



Сдать решение задачи D

Полный балл:	2
Бонусные баллы:	
Ограничение времени:	1 с
Ограничение реального времени:	5 с
Ограничение памяти:	64М

Счетчики

Напишите декоратор @counter, который позволял бы посчитать глубину рекурсии функции и количество рекурсивных вызовов функции.

Декоратор должен создать два атрибута для функции:

- ncalls — число вызовов функции;
- rdepth — глубина рекурсии.

Счётчики ncalls и rdepth должны обнуляться при каждом новом входе в рекурсию, см. примеры.

Примеры

Входные данные

```
@counter
def func1():
    return

if __name__ == "__main__":
    func1()
    print(func1.ncalls, func1.rdepth)

    func1()
    print(func1.ncalls, func1.rdepth)
```

Результат работы

```
1 1
1 1
```

Входные данные

```
@counter
def func2(n, steps):
    if steps == 0:
        return

    func2(n + 1, steps - 1)
    func2(n - 1, steps - 1)

if __name__ == "__main__":
    func2(0, 5)
    print(func2.ncalls, func2.rdepth)

    func2(0, 3)
    print(func2.ncalls, func2.rdepth)
```

Результат работы

```
63 6
15 4
```

Примечания

Не забудьте перекрыть имя декорируемой функции через декоратор из модуля functools.

Функция, обернутая в декоратор, должна возвращать то же значение, что и функция без декоратора. В задаче НЕ гарантируется, что оборачиваемая функция не принимает на вход аргументов.

Сдать решение

Язык: python3 - Python3 3.6.5

Файл No file chosen

Отправить!

Предыдущие решения этой задачи

Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть исходный текст	Посмотреть протокол
706	147:57:38	449	D	python3	ОК	4	2	Просмотр	Просмотр

A B C D E F G H I

ejudge 3.7.6 #1 (2018-08-14 14:12:28).

Copyright © 2000-2018 Alexander Chernov.