# МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ОТЧЁТ О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ**

**Б2.В.01.01 (У) научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)**

по направлению подготовки

02.03.02 Фундаментальная информатика и информационные технологии

направленность (профиль) /специализация

Математическое и программное обеспечение компьютерных технологий

период с 06 .07.2023 г. по 19.07.2023 г.

Выполнил:

студент 16 группы ОФО \_ \_\_\_ Агаджанян А.С.

*(подпись) (Ф.И.О. студента)*

Руководитель практики:

к.т.н., доцент Руденко О.В.

*(ученое звание, должность) (подпись) (Ф.И.О)*

Оценка по итогам защиты практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

Краснодар

2023

ФГБОУ ВО «КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ, ВЫПОЛНЯЕМОЕ В ПЕРИОД ПРОВЕДЕНИЯ**

**УЧЕБНОЙ ПРАКТИКИ**

**Б2.В.01.01 (У) НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА**

Студент Агаджанян Алёна Самвеловна

*(фамилия, имя, отчество полностью)*

Направление подготовки 02.03.02 Фундаментальная информатика информационные технологии

Место прохождения практики: ФГБОУ ВО «Кубанский государственный университет»

Срок прохождения практики с 06 .07.2023 г. по 19.07.2023 г.

Цель практики – закрепление полученных теоретических знаний по дисциплинам направления и специальным дисциплинам программы бакалавриата Фундаментальная информатика информационные технологии, овладение необходимыми профессиональными компетенциями по избранному направлению подготовки.

Формирование компетенций, регламентируемых ФГОС ВО:

| **Код компетенции** | **Содержание компетенции (или её части)** | **Планируемые результаты при прохождении практики** |
| --- | --- | --- |
| **УК-1** | Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач | Анализ существующих и выбор методов решения задач |
| **УК-2** | Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений | Выбор оптимального способа решения поставленных задач |
| **УК-3** | Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде | Защита отчета по учебной практике |
| **УК-4** | Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах) | Письменный отчет о прохождении практики |
| **УК-6** | Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни | Выполнение задач в установленные сроки |
| **ОПК-1** | Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности | Применение математических методов для решения поставленных задач |
| **ОПК-2** | Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности | Разработанные программы с применением современного программного обеспечения |
| **ПК-1** | Способен понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии | Реализация математических методов с помощью современных языков программирования |
| **ПК-2** | Способен проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности | Проведение анализа эффективности использованных математических методов |

**Перечень вопросов (заданий, поручений) для прохождения практики:**

1. Дан многочлен степени  в форме . Пользователь вводит значения , найти 

2. Дана определенная система линейных алгебраических уравнений в форме . Где  – квадратная матрица размерности , – вектор столбец размерности . Решить ее методом Крамера.

3. Из файла считывается текст на русском языке со знаками препинания. Построить кодирование этого текста методом Лемпеля-Зива на основании частоты встречаемости символов в тексте. Построить кодирование равномерным двоичным кодом, оценить выигрыш.

4. Реализовать программный продукт, позволяющий хранить и отображать информацию о пользователях. При запуске программного обеспечения вся информация о пользователях считывается из файла, пользователь может выбрать одну из следующих альтернатив:

* 1. Посмотреть список пользователей
  2. Добавить пользователя
  3. Удалить пользователя
     1. По фамилии-имени
     2. По логину
     3. По номеру телефона
  4. Изменить пользователя
     1. По фамилии-имени
     2. По логину
     3. По номеру телефона
  5. Сохранить изменения в файл
  6. Отправить сообщение на e-mail пользователя
     1. По фамилии-имени
     2. По логину
     3. По номеру телефона
  7. Отсортировать по выбранному полю
  8. Выход

**План-график выполнения работ**:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Этапы работы (виды деятельности) при прохождении практики | Сроки | Отметка руководителя  практики о выполнении (подпись) |
| 1 | Получение задания на практику.  Инструктаж по технике безопасности.[[1]](#footnote-1)\* | 06.07.2023 |  |
| 2 | Решение первой задачи учебной практики на основе курса «Алгебра» | 07.7.2023 – 10.07.2023 |  |
| 3 | Решение второй задачи учебной практики на основе курсов «Дифференциальное исчисление» и «Интегральное исчисление» | 11.07.2023 – 12.07.2023 |  |
| 4 | Решение третьей задачи учебной практики на основе курса «Дискретная математика» | 13.07.2023 – 14.07.2023 |  |
| 5 | Решение четвертой задачи учебной практики | 15.07.2023 – 16.07.2023 |  |
| 5 | Оформление результатов выполненных заданий и их согласование с руководителем (составление отчета о прохождении производственной практики) | 17.07.2023 – 18.07.2023 |  |
| 6 | Защита отчета | 19.07.2023 |  |

*Ознакомлен*

*(подпись студента) ( расшифровка подписи)*

« 06 » июля 2023 г.

Руководитель практики \_Руденко О.В.\_\_\_

*(подпись) (Ф.И.О. руководителя)*

**ОЦЕНОЧНЫЙ ЛИСТ**

**результатов прохождения   
УЧЕБНОЙ ПРАКТИКИ**

**(НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА)**

**по направлению подготовки**

**02.03.02 Фундаментальная информатика и информационные технологии**

Фамилия И.О. студента

Курс 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | ОБЩАЯ ОЦЕНКА  (отмечается руководителем практики) | Оценка | | | |
| 5 | 4 | 3 | 2 |
| 1. | Уровень подготовленности студента к прохождению практики |  |  |  |  |
| 2. | Умение правильно определять и эффективно решать основные  задачи |  |  |  |  |
| 3. | Степень самостоятельности при выполнении задания по  практике |  |  |  |  |
| 4. | Оценка трудовой дисциплины |  |  |  |  |
| 5. | Соответствие программе практики работ, выполняемых  студентом в ходе прохождении практики |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | СФОРМИРОВАННЫЕ В РЕЗУЛЬТАТЕ УЧЕБНОЙ ПРАКТИКИ КОМПЕТЕНЦИИ  (отмечается руководителем практики от университета) | Оценка | | | |
| 5 | 4 | 3 | 2 |
|  | УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач |  |  |  |  |
|  | УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений |  |  |  |  |
|  | УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде |  |  |  |  |
|  | УК-4 Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах) |  |  |  |  |
|  | УК-6 Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни |  |  |  |  |
|  | ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности |  |  |  |  |
|  | ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности |  |  |  |  |
|  | ПК-1 Способен понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии |  |  |  |  |
|  | ПК-2 Способен проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности |  |  |  |  |

Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Руденко О.В.

*(подпись) (расшифровка подписи)*

**СВЕДЕНИЯ**

**о прохождении инструктажа по ознакомлению с требованиями охраны труда, технике безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка**

Предприятие Федеральное государственное бюджетное образовательное учреждение высшего образования «Кубанский государственный университет»

Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

|  |  |
| --- | --- |
| Студент | Агаджанян Алёна Самвеловна, 19 лет |
|  | (ФИО, возраст) |

Дата 06 июля 2023 г.

**1.     Инструктаж по требованиям охраны труда**

|  |  |
| --- | --- |
| Провел | канд. тех. наук, доцент Руденко О.В. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Агаджанян А.С. |
|  | (ФИО, подпись студента) |

**2.     Инструктаж по технике безопасности**

|  |  |
| --- | --- |
| Провел | канд. тех. наук, доцент Руденко О.В. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Агаджанян А.С. |
|  | (ФИО, подпись студента) |

**3.     Инструктаж по пожарной безопасности**

|  |  |
| --- | --- |
| Провел | канд. тех. наук, доцент Руденко О.В. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Агаджанян А.С. |
|  | (ФИО, подпись студента) |

**4. Инструктаж по правилам внутреннего трудового распорядка**

|  |  |
| --- | --- |
| Провел | канд. тех. наук, доцент Руденко О.В. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Агаджанян А.С. |
|  | (ФИО, подпись студента) |

**ОТЗЫВ**

**руководителя о прохождении учебной практики**

**(НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА)**

|  |  |
| --- | --- |
| Студент | Агаджанян Алёна Самвеловна |
|  | *(фамилия, имя, отчество полностью)* |

Направление подготовки 02.03.02. Фундаментальная информатика и информационные технологии

За время прохождения учебной практики (НИР) студент Агаджанян А.С. проявил себя как способный и самостоятельный студент.

Студентом были решены следующие задачи и получены результаты:

|  |
| --- |
| 1. Изучены возможности языка Python, позволяющие решать поставленные задачи |
| 1. Реализован алгоритм нахождения решения матричного уравнения методом Крамера, алгоритм вычисления определённого интеграла для полинома   на отрезке [a ; b], а также алгоритм построения кодирования текста на русском языке методом Лемпеля-Зива. |
| 1. Изучена информация о способах представления многочленов в программе, о методах работы со строками. |
| 1. Реализована БД для хранения информации о пользователях. |
| 1. Разработаны приложения, позволяющие вычислить определённый интеграл для полинома, решать матричные уравнения методом Крамера, кодировать текст методом Лемпеля-Зива. |

Все поставленные задачи выполнены, степень проработанности материала считаю достаточной.

Результаты практики оцениваю на оценку "\_\_\_\_\_\_\_\_\_\_\_\_".

Руководитель практики Руденко О.В.

*(подпись) (расшифровка подписи)*

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 11](#_Toc38710752)

[Задача 1 Ошибка! Закладка не определена.](#_Toc38710753)

[Задача 2 Ошибка! Закладка не определена.](#_Toc38710754)

[Задача 3 Ошибка! Закладка не определена.](#_Toc38710755)

[Задача 4 Ошибка! Закладка не определена.](#_Toc38710756)

ЗАКЛЮЧЕНИЕ [Ошибка! Закладка не определена.](#_Toc38710757)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ Ошибка! Закладка не определена.](#_Toc38710758)

[ПРИЛОЖЕНИЕ Ошибка! Закладка не определена.](#_Toc38710759)

# ВВЕДЕНИЕ

Целью практики является получение первичных навыков самостоятельного осуществления научно-исследовательской работы; развитие профессиональных знаний в области прикладной математики и информатики; овладение необходимыми профессиональными компетенциями по избранному направлению подготовки.

Место прохождения практики: Россия, Краснодарский край, г. Краснодар, ФГБОУ ВО «Кубанский государственный университет».

Продолжительность практики: 2 недели.

Перечень основных работ: разработать программу, вычисляющую определённый интеграл полинома, программу для вычисления корней системы линейных алгебраических уравнений методом Крамера, программу, кодирующую русский текст методом Лемпеля-Зива.

**Задача 1**

**1. Математическая постановка задачи**

**Решаемая задача:** Дан многочлен степени  в форме . Пользователь вводит значения , найти 

Пусть функция y=f(x) определена на отрезке [a; b], a<b. Разобьём этот отрезок на n произвольных частей точками:

a=x0<x1<x2<…<xi-1<xi<…<xn=b

Обозначим это разбиение через τ, а точки x1,…,xn , будем называть точками разбиения. В каждом из полученных частичных отрезков [xi-1; xi] выберем произвольную точку ξ(xi-1 ≤ ξi ≤ xi). Через ∆xi обозначим разность xi – xi-1, которую условимся называть длиной частичного отрезка [xi-1; xi].

Образуем сумму:

Ϭ = f(ξ1) ∆x1 + f(ξ2) ∆x2 + … + f(ξn) ∆xn = 

которую назовём интегральной суммой для функции f(x) на [a; b], соответствующей данному разбиению [a; b] на частичные отрезки и данному выбору промежуточных точек ξi.

