Поліморфізм

№ уроку: 3 **Курс:** Python Базовий

Засоби навчання: Персональний комп'ютер/ноутбук стандартної продуктивності

Огляд, мета та призначення уроку

Ознайомитися з тим, що таке поліморфізм і як він реалізований у Python. У цьому уроці буде розглянуто саме поняття поліморфізму і після, на практичних прикладах, буде показано, як він реалізований у Python.

Вивчивши матеріал даного заняття, учень зможе:

- Розуміти, що таке поліморфізм і навіщо він потрібний.
- Знати, як він реалізований в Python і вміти застосувати його на практиці.

Зміст уроку

1. Що таке поліморфізм
2. Як він застосуємо на практиці
3. Розв'язання задач

Резюме

- Поліморфізм, напевно, одне з найважливіших понять (властивостей) в ООП та програмуванні в цілому. Кожна система прагне бути ефективною та програмні продукти, з погляду написання коду, не виняток. Одна з найбільших проблем використання класичного декларативного підходу в написанні коду є його дублювання. Функціональне програмування, частково, це вирішує, але не надто ефективно. ООП, у комбінації наслідування та гнучкості мови, дає таку властивість як поліморфізм.
- Поліморфізм (тобто різноманіття форм) є властивістю функцій і класів, коли та сама функція у різних класах поводиться по-різному. Наприклад, якби ми програмували тварин, то у них у всіх був би загальний метод "голос", який кожна конкретна тварина успадкувала б від материнського класу Animal, але при цьому ми знаємо, що кішки кажуть "мяу", а собаки "гав" . У цьому полягає поліморфізм.
- ● Поліморфізм досягається двома способами: перевизначення або перевантаження методу. Перевизначаючи метод, ви у дочірнього класу можете повністю змінити його, а перевантажуючи ви заздалегідь визначаєте аргументи і у разі використання їх усіх функція вже поводиться інакше.
- Ви не зможете перевизначити аргументи методу дочірнього класу (тільки якщо це не конструктор). Але ви можете змінювати внутрішню логіку самого методу (це і є перевизначення або overwriting).
- Перевантаження методу або overlaoding - це заздалегідь визначений алгоритм усередині самого методу, завдяки якому при різній комбінації задіяних аргументів функція веде себе по-різному. Наприклад, є функція з 2 аргументами name та age. Якщо передається лише name, то виводиться “Hello {name}”, і якщо обидва аргументи передаються, то “Hello {name}, you are {age} y.o.”. Це досягається найпростішими розгалуженнями (if/elif/else).

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Закріплення матеріалу

- Що таке поліморфізм?
- Навіщо потрібний поліморфізм?
- Чи є в Python ключові слова, які роблять поліморфною функцію?
- Які є 2 способи досягнення поліморфізму?

Додаткове завдання

Вивчити теорію поліморфізму. Як поліморфізм допомагає у розробці програмних продуктів? Яке його призначення?

Самостійна діяльність учня

1. Написати клас User, в конструкторі будуть визначатися поля age, name, user_type, а метод буде access_database.
2. Зробити спосіб таким, щоб якщо self.user_type дорівнював “superuser”, то спосіб виводив у консоль “access granted”, якщо це просто користувач, то виводило “access denied”.
3. Для суперюзера зробити успадкований клас SuperUser від User.

Рекомендовані ресурси

- https://www.edureka.co/blog/polymorphism-in-python/#:~:text=Polymorphism%20in%20python%20defines%20methods,inherited%20from%20the%20parent%20class.
- https://www.digitalocean.com/community/tutorials/how-to-apply-polymorphism-to-classes-in-python-3
- https://www.askpython.com/python/oops/polymorphism-in-python
- https://codecamp.ru/blog/python-polymorphism/

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com