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

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе**

по дисциплине «Тестирование и верификация ПО»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-15-22 | Ератин Н.В. |
| **Проверил:**  Доцент | Чернов Е.А. |

2024 г.

# техническое задание

Техническое задание (ТЗ)

1. Название проекта:

Программа для анализа и шифрования текстовых файлов

2. Цель:

Создать графическое приложение для работы с текстовыми файлами, которое позволяет:

Открывать текстовые файлы и отображать их содержимое.

Подсчитывать количество строк, слов и символов в файле.

Шифровать содержимое файла с помощью методов шифрования (Цезарь и простая замена).

Сохранять зашифрованный текст в новый файл.

3. Основные функции:

3.1 Загрузка файла:

Пользователь создает файл в формате .txt на рабочем столе.

Пользователь вводит название файла в текстовое поле.

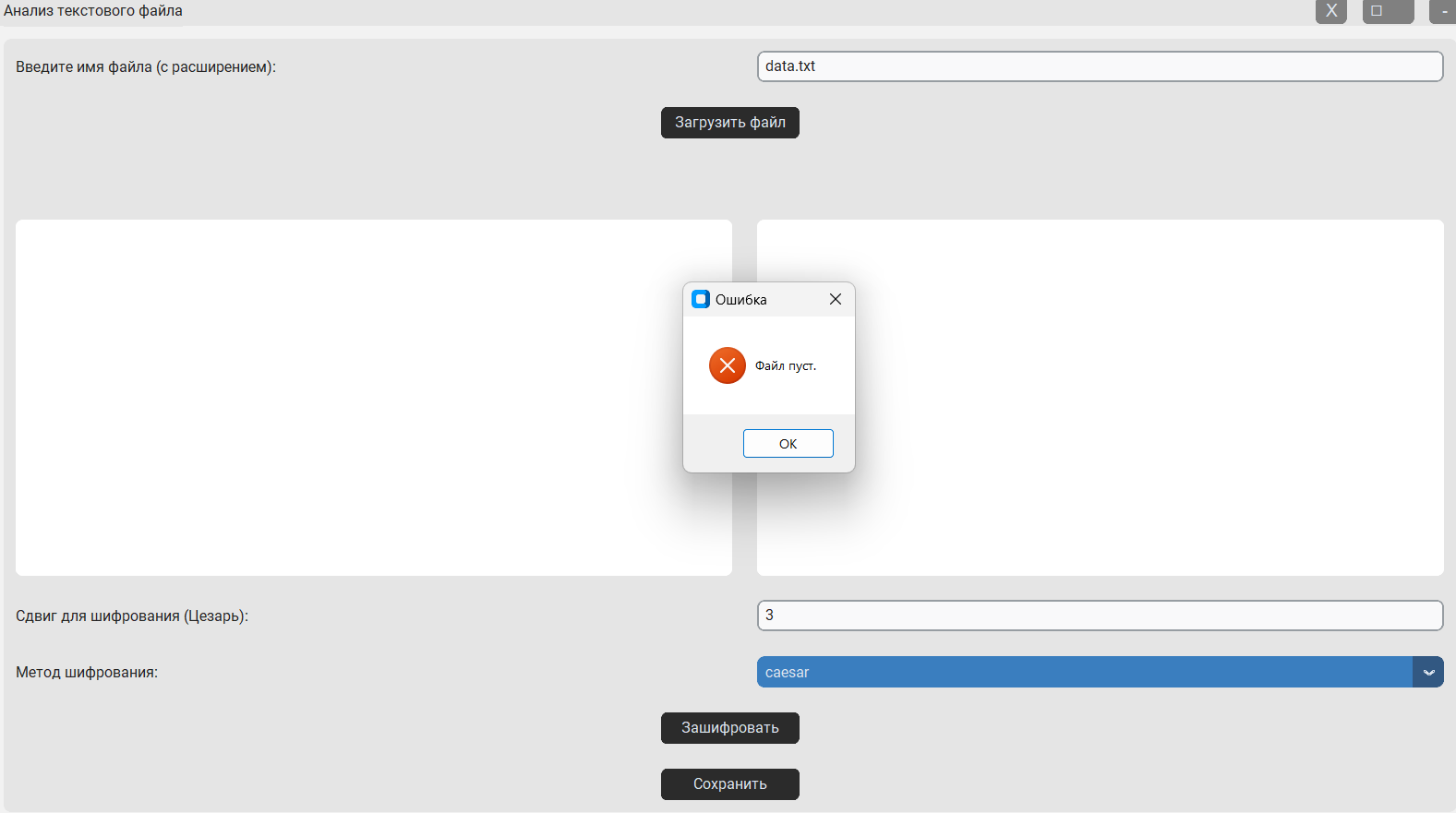


Рисунок 1 – Пример ошибки при пустом файле

Программа загружает файл, подсчитывает количество строк, слов и символов, и выводит эту информацию на экран.

Содержимое файла отображается в текстовом поле.

3.2 Методы шифрования:

Шифр Цезаря: Пользователь может задать сдвиг для шифрования текста (значение по умолчанию — 3).

