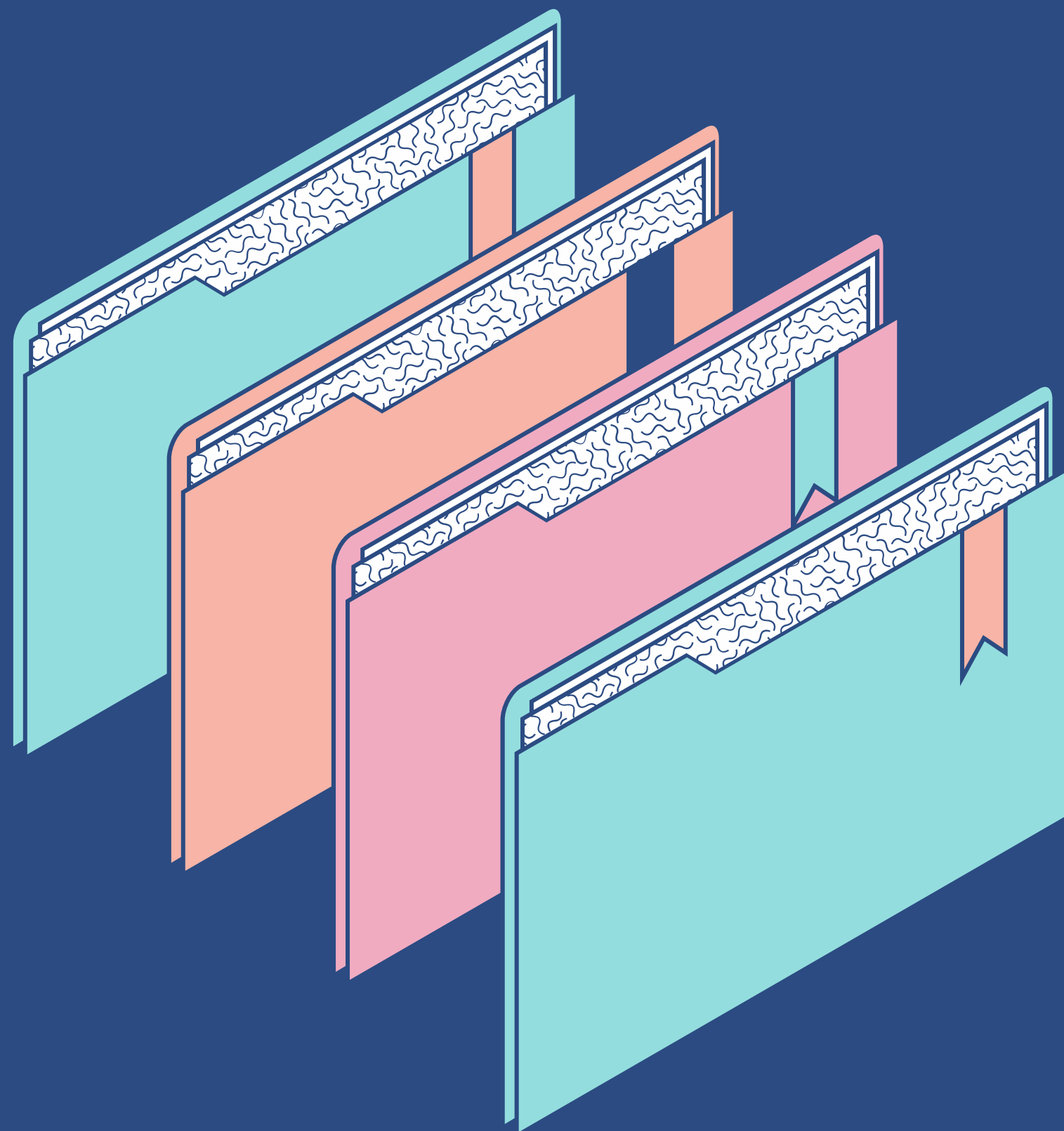




КАФЕДРА №13.