Обозначим через λ длину наибольшего частичного отрезка разбиения τ: 

Определение:

Если существует конечный предел I интегральной суммы при λ→0, то этот предел называется **определённым интегралом** от функции f(x) по отрезку

[a; b] и обозначается:



**2. Описание алгоритма решения**

Чтобы вычислить определённый интеграл воспользуемся формулой Ньютона-Лейбница:



где F(x) – первообразная функции f(x) в точке x.

На начальном этапе у нас многочлен n-ой степени. Создаём 2 списка coefficient и degree, которые содержат соответствующие коэффициенты и степени для каждой переменной в полиноме. Затем мы проходимся по списку degree и увеличиваем каждое значение на 1. Далее мы идём по списку coefficient и делим каждое значение на соответствующее ему значение из списка degree. Теперь coefficient и degree содержат соответствующие коэффициенты и степени для каждой переменной первообразной полинома. Далее мы вычисляем F(b):



и F(a):



Посчитаем определённый интеграл:



Асимптотическая сложность алгоритма: O(n), так как данный алгоритм не содержит вложенных циклов.

**3. Техническое описание программного продукта**

В программе подключён класс Fraction, который позволяет представлять и хранить данные в виде обыкновенных дробей.

Данныйкодоткрывает файл “matan.txt” и считывает строку. Затем она удаляет все символы “\*”, ”x”, “,”, “ “, “\n” из этой строки и преобразует полученную строку в список.

Далее код проверяет, есть ли в списке символ “[“, и если есть, то сохраняет индекс этого элемента в переменную index1. Затем создаём новый список sl, который содержит все элементы списка sp, начиная с индекса index1+1 и заканчивая символом “]”, разбивая их по символу “;”. Затем каждый элемент списка sl преобразуется в целое число.

Далее код создаёт новый список step2, который содержит индексы всех “^”, “-“, “+” в списке sp.

Затем создаются новые списки coefficient и degree, которые будут содержать соответствующие коэффициенты и степени для каждой переменной в полиноме.

Далее код проходит по списку step2 и для каждой пары символов "^", "-", "+" в списке sp создает новый элемент в списке coefficient, который равен разности индексов этих символов в списке sp. Затем он создает новый элемент в списке degree, который равен индексу следующего символа в списке sp после символа "^", "-" или "+".

Далее каждый элемент списка degree увеличивается на 1 и каждый элемент списка coefficient делится на соответствующий элемент списка degree. В переменную sumb суммируется coefficient[i]\*b^degree[i], а в переменную suma суммируется coefficient[i]\*a^degree[i], где a,b – это границы интегрирования, хранящиеся в списке sl.

Наконец, результат работы программы, а именно sumb-suma, записывается в файл “matan.txt”.

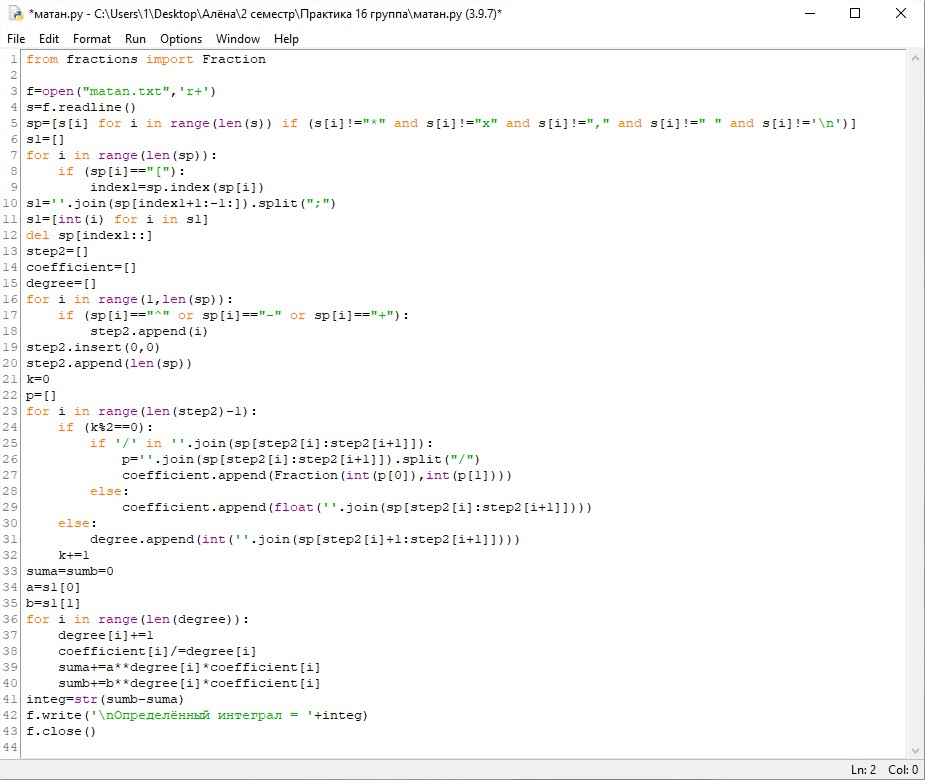


Рисунок 1 – главное окно программы

**4. Инструкция по эксплуатации**

Данные на вход подаются из файла matan.txt, который находится в той же директории, что и сам код программы. Данные в файле должны быть представлены в виде строки: Fn(x) = a0\*x0 + a1\*x1 + … + an\*xn, [a;b].

Результаты будут представлены в том же файле.

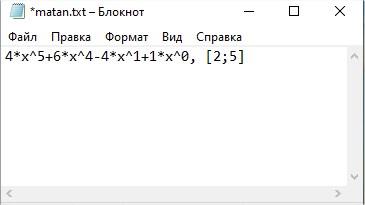


Рисунок 2 – файл с входными данными до выполнения программы

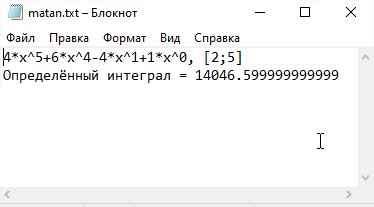


Рисунок 3 – файл с данными после выполнения программы

**5. Набор информации для тестирования**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Задача 2**

1. **Математическая постановка задачи**

**Решаемая задача:** Дана определенная система линейных алгебраических уравнений в форме . Где  – квадратная матрица размерности , – вектор столбец размерности . Решить ее методом Крамера.

Системы из n уравнений с n неизвестными называют **Крамеровской системой уравнений**, если определитель матрицы коэффициентов не равен 0.

**Вектор-столбец** – это матрица, состоящая из 1 столбца.

Система линейных алгебраических уравнений называется совместной, если она имеет хотя бы 1 решение.

Совместная система может быть:

1. определённой (имеет единственное решение)
2. неопределённой (имеет множество решений)
3. **Описание алгоритма решения**

1) Вычисляем определитель матрицы A. Обозначим его как ∆.

2) Последовательно вставляем вектор-столбец B вместо i-того столбца матрицы A и вычисляем его определитель. Обозначим его как ∆i.

3) Если ∆=0 и все ∆i=0, то система имеет множество решений.

Если ∆=0 и хотя бы 1 ∆i0, то система не имеет решений.

Если ∆0, то xi =.

Асимптотическая сложность “основного” алгоритма: O(n2)

1. **Техническое описание программного продукта**

В начале программы открывается файл “alg.txt”, в котором находится система линейных уравнений в виде .

Для хранения основных данных используется несколько матриц: matrix – матрица коэффициентов, b – матрица, хранящая вектор-столбец B.

Функция matrix\_det(), формальными переменными которой являются матрица и её размер, вычисляет определитель методом разложения по строке-столбцу.

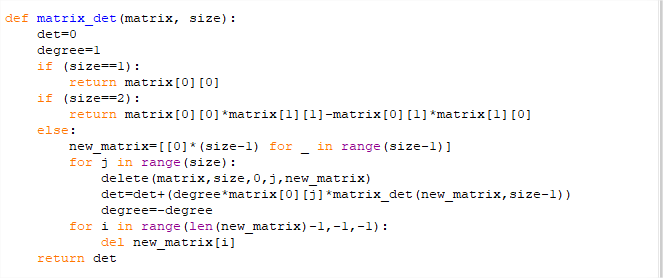


Рисунок 4 - алгоритм функции matrix\_det()

Для удаления строки и столбца из матрицы используется функция delete(), формальными переменными которой являются матрица, её размер, строка и столбец, которые мы удаляем, а также переменная new\_matrix, которая будет являться результатом работы этой функции.

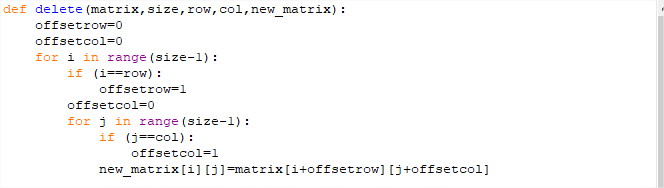


Рисунок 5 - алгоритм функции delete()

У функции zam() формальными переменными являются матрица и столбец, который мы будем заменять. Функция zam() заменяет столбец col матрицы matrix на вектор b.

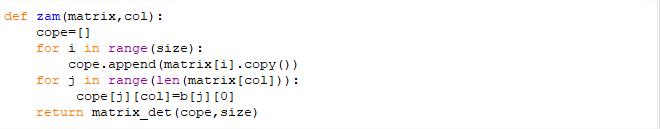


Рисунок 6 - алгоритм функции zam()

После замены столбца col матрицы matrix на вектор b мы считаем определитель получившейся матрицы ∆i и записываем его в список zero.

Результат работы программы записывается в файл “alg print.txt”.



Рисунок 7 – главное окно программы

**4. Инструкция по эксплуатации**

Данные на вход подаются из файла “alg.txt”. Результат будет представлен в файле “alg print.txt”.

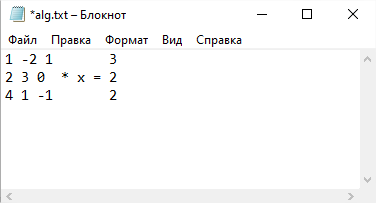


Рисунок 8 – файл с входными данными до выполнения программы

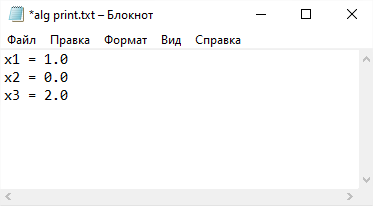


Рисунок 9 – файл с выходными данными после выполнения программы

**5. Набор информации для тестирования**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Задача 3**

**1. Математическая постановка задачи**

**Решаемая задача:** Из файла считывается текст на русском языке со знаками препинания. Построить кодирование этого текста методом Лемпеля-Зива на основании частоты встречаемости символов в тексте. Построить кодирование равномерным двоичным кодом, оценить выигрыш.

**Алгоритм Лемпеля-Зива** – это универсальный алгоритм сжатия данных без потерь.

**Равномерный код** – это код, в котором все кодовые слова имеют одинаковую длину.

**Двоичный код** – это способ представления данных в виде кода, в котором каждый разряд принимает одно из двух возможных значений, обычно обозначаемых цифрами «0» и «1».

**2. Описание алгоритма решения**

Процесс сжатия выглядит следующим образом. Последовательно считываются символы входного потока и происходит проверка, существует ли в созданной таблице такая строка. Если такая строка существует , считывается следующий символ, а если строка не существует, в поток заносится код для предыдущей найденной строки, строка заносится в таблицу, а поиск начинается снова.

**Псевдокод алгоритма:**

1. Инициализация словаря всеми возможными односимвольными фразами. Инициализация входной фразы ω первым символом сообщения.