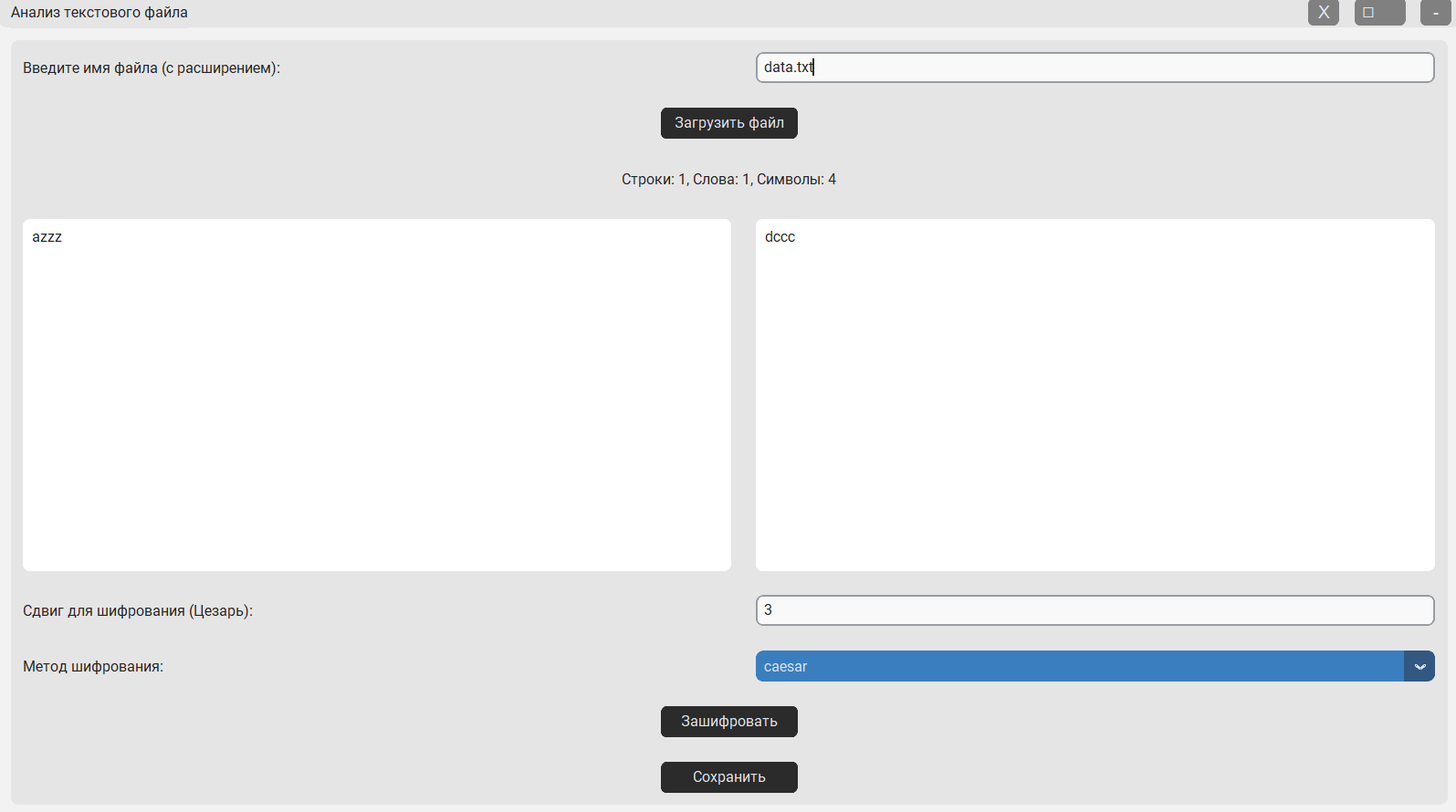


Рисунок 2 – Пример правильной работы метода шифрования

Метод замены: Каждая буква заменяется на другую (например, сдвиг на 13 позиций).

