Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«Сравнение разных методов сортировки чисел»**

**Выполнил**:

Студент группы 3824Б1ПМ1

Сорокин Д.Д.

**Проверила**:

Бусько П.В.

Нижний Новгород

2025

**Содержание**

[Введение 3](#_Toc185943764)

[Постановка задачи 4](#_Toc185943765)

[Руководство пользователя 5](#_Toc185943766)

[Описание программной реализации 6](#_Toc185943767)

[Результаты экспериментов 12](#_Toc185943768)

[Заключение 13](#_Toc185943769)

[Литература 14](#_Toc185943770)

[Приложение 15](#_Toc185943771)

# Введение

Системы линейных алгебраических уравнений (СЛАУ) играют ключевую роль в прикладных науках и программировании. Они используются для моделирования физических явлений, анализа электрических схем, оптимизации процессов, обработки данных, машинного обучения и многих других областей.

Метод Гаусса — один из самых популярных и универсальных способов решения СЛАУ. Его суть заключается в последовательном преобразовании расширенной матрицы системы к верхнетреугольному виду, после чего решение находится обратной подстановкой. Благодаря простоте, эффективности и наглядности этот метод широко применяется как в учебных программах, так и в реальных инженерных задачах.

Разработка программной реализации метода Гаусса имеет большое значение для обучения. Она помогает студентам развивать навыки программирования, работать с многомерными массивами, организовывать ввод и вывод данных, а также учитывать точность и устойчивость численных алгоритмов. Анализ систем на совместность и определение количества решений углубляют понимание математических основ линейной алгебры.

Таким образом, выполнение этой лабораторной работы способствует освоению студентами фундаментальных принципов численных методов и разработки программного обеспечения для решения практических задач анализа и моделирования.

# Постановка задачи

1. Реализовать шаблонный класс Vector для работы с одномерными массивами
2. Создать класс Matrix для представления квадратных матриц
3. Разработать класс SLAU, наследующий от Matrix, с методом Гаусса
4. Реализовать функции для перестановки строк и элементов

.

# Руководство пользователя

Программа сделана максимально просто. Пользователь выбирает, размерность матрицы (рис. 1). Для примера я выбрал размерность 2x2. Затем пользователь вводит коэффициенты матрицы (слева направо, сверху вниз) (рис. 2) Для примера, это будут числа 1,2,3,4. Затем пользователь вводит правую часть (рис. 3) Для примера введем 5,6. В конце пользователь получает решение (или его отсутствие) системы уравнений (рис. 4). Рисунок 5 – пример системы.



Рис.1 – Пользователь выбирает размерность.

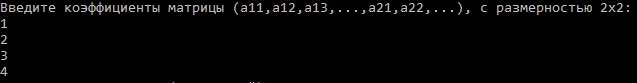


Рис.2 – Пользователь вводит коэффициент.



Рис.3 – Пользователь вводит правую часть.



Рис.4 – Пользователю выводится решение.

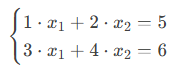


Рис.5 – Пример системы.

# Описание программной реализации

Требовалось реализовать решение через классы. Это и было сделано.

Список классов:

1/ Vector

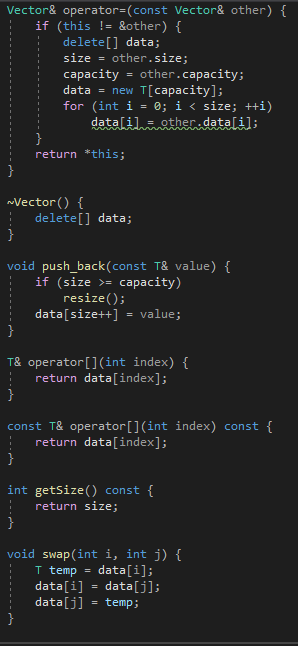
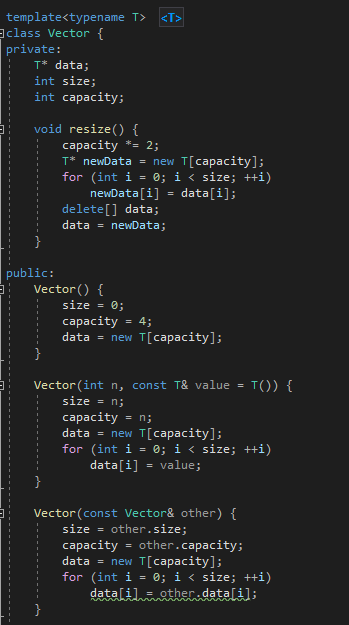


