|  |
| --- |
|  |
| 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 № 5** |
| **по дисциплине** |
| **«Тестирование и верификация программного обеспечения»** |

|  |  |
| --- | --- |
| Выполнили студенты группы ИКБО-04-22 | Егоров Л.А. |
| Принял ассистент | Петрова А.А. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2024

# 1 Статический анализ

Статический анализ кода — это процесс анализа исходного кода программы без его исполнения, направленный на поиск ошибок, уязвимостей, нарушений стандартов и других потенциальных проблем. Он проводится с использованием специальных инструментов (статических анализаторов), которые могут анализировать код на разных уровнях: от синтаксического до логического.

## 1.1 Анализ кода на языке Python

Для анализа был использован код электромагнитного алгоритма, который представлен в Листинге А.1.

Для анализа кода на языке Python использовалось три анализатора – MyPy, Pylint, flake8.

MyPy – простой статический анализатор, который проверяет код на наличие синтаксических ошибок и части ошибок оформления по PEP8. Результат анализа этой программой представлен на Рисунке 1.1.1.

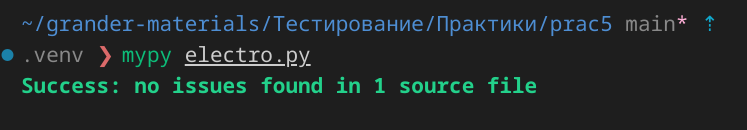


Рисунок 1.1.1 – Анализ программой MyPy

Pylint – статический анализатор, который позволяет обнаружить большое количество ошибок как синтаксических, так и касающихся оформления кода (например, отсутствие документации к функциям или использование глобальных переменных). Результат анализа этой программой представлен на Рисунке 1.1.2.

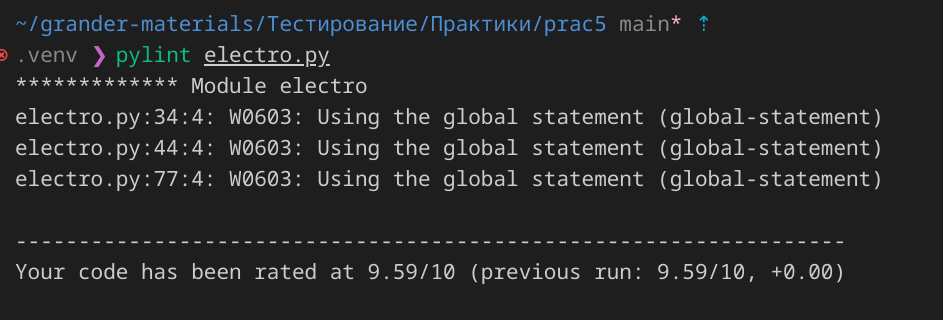


Рисунок 1.1.2 – Анализ программой pylint

Flake8 – статический анализатор, который является более легковесным по сравнению с Pylint, но также отлично подходит для нахождения ошибок в синтаксисе и в оформлении кода. Результат анализа этой программой представлен на Рисунке 1.1.3.

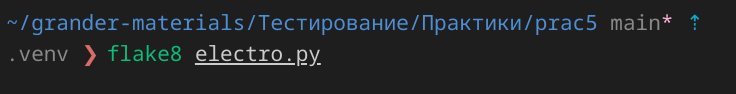


Рисунок 1.1.3 – Анализ программой flake8

### 1.1.1 Добавление ошибок в код

В код исходной программы добавлено 5 ошибок:

1. Из функций были убраны строки документации
2. Одна строка была сделана большой длины (больше 79 символов)
3. Добавлена неиспользуемая переменная
4. Добавлена неиспользуемая функция
5. Между функциями поставлено по две лишние строки

Результаты анализа представлены на Рисунках 1.1.1.1 - 1.1.1.3.