# Понятие класса и объекта класса

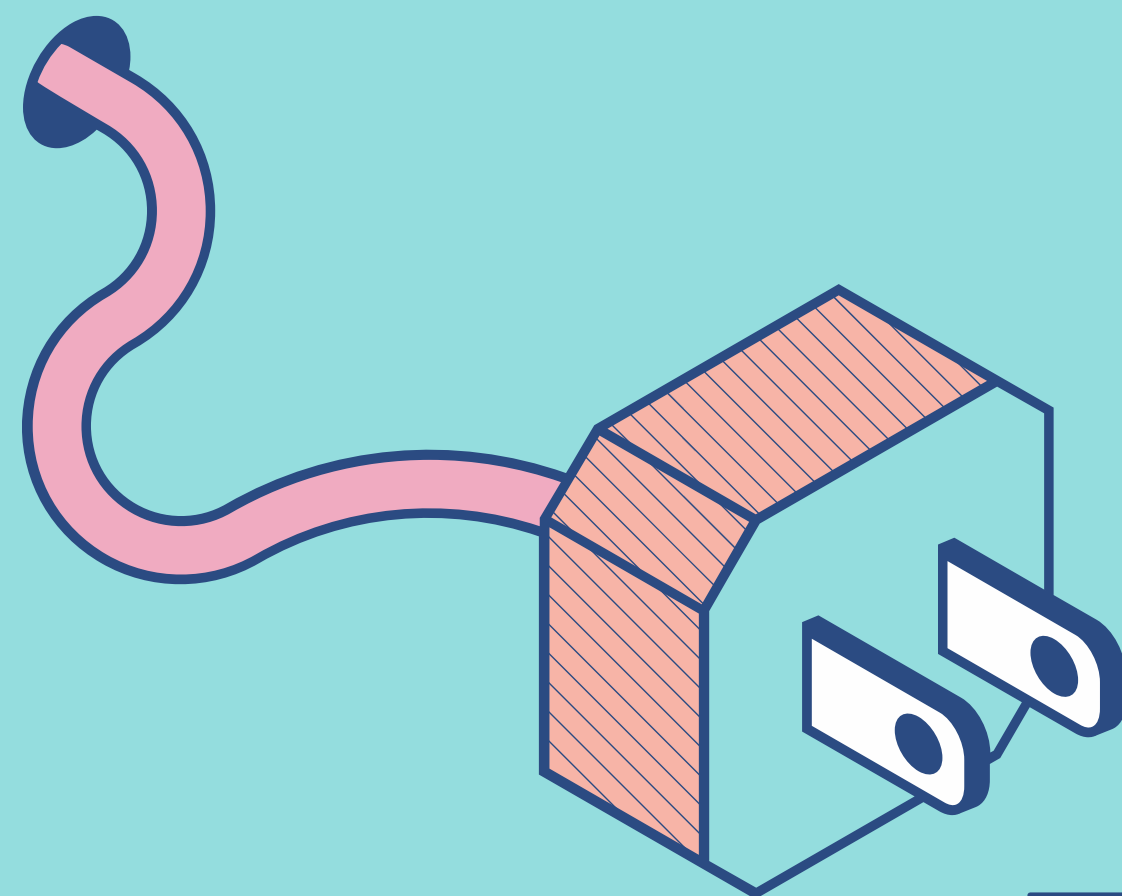
Литвинов Владислав



# ПОВЕСТКА

## КЛЮЧЕВЫЕ МОМЕНТЫ В ЭТОЙ ПРЕЗЕНТАЦИИ

- Основы объектно-ориентированного программирования.
- Понятие класса и объекта класса.
- Методы класса. Конструкторы и деструкторы.
- Атрибуты класса.



**КЛАСС**

**ОБЪЕКТ**

**МЕТОДЫ**

**АТТРИБУТЫ**

**ОБЪЕКТ**

**МЕТОДЫ**

**АТТРИБУТЫ**

**ОБЪЕКТ**

**МЕТОДЫ**

**АТТРИБУТЫ**

# СТРУКТУРА ООП

ООП помогает организовывать код так, чтобы он был более структурированным, понятным и поддерживаемым. Это особенно полезно для крупных проектов, где важна модульность и повторное использование кода.



# Что такое класс?

- Шаблон для создания объектов
- Определяется с помощью слова **class**
- Класс в простом понимании – это как что-то общее: например, есть ноутбук и системный блок + монитор – всё это является компьютерами
- Получается, что у нас появился класс **Персональный компьютер**

# Что такое объект класса?

- Объект – экземпляр класса. Те самые **ноутбук, системный блок, смартфон**
- Каждый объект не связан с другими объектами
- У объекта присутствуют все **атрибуты** и **методы** класса. Их нельзя переопределить или удалить/добавить, но можно использовать



