**Тема: ООП**

В ООП все описывается классами и объектами

Каждый объект является **самостоятельным и отвечает сам за себя**

Объекты могут взаимодействовать друг с другом!!

**Класс -** абстракция над объектом. У класса может быть неограниченное количество объектов. Например, чертеж дома - класс, а сам дом - объект этого класса

**Плюсы и минусы ООП**

**Плюсы**

1.В парадигме объектов легче писать код.

2.Читать код гораздо проще

3.Код легче обновлять и расширять.

4.Программистам легче работать в команде

5.Шаблоны проектирования

**Минусы**

1.Сложность в усвоении

2.Громоздкость

3.Низкая производительность

**Нюансы создания классов**

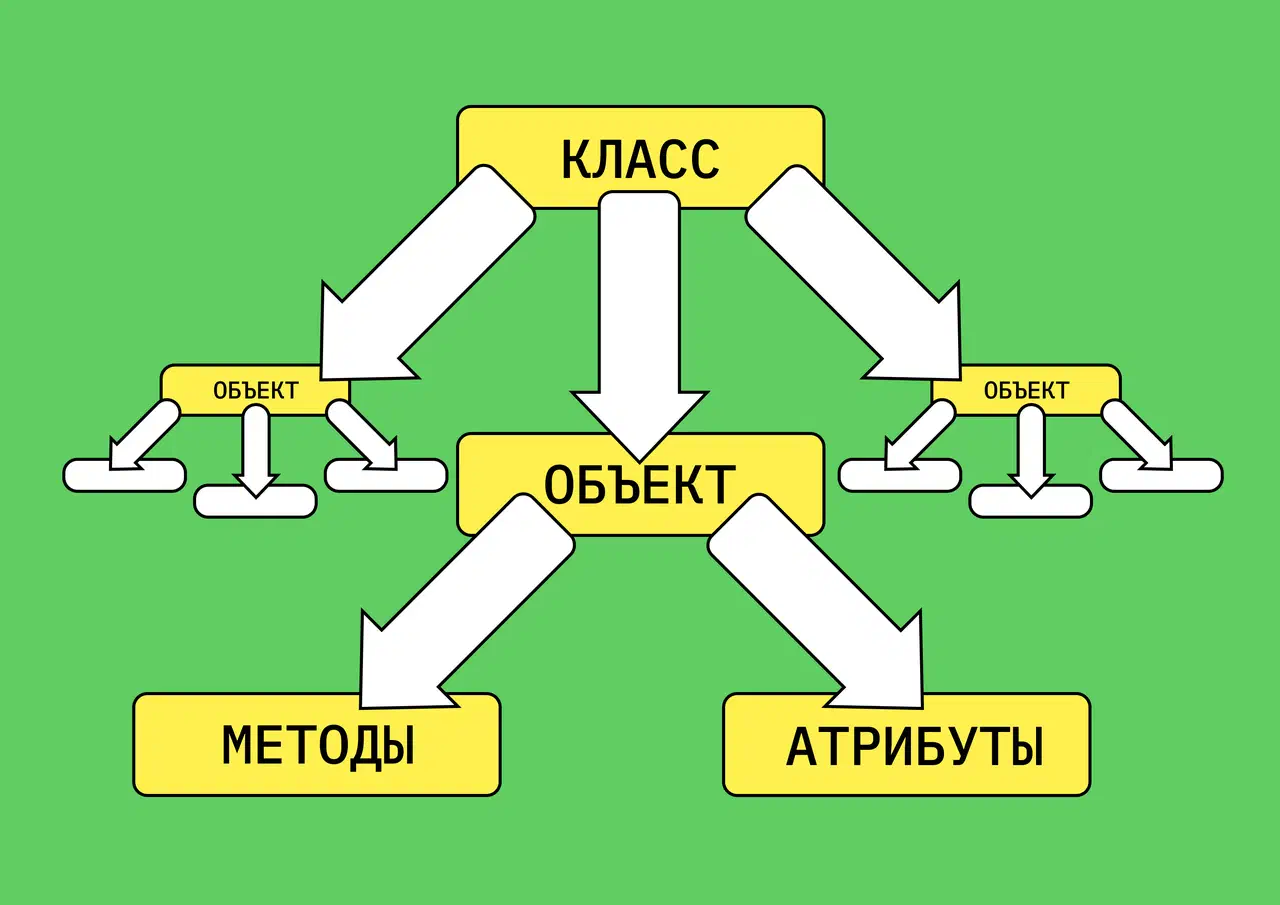
1.Класс должен описывать только один объект

2.Класс не может называться **глаголом**

3. Нужно использовать понятный интерфейс для общения с пользователем

4.Хорошая реализация **инкапсуляции** .Хорошо прописанные **свойства и методы**

5.Правильная реализация



**Поля и методы внутри классов**

**Виды методов**

**1.Метод экземпляра(один объект)** - метод, которым мы вылезаем из под объекта

**2.Статические методы -** данные методы вызываются из самого класса, общий для всех объектов

**3.Метод-конструктор -** Вызывается при создании объекта

**4.Методы мутаторы и акссесоры(сэтр и гэтр соответственно)**

**5.Виртуальный и абстрактный метод -** абстрактный - метод, который в обязательном порядке нужно переопределить фв

**Правила создания методов**

***Метод -*** *функция внутри класса*

***Поля -*** *свойства класса*

1.Один метод - одна задача

2.Имя метода должно четко отражать его задачу

3.Правильно использовать параметры метода

4.Не более 7 параметров

5.Не использовать параметры в качестве рабочих переменных

6.Необходимо проверять входные данные

**Основы ООП**

**Наследование -** возможность создавать более спец. классы на основе уже существующих

**Множественное наследование -** когда “наследник” **(SubClass)**  может наследоваться от множества “родителей”**(SuperClass)**.