

4.2 Функция

`issubclass()`.

Наследование 6 out of 10 steps passed 10 out of 20 points received
от встроенных типов

Добрый, добрый Python
ООП - обучающий курс от
Сергея Балакирева (/course
/116336)

Course progress: 356/695

1 Первые шаги в ООП

Видео-разбор подвига (решение смотреть только после своей попытки):
<https://youtu.be/aFKqv4b2QZ0> (<https://youtu.be/aFKqv4b2QZ0>)

Подвиг 4. Разрабатывается интернет-магазин. Каждый товар предполагается представлять классом `Thing`, объекты которого создаются командой:

```
thing = Thing(name, price, weight)
```

где name - наименование товара (строка); price - цена (вещественное число); weight - вес товара (вещественное число). В каждом объекте этого класса создаются аналогичные атрибуты: `name`, `price`, `weight`.

Класс `Thing` необходимо определить так, чтобы его объекты можно было использовать в качестве ключей словаря, например:

```
d = {}  
d[thing] = thing
```

И для каждого уникального набора данных `name`, `price`, `weight` должны формироваться свои уникальные ключи.

Затем, вам необходимо объявить класс словаря `DictShop`, унаследованный от базового класса `dict`. В этом новом словаре ключами могут выступать только объекты класса `Thing`. При попытке указать любой другой тип, генерировать исключение командой:

```
raise TypeError('ключами могут быть только объекты класса Thing')
```

Объекты класса `DictShop` должны создаваться командами:

```
dict_things = DictShop() # пустой словарь  
dict_things = DictShop(things) # словарь с набором словаря things
```

где `things` - некоторый словарь. В инициализаторе следует проверять,

2 Режимы доступа, свойс...

3 Магические методы кла...

3.7 М... (/lesson



чтобы аргумент thing был словарем, если не так, то выбрасывать исключение:

```
raise TypeError('аргумент должен быть словарем')
```

И проверять, чтобы все ключи являлись объектами класса Thing. Если это не так, то генерировать исключение:

```
raise TypeError('ключами могут быть только объекты класса Thing')
```

Дополнительно в классе DictShop переопределить метод:

```
__setitem__()
```

с проверкой, что создаваемый ключ является объектом класса Thing. Иначе, генерировать исключение:

```
raise TypeError('ключами могут быть только объекты класса Thing')
```

Пример использования классов (эти строчки в программе не писать):

```
th_1 = Thing('Лыжи', 11000, 1978.55)
th_2 = Thing('Книга', 1500, 256)
dict_things = DictShop()
dict_things[th_1] = th_1
dict_things[th_2] = th_2

for x in dict_things:
    print(x.name)

dict_things[1] = th_1 # исключение TypeError
```

P.S. В программе нужно объявить только классы. На экран выводить ничего не нужно.

To solve this problem please visit
<https://stepik.org/lesson/701996/step/5>

Добрый, добрый Python
ООП - обучающий курс от
Сергея Балакирева (/course
/116336)

Course progress: 356/695

1 Первые шаги в ООП

2 Режимы доступа, свойс...

3 Магические методы кла...

3.7 M... (/lesson

