

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | |
| Направление подготовки | | 09.03.03 | Прикладная информатика | | |
| Направленность (профиль) | |  | Прикладная информатика в химии | | |
| Форма обучения | |  | очная | | |
|  | |  |  | | |
| Факультет | |  | Информационных технологий и управления | | |
| Кафедра | |  | Систем автоматизированного проектирования и управления | | |
| Учебная дисциплина | |  | Разработка программных систем | | |
| Курс | II | | | Группа | 425 |

Отчёт по лабораторной работе № 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Исполнители: |  |  |  |  |
| обучающиеся группы 425 |  |  |  | Смирнова С. А. |
|  |  | (дата, подпись) |  |  |
|  |  |  |  |  |
| Проверили: |  |  |  | Дамрин А. О.  Соболевский В. А. |
|  |  | (дата, подпись) |  |  |
|  |  |  |  |  |

СОДЕРЖАНИЕ

[1 Цель работы и задание 3](#_Toc153291233)

[Цель работы 3](#_Toc153291234)

[Задание 3](#_Toc153291235)

[2 Описание хода выполнения 4](#_Toc153291236)

[3 Результаты тестирования 4](#_Toc153291238)

[4 Описание графического интерфейса 5](#_Toc153291239)

[Вывод 8](#_Toc153291240)

# Цель работы и задание

## Цель работы

Разработка для программы сортировки графического пользовательского интерфейса и интеграция базы данных.

|  |  |  |
| --- | --- | --- |
| 7 | Сортировка Шелла / Shell sort | Смирнова |

## Задание

1. Для проекта, созданного в рамках 2 лабораторной, в выбранной системе контроля версий создать новую ветку (branch). Все дальнейшие модификации должны вноситься исключительно в созданную ветку, исходная ветка со 2 лабораторной не должна быть изменена. Каждый участник учебной группы в рамках работы над лабораторной должен внести хотя бы одно изменение в проект.
2. Реализовать 5 программных тестов работы с базой данных:

1) тест добавления 100 массивов в базу данных программы (размер и наполнение массивов генерируется случайным образом), который в качестве ответа выводит флаг успешного / не успешного выполнения и время работы;

2) тест добавления 1000 массивов в базу данных программы (размер и наполнение массивов генерируется случайным образом), который в качестве ответа выводит флаг успешного / не успешного выполнения и время работы;

3) тест добавления 10000 массивов в базу данных программы (размер и наполнение массивов генерируется случайным образом), который в качестве ответа выводит флаг успешного / не успешного выполнения и время работы;

4) тест выгрузки и сортировки 100 случайных массивов из базы данных, который в качестве ответа выводит флаг успешного / не успешного выполнения, общее время работы теста и среднее время работы с 1 массивом (данный тест должен быть запущен 3 раза для базы на 100, 1000 и 10000 записей);

5) тест очистки базы данных, который в качестве ответа выводит флаг успешного / не успешного выполнения и общее время работы теста (данный тест должен быть запущен 3 раза для базы на 100, 1000 и 10000 записей). Тесты должны обращаться к программным функциям создаваемой программы, но не должны быть частью самой программы.

1. Развернуть базу данных (любую на выбор, в том числе можно использовать в качестве базы данных текстовый файл, SQL или NoSQL СУБД). Создать структуру базы данных, реализующую хранение массивов (как отсортированных, так и не отсортированных). Протестировать работу базы с помощью тестов, написанных в предыдущем пункте. Результаты тестирования записать в отчёт.
2. Реализовать графический пользовательский интерфейс (стек технологий любой на выбор). Интерфейс должен иметь функции:

1) ввода/редактирования массива для сортировки с клавиатуры;

2) сохранение выведенного в интерфейсе массива в базу данных;

3) вывод списка всех массивов, хранящихся в базе (без возможности редактирования и сортировки);

4) выгрузка массива из базы для редактирования и сортировки;

5) сортировка массива по алгоритму реализованному во 2 лабораторной.

Получившийся интерфейс должен хотя бы отчасти соответствовать следующий эвристикам ui / ux :

1. видимость статуса системы;
2. соответствие между системой и реальным миром;
3. управляемость;
4. согласованность;
5. гибкость;
6. минимализм в дизайне;
7. помощь пользователю в выявлении ошибок;
8. наличие справки.

В отчёте должны быть даны пояснения, какие элементы графического интерфейса какой эвристике соответствуют.

1. Написать отчёт, в котором представить результат работы программы и все необходимые пояснения, указанные в предыдущих пунктах работы.

