coursera

Задание по программированию: Задача по созданию модульного теста функции factorize

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

ⓘ Похоже, это ваше первое задание по программированию. Подробнее

Срок сдачи Сдайте это задание до 4 окт. г., 9:59 MSK

Инструкции Моя работа Обсуждения

Дана функция factorize(x) со следующим контрактом:

```
def factorize(x):
    """
    Factorize positive integer and return its factors.
    :type x: int,>=0
    :rtype: tuple[N],N>0
    """
    pass
```

Необходимо написать комплект тестов используя модуль **unittest** стандартной библиотеки **Python**. Имя тестового класса - **TestFactorize**.

Описание тестов:

1. **test_wrong_types_raise_exception** - проверяет, что передаваемый в функцию аргумент типа **float** или **str** вызывает исключение **TypeError**. Тестовый набор входных данных: *'string', 1.5*
2. **test_negative** - проверяет, что передача в функцию **factorize** отрицательного числа вызывает исключение **ValueError**. Тестовый набор входных данных: *-1, -10, -100*
3. **test_zero_and_one_cases** - проверяет, что при передаче в функцию целых чисел 0 и 1, возвращаются соответственно кортежи (0,) и (1,). Набор тестовых данных: *0 → (0,), 1 → (1,)*
4. **test_simple_numbers** - что для простых чисел возвращается кортеж, содержащий одно данное число. Набор тестовых данных: *3 → (3,), 13 → (13,), 29 → (29,)*
5. **test_two_simple_multipliers** — проверяет случаи, когда передаются числа для которых функция **factorize** возвращает кортеж с числом элементов равным 2. Набор тестовых данных: *6 → (2, 3), 26 → (2, 13), 121 --> (11, 11)*
6. **test_many_multipliers** - проверяет случаи, когда передаются числа для которых функция **factorize** возвращает кортеж с числом элементов больше 2. Набор тестовых данных: *1001 → (7, 11, 13) , 9699690 → (2, 3, 5, 7, 11, 13, 17, 19)*

ВАЖНО! *Все входные данные должны быть такими, как указано в условии. Название переменной в каждом тестовом случае должно быть именно "x".* При этом несколько различных проверок в рамках одного теста должны быть обработаны как подслучаи с указанием x: subTest(x=...). *В задании необходимо реализовать ТОЛЬКО класс **TestFactorize**, кроме этого реализовывать ничего не нужно. Импортировать **unittest** и вызывать **unittest.main()** в решении также не нужно.*

Ссылка на часто задаваемые вопросы по этому заданию - FAQ.

Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.