

2.3 Дескрипторы (data descriptor и non-data descriptor)

10 из 11 шагов пройдено 17 из 20 баллов получено

Видео-разбор подвига (решение смотреть только после своей попытки):

<https://youtu.be/S4CDXCG9nbA> (<https://youtu.be/S4CDXCG9nbA>)

Подвиг 9 (на повторение). Необходимо объявить класс Bag (рюкзак), объекты которого будут создаваться командой:

```
bag = Bag(max_weight)
```

где `max_weight` - максимальный суммарный вес вещей, который выдерживает рюкзак (целое число).

В каждом объекте этого класса должен создаваться локальный приватный атрибут:

`__things` - список вещей в рюкзаке (изначально список пуст).

Сам же класс Bag должен иметь объект-свойство:

`things` - для доступа к локальному приватному атрибуту `__things` (только для считывания, не записи).

Также в классе Bag должны быть реализованы следующие методы:

`add_thing(self, thing)` - добавление нового предмета в рюкзак (добавление возможно, если суммарный вес (`max_weight`) не будет превышен, иначе добавление не происходит);

`remove_thing(self, indx)` - удаление предмета по индексу списка `__things`;

`get_total_weight(self)` - возвращает суммарный вес предметов в рюкзаке.

Каждая вещь описывается как объект класса Thing и создается командой:

```
t = Thing(название, вес)
```

где название - наименование предмета (строка); вес - вес предмета (целое или вещественное число).

В каждом объекте класса Thing должны формироваться локальные атрибуты:

`name` - наименование предмета;

`weight` - вес предмета.

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

```
bag = Bag(1000)
bag.add_thing(Thing("Книга по Python", 100))
bag.add_thing(Thing("Котелок", 500))
bag.add_thing(Thing("Спички", 20))
bag.add_thing(Thing("Бумага", 100))
w = bag.get_total_weight()
for t in bag.things:
    print(f"{t.name}: {t.weight}")
```

P.S. В программе требуется объявить классы с описанным функционалом. На экран в программе выводить ничего не нужно.

Чтобы решить это задание откройте
<https://stepik.org/lesson/701985/step/10>