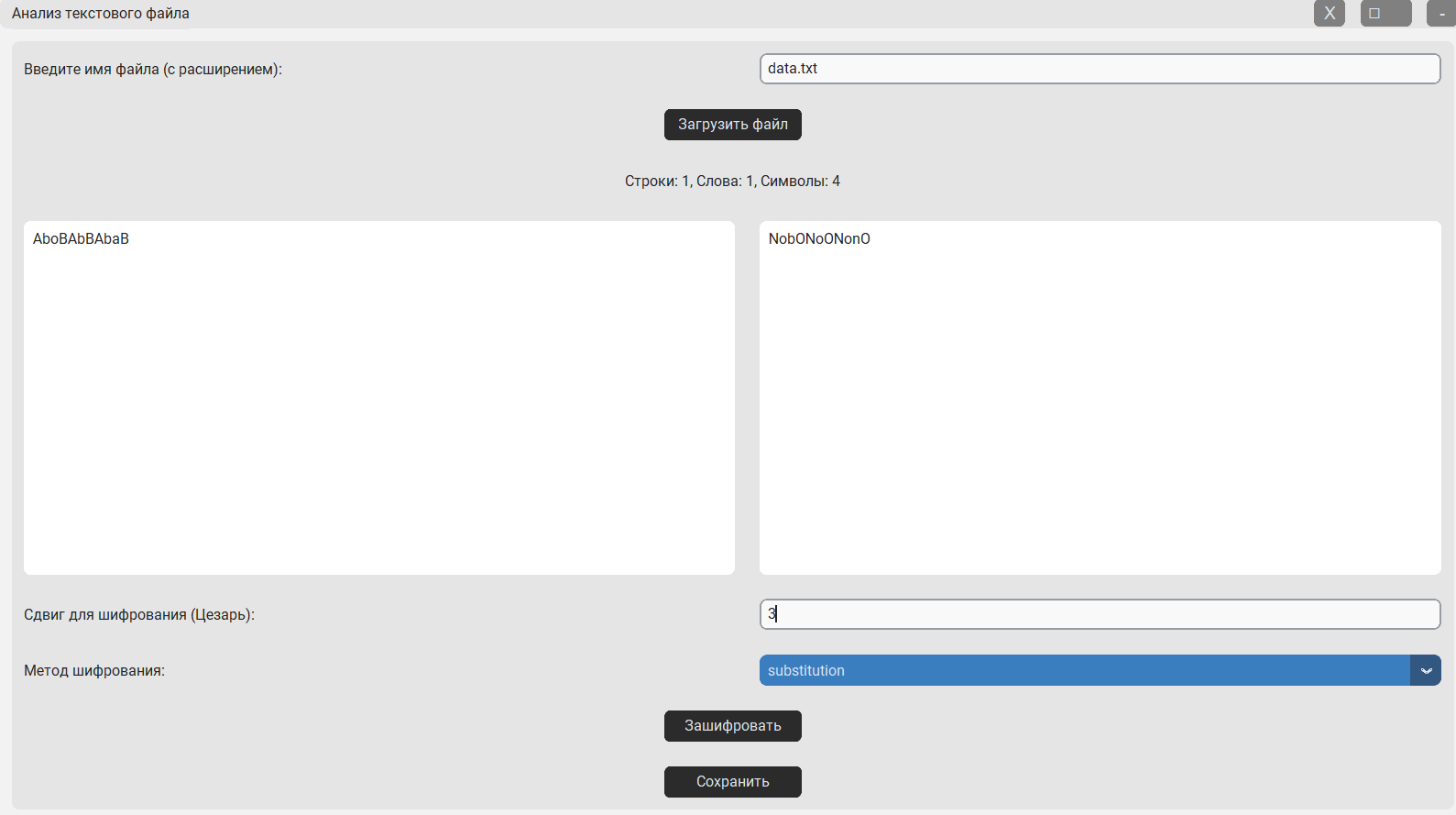


Рисунок 3 – Пример правильной работы метода шифрования

Пользователь может выбрать метод шифрования с помощью выпадающего меню.

3.3 Шифрование текста:

Программа шифрует текст, который был загружен, в зависимости от выбранного метода шифрования.

Зашифрованный текст отображается в отдельном текстовом поле.

3.4 Сохранение зашифрованного текста:

Пользователь может сохранить зашифрованный текст в новый файл через диалоговое окно.

4. Интерактивные элементы интерфейса:

4.1 Основные элементы:

Поле ввода имени файла: Текстовое поле для названия файла.

Кнопка "Загрузить файл": запускает процесс загрузки и анализа файла.

Текстовые поля:

Левое поле для отображения содержимого файла.

Правое поле для отображения зашифрованного текста.

4.2 Элементы для шифрования:

Поле ввода сдвига для шифра Цезаря: позволяет ввести целое число для сдвига шифра.

Выпадающее меню выбора метода шифрования: позволяет выбрать метод (Цезарь или замена).

Кнопка "Зашифровать": запускает процесс шифрования текста.

4.3 Элементы для сохранения:

Кнопка "Сохранить": открывает диалог для сохранения зашифрованного текста в файл.

5. Окно приложения:

Заголовок окна: "Анализ текстового файла".

Размер окна: 1280x720 пикселей.

Режим отображения: Окно всегда поверх других окон (опция -topmost).

Управление окном: Отключён системный заголовок, добавлены кнопки для сворачивания, разворачивания и закрытия окна.

6. Технические детали:

Библиотеки:

customtkinter — для создания пользовательского интерфейса.

tkinter.messagebox — для вывода сообщений об ошибках и уведомлений.