Рис. 6-7 – Реализация класса Vector

Класс Vector представляет собой упрощённую реализацию динамического массива (аналог std::vector из стандартной библиотеки C++). Он предназначен для:

* Хранения элементов произвольного типа (благодаря шаблону template<typename T>).
* Автоматического управления памятью (динамическое расширение при добавлении элементов).
* Предоставления базовых операций для работы с последовательностями данных.

Содержит несколько конструкторов и деструктор. Также есть несколько функций (getSize – узнать размер, swap – поменять 2 элемента местами и push\_back – добавить элемент в конец вектора).

2/ Matrix

представляет собой квадратную матрицу с элементами произвольного типа T

Конструктор создает матрицу размера n на n, как вектор векторов.

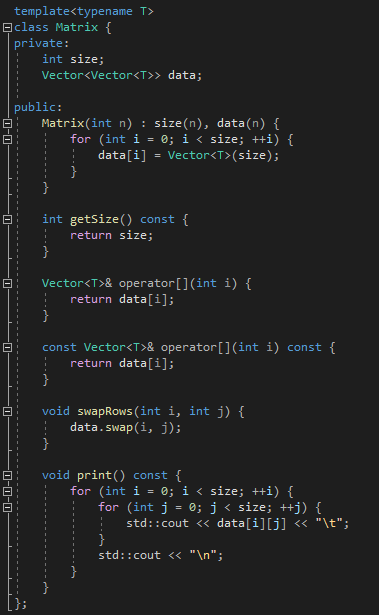
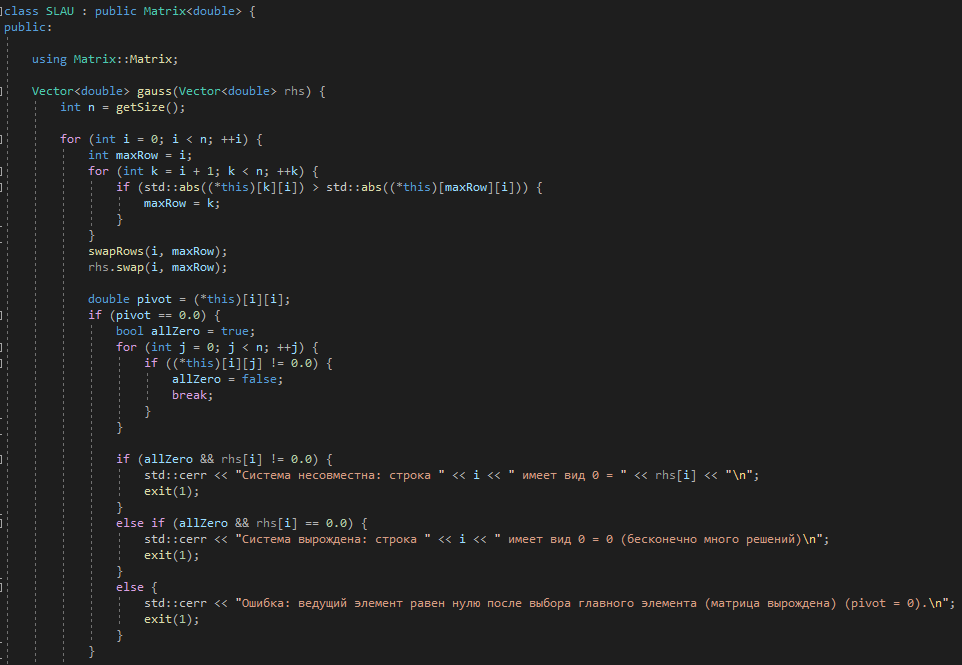
Есть функции: print – вывод матрицы, swapRows- поменять строки местами, getSize – вывести размер.  


