

## Что такое classmethod?

`classmethod` — это метод, который привязан к самому классу, а не к его экземпляру. В отличие от обычных методов, которые принимают первым параметром `self` (ссылающийся на экземпляр класса), методы класса принимают первым параметром `cls`, который ссылается на сам класс.

## Зачем нужен classmethod?

Методы класса полезны, когда нам нужно выполнять операции, которые относятся к самому классу, а не к конкретному экземпляру. Например, они часто используются для создания альтернативных конструкторов или для выполнения операций, которые зависят от класса в целом, а не от его отдельных объектов.

## Какие задачи решает classmethod?

1. **Работа с данными класса:** Позволяет изменять и читать данные, относящиеся ко всему классу, а не к отдельным объектам.
2. **Обеспечение инкапсуляции логики:** Логика, относящаяся ко всему классу, может быть инкапсулирована в методах класса.
3. **Создание альтернативных конструкторов:** Позволяет создавать объекты несколькими способами.

## Применение classmethod

Для создания метода класса используется декоратор `@classmethod`.  
Пример базового использования:

```
class MyClass:
    class_attribute = 0

    def __init__(self, value):
        self.instance_attribute = value

    @classmethod
    def class_method(cls):
        return f"Class attribute value: {cls.class_attribute}"
```

## Примеры использования

Пример 1: Работа с данными класса

```
class MyClass:
    class_counter = 0

    def __init__(self):
        MyClass.class_counter += 1

    @classmethod
    def get_instance_count(cls):
        return cls.class_counter

# Создание экземпляров
obj1 = MyClass()
obj2 = MyClass()
obj3 = MyClass()

# Получение количества созданных экземпляров
print(f"Number of instances: {MyClass.get_instance_count()}")
```

## Пример 2: Альтернативный конструктор

```
class Date:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    @classmethod
    def from_string(cls, date_string):
        year, month, day = map(int, date_string.split('-'))
        return cls(year, month, day)

# Использование основного конструктора
date1 = Date(2023, 5, 19)
print(f"Date1: {date1.year}-{date1.month}-{date1.day}")

# Использование альтернативного конструктора
date2 = Date.from_string('2023-05-19')
print(f"Date2: {date2.year}-{date2.month}-{date2.day}")
```

## Задания для закрепления

1. Создайте класс `Student` с методом класса `from_dict`, который создает экземпляр студента из словаря с ключами 'name', 'age' и 'grade'.
2. Создайте класс `Pizza` с методом класса `margherita` и `pepperoni`, которые создают объекты пиццы с фиксированными ингредиентами.
3. Создайте класс `Book` с методами `from_dict` и `display_info`. Метод `from_dict` должен создавать объект книги из словаря с ключами 'title', 'author' и 'isbn'.
4. Создайте класс `Employee` с методом класса `from_string`, который создает экземпляр сотрудника из строки формата "Имя,Возраст,Должность".