* Текст программы должен быть хорошо прокомментирован. По всем не прокомментированным участкам программы будут заданы вопросы.

# Описание хода выполнения

Результат выполнения задания выложен по адресу (ветка ):

Для разработки графического интерфейса была использована стандартная графическая библиотека Python 3 – Tkinter. Для создания базы была выбрана SQLite, управляемая из стандартного модуля Python 3 – sqlite3.

Tkinter не имеет особых преимуществ над другими библиотеками, такими как PyQt5, но обладает достаточным функционалом для быстрого прототипирования интерфейса несложных приложений, как в данной контрольной работе.

SQLite был выбран в связи с простотой создания базы данных и её переноса данных между несколькими устройствами.

# Результаты тестирования

База данных имеет структуру, содержащую одну таблицу с двумя полями: sequence (содержит массив), type (содержит тип последовательности, raw – обычная, sorted – отсортированная).

Результат проведения тестов работы с базой данных представлен ниже. Время работы указано в секундах. Последовательности составлялись из чисел длиной до 15 знаков в количестве до 100 штук.

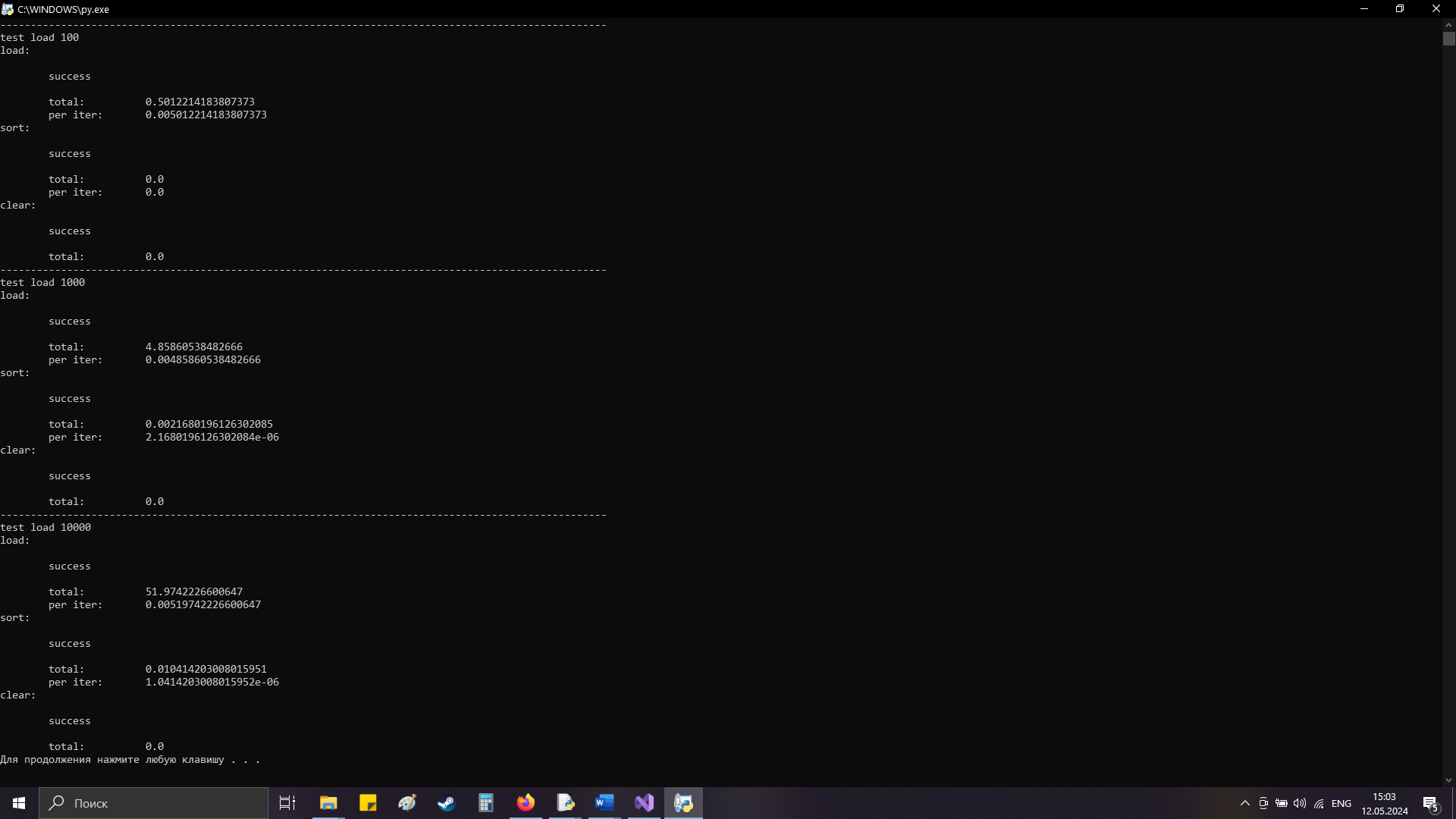


Рисунок 1 – Результаты тестирования

# Описание графического интерфейса

Основные окна представлены на рисунках 2-4.