tkinter.filedialog — для открытия и сохранения файлов через диалоговые окна.

Обработка ошибок: Проверка на наличие файла.

Обработка ошибок, связанных с пустыми файлами или некорректными путями.

Обработка ошибок при вводе сдвига для шифра Цезаря.

7. Ожидаемые результаты:

Удобный графический интерфейс, позволяющий работать с текстовыми файлами.

Возможность анализа содержимого файла (количество строк, слов, символов).

Шифрование текста выбранным методом и сохранение результата.

8. Пользовательские требования:

Операционная система Windows, клавиатура, мышь.

Программа должна работать с текстовыми файлами в кодировке UTF-8.

# описание добавленных ошибок

1. Метод Цезаря неправильно шифрует данные. Чтобы это обнаружить нужно подготовить несколько тестовых записей и вручную зашифровать их методом Цезаря, после чего сравнить полученный шифр с результатом работы программы.
2. Программа не выдает ошибку при пустом анализируемом файле. В соответствии с ТЗ программа должна вывести сообщение об ошибке, если анализируемый файл пуст, но этого не происходит. Чтобы обнаружить ошибку нужно загрузить пустой файл, после чего нажать кнопку «Зашифровать».
3. Программа неправильно считает количество строк. Для обнаружения требуется сравнить количество строк в файле с соответствующей записью данных о файле.
4. Окно нельзя переместить. Для обнаружения данной ошибки требуется проверять разные способы взаимодействия с интерфейсом программы.
5. Окно может растягиваться, но не может сворачиваться. Для обнаружения данной ошибки также требуется проверять разные способы взаимодействия с интерфейсом программы.

