

# Неделя-3

## 3-2: Классы и наследование

Как правило задачи про классы не носят вычислительный характер. Обычно нужно написать классы, которые отвечают определенным интерфейсам. Насколько удобны эти интерфейсы и как сильно связаны классы между собой, определит легкость их использования в будущих программах.

Предположим есть данные о разных автомобилях и спецтехнике. Данные представлены в виде таблицы с характеристиками. Вся техника разделена на три вида: спецтехника, легковые и грузовые автомобили. Обратите внимание на то, что некоторые характеристики присущи только определенному виду техники. Например, у легковых автомобилей есть характеристика «кол-во пассажирских мест», а у грузовых автомобилей — габариты кузова: «длина», «ширина» и «высота».

Тип (car_type)	Марка (brand)	Кол-во пассажирски х мест (passenger_se ats_count)	Фото (photo_fil e_name)	Кузов ДхШх В, м (body_ whl)	Грузоподъе мость, Тонн (carrying)	Дополните льно (extra)
car	Nissan xTtrail	4	f1.jpeg		2.5	
truck	Man		f2.jpeg	8x3x2.5	20	
car	Mazda 6	4	f3.jpeg		2.5	
spec_mac hine	Hitachi		f4.jpeg		1.2	Легкая техника для уборки снега

Вам необходимо создать свою иерархию классов для данных, которые описаны в таблице. Классы должны называться **CarBase** (базовый класс для всех типов машин), **Car** (легковые автомобили), **Truck** (грузовые автомобили) и **SpecMachine** (спецтехника). Все объекты имеют обязательные атрибуты:

- **car\_type**, значение типа объекта и может принимать одно из значений: «car», «truck», «spec\_machine».
- **photo\_file\_name**, имя файла с изображением машины, допустимы названия файлов изображений с расширением из списка: «.jpg», «.jpeg», «.png», «.gif»
- **brand**, марка производителя машины
- **carrying**, грузоподъемность

В базовом классе **CarBase** нужно реализовать метод **get\_photo\_file\_ext** для получения расширения файла изображения. Расширение файла можно получить при помощи `os.path.splitext`.

Для грузового автомобиля необходимо в конструкторе класса определить атрибуты: **body\_length**, **body\_width**, **body\_height**, отвечающие соответственно за габариты кузова — длину, ширину и высоту. Габариты передаются в параметре **body\_whl** (строка, в которой размеры разделены латинской буквой «x»). Обратите внимание на то, что характеристики кузова должны быть вещественными числами и характеристики кузова могут быть не валидными (например, пустая строка). В таком случае всем атрибутам, отвечающим за габариты кузова, присваивается значение равное нулю.

Также для класса грузового автомобиля необходимо реализовать метод **get\_body\_volume**, возвращающий объем кузова.

В классе **Car** должен быть определен атрибут **passenger\_seats\_count** (количество пассажирских мест), а в классе **SpecMachine** — **extra** (дополнительное описание машины).

Полная информация о атрибутах классов приведена в таблице ниже, где **1** - означает, что атрибут обязателен для объекта, **0** - атрибут должен отсутствовать.

	Car	Truck	SpecMachine
car_type	1	1	1
photo_file_name	1	1	1
brand	1	1	1
carrying	1	1	1
passenger_seats_count	1	0	0
body_width	0	1	0
body_height	0	1	0
body_length	0	1	0
extra	0	0	1

Обратите внимание, что у каждого объекта из иерархии должен быть свой набор атрибутов и методов. Например, у класса легковой автомобиль не должно быть метода **get\_body\_volume** в отличие от класса грузового автомобиля. Имена атрибутов и методов должны совпадать с теми, что описаны выше.

Далее вам необходимо реализовать функцию **get\_car\_list**, на вход которой подается имя файла в формате **csv**. Файл содержит данные, аналогичные строкам из таблицы. Вам необходимо прочитать этот файл построчно при помощи модуля стандартной библиотеки **csv**. Затем проанализировать строки на валидность и создать список объектов с автомобилями и специальной техникой. Функция должна возвращать список объектов.

Вы можете использовать для отладки работы функции **get\_car\_list** следующий csv-файл:

```
coursera_week3_cars.csv
```

Первая строка в исходном файле — это заголовок **csv**, который содержит имена колонок. Нужно пропустить первую строку из исходного файла. Обратите внимание на то, что в некоторых строках исходного файла , данные могут быть заполнены некорректно, например, отсутствовать обязательные поля или иметь не валидное значение. В таком случае нужно проигнорировать подобные строки и не создавать объекты. Строки с пустым или не валидным значением для **body\_whl** игнорироваться не должны. Вы можете

использовать стандартный механизм обработки исключений в процессе чтения, валидации и создания объектов из строк csv-файла. Проверьте работу вашего кода с входным файлом, прежде чем загружать задание для оценки.

Пример кода, демонстрирующего чтение csv файла:

```
import csv
with open(csv_filename) as csv_fd:
    reader = csv.reader(csv_fd, delimiter=';')
    next(reader) # пропускаем заголовок
    for row in reader:
        print(row)
```

Ниже приведен шаблон кода для выполнения задания:

```
class CarBase:
    def __init__(self, brand, photo_file_name, carrying):
        pass
class Car(CarBase):
    def __init__(self, brand, photo_file_name, carrying,
                 passenger_seats_count):
        pass
class Truck(CarBase):
    def __init__(self, brand, photo_file_name, carrying, body_whl):
        pass
class SpecMachine(CarBase):
    def __init__(self, brand, photo_file_name, carrying, extra):
        pass
def get_car_list(csv_filename):
    car_list = []
    return car_list
```

Несколько примеров работы:

```
>>> from solution import *
>>> cars = get_car_list('cars_week3.csv')
>>> len(cars)
4
>>> for car in cars:
...     print(type(car))
...
<class 'solution.Car'>
<class 'solution.Truck'>
<class 'solution.Truck'>
<class 'solution.Car'>
>>> cars[0].passenger_seats_count
4
>>> cars[1].get_body_volume()
60.0
>>> >>> from solution import *
>>> cars = get_car_list('cars_week3.csv')
>>> len(cars)
4
>>> for car in cars:
...     print(type(car))
...
<class 'solution.Car'>
<class 'solution.Truck'>
<class 'solution.Truck'>
<class 'solution.Car'>
>>> cars[0].passenger_seats_count
4
>>> cars[1].get_body_volume()
60.0
>>>
```

Успехов! Часто задаваемые вопросы и полезные ссылки по данному заданию - [FAQ](#).