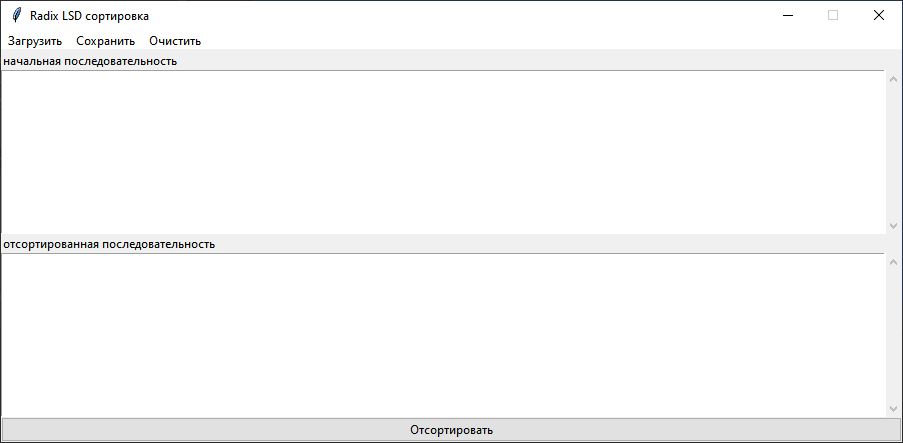
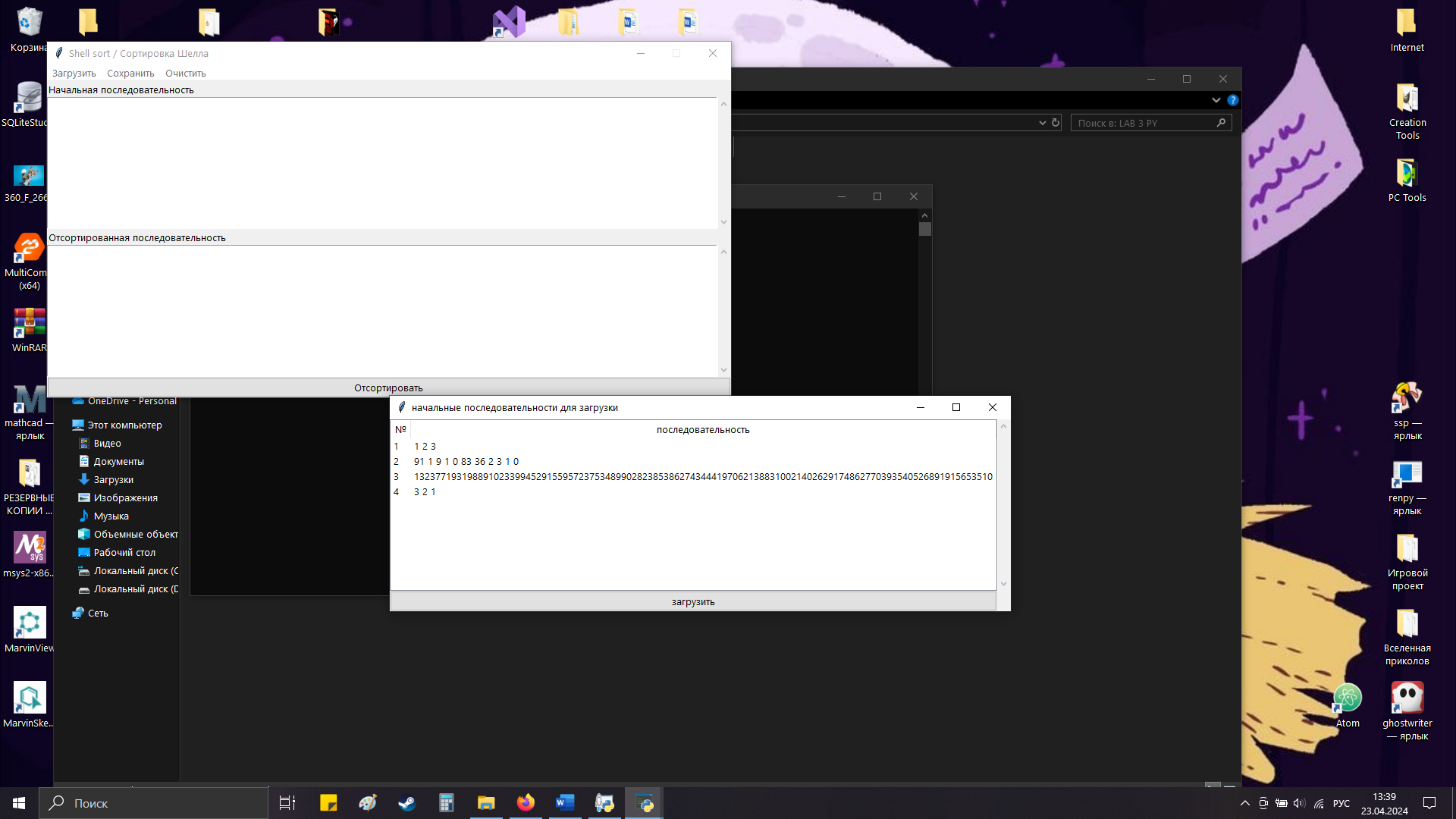


