|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

|  |
| --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2** |
| **по дисциплине** |
| **«Тестирование и верификация программного обеспечения»** |

|  |  |
| --- | --- |
| Выполнил:  студент группы ИКБО-36-22 | Шумахер М.Е. |
| Проверил:  Доцент | Чернов Е.А. |

Москва 2024

1 Документация по программе

1. Функция reverse\_string принимает строку и возвращает ее символы в обратном порядке.

* Параметры: одна строка.
* Возвращаемое значение: строка с символами в обратном порядке.
* Как это работает: с помощью срезов строка разворачивается и возвращается.

1. Функция to\_uppercase преобразует все символы строки в верхний регистр.

* Параметры: одна строка.
* Возвращаемое значение: строка, где все символы в верхнем регистре.
* Как это работает: используется метод upper(), который преобразует каждый символ строки в заглавную версию.

1. Функция is\_substring проверяет, является ли одна строка подстрокой другой.

* Параметры: две строки, основная строка и подстрока.
* Возвращаемое значение: True, если подстрока встречается в основной строке, иначе False.
* Как это работает: используется оператор in для проверки наличия подстроки в строке.

1. Функция concatenate\_strings объединяет две строки и возвращает их как одну строку.

* Параметры: две строки.
* Возвращаемое значение: новая строка, являющаяся результатом конкатенации двух входных строк.
* Как это работает: используется оператор сложения для соединения двух строк.

1. Функция count\_vowels подсчитывает количество гласных в строке.

* Параметры: одна строка.
* Возвращаемое значение: целое число, которое указывает количество гласных.
* Как это работает: для каждой буквы в строке проверяется, является ли она гласной, и если да, она добавляется в счетчик.

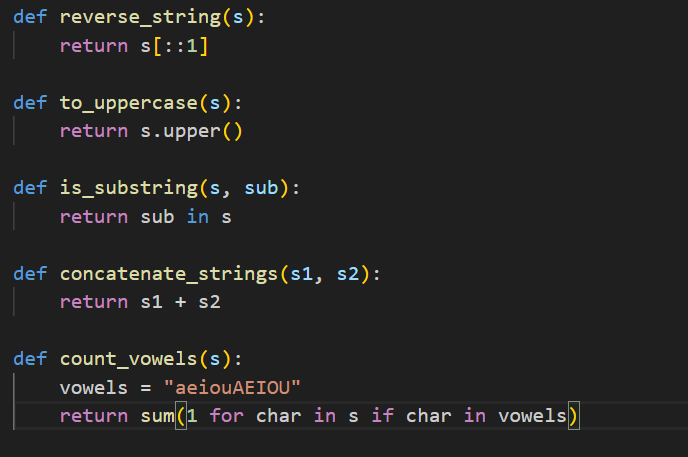


Рисунок 1 – Код программы

2 Ошибки, найденные у коллеги



Рисунок 2 – Часть кода с тестами

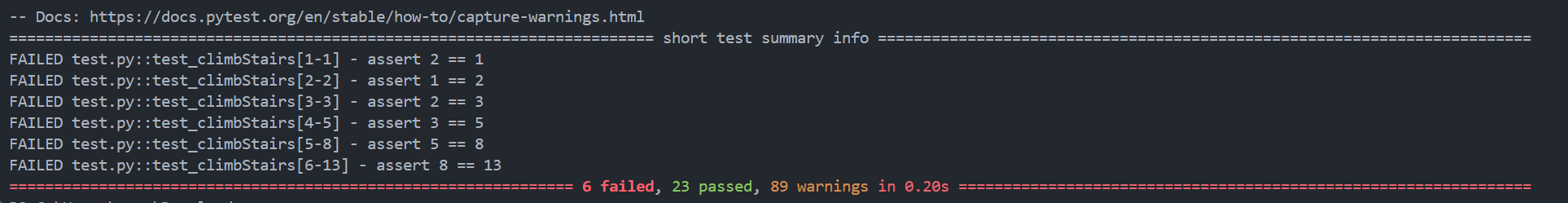


Рисунок 3 – Тесты не пройдены

**Краткое описание ошибки**: Неверное вычисление количества способов подъёма по лестнице.

**Статус ошибки**: открыта («Open»).

**Категория ошибки**: серьезная («Major»).

**Тестовый случай**: Тестирование функции climbStairs.

**Описание ошибки**:

1. Загрузить программу с функцией climbStairs.
2. Вызвать функцию с параметром n = 1.
3. Ожидаемый результат: 1.
4. Полученный результат: 2.

Другие тесты:

n=2, ожидаемый результат: 2, фактический результат: 1.

n=3, ожидаемый результат: 3, фактический результат: 2.

n=4, ожидаемый результат: 5, фактический результат: 3.

n=5, ожидаемый результат: 8, фактический результат: 5.

n=6, ожидаемый результат: 13, фактический результат: 8.

Ожидаемый результат: правильное количество способов для каждого значения n.

Исправим ошибку в программе и протестируем еще раз.

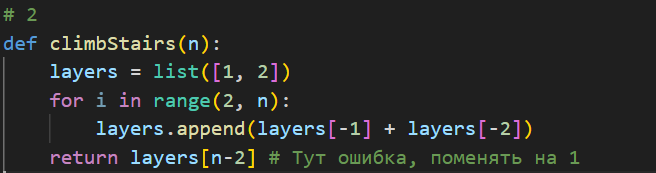


Рисунок 4 – Исправление ошибки

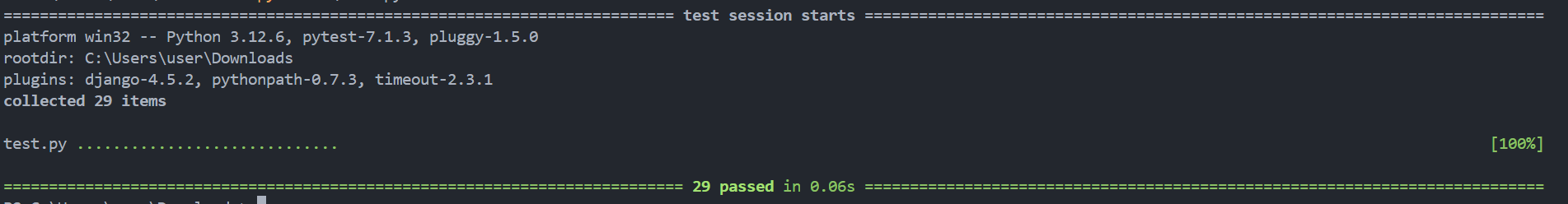


Рисунок 5 – Успешное прохождение тестов

Вывод

В ходе работы был создан простой модуль программы и написано техническое задание на программный продукт, а также подготовлена необходимая документация. Программный продукт второй члена команды был протестирован юнит-тестами. В результате тестирования были выявлены ошибки, допущенные при разработке, которые были описаны с указанием фактических результатов и описанием ошибок, затем все ошибки были исправлены. По итогам анализа сделан вывод о необходимости совершенствования процесса тестирования и более строгого соблюдения требований технического задания.