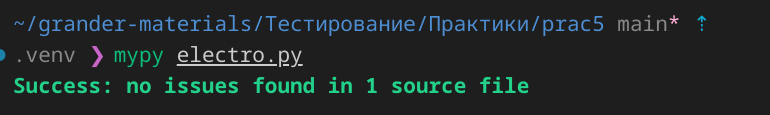


Рисунок 1.1.1.1 – Анализ программой mypy

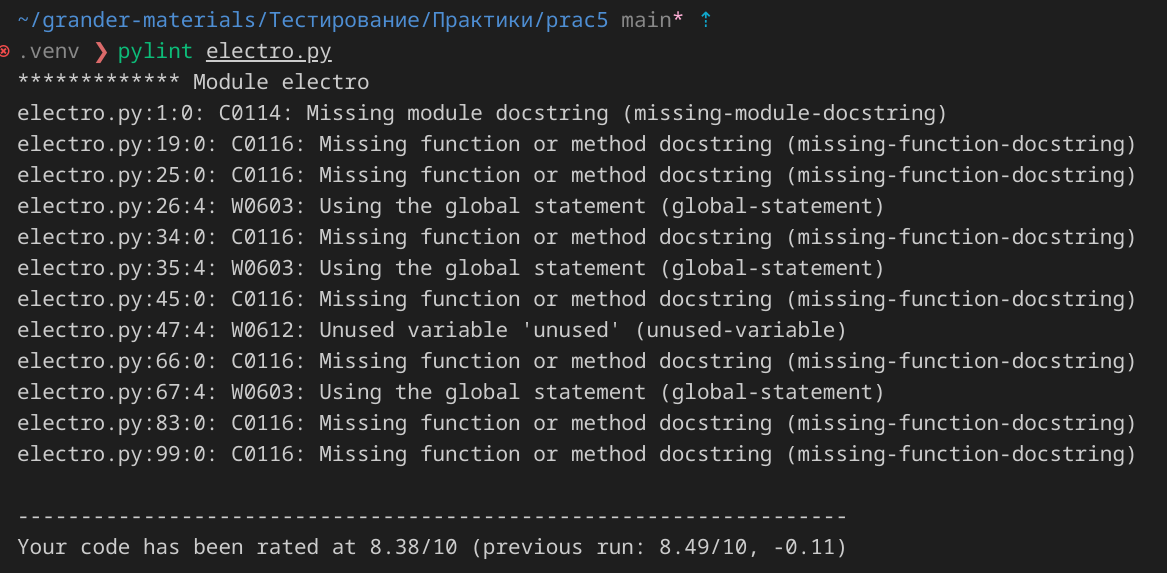


Рисунок 1.1.1.2 – Анализ программой pylint

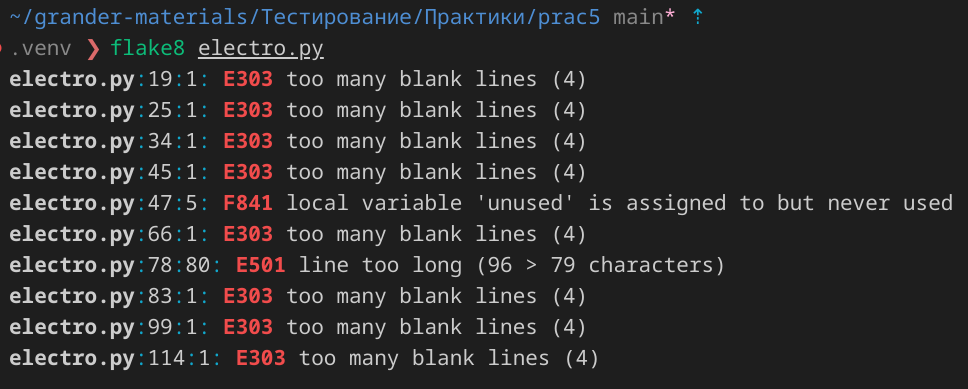


Рисунок 1.1.1.3 – Анализ программой flake8

В результате программа mypy не увидела ошибок, допущенных в коде. Pylint больше указала на стилистические ошибки, а flake8 указала те ошибки, которые допущены в оформлении по PEP8.

## 1.2 Анализ кода на языке C

Для анализа взята программа на языке C, выполняющая операцию добавления строки в матрицу. Её код представлен в Листинге А.2.

Для анализа кода используются статические анализаторы cppcheck, clang-tidy и gcc fanalyzer.

Cppcheck – простой статический анализатор, который позволяет

# 2 Динамический анализ

## 2.1 Анализ кода на языке Python

## 2.2 Анализ кода на языке C