

# Анализ социальных сетей. Практика 6

## Основы объектно-ориентированного программирования

Михаил Пожидаев

27 апреля 2024 г.

# Классы и объекты

## Объектно-ориентированное программирование

### Классы!

Классы определяют шаблон объектов. С классом нельзя работать напрямую, он только определяет создаваемые на его основе объекты. При наследовании наследуемый класс получает все свойства своего родителя.

### Объекты!

Объекты являются экземплярами классов для работы. Все структуры, из которых состоит классов, для каждого объекта копируются таким образом, что изменение данных в одном объекте не влечёт изменение данных в другом объекте этого же класса.

# Яблоко

## *Простой класс*

```
class apple:
    def __init__(self, comment):
        self.comment = comment

    def info(self):
        print("Это", self.type(), self.comment)

    def type(self):
        return "яблоко"

a = apple("для гостей")
a.info()
```

# Апельсин

## *Ещё один простой класс*

```
class orange:
    def __init__(self, comment):
        self.comment = comment

    def info(self):
        print("Это", self.type(), self.comment)

    def type(self):
        return "апельсин"

a = orange("для хозяев")
a.info()
```

# Фрукт

*Родительский класс для яблока и апельсина*

```
class fruit:
    def __init__(self, comment):
        self.comment = comment

    def info(self):
        print("Это", self.type(), self.comment)
```

# Яблоко как фрукт

## *Наследование класса яблока*

```
class apple(fruit):  
    def type(self):  
        return "яблоко"  
  
a = apple("для меня")  
a.info()
```

# Апельсин как фрукт

## *Наследование класса апельсина*

```
class orange(fruit):  
    def type(self):  
        return "апельсин"  
  
a = orange("для меня")  
a.info()
```

# Спасибо за внимание!

Всё о курсе: <https://marigostra.ru/materials/networks.html>

E-mail: [msp@luwrain.org](mailto:msp@luwrain.org)

Канал в Телеграм: <https://t.me/MarigostraRu>