Полиморфизм

№ урока:	3	**Курс:**	Python Базовый

Средства обучения:	Персональный компьютер/ноутбук стандартной производительности

Обзор, цель и назначение урока

Познакомиться с тем, что такое полиморфизм и как он реализован в Python. В данном уроке будет рассмотрено само понятие полиморфизма и после, на практических примерах, будет показано как он реализован в Python.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое полиморфизм и зачем он нужен.
- Знать, как он реализован в Python и уметь применить его на практике.

Содержание урока

1. Что такое полиморфизм
2. Как он применим на практике
3. Решение задач

Резюме

- Полиморфизм, наверное, одно из самых важных понятий (свойств) в ООП и в программировании в целом. Каждая система стремится быть эффективной и программные продукты, с точки зрения написания кода, не исключение. Одна из самых больших проблем в использовании классического декларативного подхода в написании кода является его дублирование. Функциональное программирование, отчасти, это решает, но не слишком эффективно. ООП, в комбинации наследования и гибкости языка, дает такое свойство как полиморфизм.
- Полиморфизм (т.е. многообразие форм) является свойством функций и классов, когда одна и та же функция в разных классах ведет себя по-разному. Например, если бы мы программировали животных, то у них у всех был бы общий метод “голос”, который каждое конкретное животное наследовало бы от материнского класса Animal, но при этом мы знаем, что кошки говорят “мяу”, а собаки “гав”. В этом и заключается полиморфизм.
- Полиморфизм достигается двумя способами: переопределение или перегрузка метода. Переопределяя метод, вы у дочернего класса можете полностью изменить его, а перегружая вы заранее определяете аргументы и в случае использования их всех функция уже ведет себя иначе.
- Вы не сможете переопределить аргументы метода дочернего класса (только если это не конструктор). Но вы можете менять внутреннюю логику самого метода (это и есть переопределение или overwriting).
- Перегрузка метода или overlaoding это заранее определенный алгоритм внутри самого метода, благодаря которым при разной комбинации задействованных аргументов функция ведет себе по-разному. Например, есть функция с 2 аргументами name и age.

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

Если передается только name, то выводится “Hello {name}”, а если оба аргумента передаются, то “Hello {name}, you are {age} y.o.”. Это достигается простейшими ветвлениями (if/elif/else).

Закрепление материала

- Что такое полиморфизм?
- Зачем нужен полиморфизм?
- Есть ли в Python ключевые слова, делающие функцию полиморфной?
- Какие есть 2 способа достижения полиморфизма?

Дополнительное задание

Изучить теорию полиморфизма. Каким образом полиморфизм помогает в разработке программных продуктов? Каково его назначение?

Самостоятельная деятельность учащегося

1. Написать класс User, у него будут в конструкторе определяться поля age, name, user_type, а метод будет access_database.
2. Сделать метод таким, чтобы если self.user_type был равен “superuser”, то метод выводил в консоль “access granted”, в случае если это просто юзер, то выводило “access denied”.
3. Для суперюзера сделать унаследованный класс SuperUser от User.

Рекомендуемые ресурсы

- 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