2. Считается очередной символ К из кодируемого сообщения.

3. Если КОНЕЦ\_СООБЩЕНИЯ, то выдать код для ω, иначе:

4. Если фраза ω(К) уже есть в словаре, присвоить входной фразе значение ω(К) и перейти к шагу 2, иначе выдать код ω, добавить ω (К) в словарь, присвоить входной фразе значение К и перейти к шагу 2.

5. Конец.

Асимптотическая сложность алгоритма: О(n), так как, чтобы закодировать текст, мы проходимся по нему 1 раз.

**3. Техническое описание программного продукта**

Содержимое файла “discretka.txt” записывается в строку s. Затем определяется набор уникальных символов в тексте и записывается в множество myset.

Создаётся словарь, в котором изначально будут находиться уникальные символы и каждому из них будет присвоен свой код в двоичной системе счисления. Количество бит, необходимых для представления кода каждого символа, определяется с помощью функции “math.log2”, для которой мы заранее подключили библиотеку math.

Затем мы проходимся по исходному тексту посимвольно, и если текущей комбинации символов, которая хранится в строковой переменной news, нет в словаре, то она добавляется в словарь и получает новый код. Иначе мы считываем новый символ.

Изначально мы кодировали текст равномерно n битами, так как столькими битами мы закодировали каждый уникальный символ текста. Если при дальнейшем добавлении слов в словарь n бит не хватает для кодировки, то новые элементы словаря будем кодировать (n+1) битами и текст будем кодировать также равномерно, но уже (n+1) битами.

Закодированный текст записывается в список sp.

Далее вычисляем количество бит, необходимых для представления исходного текста, как произведение количества символов в тексте на количество бит, необходимых для кодирования одного символа. И записываем получившееся значение в переменную bit1.

Затем вычисляем количество бит, необходимых для представления закодированного текста (в данном случае это будем длина закодированного текста) и записываем это значение в переменную bit2.

Коэффициент сжатия вычисляется как процентное отношение разницы между количеством бит в исходном и закодированных текстах к количеству бит в исходном тексте.

Закодированный текст записывается в файл "discretka print.txt" вместе с информацией о количестве бит, необходимых для представления исходного и закодированного текстов, а также о коэффициенте сжатия текста.



Рисунок 10 - главное окно программы

**4. Инструкция по эксплуатации**

Ввод исходных данных осуществляется из файла“discretka.txt”. В файле содержится текст, написанный на русском языке, со знаками препинания.

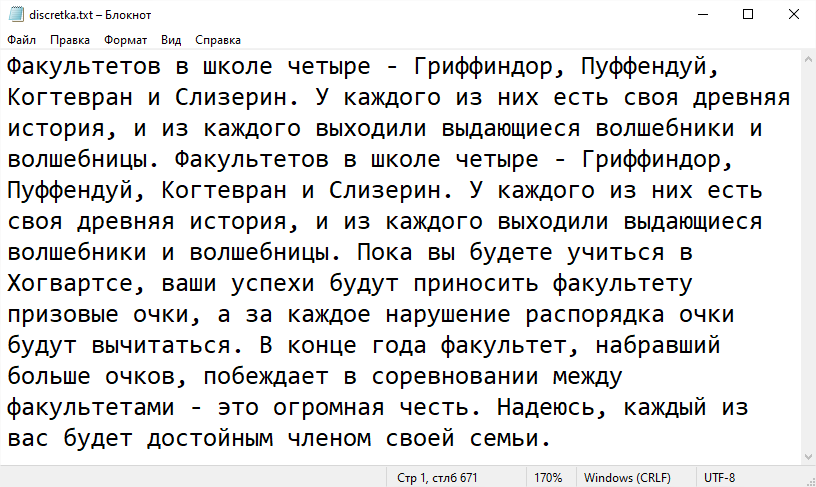


Рисунок 11 – файл с входными данными до выполнения программы

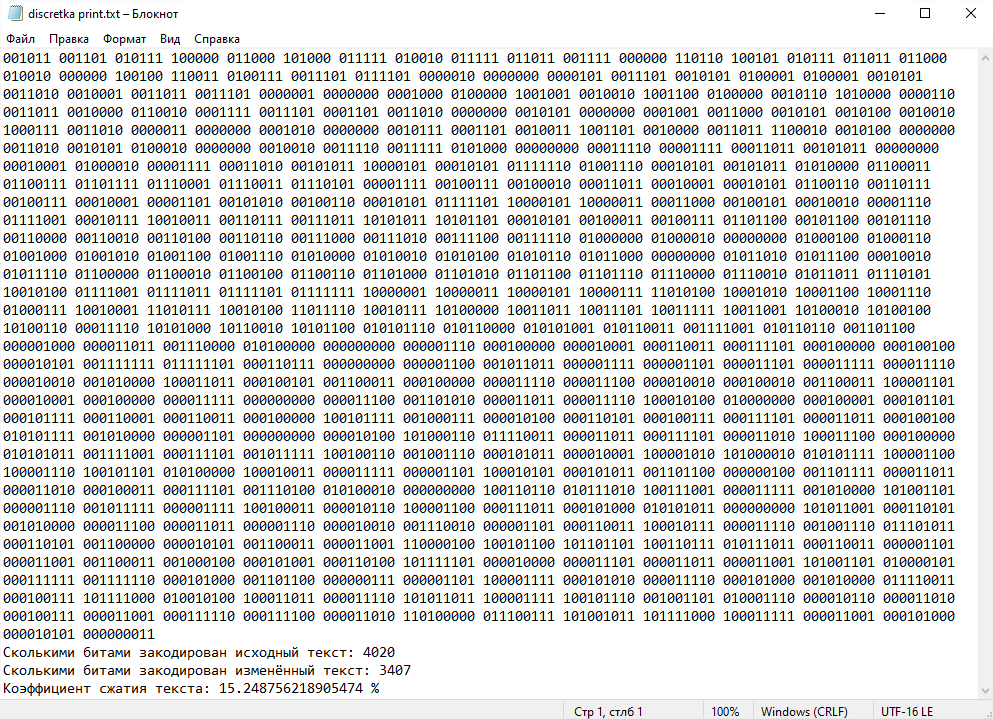


Рисунок 12 – файл с выходными данными после выполнения программы

**5. Набор информации для тестирования**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Задача 4**

**1. Постановка задачи**

**Решаемая задача:** Реализовать программный продукт, позволяющий хранить и отображать информацию о пользователях. При запуске программного обеспечения вся информация о пользователях считывается из файла, пользователь может выбрать одну из следующих альтернатив:

1. Посмотреть список пользователей
2. Добавить пользователя
3. Удалить пользователя
   * 1. По фамилии-имени
     2. По логину
     3. По номеру телефона
4. Изменить пользователя
   * 1. По фамилии-имени
     2. По логину
     3. По номеру телефона
5. Сохранить изменения в файл
6. Отправить сообщение на e-mail пользователя
   * 1. По фамилии-имени
     2. По логину
     3. По номеру телефона
7. Отсортировать по выбранному полю
8. Выход

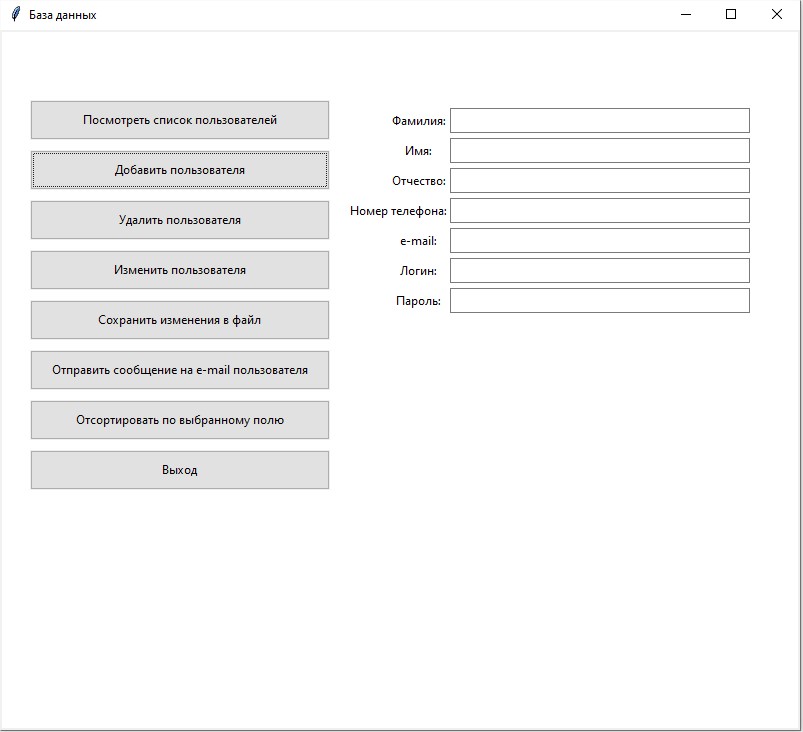
**2. Техническое описание программного продукта**

При запуске программного продукта вся информация о пользователях считывается из файла, пользователю же предоставляется интерфейс программы, в котором отображаются действия, указанные в условии. Все действия с данными в программе осуществляются через графический интерфейс.

Все данные о пользователях заносятся в программу в конкретной форме, правильность вводимых данных проверяется с помощью регулярных выражений, позже данные преобразовываются в форму, подходящую для хранения.



Рисунок 13 – функция, обеспечивающая ввод полей



Функции regex\_SNO(), regex\_phone(), regex\_email(), regex\_password(), формальной переменной которых будет строка s, являются функциями, возвращающими единицу, если строка с фамилией-именем-отчеством, номером телефона, электронной почтой и паролем соответственно является корректной, и ноль, если требуемые условия к полям не выполняются.

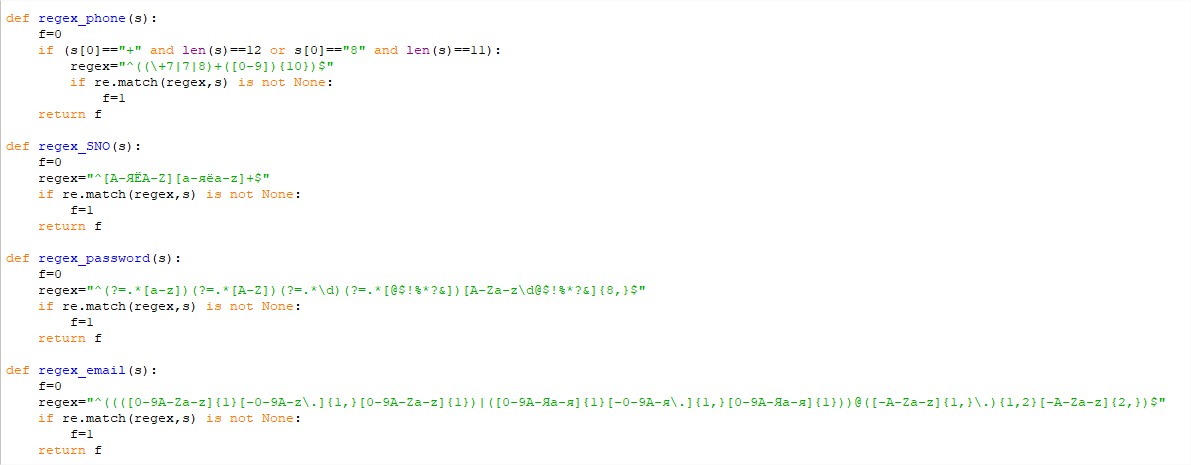


Рисунок 14 – функции, проверяющие корректность введённых данных

С помощью функции see() мы можем посмотреть список всех пользователей.

