

Programming Assignment: Классы и наследование

✓ Passed · 1/1 points

Deadline Pass this assignment by March 18, 12:59 AM PDT

Instructions My submission Discussions

Классы и наследование

Как правило задачи про классы носят не 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. Обратите внимание на то, что некоторые колонки присущи только легковым автомобилям, например, кол-во пассажирских мест. В свою очередь только у грузовых автомобилей есть длина, ширина и высота кузова.

Тип (car_type)	Марка (brand)	Кол-во пассажирских мест (passenger_seats_count)	Фото (photo_file_name)	Кузов ДхШхВ, м (body_whl)	Грузоподъемность, Тонн (carrying)	Дополнитель (extra)
car	Nissan xTrail	4	f1.jpeg		2.5	
truck	Man		f2.jpeg	8x3x2.5	20	
car	Mazda 6	4	f3.jpeg		2.5	
spec_machine	Hitachi		f4.jpeg		1.2	Легкая техни для уборки снега

Вам необходимо создать свою иерархию классов для данных, которые описаны в таблице.

BaseCar

Car(BaseCar)

Truck(BaseCar)

SpecMachine(BaseCar)

У любого объекта есть обязательный атрибут car_type. Он означает тип объекта и может принимать одно из значений: car, truck, spec_machine.

Также у любого объекта из иерархии есть фото в виде имени файла — обязательный атрибут photo_file_name.

В базовом классе нужно реализовать метод get_photo_file_ext для получения расширения файла (".png", ".jpeg" и т.д.) с фото. Расширение файла можно получить при помощи os.path.splitext.

Для грузового автомобиля необходимо разделить характеристики кузова на отдельные составляющие body_length, body_width, body_height. Разделитель — латинская буква x. Характеристики кузова могут быть заданы в виде пустой строки, в таком случае все составляющие равны 0. Обратите внимание на то, что характеристики кузова должны быть вещественными числами.

Также для класса грузового автомобиля необходимо реализовать метод get_body_volume, возвращающий объем кузова в метрах кубических.

Все обязательные атрибуты для объектов Car, Truck и SpecMachine перечислены 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

Далее необходимо реализовать функцию, 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 и создать список нужных объектов с автомобилями и специальной техникой. Функция должна возвращать список объектов.

Не важно как вы назовете свои классы, главное чтобы их атрибуты имели имена:

```
1 car_type
2 brand
3 passenger_seats_count
4 photo_file_name
5 body_width
6 body_height
7 body_length
8 carrying
9 extra
```

И методы:

get_photo_file_ext и get_body_volume

У каждого объекта из иерархии должен быть свой набор атрибутов и методов. У класса легковой автомобиль не должно быть метода get_body_volume в отличие от класса грузового автомобиля.

Функция, которая парсит строки входного массива, должна называться get_car_list. Для проверки работы своей реализации функции get_car_list и всех созданных классов вам необходимо использовать 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 исходный файл с данными содержит некорректные строки, которые нужно пропустить. Если возникают исключения в процессе создания объектов из строк csv-файла, то требуется их корректно обработать стандартным способом. Проверьте работу вашего кода с входным файлом, прежде чем загружать задание для оценки.

Пример чтения csv файла:

```
1 import csv
2
3 with open(csv_filename) as csv_fd:
4     reader = csv.reader(csv_fd, delimiter=',')
5     next(reader) # пропускаем заголовок
6     for row in reader:
7         print(row)
```

Также обратите внимание, что все значения в csv файле при чтении будут python-строками. Нужно преобразовать строку в int для passenger_seats_count, во float для carrying, а также во float для body_width body_height, body_length.

Также ваша программа должна быть готова к тому, что в некоторых строках данные могут быть заполнены некорректно. Например, число колонок меньше . В таком случае нужно проигнорировать подобные строки и не создавать объекты. Строки с пустым значением для body_whl игнорироваться не должны. Вы можете использовать механизм исключений для обработки ошибок.

Ниже приведен пример с заготовкой кода для выполнения задания.

```
1 class CarBase:
2     def __init__(self, brand, photo_file_name, carrying):
3         pass
4
5
6 class Car(CarBase):
7     def __init__(self, brand, photo_file_name, carrying,
8                 passenger_seats_count):
9         pass
10
11 class Truck(CarBase):
12     def __init__(self, brand, photo_file_name, carrying, body_whl):
13         pass
14
15
16 class SpecMachine(CarBase):
17     def __init__(self, brand, photo_file_name, carrying, extra):
18         pass
19
20
21 def get_car_list(csv_filename):
22     car_list = []
23     return car_list
```

Вам необходимо расширить функционал исходных классов, дополнить методы нужным кодом и реализовать функцию `get_car_list`.

Успехов!

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.

