Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования   
«Алтайский государственный технический университет

им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой

Преподаватель

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Отчет

по лабораторной работе №1

по дисциплине «Верификация и тестирование программного обеспечения»

Студент гр. ПИ-81

Рукин Н. В.

Преподаватель доцент,

Еремин Я. Б.

Барнаул 2022

Задание

1. Зарегистрироваться на Github и прислать мне свой логин, чтобы я мог добавить вас в репозиторий.  
  
2. Склонировать репозиторий, соответствующий вашему потоку, список репозиториев вот тут: https://github.com/yakov-eremin  
  
3. Разбиться на пары для выполнения работы.  
  
4. Выбрать какой-нибудь из ваших прошлых курсовых проектов, в котором есть графический интерфейс с достаточно большим набором контролов. Если в проекте всего пять виджетов и один пользовательский сценарий — этого недостаточно. В папке с проектом должна быть инструкция, как его собирать и запускать.  
  
5. Первый человек из пары создает новую ветку в git с названием, содержащим фамилии, например ivanov\_petrov, и заливает туда папку со своим проектом.  
  
6. Второй человек из пары переключается на эту же ветку, и заливает папку со своим проектом.  
  
7. Запустить проект своего напарника, сделать скриншоты, написать спецификацию в виде сценариев использования (стр. 23-26).

Спецификацию пишем не по фактической работе программы, а по тому, как она должна работать, по-вашему мнению.

Как будто у вас есть только макет интерфейса и описание требуемого функционала.    
  
8. Создать спецификацию в виде Use Case UML диаграмм (стр. 27-29 и https://habr.com/ru/post/566218/).

Делать можно в любом редакторе, например https://online.visual-paradigm.com/diagrams/solutions/free-use-case-diagram-tool/  
  
9. Залить в репозиторий папку с созданными файлами, внутрь каталога напарника, чтобы в итоге получилась примерно такая структура:  
  
  
  
Это не жесткое требование к структуре, можно сделать спецификацию в едином документе (по сути это всё равно нужно для отчета), можно разбить на подпапки и т.д. Главное — содержание, а не форма.  
  
10. Сделать отчет о проделанной работе и залить на ИЛИАС.

**Сценарии использования:**

***1. Прямые методы решения СЛАУ.***

Действующее лицо: пользователь приложения.

Предусловие: пользователь во вкладке "Прямые методы решения СЛАУ".

Основной сценарий (Решение методом Гаусса):

• В полях для матрицы и вектора вписать значения согласно инструкциям прямо над полями ввода.

• Нажать кнопки "Сохранить матрицу" и "Сохранить вектор".

• Используемый метод решения выбрать "Метод Гаусса".

• Нажать «Решить».

Постусловие: справа появится решение системы, число решений, оценка точности решения, обратная матрица, невязки, произведение норм и определитель.

Альтернативный сценарий (Решение методом квадратного корня):

• В полях для матрицы и вектора вписать значения согласно инструкциям прямо над полями ввода. Матрица должна быть симметричной.

• Нажать кнопки "Сохранить матрицу" и "Сохранить вектор".

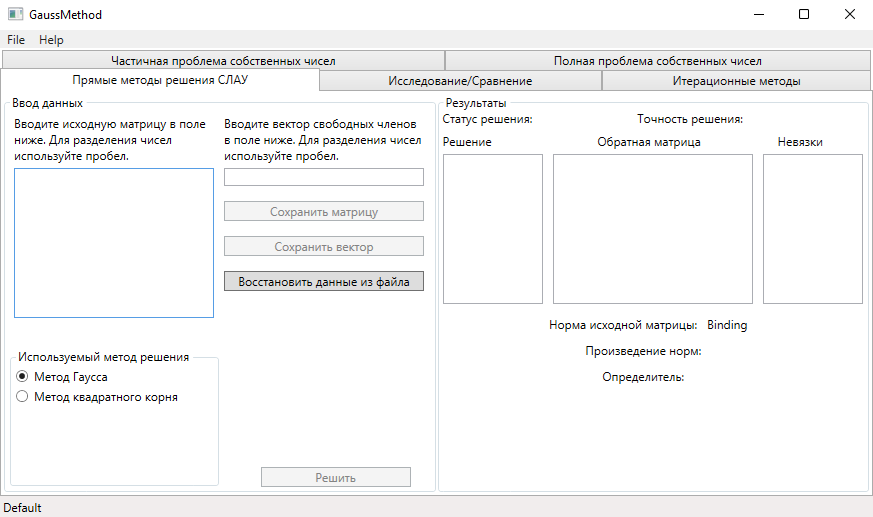
• Используемый метод решения выбрать "Метод квадратного корня".