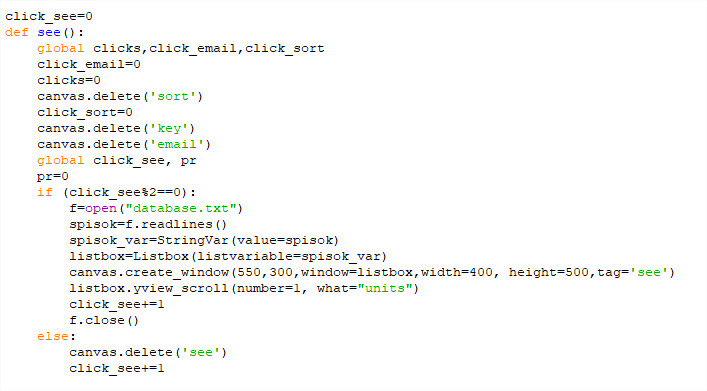
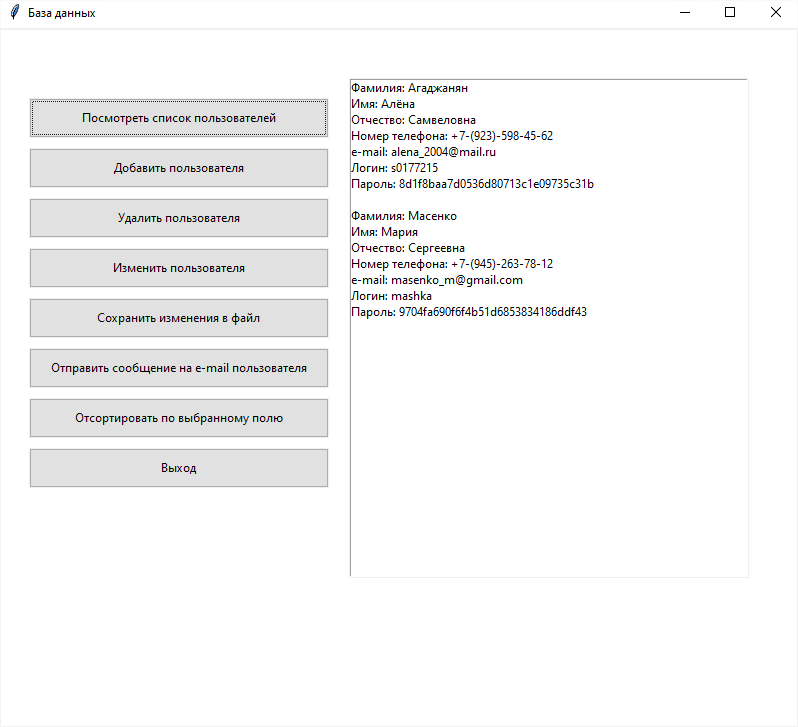
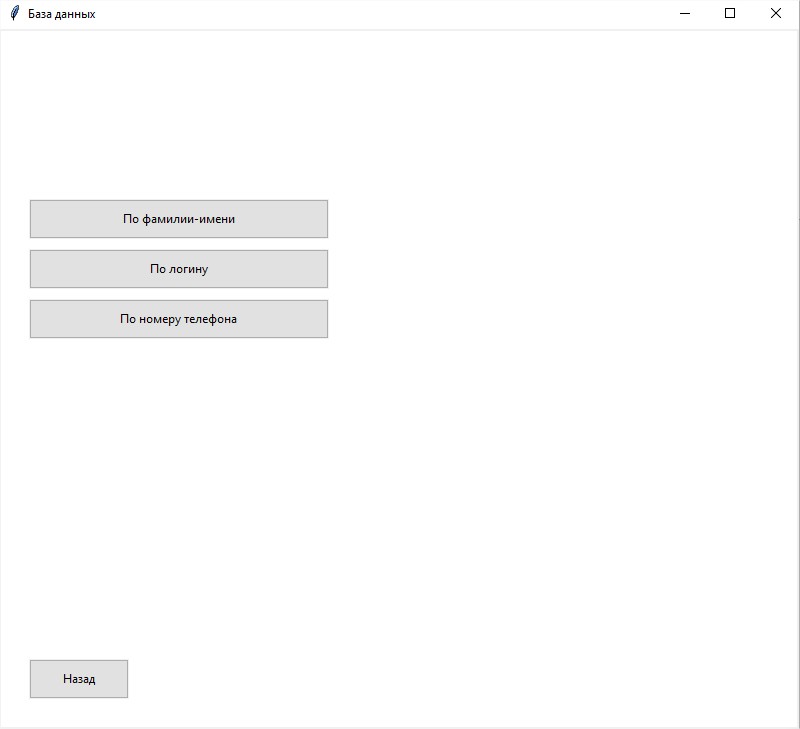


Рисунок 15 – функция, позволяющая посмотреть список всех пользователей



При нажатии на кнопку «удалить пользователя» вам предоставляется дополнительный выбор.



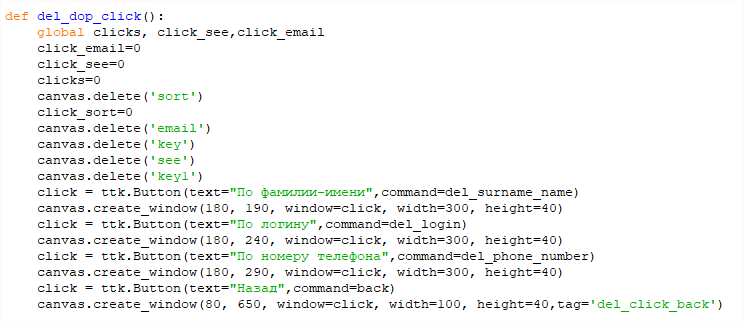


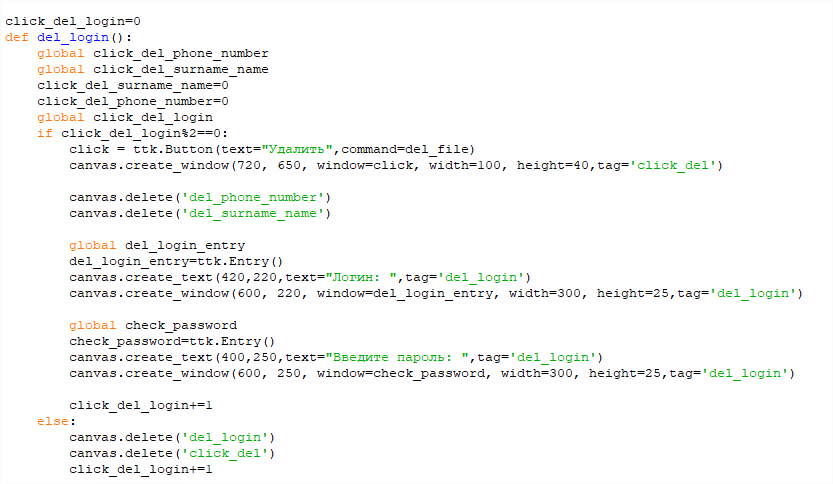
Рисунок 16 – функция, предоставляющая дополнительный выбор

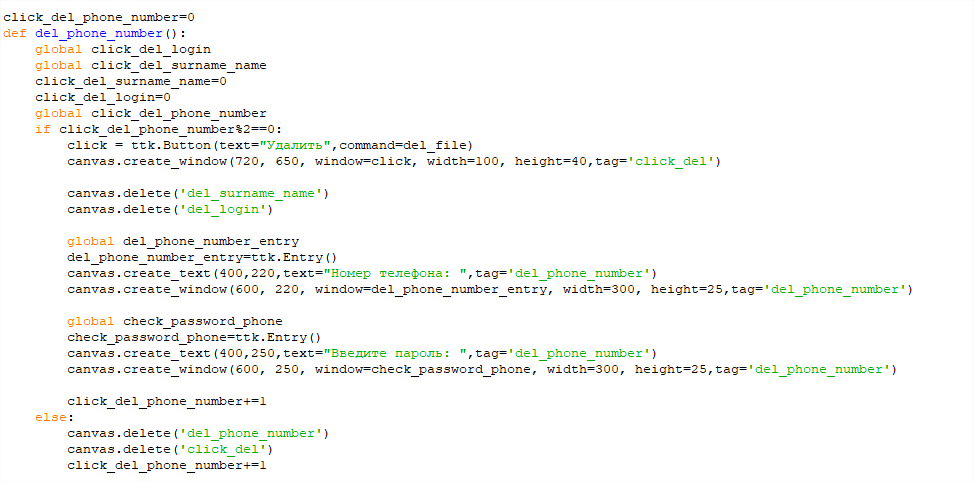


Рисунок 17 – реализация функции back(), позволяющей вернутся к исходному меню

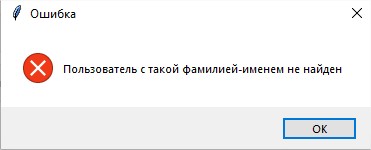
При нажатии на одну из этих кнопок функции del\_surname\_name(), del\_login(), del\_phone\_number() предоставляют вам поля ввода фамилии-имени/логина/телефона соответственно, а также поле ввода пароля.







Если пользователь не найден, на экран выведется соответствующее сообщение. Для вывода ошибки была импортирована из библиотеки tkinter.messagebox функция showerror.



Если же пользователь найден, то происходит проверка пароля. Пароль хранится в форме ХЭШ-кода. Для того, чтобы сравнить пароли мы хэшируем введённый пароль с помощью вспомогательной функции hashlib.md5() и если захэшированные пароли совпадают, то после нажатия соответствующей кнопки «удалить» информация о пользователе удаляется.

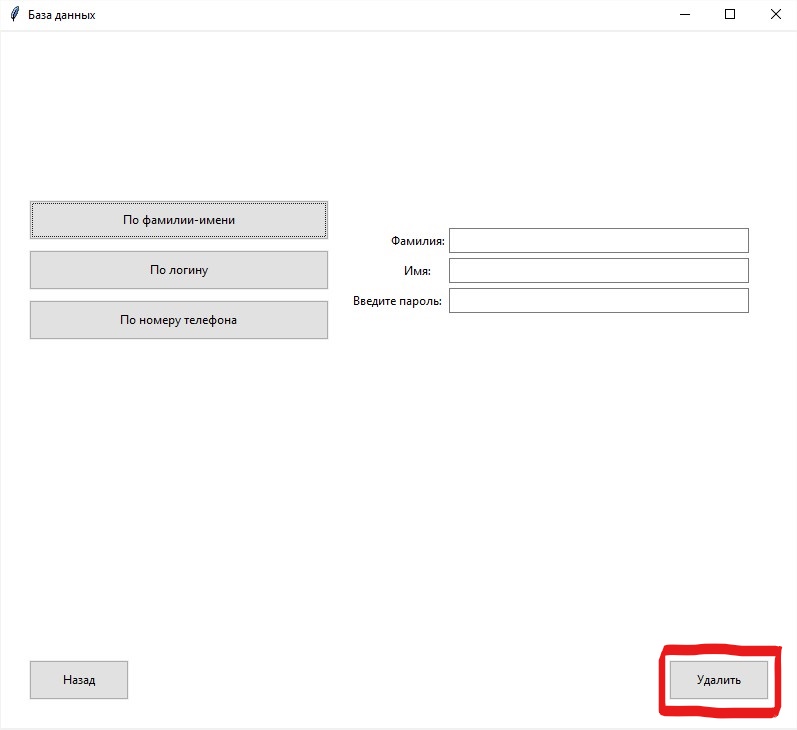






Рисунок 18 – функция, удаляющая пользователя

При нажатии на кнопку «изменить пользователя» вам предоставляется дополнительный выбор.

