належність якого виконується перевірка. Якщо об'єкт представляє вказаний тип, функція повертає True. Наприклад, візьмемо вище описану ієрархію класів:

for person in people:

if isinstance(person, Student):

print(person.university)

elif isinstance(person, Employee):

print(person.company)

else:

print(person.name)

print()

1. Що таке поліморфізм?
2. Що таке віртуальний метод?
3. Що таке раннє зв’язування?
4. Що таке пізнє зв’язування?
5. Чи потрібен спільний нащадок для класів в Python, щоб забезпечити поліморфізм?
6. Яким чином відбувається доступ до елементів базового класу?
7. Як перекрити реалізацію метода в базовому класі?
8. Яка функція дає можливість перевірити тип об’єкта?
9. Чи може клас нащадок викликати методи базового класу?
10. Чи всі методи в python є віртуальними?
11. Чи є якісь відмінності у використанні поліморфізму в python та інших мовах?
12. Чи може програма існувати без поліморфізму?
13. Чи можна не використовувати поліморфізм?
14. Чи явно використовується поліморфізм?
15. Чи застосовується поліморфізм при наслідування?

Джерела:

https://metanit.com/python/tutorial