

Объектно ориентированное программирование

Набор техник программирования, основанных на использовании концепции «объекта». Объекты реального мира могут быть представлены в виде некоторых абстрактных чертежей – классов – на основе которых создаются и взаимодействуют объекты в памяти.

🔪 **Класс** – структурная единица ООП, представляет собой некую абстракцию, описанную с помощью кода.

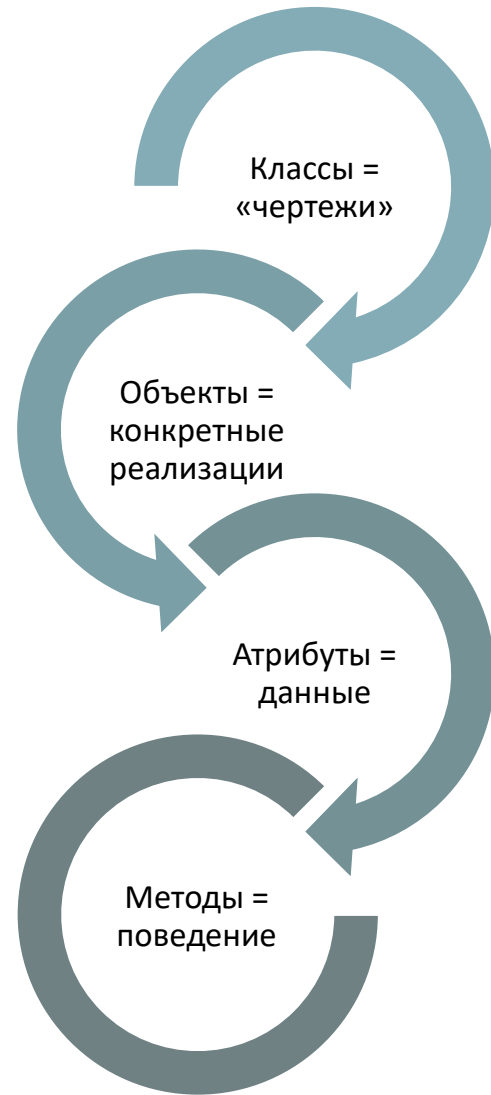
🔪 **Экземпляр класса** – объект, конкретный представитель некоторого класса.

🔪 **Атрибут** – переменная, принадлежащая классу или экземпляру.

🔪 **Метод** – функция, обычно доступная через атрибут.

Терминология

Данные концепции
общеприняты и не зависят от
языка программирования



Иерархия ООП

Нужно учитывать, что атрибуты и методы могут быть как у конкретного экземпляра, так и у самого класса

Класс

- ✂ Инструкция **class** создаёт объект класса и присваивает ему имя
- ✂ Присваивания внутри инструкции **class** создают атрибуты
- ✂ Атрибуты сообщают состояние и способы поведения всем экземплярам класса

Экземпляр класса

- 🐾 Вызов объекта класса как функции создаёт новый экземпляр этого класса
- 🐾 Каждый экземпляр наследует атрибуты класса и получает собственное пространство имён
- 🐾 Изменения атрибутов экземпляра не влияет на сам класс

```
class DataPrinter:
    data = "test"  #атрибут класса

    def print(self):  #метод
        print(self.data)  #атрибут
                           экземпляра

    def update_data(self, new_data):
        self.data = new_data
```

Синтаксис

Аргумент self позволяет
получить доступ к конкретному
экземпляру класса

Простейшие классы



Абстракция



Инкапсуляция



Наследование



Полиморфизм

Принципы ООП