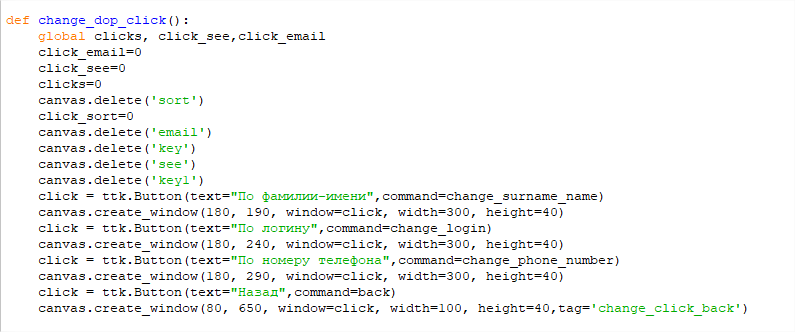


Рисунок 19 – функция, предоставляющая дополнительный выбор

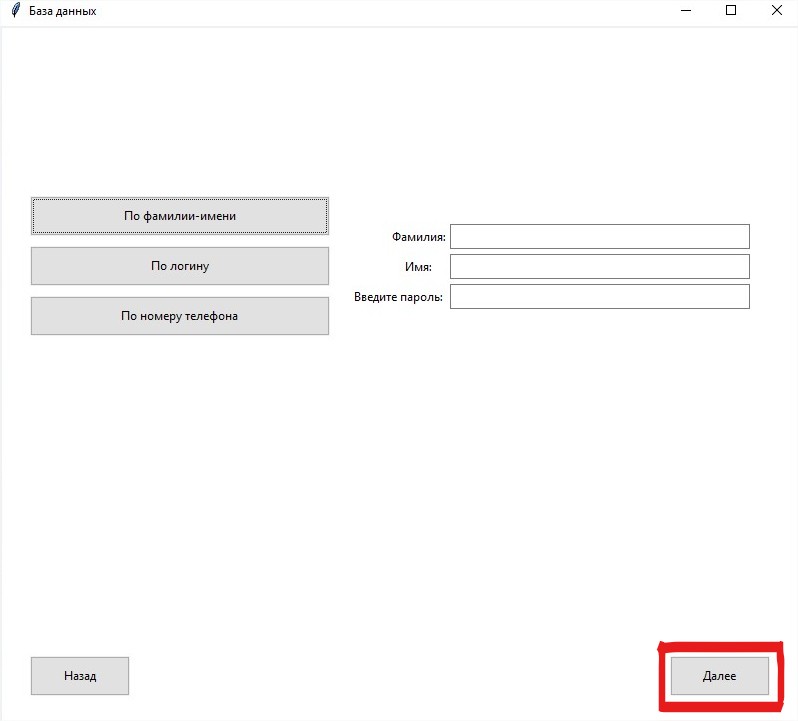
При нажатии на одну из этих кнопок функции change\_surname\_name(), change\_login(), change\_phone\_number() предоставляют вам поля ввода фамилии-имени/логина/телефона соответственно, а также поле ввода пароля.

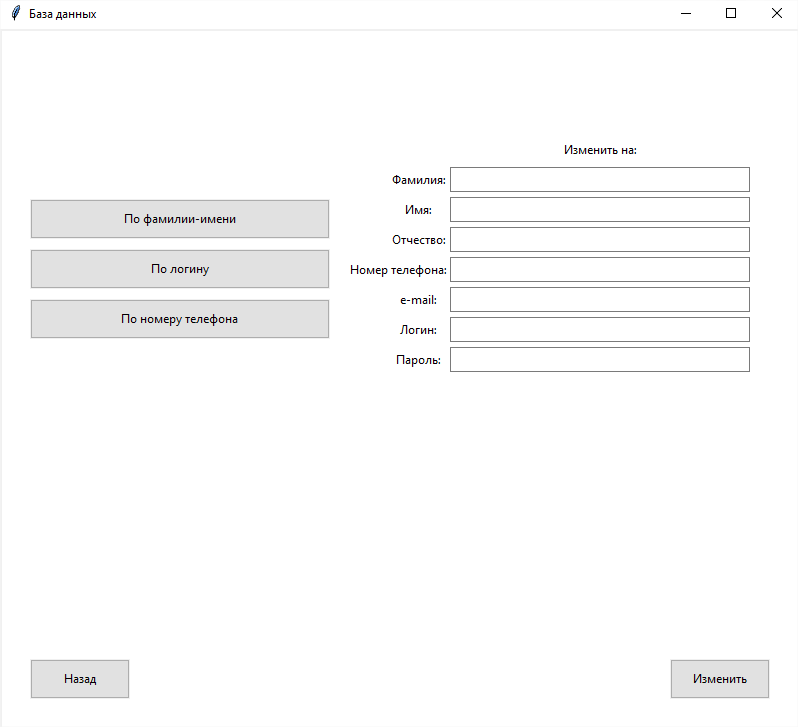


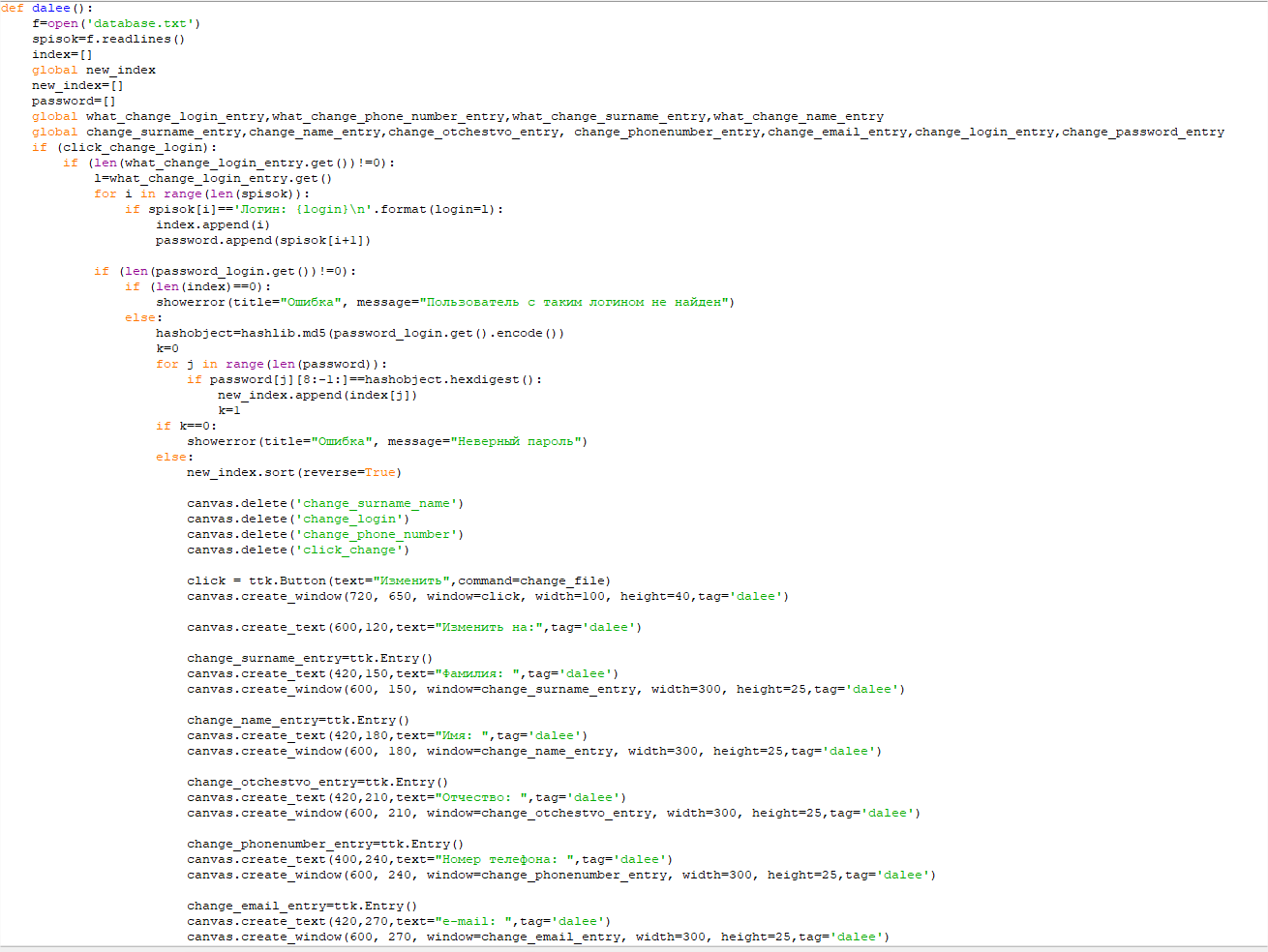




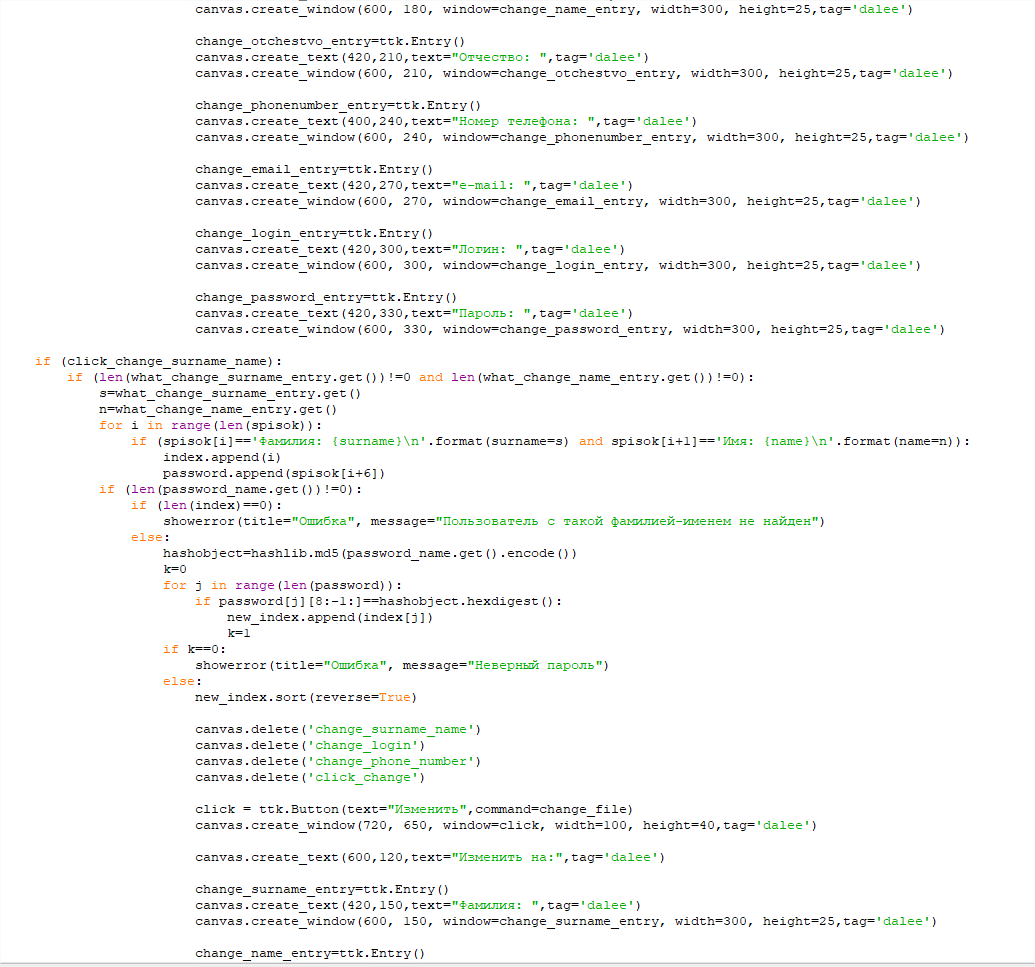
Если пользователь не найден, на экран выведется соответствующее сообщение. Если же пользователь найден, то происходит проверка пароля. Пароль хранится в форме ХЭШ-кода. Для того, чтобы сравнить пароли мы хэшируем введённый пароль с помощью вспомогательной функции hashlib.md5() и если захэшированные пароли совпадают, то после нажатия соответствующей кнопки «далее» вам предоставляются поля ввода новой информации.











