Основы Python

GeekBrains

Урок 6

ООП. Введение

На этом уроке

1. Плюсы и минусы механизма ООП.
2. Классы, объекты, атрибуты.
3. Конструкторы, методы.
4. Локальные и глобальные переменные.
5. Модификаторы доступа.
6. Инкапсуляция.
7. Наследование.
8. Множественное наследование.
9. Полиморфизм.

Плюсы и минусы механизма ООП

- Повторное использование кода.
- Повышение читаемости и гибкости кода.
- Ускорение процесса поиска ошибок и их исправления.
- Повышение безопасности проекта.

- Необходимость хорошего понимания предметной области.
- Необходимость хорошего представления структуры приложения.
- Сложность в разбиении проекта на классы.
- Сложность в модификации проекта.

Классы, объекты, атрибуты

Класс →

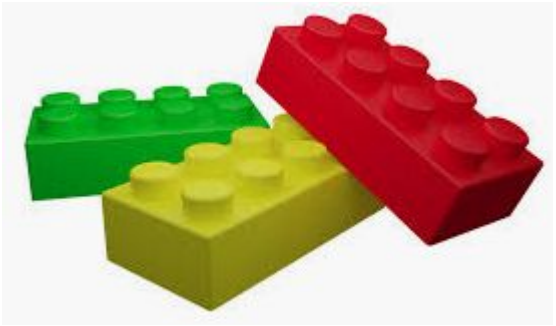
Объект →

Атрибут →

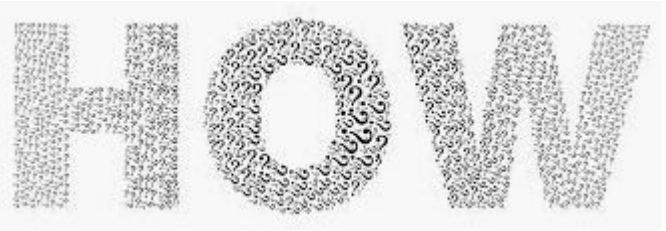
Цвет автомобиля

Конструкторы, методы

```
def __init__(self):
```



```
def get_class_info(self):
```



Переменные

Локальные

```
class Auto:
    def on_start(self):
        info = "Автомобиль заведен"
        return info
```

Глобальные

```
class Auto:
    info_1 = "Автомобиль заведен"

    def on_start(self):
        info_2 = "Автомобиль заведен"
        return info_2
```

Модификаторы доступа

```
self.auto_name = "Mazda"
self._auto_year = 2019
self.__auto_model = "CX9"
```

Публичный (Public)

Защищённый (Protected)

Приватный (Private)

Инкапсуляция

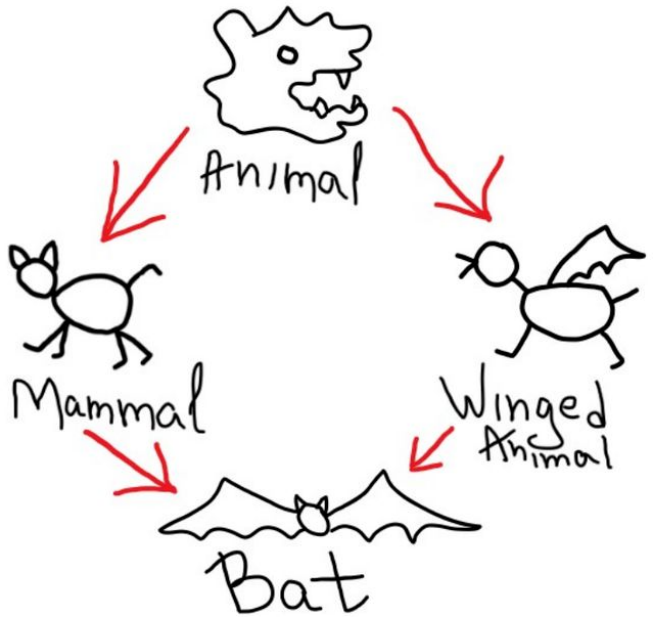
```
class MyClass:
    _attr = "значение"
    def _method(self):
        print("Это защищенный метод!")
```

```
class MyClass:
    __attr = "значение"
    def __method(self):
        print("Это защищенный метод!")
```