# анализируемое техническое задание

Техническое задание на создание игры «Змейка»

Общие сведения

Игра «Змейка» представляет собой классическую аркадную игру, в которой игрок управляет змейкой, цель которой — собирать яблоки, увеличивая свою длину и набирая очки. Игра проходит на ограниченной плоскости, где игрок должен избегать столкновений со стенами и самой собой.

Цели и назначение создания автоматизированной системы

Цель создания автоматизированной системы — разработка интерактивной игры, которая обеспечит увлекательный и развлекательный опыт для пользователей. Игра будет способствовать развитию реакций, координации и стратегического мышления.

Характеристика объектов автоматизации

Змейка: главный объект игры, представляющий собой линию, состоящую из сегментов. Длина змейки увеличивается при поедании яблок.

Яблоки: объекты, которые необходимо собирать для получения очков. Каждое съеденное яблоко увеличивает длину змейки.

Стены: границы игрового поля, при столкновении с которыми игра заканчивается.

Игровое поле: ограниченная плоскость, на которой происходит игра.

Требования к автоматизированной системе

Игрок должен иметь возможность управлять движением змейки в четырех направлениях: вверх, вниз, влево, вправо.

Змейка должна увеличиваться в размерах при поедании яблок.

Игра должна завершаться при столкновении со стеной, самой собой или выходе за границы игрового поля.

Очки должны начисляться в равном объеме за каждое съеденное яблоко.

Игровое поле должно быть визуально доступным для игрока.

Состав и содержание работ по созданию автоматизированной системы

Проектирование игрового интерфейса.

Разработка механики управления змейкой.

Создание логики появления и исчезновения яблок.

Реализация системы подсчета очков.

Тестирование игры на наличие ошибок и недочетов.

Порядок разработки автоматизированной системы

Исследование и анализ требований к игре.

Проектирование архитектуры системы.

Разработка прототипа игры.

Проведение тестирования и отладка.

Завершение разработки и подготовка к запуску.

Порядок контроля и приемки автоматизированной системы

Проведение промежуточных проверок на каждом этапе разработки.

Тестирование готового продукта с привлечением тестировщиков.

Оценка соответствия конечного продукта первоначальным требованиям.

Подготовка отчета о приемке системы.

Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие

Обучение пользователей основам управления игрой.

Подготовка документации по эксплуатации игры.

Проведение финального тестирования перед запуском.

Требования к документированию

Вся документация должна быть оформлена в соответствии с установленными стандартами.

Необходимо создать пользовательское руководство и техническую документацию.

Документация должна быть доступна для всех участников проекта.

Источники разработки

Литература по разработке игр и программированию.

Существующие открытые проекты и библиотеки для создания игр.

Онлайн-курсы и ресурсы по разработке игр и программированию на выбранном языке.

# описание найденных ошибок

1. Змейка выходит за границы экрана. При попытке закончить игру столкновением с границами игрового поля было обнаружено, что змейка проходит сквозь границы, игра не заканчивается.
2. Змейка может проходить через себя. При попытке закончить игру столкновением змейки с ее хвостом было обнаружено, что змейка проходит себя насквозь, игра при этом не заканчивается.
3. Кнопка выхода работает некорректно. При попытке выхода из игры нажатием кнопки “Exit” обнаружено, что кнопка “Exit” делает то же самое что и кнопка “Restart”.
4. Случайный счет. При подборе каждого яблока числовое поле “score” либо не меняет значение, либо меняет на случайное. Таким образом, поле, отвечающее за счет игры работает некорректно.
5. Программа завершается при съедании десяти яблок. При фактическом счете в десять яблок программа завершается. Обнаружено путем нескольких попыток съесть больше десяти яблок.
6. При перезапуске игра каждый раз ускоряется. Обнаружено при попытке перезапуска игры.

# заключение

В ходе работы были получены навыки тестирования по стратегии черного ящика. Также были изучены правила написания технического задания для программных продуктов.