• Нажать «Решить».

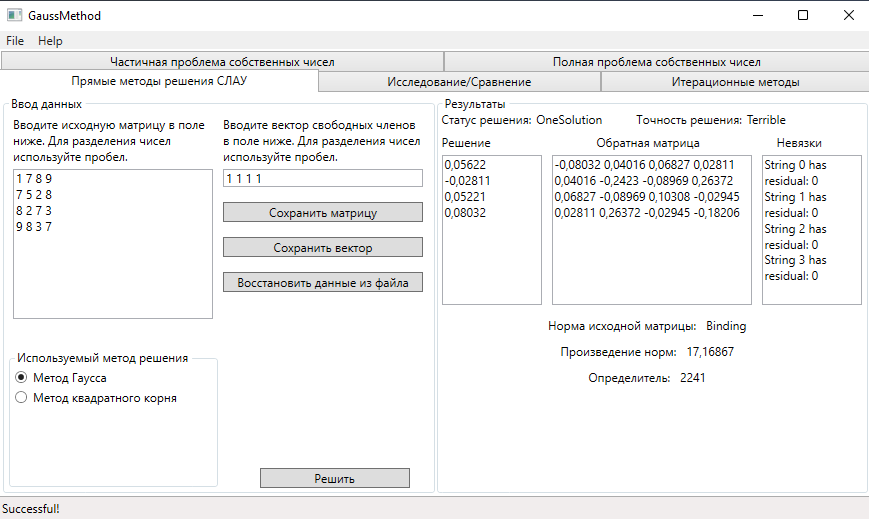
Постусловие: справа появится решение системы, число решений, оценка точности решения, обратная матрица, невязки, произведение норм и определитель.

Скриншоты:

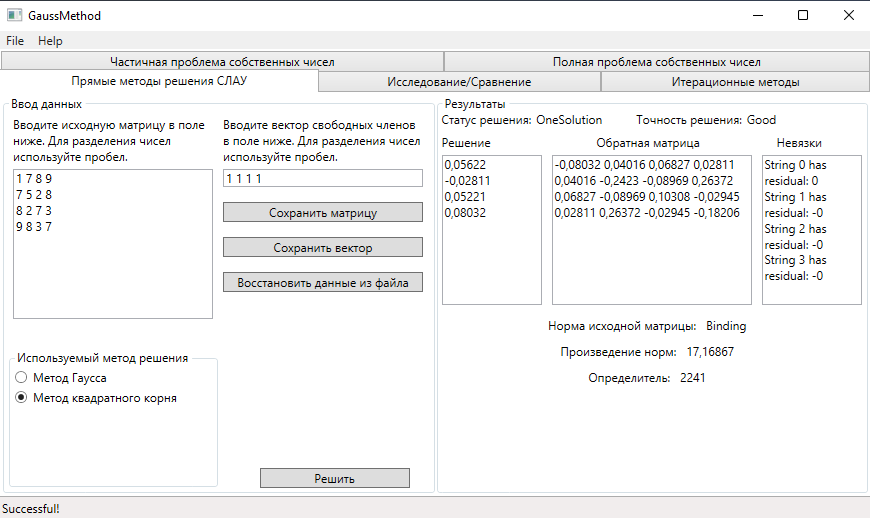
До



После



ПОСЛЕ (альтернативный сценарий)



***2. Сравнение решений методом Гаусса и квадратного корня.***

Действующее лицо: пользователь приложения.

Предусловие: пользователь во вкладке "Исследование/Сравнение". Были сохранены матрица и вектор во вкладке "Прямые методы решения СЛАУ".

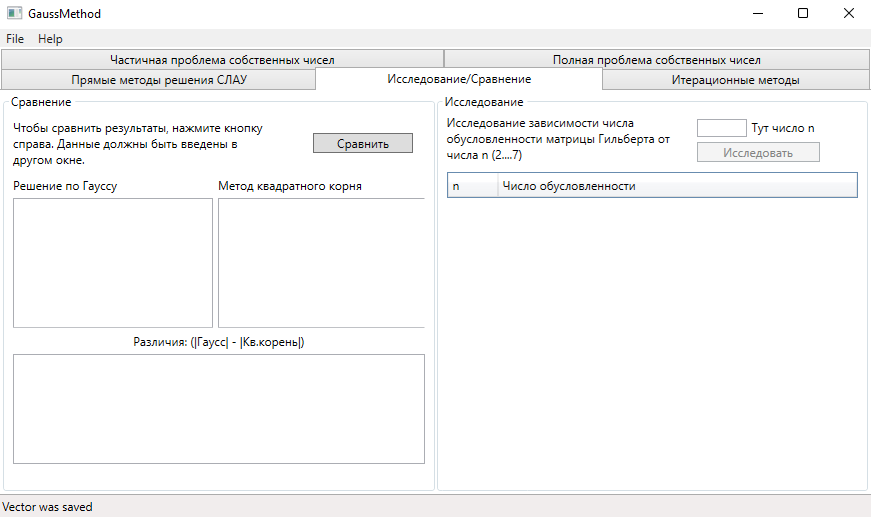
Основной сценарий:

• Нажать "Сравнить".

