Введение в ООП. Наследование

№ урока:	1	**Курс:**	Python Базовый

Средства обучения:	Персональный компьютер/ноутбук стандартной производительности

Обзор, цель и назначение урока

Познакомиться с понятием объектно-ориентированного программирования (ООП) и его особенностями в Python. Рассмотреть один из четырех базовых принципов ООП - наследование.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое ООП и зачем оно нужно.
- Понимать, что такое наследование в ООП и как его применить на практике.

Содержание урока

1. О чем курс
2. Что такое ООП
3. Что такое наследование
4. Примеры использования ООП и наследования на практике

Резюме

- ООП или объектно-ориентированное программирование — это одна из многих парадигм в программировании и в языке Python она тоже реализована, и является очень полезным и мощным инструментом в разработке программ. ООП строится на 4 основных принципах: наследование, инкапсуляция, полиморфизм и абстракция.
- Ключевыми понятиями в ООП являются класс и объект класса. Класс - общее понятие (растение, предмет мебели), а объект или объект класса — это уже конкретное понятие (одуванчик (растение), стул (предмет мебели)). Данная концепция лежит в основе ООП.
- ООП не является единственным стилем программирования, можно писать программы и без него. Но владение им дает неоспоримое преимущество в качестве написания кода и несет в себе массу полезных свойств.
- Наследование является относительно простой идеей. Вы можете создать иерархию классов, где есть материнские (от которых наследуются) и дочерние классы (которые наследуются от материнских).
- Ключевая идея наследования такая же, как и в биологии. Дочерние классы-наследники “перенимают” от материнских их атрибуты и методы. То есть вам не нужно заново их прописывать. Но какой смысл наследования одного и того же без изменений? Никакого.
- Если вы используете наследование, то всегда должны в дочерних классах что-то изменять или добавлять.
- Например, материнский класс Car (автомобиль) может ездить (вперед, назад, поворачивать). Если вы от него унаследуете класс FlyingCar (летающая машина), то можете ему добавить функцию полета, при этом летать она сможет так же - вперед, назад и поворачивать, как и материнский класс.

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

- Также существует множественное наследование, когда материнских классов у дочернего не один, а несколько. Например, класс SuperUser с расширенными правами может быть унаследован от класса Admin и User, при этом имея функции юзера и права администратора.

Закрепление материала

- Что такое ООП?
- Зачем нужно ООП?
- Что такое наследование?
- Может ли быть множественное наследование?

Дополнительное задание

Изучить теорию наследования. Каким образом наследование помогает сократить дублирование кода? Ответьте сами себе на этот вопрос.

Самостоятельная деятельность учащегося

1. Написать класс Cat, создать ему атрибуты size, color, cat_type.
2. При создании объекта класса передавать в конструктор color и cat_type, которые записываются в соответсвующие атрибуты.
3. Сделать метод set_size, в котором если self.cat_type это “indoor”, то self.size = ‘small’ иначе self.size=’undefined’. Протестируйте разные варианты.
4. Сделать класс Tiger, унаследованный от класса Cat.
5. Переопределить метод set_size таким образом чтобы если self.cat_type это ‘wild’, то self.size = ‘big’ иначе self.size=’undefined’.

Рекомендуемые ресурсы

- Википедия (о принципах ООП и о нем в целом)
- Наследование (реализация в Python)
- Inheritance (наследование, англ)

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