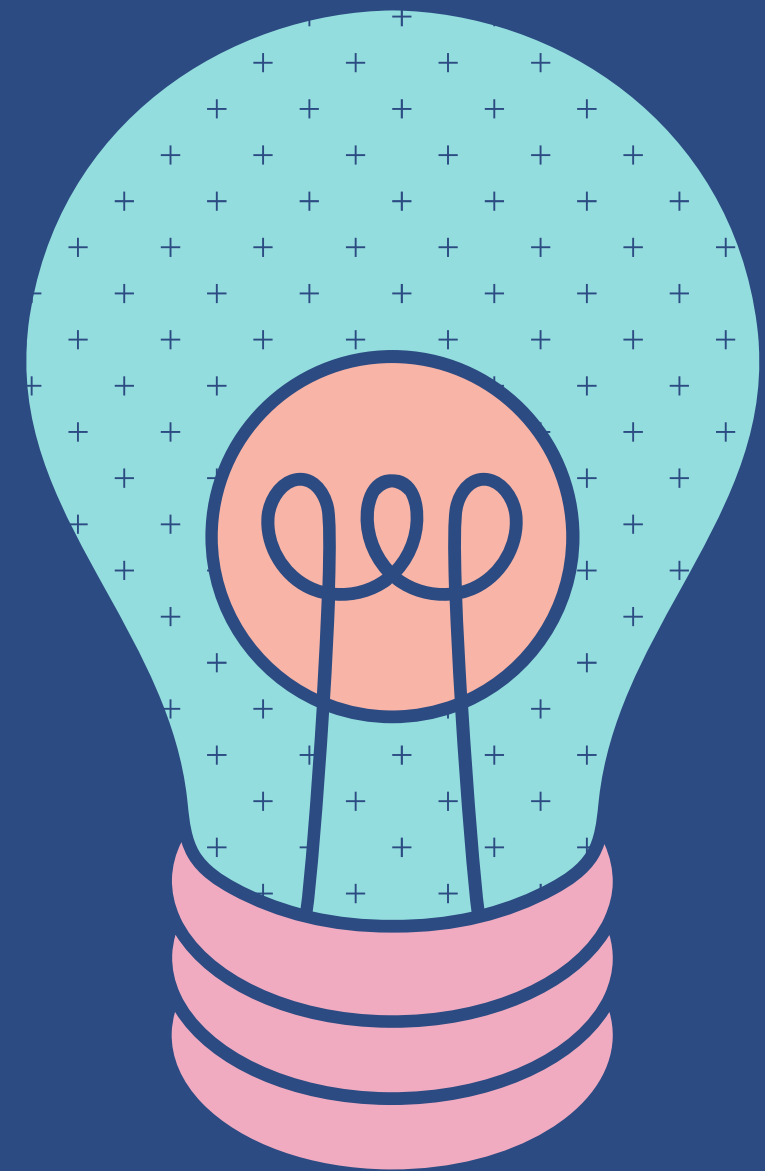
# Что такое метод?

- Метод класса – это поведение, действие объекта. По сути, это функция внутри класса.
- Определяется как и функция с помощью слова **def**.
- Среди методов есть пара особых – **конструктор** и **деструктор**. Первый – вызывается при создании объекта, второй – при удалении.
- Вспоминая класс **Компьютер**, метод может быть **Включить компьютер**.

