

# Lesson 16 - OOP

## Повторение

1. Теория из прошлого урока
2. Разбор домашнего задания

## Legend

### Классы и объекты

Ресурс: <https://metanit.com/python/tutorial/7.1.php>

### Задачи

#### 1. Создание класса

Создайте класс `Car` с атрибутами:

- `brand` (марка машины)
- `model` (модель)
- `year` (год выпуска)

Добавьте метод `display_info`, который выводит информацию о машине.

#### 2. Методы и атрибуты

Дополните класс `Car` методом `is_old`, который возвращает `True`, если машине больше 10 лет, иначе – `False`.

#### 3. Конструктор

Создайте класс `Rectangle` с атрибутами:

- `width` (ширина)
- `height` (высота)

Добавьте методы:

- `area` – для вычисления площади.
- `perimeter` – для вычисления периметра.

#### 4. Класс `Book`

Создайте класс `Book` с атрибутами:

- `title` (название книги)
- `author` (автор книги)
- `year` (год публикации)

Добавьте метод `display_info`, который выводит информацию о книге в формате:

"Название: <title>, Автор: <author>, Год: <year>".

#### 5. Класс Student

Создайте класс Student с атрибутами:

- name (имя студента)
- age (возраст)
- grades (список оценок)

Добавьте методы:

- add\_grade(grade) – добавляет оценку в список.
- get\_average() – возвращает среднюю оценку.

#### 6. Класс Circle

Создайте класс Circle с атрибутом:

- radius (радиус круга).

Добавьте методы:

- area() – возвращает площадь круга.
- circumference() – возвращает длину окружности.

#### 7. Класс Person

Создайте класс Person с атрибутами:

- name (имя)
- age (возраст)

Добавьте метод is\_adult, который возвращает True, если возраст больше или равен 18, иначе False.

#### 8. Класс Rectangle

Создайте класс Rectangle с атрибутами:

- width (ширина)
- height (высота)

Добавьте методы:

- is\_square() – возвращает True, если ширина равна высоте.
- resize(new\_width, new\_height) – изменяет ширину и высоту.

#### 9. Класс BankAccount

Создайте класс BankAccount с атрибутами:

- owner (владелец счета)
- balance (баланс)

Добавьте методы:

- deposit(amount) – увеличивает баланс.
- withdraw(amount) – уменьшает баланс, если денег достаточно.
- check\_balance() – выводит текущий баланс.

#### 10. Класс Dog

Создайте класс Dog с атрибутами:

- name (имя)
- age (возраст)
- breed (порода)

Добавьте метод bark, который выводит строку: "Гав-гав, меня зовут <name>!".

#### 11. Класс Temperature

Создайте класс Temperature с атрибутом:

- celsius (температура в градусах Цельсия).

Добавьте методы:

- to\_fahrenheit() – переводит температуру в Фаренгейты.
- to\_kelvin() – переводит температуру в Кельвины.

#### 12. Класс Triangle

Создайте класс Triangle с атрибутами:

- a, b, c (стороны треугольника).

Добавьте методы:

- is\_valid() – проверяет, существует ли треугольник (сумма любых двух сторон должна быть больше третьей).
- perimeter() – возвращает периметр треугольника.

#### 13. Класс Laptop

Создайте класс Laptop с атрибутами:

- brand (бренд)
- model (модель)
- price (цена)

Добавьте метод apply\_discount(percent), который уменьшает цену на указанное количество процентов.

#### 14. Класс Movie

Создайте класс Movie с атрибутами:

- title (название)
- director (режиссер)
- rating (рейтинг)

Добавьте метод is\_high\_rated, который возвращает True, если рейтинг больше 8.

#### 15. Класс Clock

Создайте класс `Clock` с атрибутами:

- `hours`
- `minutes`

Добавьте метод `add_minutes(minutes)`, который добавляет указанное количество минут и корректно обновляет часы.

## ДЗ-16

### 4. Класс `Book`

Создайте класс `Book` с атрибутами:

- `title` (название книги)
- `author` (автор книги)
- `year` (год публикации)

Добавьте метод `display_info`, который выводит информацию о книге в формате: "Название: <title>, Автор: <author>, Год: <year>".

### 5. Класс `Student`

Создайте класс `Student` с атрибутами:

- `name` (имя студента)
- `age` (возраст)
- `grades` (список оценок)

Добавьте методы:

- `add_grade(grade)` – добавляет оценку в список.
- `get_average()` – возвращает среднюю оценку.

### 6. Класс `Circle`

Создайте класс `Circle` с атрибутом:

- `radius` (радиус круга).

Добавьте методы:

- `area()` – возвращает площадь круга.
- `circumference()` – возвращает длину окружности.

### 7. Класс `Person`

Создайте класс `Person` с атрибутами:

- `name` (имя)
- `age` (возраст)

Добавьте метод `is_adult`, который возвращает `True`, если возраст больше или равен 18, иначе `False`.

### 8. Класс `Rectangle`

Создайте класс `Rectangle` с атрибутами:

- `width` (ширина)
- `height` (высота)

Добавьте методы:

- `is_square()` – возвращает `True`, если ширина равна высоте.
- `resize(new_width, new_height)` – изменяет ширину и высоту.

## 9. Класс `BankAccount`

Создайте класс `BankAccount` с атрибутами:

- `owner` (владелец счета)
- `balance` (баланс)

Добавьте методы:

- `deposit(amount)` – увеличивает баланс.
- `withdraw(amount)` – уменьшает баланс, если денег достаточно.
- `check_balance()` – выводит текущий баланс.

## 10. Класс `Dog`

Создайте класс `Dog` с атрибутами:

- `name` (имя)
- `age` (возраст)
- `breed` (порода)

Добавьте метод `bark`, который выводит строку: "Гав-гав, меня зовут <name>!".