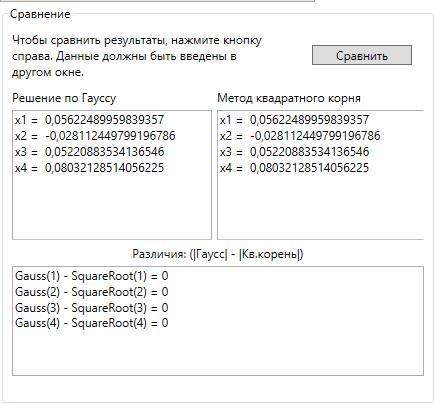
Постусловие: Будут выведены точные решения из файлов, сохраненных на предыдущем шаге. Показаны различия в ответах.

Скриншоты:

До



После



***3. Исследование зависимости числа обусловленности матрицы Гильберта от числа.***

Действующее лицо: пользователь приложения.

Предусловие: пользователь во вкладке "Исследование/Сравнение".

Основной сценарий:

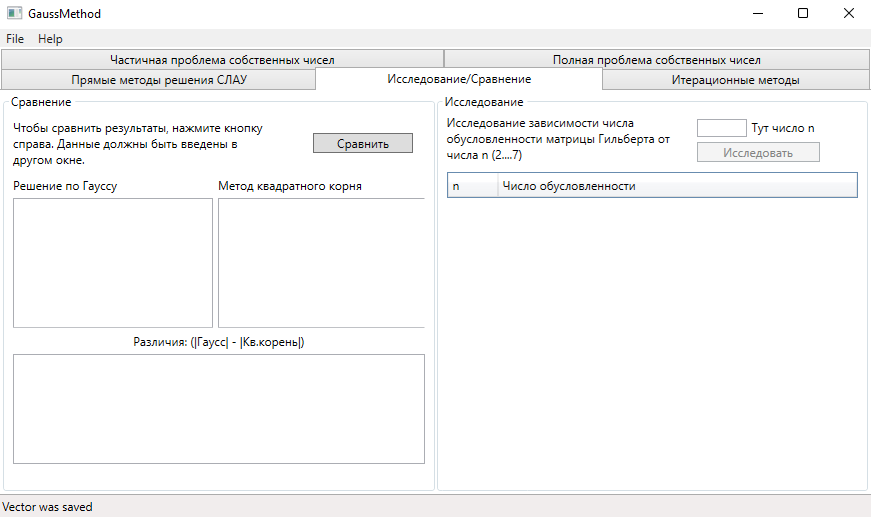
• Ввести число n (от 2 до 7)

• Нажать "Исследовать".

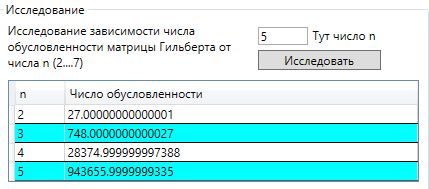
Постусловие: Будут выведены числа обусловленности матрицы Гильберта от n=2 до выбранного n.

Скриншоты:

До



После



***4. Решение частичной проблемы собственных чисел***

Действующее лицо: пользователь приложения.

Предусловие: пользователь во вкладке "Частичная проблема собственных чисел".

Основной сценарий:

• В полях для матрицы и вектора вписать значения согласно инструкциям прямо над полями ввода.

• Нажать кнопки "Сохранить матрицу" и "Сохранить вектор".

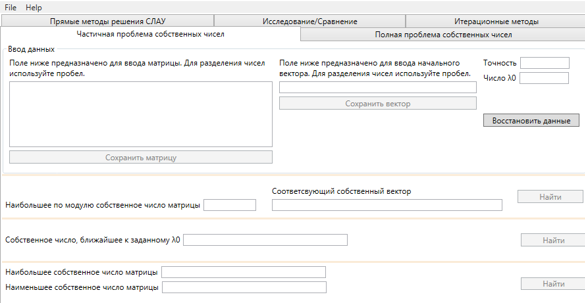
• Справа ввести число лямбда и точность вычислений.

• Справа нажать "Найти" для вывода данных, полученных данным способом.