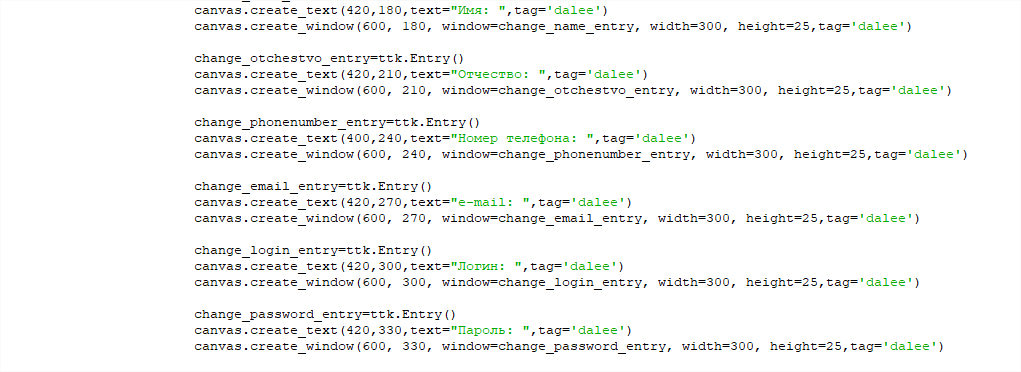
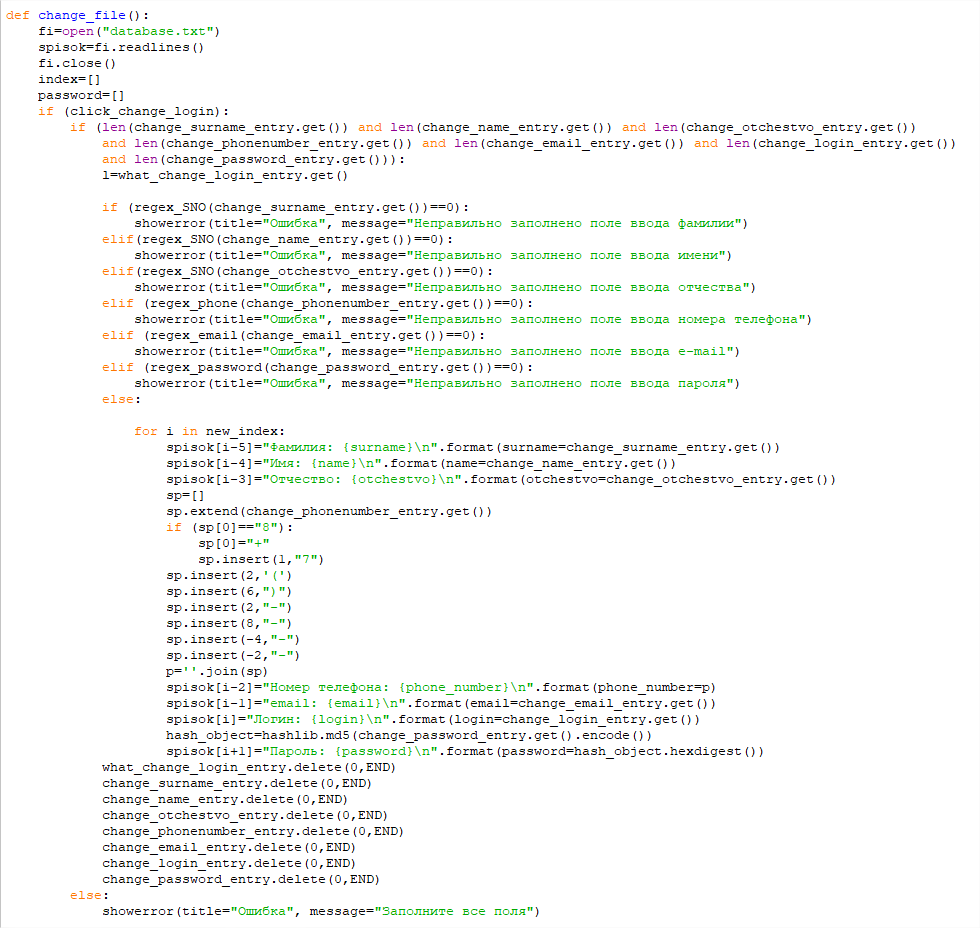


Рисунок 20 – реализация функции dalee()

После нажатия кнопки «изменить» информация о пользователе заменяется на новую.







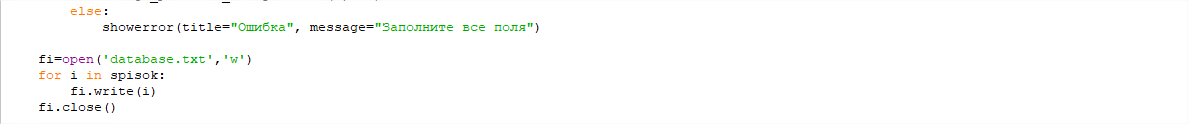
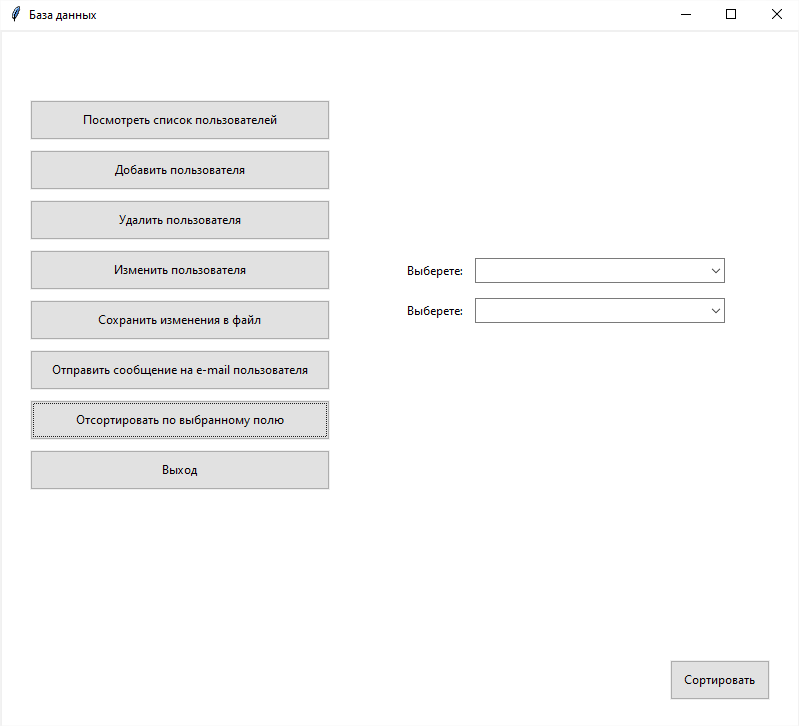


Рисунок 21 – реализация функции, изменяющей информацию о пользователе

При нажатии на кнопку «Отсортировать по выбранному полю» вам предоставляется выбор, по какому полю сортировать и как (по возрастанию или по убыванию).



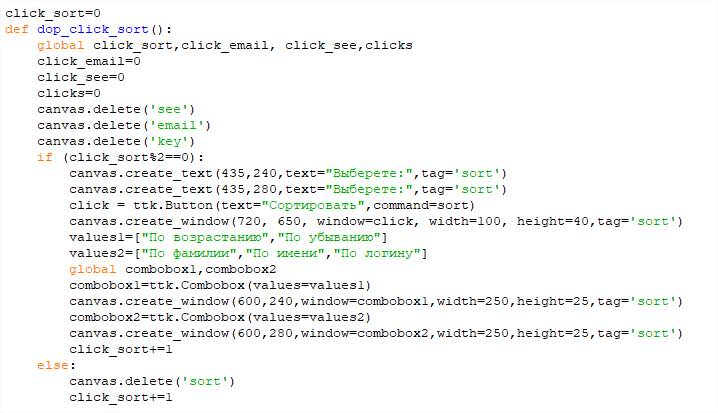


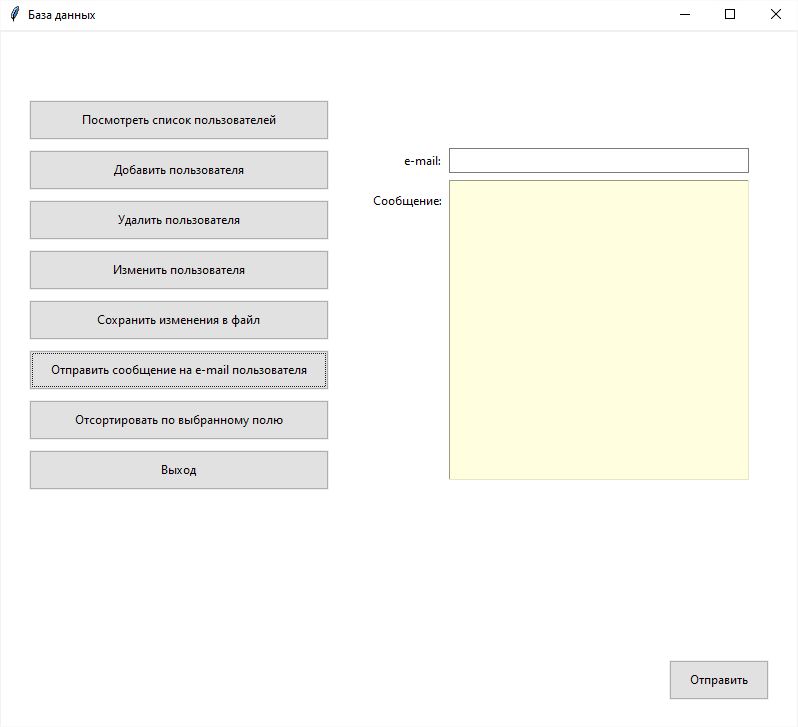
Рисунок 22 – реализация функции dop\_click\_sort(), предоставляющая дополнительный выбор

После нажатия соответствующей кнопки «Сортировать» мы получаем отсортированные данные о пользователях.



Рисунок 23 – реализация функции сортировки

При нажатии на кнопку «Отправить сообщение на email пользователя» вы увидите следующее:



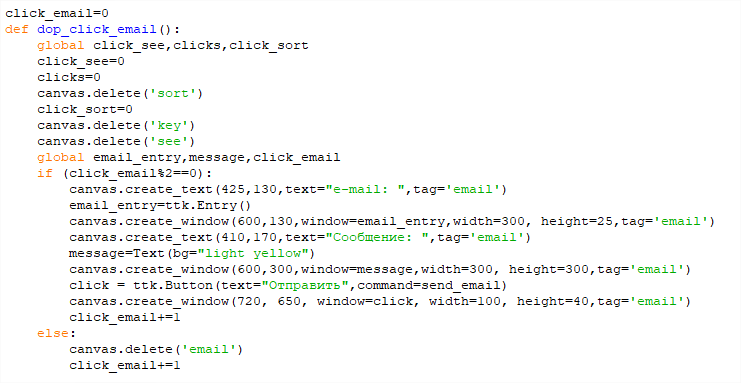


Рисунок 24 – реализация функции, предоставляющей поля ввода email и сообщения

После нажатия на кнопку «Отправить» сообщение отправляется на введённую вами электронную почту.