Наследование

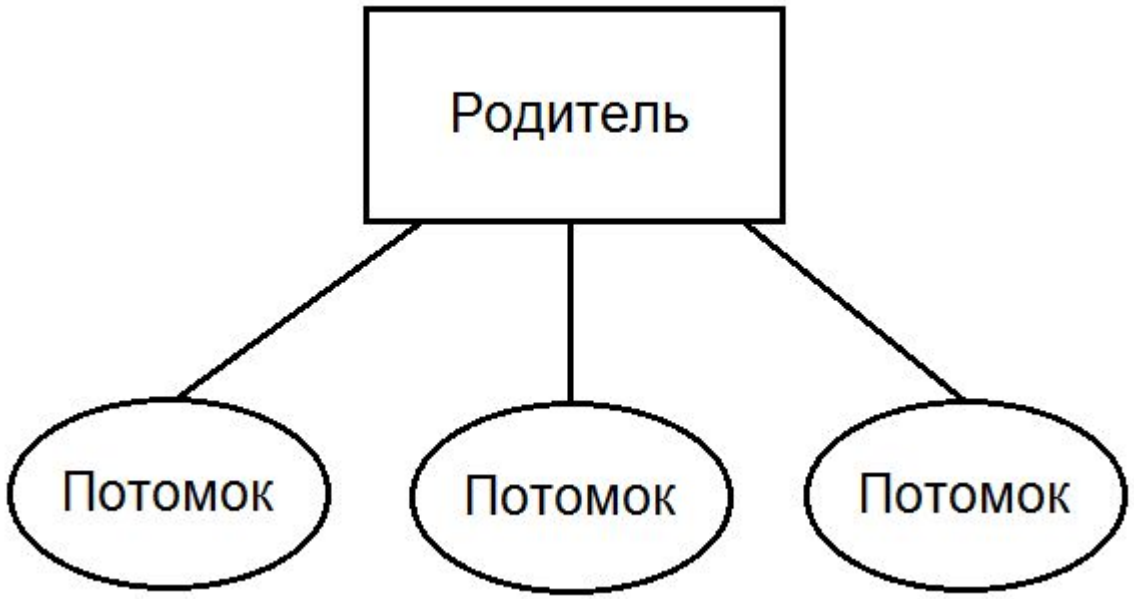
```python
# Класс Transport
class Transport:
    def transport_method(self):
        print("Это родительский метод из класса Transport")


# Класс Auto, наследующий Transport
class Auto(Transport):
    def auto_method(self):
        print("Это метод из дочернего класса")
```

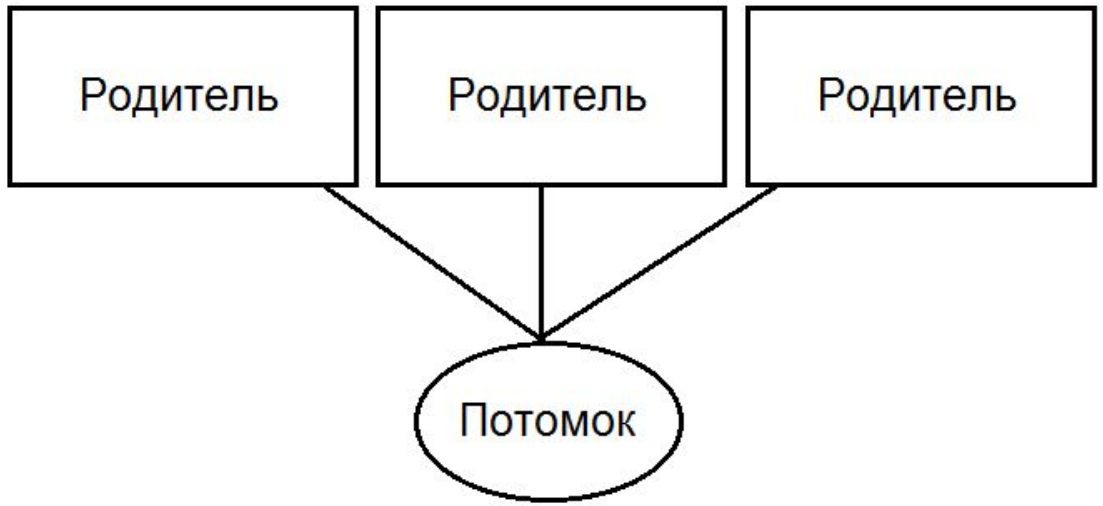


Множественное наследование

Несколько дочерних классов у одного родителя



Несколько родителей у одного класса



Полиморфизм

Перегрузка методов

Реализуется в возможности метода отражать разную логику выполнения в зависимости от количества и типа передаваемых параметров.

Переопределение методов

Переопределение методов в полиморфизме выражается в наличии метода с одинаковым названием для родительского и дочернего классов.

Итоги

1. Познакомились с основами ООП и знаете, чем отличается объект от класса.
2. Узнали, для чего нужны атрибуты и методы.
3. Рассмотрели, как определяются локальные и глобальные атрибуты классов.
4. Узнали, для чего используются модификаторы доступа.
5. Научились реализовывать в своём коде инкапсуляцию, наследование, множественное наследование и полиморфизм.
6. Разобрали достоинства и недостатки ООП.