Рисунок 2 – основное окно приложения



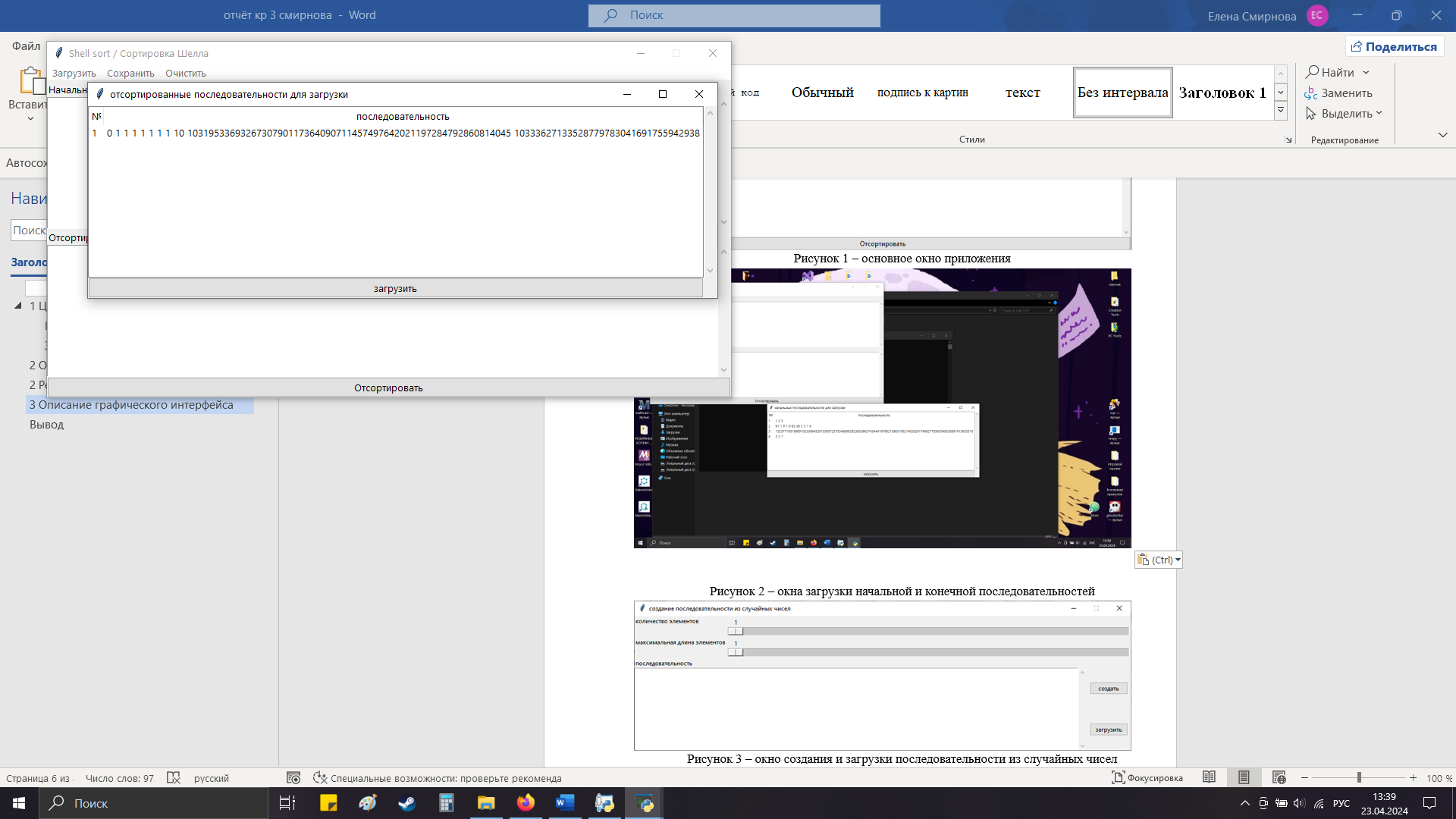


Рисунок 3 – окна загрузки начальной и конечной последовательностей

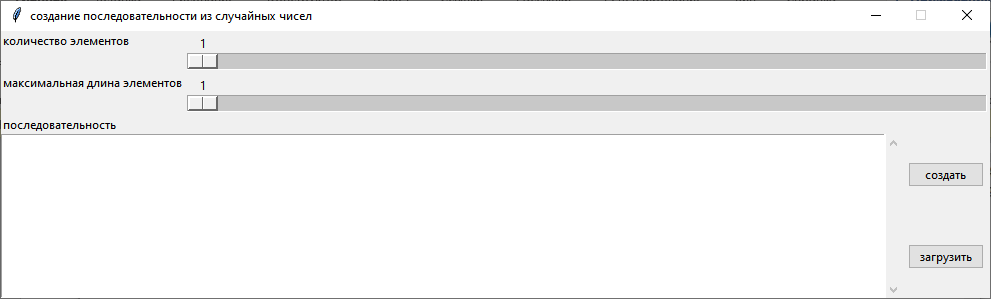


Рисунок 4 – окно создания и загрузки последовательности из случайных чисел

Видимость статуса системы можно увидеть только при загрузке очень длинной последовательности, пример представлен на рисунке 5.

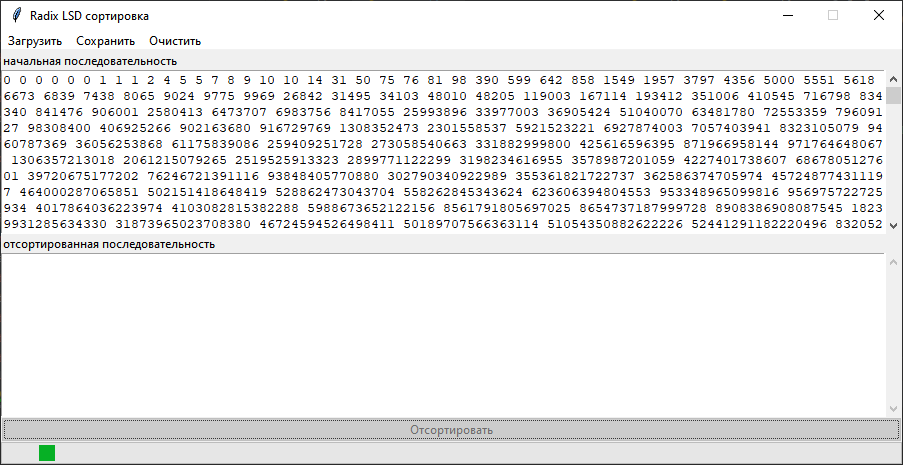


Рисунок 5 – процесс обработки очень длинного массива (худший случай алгоритмической сложности – отсортированный по порядку)

Соответствие между системой и реальным миром представлено отсутствием любых слов, значение которых было бы сложно понять. Также все подписи и элементы управления расположены вплотную друг к другу, избегая двусмысленности описания.

Согласованность представлена привычным для управления меню в шапке приложения, позволяющим выполнить дополнительные функции без длительного поиска. Также она видна в меню выбора последовательности из базы двойным нажатием.

Помощь пользователю в выявлении ошибок представлена предупреждениями о попытке выполнения неверной операции, такой как попытка загрузить последовательность без выбора.

Каждое из окон имеет минимальное количество основных элементов управления, доступных при открытии окна

# Вывод

Результат тестирования базы данных показал зависящее от количества массивов время записи и обработки данных. При использовании более мощного инструмента работы с SQL базами данных, можно было бы уменьшить это время.