

3.4 Методы `__add__`, `__sub__`, `__mul__`,
`__truediv__`

5 out of 11 steps passed 7 out of 25 points received

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

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

Подвиг 4. Известно, что в Python мы можем соединять два списка между собой с помощью оператора `+`:

```
lst = [1, 2, 3] + [4.5, -3.6, 0.78]
```

Но нет реализации оператора `-`, который бы убирал из списка все значения вычитаемого списка:

```
lst = [1, 2, 3, 4, 5, 6] - [5, 6, 7, 8, 1] # [2, 3, 4] (порядок следования  
оставшихся элементов списка должен сохраняться)
```

Давайте это поправим и создадим такой функционал. Для этого нужно объявить класс с именем `NewList`, объекты которого создаются командами:

```
lst = NewList() # пустой список  
lst = NewList([-1, 0, 7.56, True]) # список с начальными значениями
```

Реализуйте для этого класса работу с оператором вычитания, чтобы над объектами класса `NewList` можно было выполнять следующие действия:

```
lst1 = NewList([1, 2, -4, 6, 10, 11, 15, False, True])  
lst2 = NewList([0, 1, 2, 3, True])  
res_1 = lst1 - lst2 # NewList: [-4, 6, 10, 11, 15, False]  
lst1 -= lst2 # NewList: [-4, 6, 10, 11, 15, False]  
res_2 = lst2 - [0, True] # NewList: [1, 2, 3]  
res_3 = [1, 2, 3, 4.5] - res_2 # NewList: [4.5]  
a = NewList([2, 3])  
res_4 = [1, 2, 2, 3] - a # NewList: [1, 2]
```

Также в классе `NewList` необходимо объявить метод:

`get_list()` - для возвращения результирующего списка объекта класса `NewList`

Например:

```
lst = res_2.get_list() # [1, 2, 3]
```

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

To solve this problem please visit
<https://stepik.org/lesson/701989/step/5>