ООП

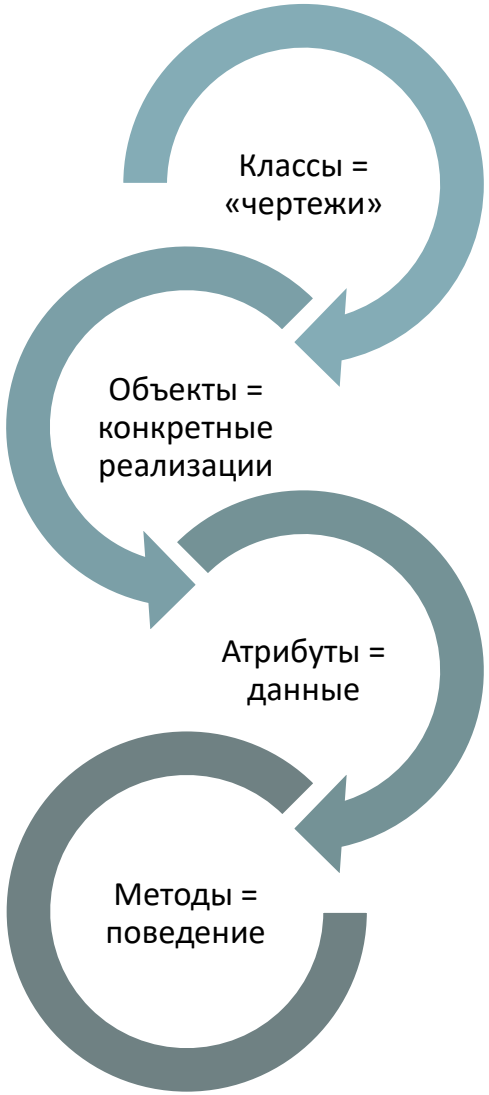
Объектно ориентированное программирование

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.

Терминология

Данные концепции общепринятые и не зависят от языка программирования

- **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.



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

Принципы ООП

Абстракция

Инкапсуляция

Наследование

Полиморфизм

Абстракция

Абстракция подразумевает, что в чертёж объекта попадают только необходимые для работы программы характеристики, при этом внутренняя структура объекта скрывается (с логической точки зрения) от пользователя, который может оперировать только высокоуровневыми понятиями объекта и набором его поведений.

Инкапсуляция

В отличии от абстракции, инкапсуляция подразумевает процесс сокрытия внутренностей объектов уже с технической точки зрения. Все объекты должны предоставлять вызывающему коду определённый набор методов-поведений, с помощью которых с ними можно взаимодействовать. **Менять данные объекта в обход его методов – дурной тон**.

```
class Employee:
        #объявление инициализатора
    def __init__(self,name,job,salary):
    #заполнение атрибутов экземпляра
            self.name = name
            self.job = job
            self.salary = salary
```

Инициализатор

Специальный метод, описывающий логику создания нового экземпляра класса