

✓ Поздравляем! Вы прошли тест!Оценка
получена **100 %**Для успешного прохождения 80 %% или
выше[Перейти к
следующему пункту](#)

Тест на модульное тестирование

Оценка последней работы: 100%

1. Что такое Test fixture в библиотеке unittest?

1 / 1 балл

- ☐ класс, описывающий тестовый случай
- ☒ среда исполнения теста
- ☐ класс, позволяющий группировать тестовые случаи
- ☐ группа запуска тестов

✓ Правильно

2. Для того чтобы была выполнена проверка тестового случая...

1 / 1 балл

- ☒ нужно импортировать библиотеку unittest

✓ Правильно

- ☒ создать класс-потомок unittest.TestCase

✓ Правильно

- ☒ для проверки успешности прохождения теста использовать методы self.assert*

✓ Правильно

- ☒ метод, осуществляющий проверку, назвать test*

✓ Правильно

- ☒ запустить unittest.main() или создать и запустить свой TestRunner

✓ Правильно

3. В одном тестовом методе можно осуществить проверку нескольких подслучаев, получая в результатах тестирования информацию по каждой, а не только по первой не пройденной проверке.

1 / 1 балл

Для этого нужно:

- ☐ несколько раз написать вызовы методов self.assertEqual()
- ☒ вызвать метод self.subTest() с любыми именованными параметрами
- ☐ использовать логическую переменную, аккумулирующую результат всех проверок в подслучаях, и только один раз вызывать метод self.assertTrue()
- ☐ Нет, это невозможно. Отдельные случаи, даже однотипные, нужно разделять на разные методы тестирующего класса.

✓ Правильно

4. Для создания среды исполнения теста в классе тестового случая требуется создать методы:

1 / 1 балл

- ☐ setUp и tearDown
- ☐ setUp и tearDown
- ☒ setUp и tearDown

☐ tearDown и setUp

✓ Правильно

5. В классе TestFactorization, потомке TestCase, создан метод test_simple_multipliers().

1 / 1 балл

Требуется написать проверку того, что a умножить на b равно x , с использованием унаследованного от TestCase метода assertEquals.

(Обратите внимание, что импортировать библиотеку и запускать unittest.main() не требуется.)

```
1 class TestFactorization(unittest.TestCase):
2     def test_simple_multipliers(self):
3         x = 77
4         a, b = factorize(x)
5         self.assertEqual(int(a)*int(b), x)
```

Выполнить

Сбросить

✓ Правильно

Good job!