Рисунок 25 – функция отправки сообщения на почту

Функция writing() записывает информацию о пользователях в файл “database.txt”.

**3. Описание применяемых алгоритмов и решений**

Для хранения информации обо всех пользователях используются списки.

Поиск, за счёт которого происходит и изменение, и удаление пользователей по заданному полю, имеет асимптотическую сложность O(n), поскольку осуществляется просмотр поля для всех пользователей в базе данных.

Для проверки правильности ввода данных используются регулярные выражения, соответствующие условиям работы программы. Для регулярных функций была подключена библиотека re.

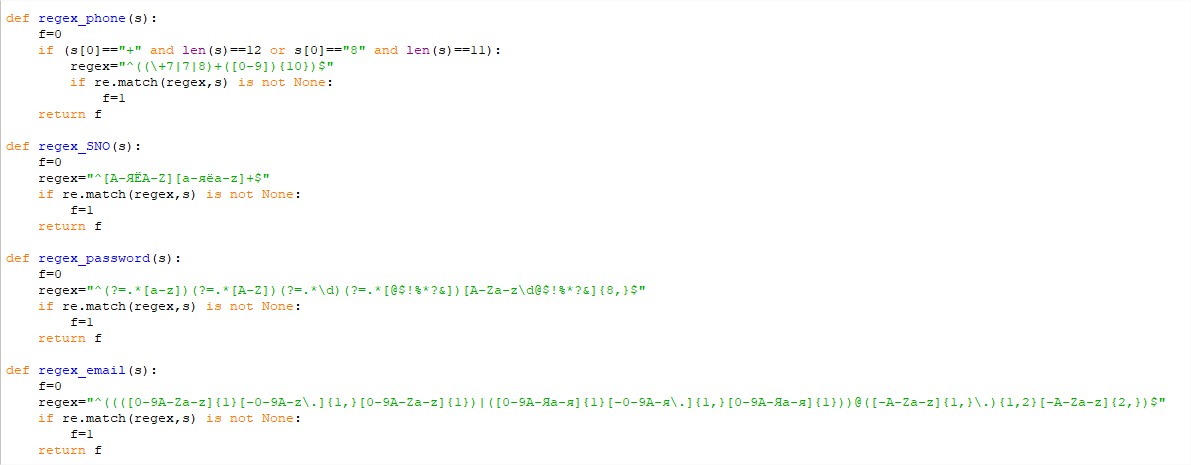


Рисунок 26 – реализация регулярных выражений

Пароли пользователей хранятся в формате ХЭШ-кодов. Для реализации хранения в данной форме используется ХЭШ-функция – функция, осуществляющая преобразование входных данных произвольной длины в выходную побитовую строку установленной длины, выполняемое определённым алгоритмом. Для написания функции подключается библиотека hashlib. С помощью функции hashlib.md5() мы хэшируем пароль и записываем его в переменную hash\_object и с помощью функции hexdigest() мы записываем захэшированный пароль в файл. Для отправки сообщений на электронную почту пользователя была подключена библиотека smtplib.



Рисунок 27 – реализация ХЭШ-функции

**4. Инструкция по эксплуатации**

При запуске программы пользователю представляется меню, которое предлагает выбрать одну из нескольких альтернатив. При выборе различных альтернатив может возникнуть предложение выбрать дальнейшую из предоставленных альтернатив.

При выборе альтернатив «Удалить пользователя» и «Изменить пользователя» пользователей должен будет выбрать одно из трёх полей для нахождения нужного пользователя. Пользователю потребуется ввести правильный пароль для выполнения действий.