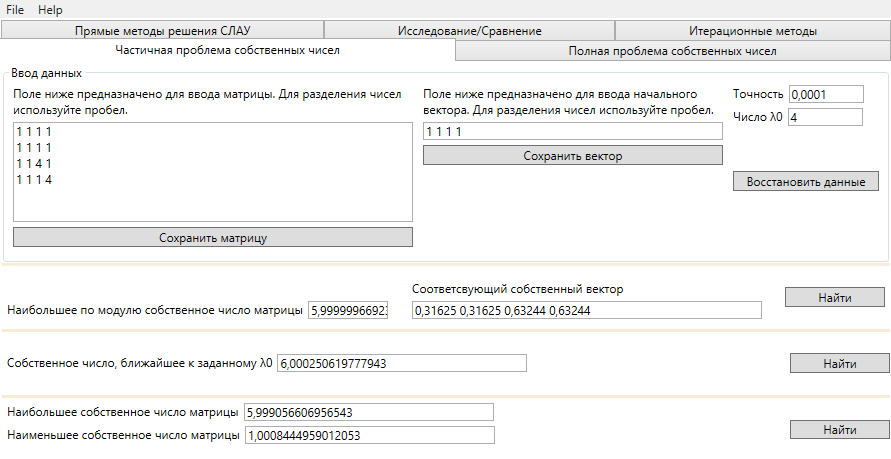
Постусловие: Будут выведены: наибольшее и наименьшее собственное число, соответствующий собственный вектор, собственное число приближенное к лямбда.

Скриншоты:

До



После



***5. Решение полной проблемы собственных чисел***

Действующее лицо: пользователь приложения.

Предусловие: пользователь во вкладке "Полная проблема собственных чисел".

Основной сценарий:

• В поле для матрицы вписать значения согласно инструкциям прямо над полями ввода.

• Нажать "Сохранить матрицу".

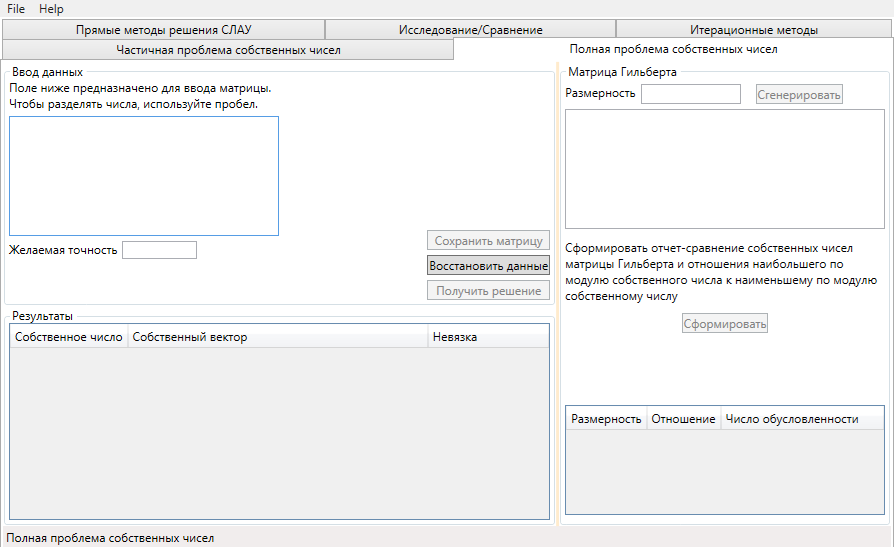
• Справа ввести точность вычислений.

• Нажать "Получить решение".

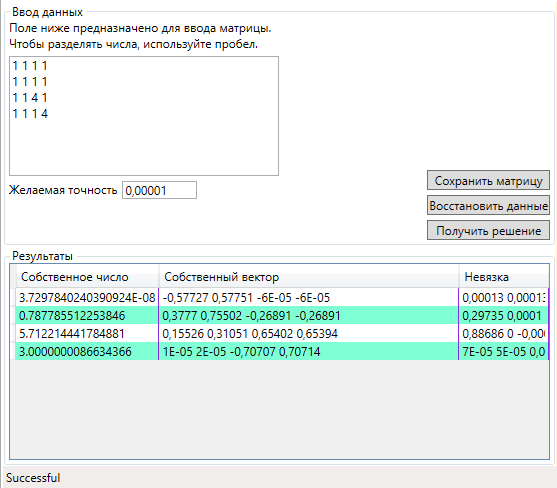
Постусловие: Будут выведены: все собственные числа матрицы, соответствующие им вектора и невязки.

Скриншоты:

До



После



UML-диаграмма

