Федеральное государственное автономное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет», ПНИПУ

ЛАБОРАТОРНАЯ РАБОТА

Конечные Автоматы

Выполнил: студент группы РИС-23-3б

Буковский Денис Владимирович

Проверила: старший преподаватель ИТАС

Рустамханова Гульшат Ильдаровна

Пермь 2024

**Постановка задачи**

Разработать программу для моделирования детерминированного конечного автомата (DFA), распознающего определенный язык. Автомат должен проверять принадлежность входных строк заданному языку.

**Функциональные требования**

* Моделирование DFA с заданной таблицей переходов
* Проверка строк на принадлежность языку автомата
* Обработка ошибок для недопустимых символов
* Тестирование на наборе входных данных

**Архитектура решения**

Программа состоит из трех модулей.

Основные модули:

* main.py - точка входа
* setup.py - конфигурация и запуск тестирования
* DFA.py - ядро системы с логикой автомата

Методы и классы

Класс DFA:

* process\_symbol() - обработка одного символа
* process\_input() - обработка всей строки
* reset() - сброс в начальное состояние
* validate\_automaton() - проверка корректности автомата

Класс DFATester:

* test\_single\_word() - тестирование одного слова
* test\_multiple\_words() - тестирование набора слов
* print\_results() - вывод результатов

Алгоритм работы

1. Инициализация автомата с заданными параметрами:
2. Начальное состояние: q0
3. Принимающие состояния: q1
4. Алфавит: {a, b, c, d}
5. Обработка входной строки посимвольно
6. Проверка конечного состояния на принадлежность принимающим
7. Валидация входных данных на соответствие алфавиту

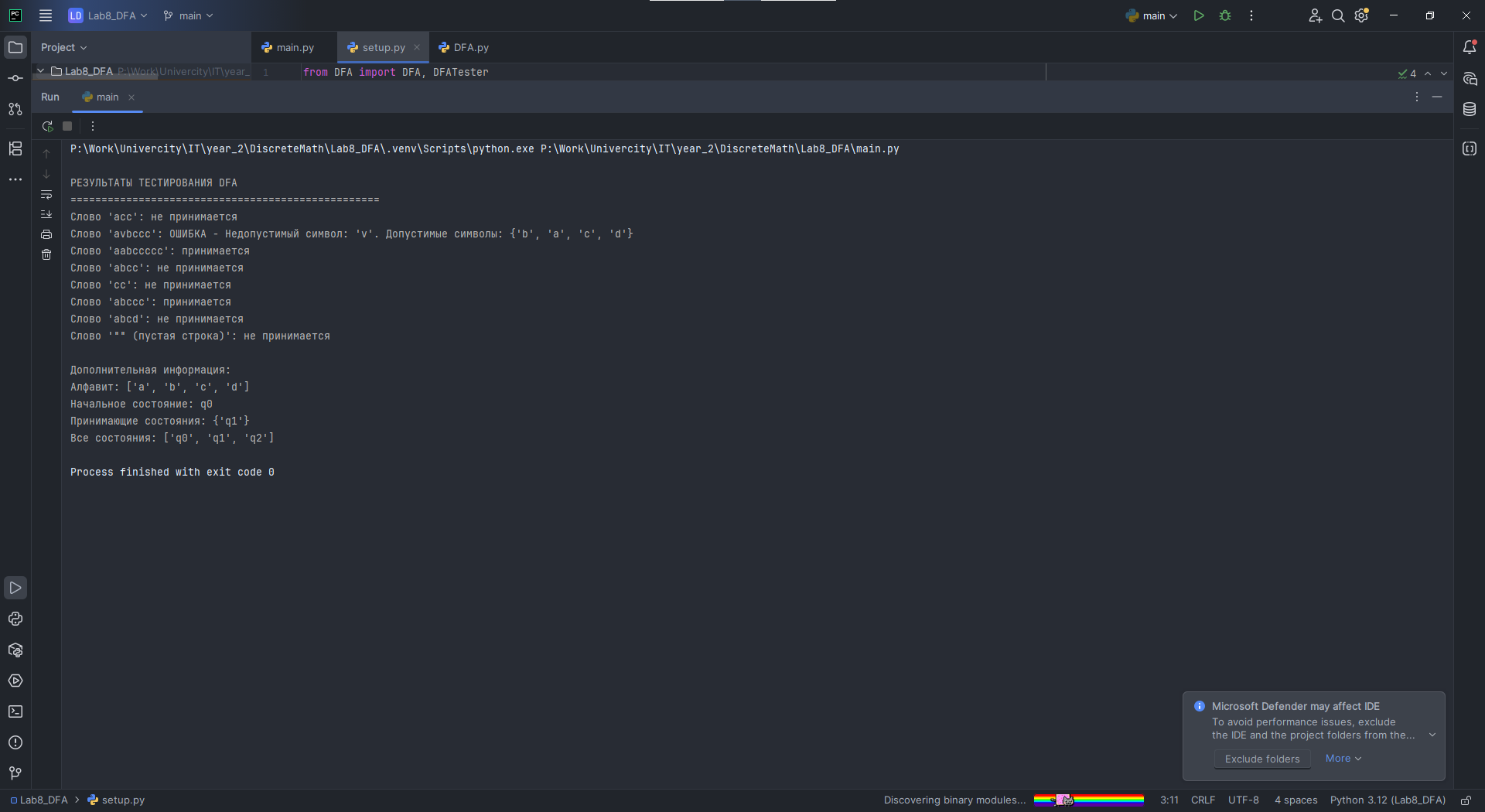
Особенности реализации

* Жесткая валидация корректности автомата при инициализации
* Обработка ошибок для недопустимых символов и переходов
* Подробное тестирование с выводом результатов
* Гибкая архитектура для легкого изменения автомата

Программа демонстрирует классический подход к реализации DFA с четким разделением ответственности между компонентами и надежной обработкой граничных случаев.

**Скриншоты**

Изображения с примерами работы программы представлены ниже



**Листинг**

Листинг программы доступен по ссылке:

<https://github.com/buksnet/DiscreteMath_Year2/tree/main/Lab8_DFA>