

## Л.Р. №8 (Лабораторная работа №8) 25.11

### Lab\_8.1

Создайте базовый класс Vehicle, у которого есть:

- метод `__init__`, принимающий название транспортного средства, пробег и вместимость. Их необходимо сохранить в атрибуты экземпляра `name`, `mileage` и `capacity` соответственно
- метод `fare`, который возвращает стоимость проезда из расчета `capacity * 100`:
- метод `display`, который печатает строку следующего вида: «Total name - <name>, mileage - <mileage>fare is: <метод fare>»

Затем создайте подкласс Bus, унаследованный от Vehicle. В нем необходимо:

- переопределить метод `__init__`. Он должен принимать два значения: название транспортного средства и пробег. Необходимо делегировать создание атрибутов `name`, `mileage` и `capacity` базовому классу, в качестве аргумента передайте `capacity` значение 50
- переопределить метод `fare`. Он должен получить стоимость проезда у родительского класса и увеличить ее на 10%.

После создайте подкласс Taxi, унаследованный от Vehicle. В нем необходимо:

- переопределить метод `__init__`. Он должен принимать два значения: название транспортного средства и пробег. Необходимо делегировать создание атрибутов `name`, `mileage` и `capacity` базовому классу, в качестве аргумента передайте `capacity` значение 4
- переопределить метод `fare`. Он должен получить стоимость проезда у родительского класса и увеличить ее на 35%.

```
sc = Vehicle('Scooter', 100, 2)
sc.display()
merc = Bus("Mercedes", 120000)
merc.display()
polo = Taxi("Volkswagen Polo", 15000)
polo.display()
```

### Lab\_8.2

В этой задаче один родительский класс Transport и три дочерних класса: Car, Boat, Plane.

В классе Transport должны быть реализованы:

- метод `__init__`, который создает атрибуты `brand`, `max_speed` и `kind`. Значения атрибутов `brand`, `max_speed`, `kind` поступают при вызове метода `__init__`. При этом значение `kind` не является обязательным и по умолчанию имеет значение `None`;
- метод `__str__`, который будет возвращать строку формата: "Тип транспорта <kind> марки <brand> может развить скорость <максимальная скорость> км/ч".

В классе Car должны быть реализованы:

- метод `__init__`, создающий у экземпляра атрибуты `brand`, `max_speed`, `mileage`, приватный атрибут `gasoline_residue` и `kind= "Car"`. Все значения этих атрибутов передаются при вызове класса `Car`. Внутри инициализации делегируйте создание атрибутов `brand`, `max_speed`, `kind` родительскому классу `Transport`;
- свойство-геттер `gasoline`, который будет возвращать строку: "Осталось бензина на `<gasoline_residue>` км";
- свойство-сеттер `gasoline`, которое должно принимать ТОЛЬКО целое число `value`, увеличивает уровень топлива `gasoline_residue` на переданное значение и затем вывести фразу 'Объем топлива увеличен на `<value>` л и составляет `<gasoline_residue>` л'. Если в значение `value` подается не целое число, вывести 'Ошибка заправки автомобиля'.

В классе `Boat` должны быть реализованы:

- метод `__init__`, принимающий три обязательных значения `brand`, `max_speed`, `owners_name` и один необязательный `kind="Boat"`. Их нужно сохранить в соответствующие атрибуты. При этом внутри инициализации нужно делегировать создание атрибутов `brand`, `max_speed`, `kind` родительскому классу `Transport`;
- метод `__str__`, который будет возвращать строку: 'Этой лодкой марки `<brand>` владеет `<owners_name>`'.

В классе `Plane` должны быть реализованы:

- метод `__init__`, создающий у экземпляра атрибуты `brand`, `max_speed`, `capacity`, `kind = "Plane"`. Внутри инициализации делегируйте создание атрибутов `brand`, `max_speed`, `kind` родительскому классу `Transport`;
- метод `__str__`, который будет возвращать строку: 'Самолет марки `<brand>` вмещает в себя `<capacity>` людей'.

```
transport = Transport('Telega', 10)
print(transport) # Тип транспорта None марки Telega может развить скорость 10 км/ч
bike = Transport('shkolnik', 20, 'bike')
print(bike) # Тип транспорта bike марки shkolnik может развить скорость 20 км/ч
first_plane = Plane('Virgin Atlantic', 700, 450)
print(first_plane) # Самолет марки Virgin Atlantic вмещает в себя 450 людей
first_car = Car('BMW', 230, 75000, 300)
print(first_car) # Тип транспорта Car марки BMW может развить скорость 230 км/ч
print(first_car.gasoline) # Осталось бензина на 300 км
first_car.gasoline = 20 # Печатает 'Объем топлива увеличен на 20 л и составляет 320 л'
print(first_car.gasoline) # Осталось бензина на 320 км
second_car = Car('Audi', 230, 70000, 130)
second_car.gasoline = [None] # Печатает 'Ошибка заправки автомобиля'
first_boat = Boat('Yamaha', 40, 'Petr')
print(first_boat) # Этой лодкой марки Yamaha владеет Petr
```