

# Синтаксис классов

## Цель

Овладеть навыком создания высококачественного и модульного программного кода с использованием принципов и инструментов объектно-ориентированного программирования

## Что нужно сделать

Решите следующие задачи на языке программирования Python

### 01 Оркестр животных и их симфония методов

Требуется создать иерархию классов для животных: базовый класс **Animal**, а затем два подкласса **Mammal** и **Bird**. Каждый из подклассов может иметь свои уникальные свойства и методы, а также наследовать общие от **Animal**.

Класс **Animal** содержит атрибуты **name** и **sound**. У него должен быть метод **make\_sound**, который выводит звук этого животного.

Создайте подкласс **Mammal**, который наследуется от **Animal**.

Добавьте ему атрибут **num\_legs** и метод **give\_birth**, который выводит сообщение о том, что млекопитающее родило потомство.

Создайте подкласс **Bird**, также наследующий от **Animal**. Добавьте ему атрибут **can\_fly** и метод **fly**, который выводит сообщение о том, что птица летит.

Попробуйте создать несколько экземпляров каждого класса и вызвать их методы.

### 02 Библиотечная

Создайте простую систему для учета книг в библиотеке, содержащую два класса: класс **Book** для представления книги и класс **Library** для управления коллекцией книг.

Класс **Book** содержит атрибуты **title**, **author** и **published\_year** (название книги, автор книги, год публикации книги).

Класс **Library** хранит список книг. У него должны быть методы для добавления новой книги, удаления книги по названию, и вывода списка всех книг в библиотеке.

Создайте несколько экземпляров книг и библиотеки, и проведите операции добавления, удаления и вывода книг.

## 03 Калькулятор

Требуется создать простую программу-калькулятор с использованием класса **Calculator**, который будет выполнять основные арифметические операции.

Класс **Calculator** должен содержать следующие методы:

**Метод `set_value`**

который позволяет установить текущее значение калькулятора

**Метод `get_value`**

который позволяет получить текущее значение калькулятора

**Методы `add`, `subtract`, `multiply`, `divide`**

для сложения, вычитания, умножения и деления текущего значения на число. В случае деления на ноль, вывести сообщение об ошибке

## 04 Калькулятор 2.0

Требуется создать наследника класса **Calculator**, класс **AdvancedCalculator**, который расширит функциональность калькулятора из предыдущей задачи. Новый калькулятор сможет выполнять более сложные операции такие как: возведение в степень и вычисление квадратного корня.

Класс **AdvancedCalculator** должен содержать следующие методы:

**Метод `power`**

для возведения значения в степень

**Метод `square_root`**

для нахождения квадратного корня из значения. В случае взятия корня из отрицательного числа, вывести сообщение об ошибке

## Результат

В ответе приложите файл с расширением `ru` для каждой задачи

## **Критерии оценивания**

**K1** Решено верно 2 задачи с выполнением всех условий **2 балла**

**K2** Решено верно 3 задачи с выполнением всех условий **3 балла**

**K3** Решено верно 4 задачи с выполнением всех условий **4 балла**

**Максимальное количество баллов** **4 балла**

**Минимальное количество баллов**  
чтобы преподаватель смог зачесть вашу работу **2 балла**