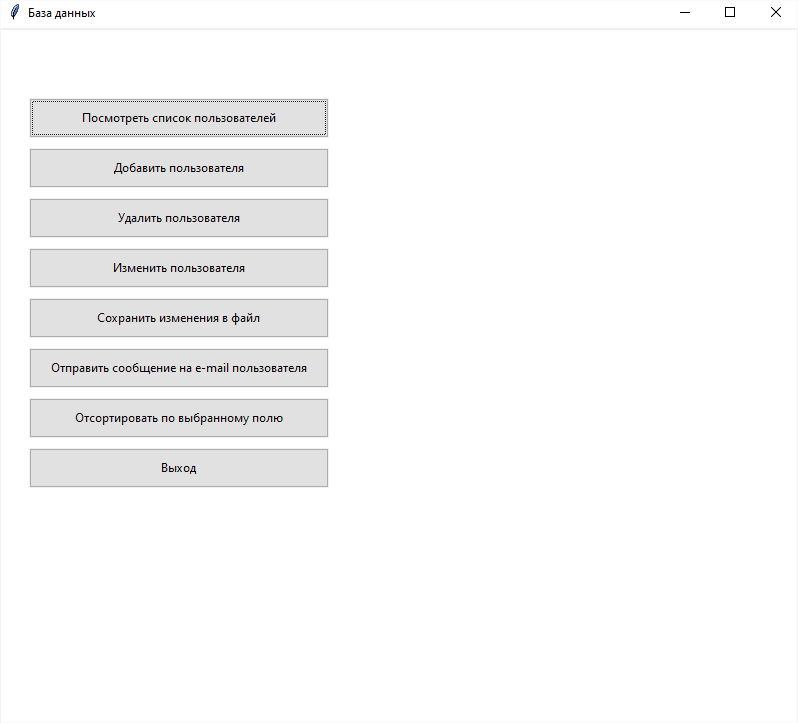


Рисунок 28 – начальное меню

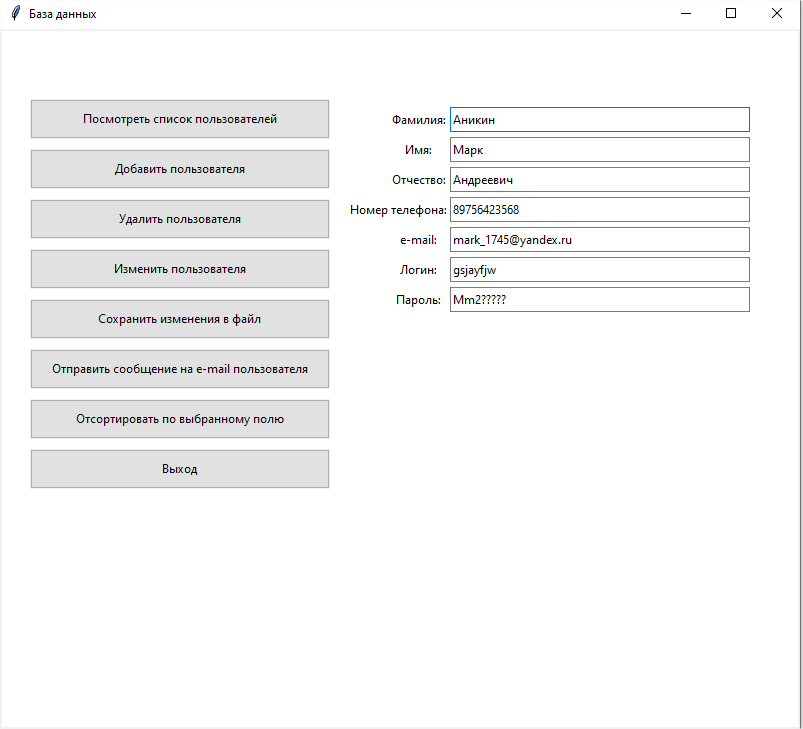


Рисунок 29 – добавление нового пользователя

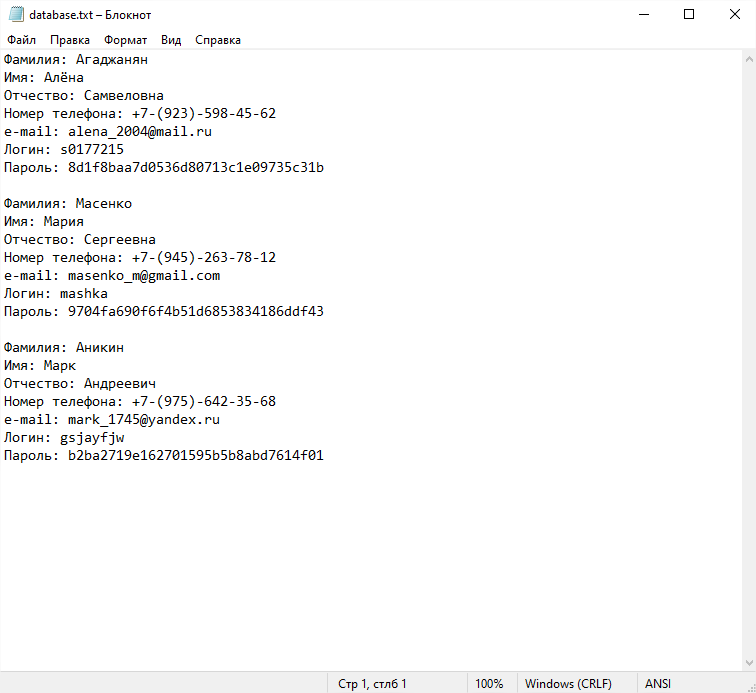


Рисунок 30 – данные в файле до удаления

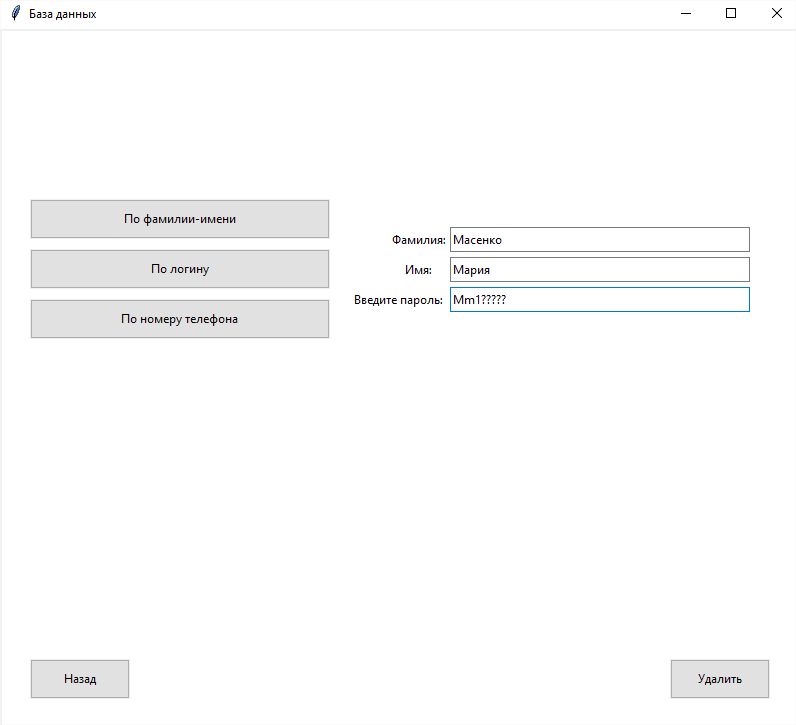


Рисунок 31 – удаление пользователя

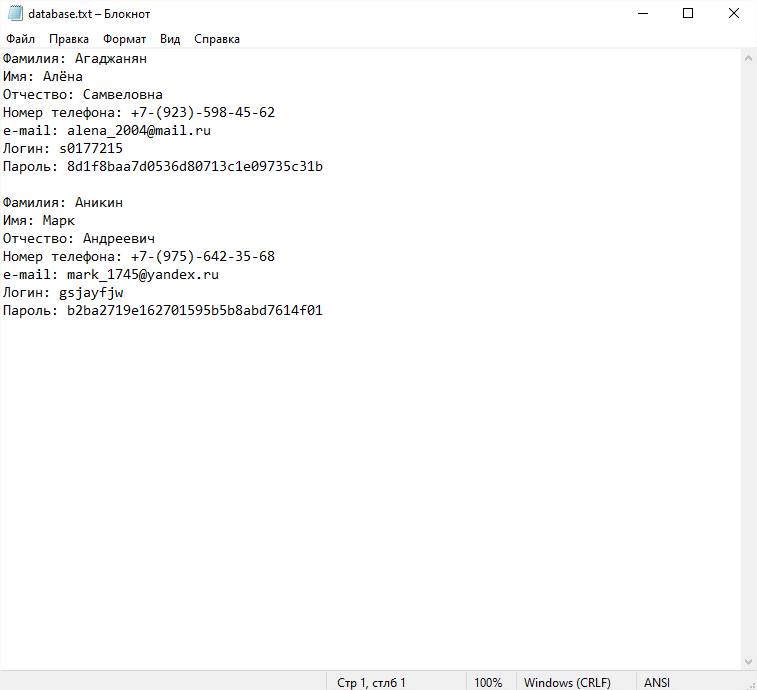


Рисунок 32 – данные в файле после удаления

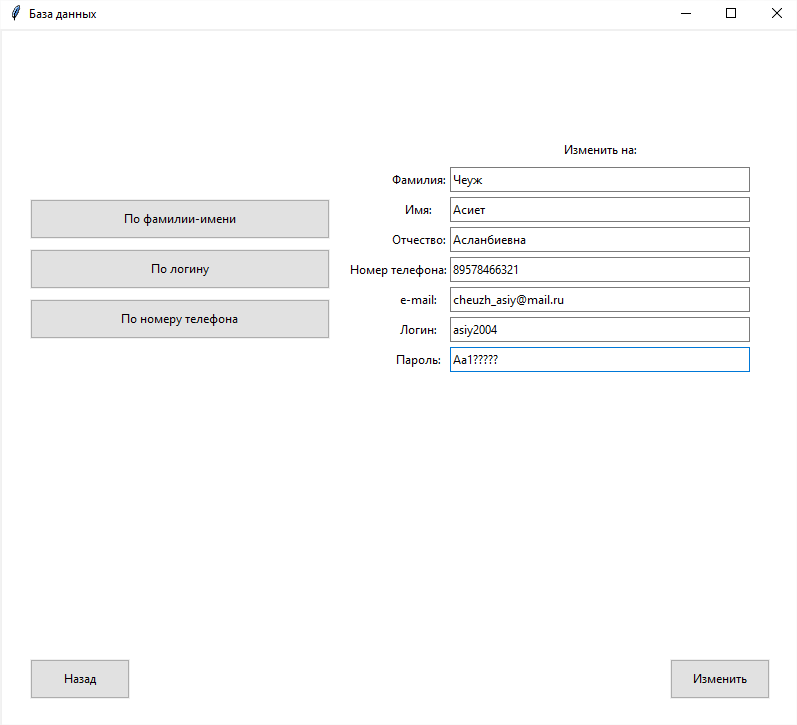


Рисунок 33 – изменение пользователя

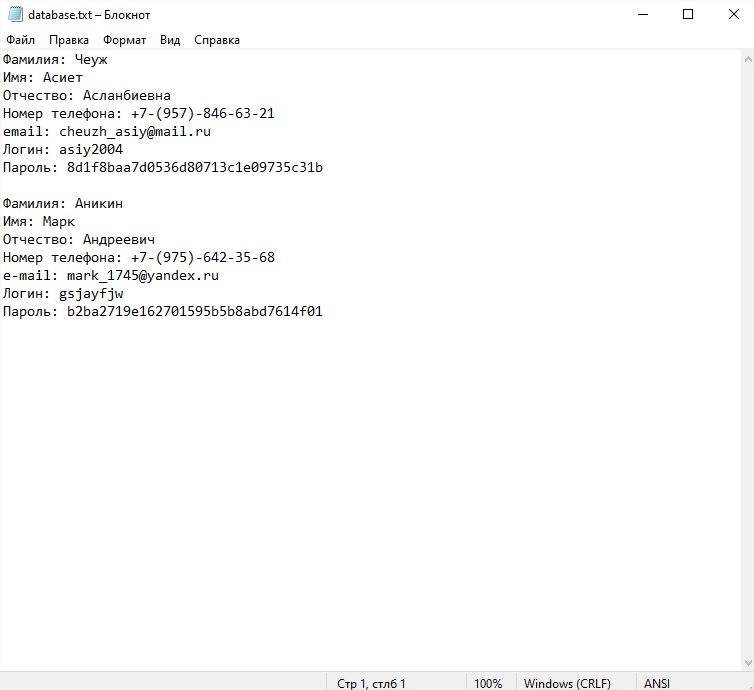


Рисунок 34 – данные в файле после изменения

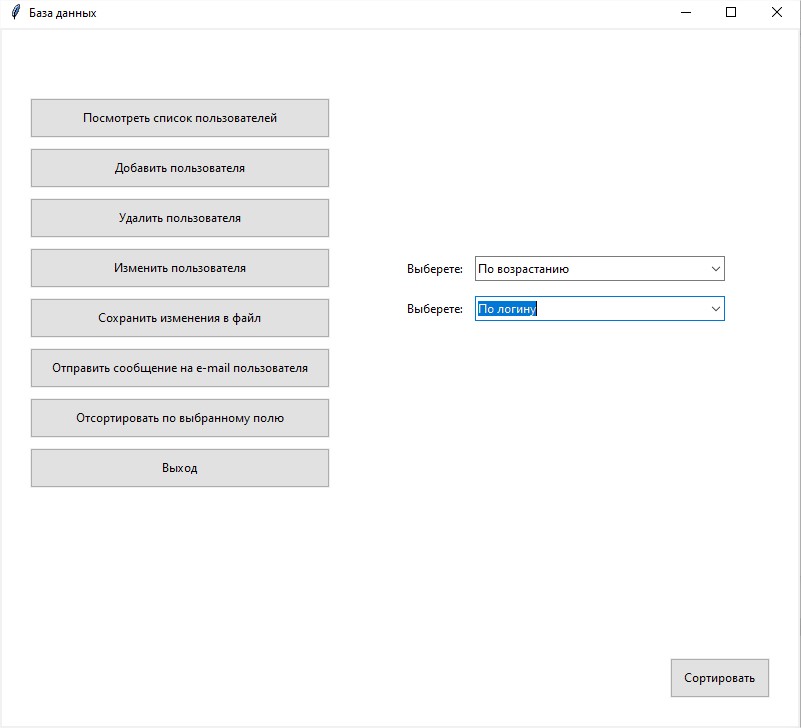


Рисунок 35 - сортировка

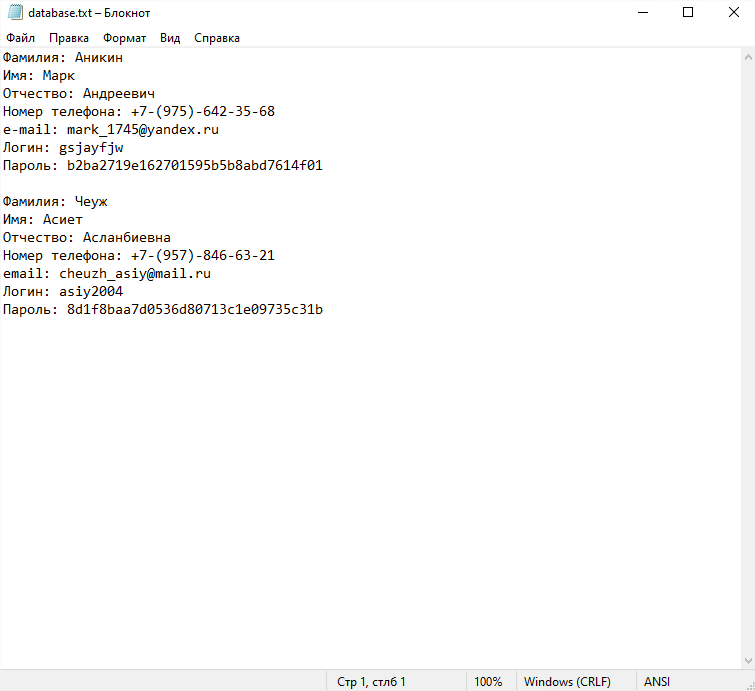


Рисунок 36 – данные в файле после сортировки

**5. Набор информации для тестирования**

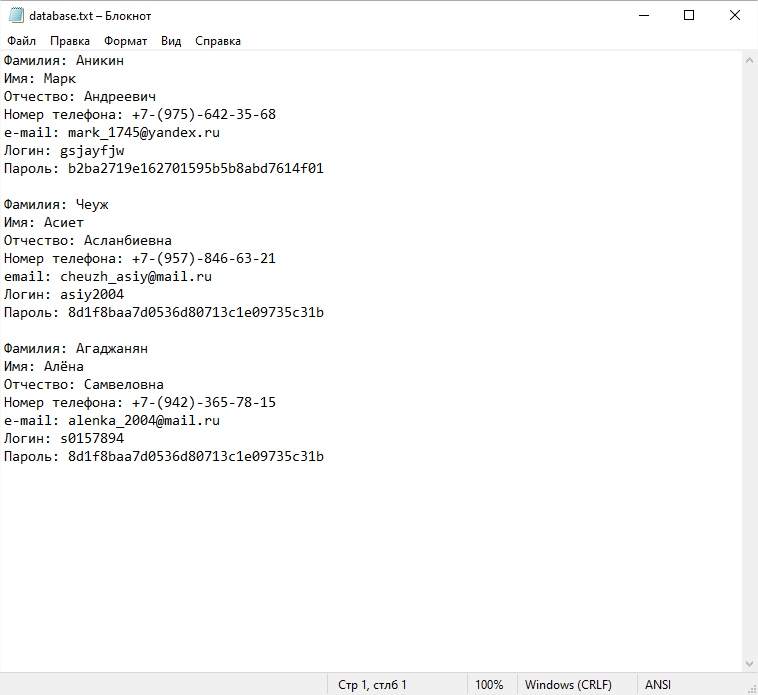


Рисунок 37 – данные в файле

**Заключение**

Приобретён опыт в реализации программных продуктов, основанных на знаниях, полученных за время обучения на первом курсе: разработана программа, которая вычисляет определённый интеграл полинома, разработан программный продукт, который находит решение системы линейных уравнений, представленных в матричном виде, методом Крамера, создано приложение для кодирования текста методом Лемпеля-Зива,, а также разработана база данных для хранения и изменения данных о пользователях.

**Список использованных источников**

1) В.С. Шипачёв. Высшая математика: Учеб. для вузов – 6-е изд., стер. – М.: Высш. Шк., 2003. – 479 с.: ил.

2) Фадеев Д. К. Лекции по алгебре: Учебное пособие. 5-е изд., стер. – СПБ.: Издательство «Лань», 2007 – 416 с.

3) Гошин, Егор Вячеславович. Теория информации и кодирования: учеб. пособие. – Самара: Изд-во Самарского университета, 2018. – 124 с.

1. Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, а также правилами внутреннего трудового распорядка. [↑](#footnote-ref-1)