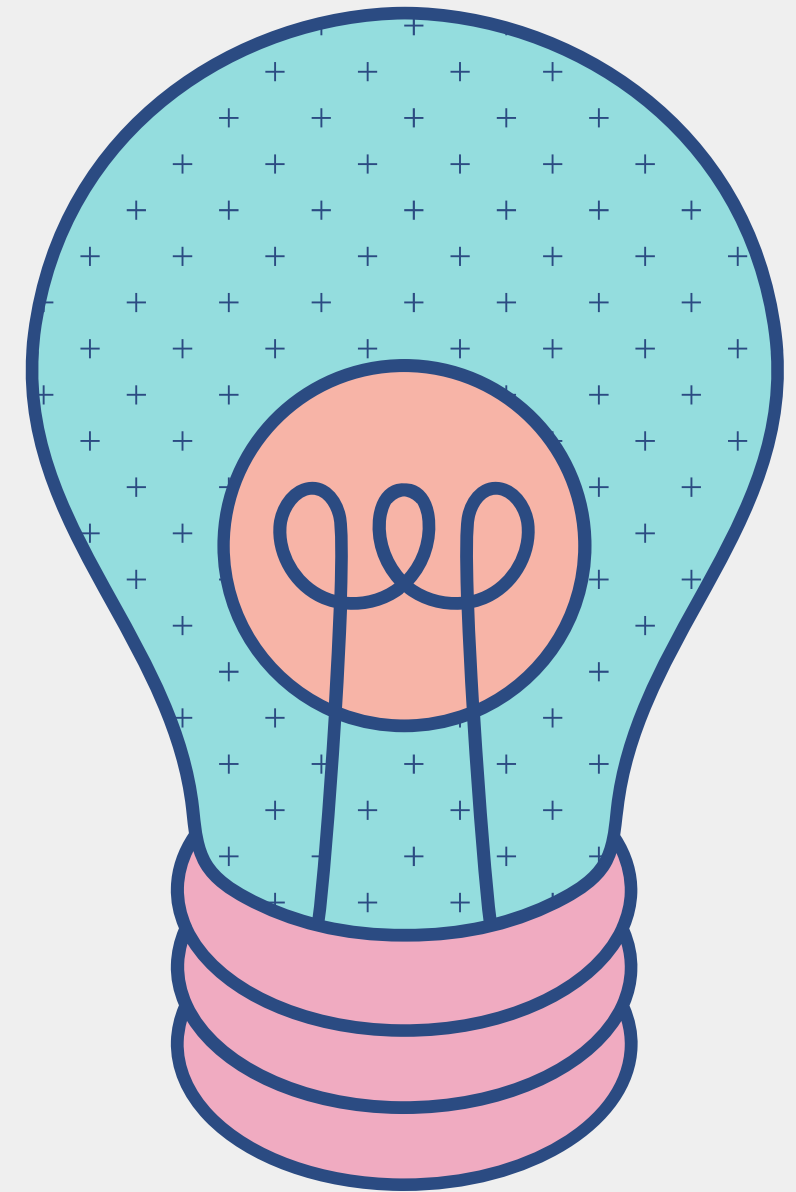
# Что такое атрибут?

- Атрибут класса – это характеристика объекта – переменная внутри класса.
- Определяется как обычная переменная.
- Если говорить про класс **Компьютер**, атрибутами будут **диагональ экрана** и **объем памяти**, например.

Конструктор, как и остальные методы, определяется с помощью **def**. Имя конструктора всегда **\_\_init\_\_**. Внутри данного метода объявляем все атрибуты класса с помощью ключевого префикса **self**.



Деструктор, как и конструктор, определяется с помощью **def**.  
Имя деструктора всегда **`__del__`**. Деструктор вызывается интерпретатором автоматически при удалении объекта.



# Создаем класс правильно

1

ШАГ

Выберите имя класса.

```
class Computer:
```

2

ШАГ

Создайте конструктор класса, перечислите атрибуты

```
def __init__(self, diag, mem):  
    self.diag = diag  
    self.mem = mem
```

3

ШАГ

Создайте деструктор класса

```
def __del__(self):  
    print("del done", self.diag)
```

4

ШАГ

Создайте все необходимые методы класса

```
def info(self):  
    print(f"diag: {self.diag},  
          mem:{self.mem}")
```

5

ШАГ

Создайте объект класса

```
notebook = Computer(14, 512)
```



```
1 class Computer:
2
3     def __init__(self, diag, memory):
4         self.diag = diag
5         self.mem = memory
6
7     def info(self):
8         print(f"diag: {self.diag}, memory:{self.mem} GB")
9
10    def __del__(self):
11        print("del done", self.diag)
12
```

✓ 0.0s

```
1 notebook = Computer(14, 512)
2 notebook.info()
```

✓ 0.0s

diag: 14, memory:512 GB

```
1 def left_func():
2     pc = Computer(32, 2048)
3     pc.info()
```

✓ 0.0s

```
1 left_func()
```

✓ 0.0s

diag: 32, memory:2048 GB  
del done 32

# Результат нашей работы

ЕСЛИ ВЫ ВСЕ СДЕЛАЛИ ПРАВИЛЬНО – ВАС ЖДУТ НЕСКОЛЬКО ЗЕЛЕННЫХ ГАЛОЧЕК!.

Что же мы сделали? Создали класс **Computer**, определили классу конструктор и деструктор, а также метод **info**, который выдает всю информацию об атрибутах данного класса: **diag** и **memory**. Также мы создали два объекта класса: **notebook** и **pc**. Вторым объектом мы создали внутри функции, и когда функция отработает, объект будет удален. Первый же объект, созданный в глобальной области, будет удален только по окончании выполнения программы.



