

# Лабораторная работа 4

---

## 1. Базовый класс

Создайте класс `Transport`, который будет описывать транспортное средство.

У класса должны быть:

- атрибуты: `brand` (марка), `speed` (скорость),
  - метод `move()`, который выводит сообщение `"Transport is moving at {speed} km/h"`,
  - магический метод `__str__`, который возвращает строку `"Transport: {brand}, Speed: {speed}"`.
- 

## 2. Наследники

Создайте два дочерних класса:

### 1. `Car`

- Дополнительный атрибут: `seats` (количество мест).
- Метод `honk()`, который выводит `"Beep beep!"`.
- Переопределите метод `move()`, чтобы он выводил `"Car {brand} is driving at {speed} km/h"`.
- В конструкторе (`__init__`) используйте `super()`, чтобы вызвать инициализацию базового класса.
- В `__str__` добавьте информацию про количество мест.

### 2. `Bike`

- Дополнительный атрибут: `type` (например, "mountain" или "road").
  - Переопределите метод `move()`, чтобы он выводил `"Bike {brand} is cycling at {speed} km/h"`.
  - Используйте `super()` в конструкторе.
  - В `__str__` добавьте информацию про тип велосипеда.
- 

## 3. Магические методы

В класс `Car` добавьте:

- `__len__` — пусть возвращает количество мест.
  - `__eq__` — сравнение двух машин по скорости.
  - `__add__` — пусть сложение двух машин возвращает их «суммарную скорость».
- 

## 4. Практика использования

1. Создайте несколько объектов разных классов (`Transport`, `Car`, `Bike`).
  2. Выведите их на экран (чтобы сработал `__str__`).
  3. Проверьте работу методов `move()` и `honk()`.
  4. Используйте `len(car)` для машины.
  5. Сравните две машины (`car1 == car2`).
  6. Сложите скорости двух машин (`car1 + car2`).
  7. Сложите машину и велосипед — что произойдёт? (объяснить поведение).
- 

## 5. Дополнительное задание

Создайте список объектов (`Transport`, `Car`, `Bike`) и с помощью цикла `for` вызовите у каждого метод `move()`.

Подумайте: какой принцип ООП тут демонстрируется?