

Задания

1. Используя class decorators (<http://docs.python.org/3.0/whatsnew/2.6.html#pep-3129>) создайте декоратор @Singleton, который будучи приписан к классу превращает его в класс Singleton, т.е. экземпляр этого класса создается только один раз, а при всех остальных вызовах конструктора возвращается существующий экземпляр.
2. Напишите скрипт, который принимает на вход имя файла написанного на Python, после чего декорирует каждый метод в этом файле таким образом, чтобы при запуске любого метода выдавалась информация вида:

```
13.126ms foo
    +0.472ms
```

Здесь 13.126 – время прошедшее с запуска программы, 0.472 – время работы функции, foo – имя функции (доступно через поле `__name__`).

После декорирования файл необходимо выполнить. Это можно сделать, например, используя функцию `exec` (<http://docs.python.org/3.0/library/functions.html#exec>)

Задания

3. Словари:

- Создайте наследника класса словаря
- Используя операции `getattr` и `setattr` добейтесь того, чтобы к элементам словаря можно было обращаться `d["key"]` и `d.key`. Если в названии ключа есть пробелы, то замените их на символ подчеркивания.
- Сделайте так, чтобы при конструировании словарь получал кортеж из «запрещенных» ключей. При попытке создать элемент с таким ключом должно порождаться исключение `DeniedKeyException`
- Добейтесь того, чтобы можно было обращаться к элементам словаря по номеру, т.е. `d[0]` должно возвращать значение первого добавленного в словарь элемента.

Задания

4. Матрицы:

- Создать класс матриц 3×3
- Создать класс трехмерных векторов.
- Создать наследника класса матриц – класс матриц поворота вокруг оси Z.
- Перегрузить операции:
 - Сложения матриц
 - Умножения матриц на число
 - Умножения матрицы на вектор
 - Умножения матрицы на другую матрицу
 - Взятия обратной матрицы (операцию \sim)