## Следующие шаги нашего обучения

- Наследование
- Инкапсуляция
- Полиморфизм

# У вас есть какие-то вопросы?

Оставьте их себе! Надеюсь, что вы узнали что-то новое.

**Позвоните мне**

+375 (33) 398-54-37

.....

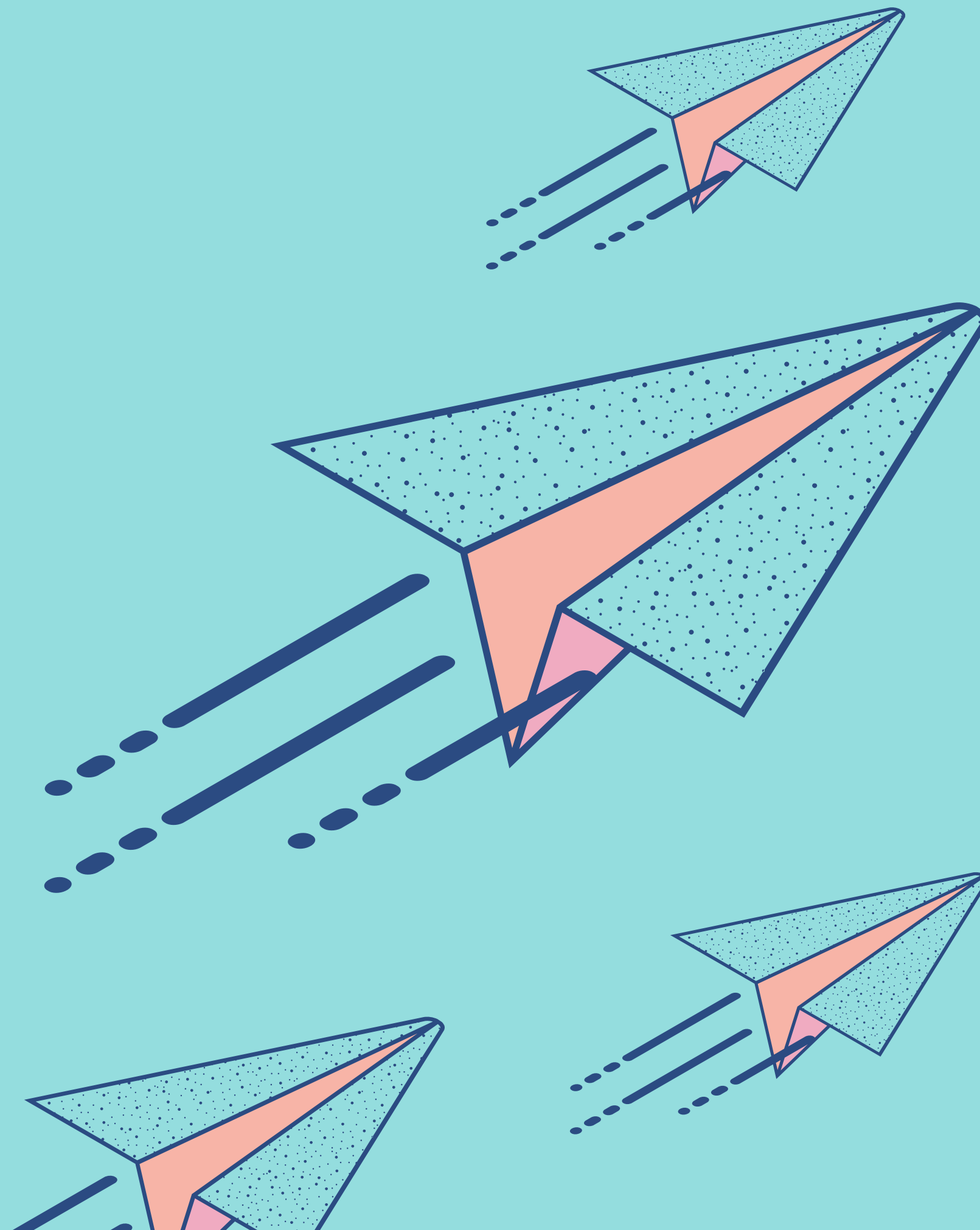
**Электронная почта**

kordebalet67@gmail.com

.....

**Telegram**

@Kordebalet





КАФЕДРА №13.

# Понятие класса и объекта класса

Литвинов Владислав