Рис. 8 – Реализация класса Matrix

3/ SLAU



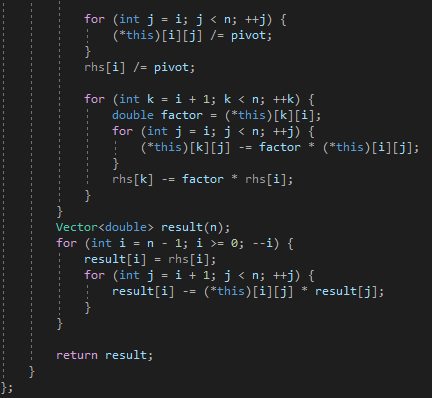


Рис. 9-10 – Реализация класса SLAU

SLAU (Система Линейных Алгебраических Уравнений), который наследуется от Matrix<double> и содержит метод gauss() для решения СЛАУ методом Гаусса с выбором главного элемента. Метод gauss Принимает вектор правых частей rhs и возвращает вектор решений.

**Ключевые особенности**

* 1. Выбор главного элемента: улучшает устойчивость метода.
  2. Проверка на совместность: обнаруживает вырожденные случаи.
  3. Треугольный вид: матрица преобразуется "на месте" для экономии памяти.
  4. Выход при ошибках: Программа завершается с сообщением, если система не имеет решений.

В main Программа должна:

1. Запросить у пользователя размер системы линейных уравнений N x N.
2. Считать матрицу коэффициентов и правую часть системы.
3. Решить систему методом Гаусса.
4. Вывести решение.

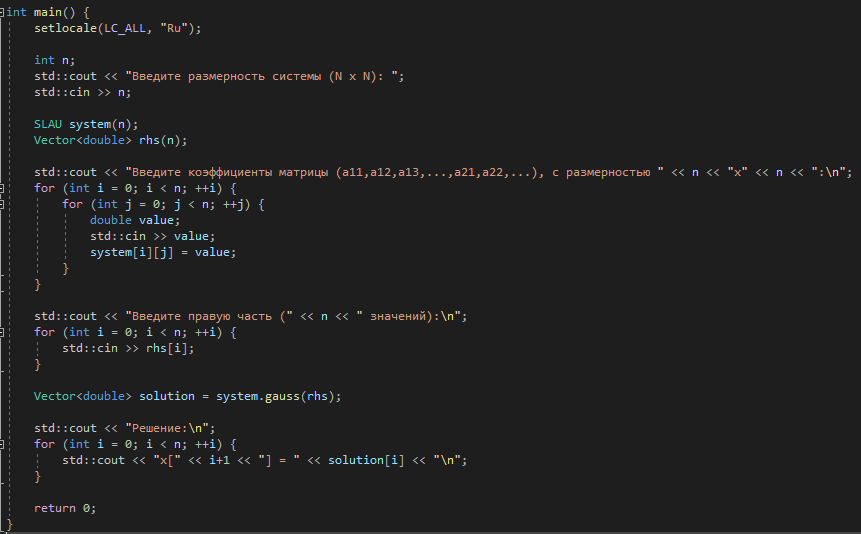


Рис. 11 – Реализация main.

# Результаты экспериментов

В рамках выполнения лабораторной работы была разработана и протестирована программа для решения систем линейных алгебраических уравнений (СЛАУ) методом Гаусса. Для проверки корректности работы алгоритма был проведен эксперимент: проверка правильности решения при известном ответе.

Например, возьмем систему:

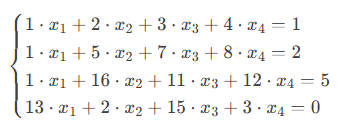


Рис. 12 – Случайная система уравнений

Проверим, какой ответ выдаст программа:

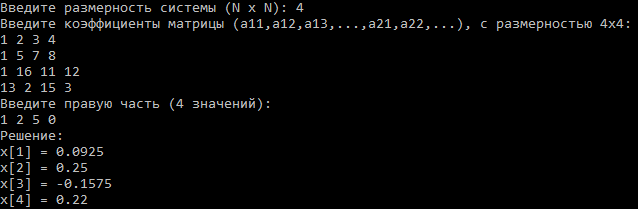


Рис. 13 – Проверка при случайной системе уравнений.

Сравним с верным ответом (там x[1], x[2], x[3], x[4] обозначены как x, y, z, t)

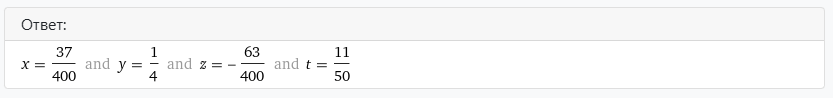


Рис. 14 – правильный ответ случайной системы уравнений.

Ответы совпали, программа работает правильно.

# Заключение

В ходе выполнения лабораторной работы была успешно реализована программа для решения систем линейных алгебраических уравнений (СЛАУ) методом Гаусса. Основные этапы работы включали:

1. **Разработку классов**:
   * Шаблонный класс Vector для хранения и управления динамическими массивами.
   * Класс Matrix, представляющий квадратные матрицы на основе Vector<Vector<T>>.
   * Класс SLAU, наследующий Matrix<double>, с методом gauss() для решения СЛАУ.
2. **Реализацию алгоритма**:
   * Метод Гаусса с выбором главного элемента для устойчивости вычислений.
   * Проверку на совместность системы (вырожденность и несовместность).
   * Обратный ход для нахождения решения.
3. **Тестирование**:
   * Проверка на корректных данных (системы с единственным решением).
   * Анализ особых случаев (вырожденные и несовместные системы).
   * Демонстрация работы программы на примерах, включая систему с заданной матрицей и правой частью.

**Результаты**:

* Программа корректно решает СЛАУ с квадратными матрицами.
* Обнаруживает и обрабатывает вырожденные и несовместные системы.
* Выводит понятные сообщения об ошибках.

# Литература

1. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования СИ //М.: Финансы и статистика. – 1992.
2. Кнут Д. Э. Искусство программирования: Сортировка и поиск. – Издательский дом Вильямс, 2000. – Т. 3.

# Приложение

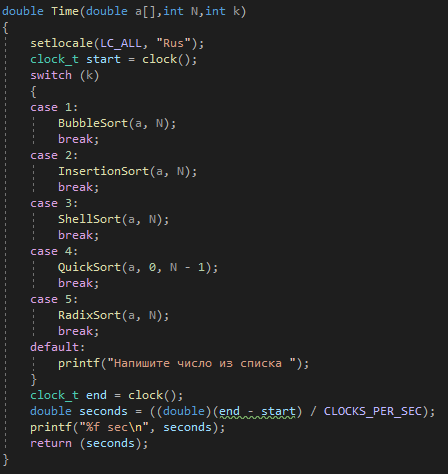


Рис.10 – Реализация функции Time

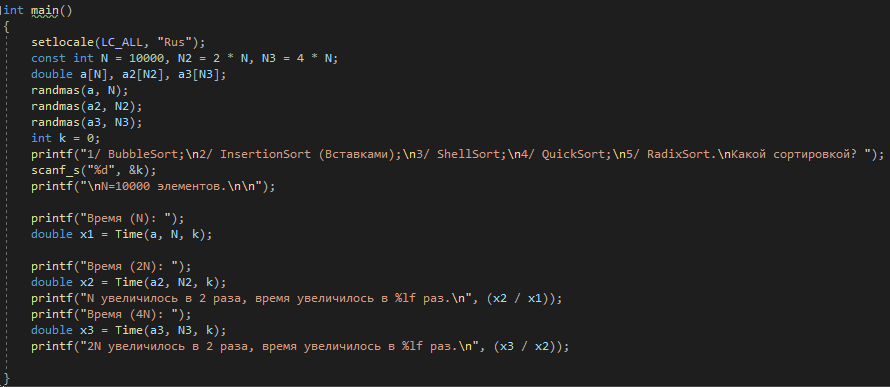


Рис.11 – Основная часть программы.