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**

**«Метод Гаусса»**

**Выполнила**:

студент группы 3821Б1ПМ2

Головин Р.М.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2022

Оглавление

[Постановка задачи 3](#_Toc105323402)

[Метод решения 4](#_Toc105323403)

[Руководство пользователя 6](#_Toc105323404)

[Описание программной реализации 7](#_Toc105323405)

[template class Container 7](#_Toc105323406)

[template class Matrix 9](#_Toc105323407)

[class SLAE 10](#_Toc105323408)

[Подтверждение корректности 12](#_Toc105323409)

[Заключение 13](#_Toc105323410)

# [Постановка задачи](file:///C:\Users\syato\1_КУРС\ЯиМП\лабараторная%20работа%201\mp1-3821B1PM2\SyatovNikita\Lab_Sort_1\Отчёт%20по%20лабароторной%20работе1.docx#_bookmark0)

Используя язык программирования Си++ реализовать шаблон класса вектор, шаблон класса матрица и СЛАУ. Класс вектор хранит в себе данные, позволяет добавлять их, менять местами и производить арифметические операции над ними. Класс матрица наследуется от класса вектор и расширяет его функционал. Класс СЛАУ включает в себя класс вектор и класс матрица. Слау должен содержать в себе метод ,решающий СЛАУ методом Гаусса и метод производящий проверку.

# [Метод решения](file:///C:\Users\syato\1_КУРС\ЯиМП\лабараторная%20работа%201\mp1-3821B1PM2\SyatovNikita\Lab_Sort_1\Отчёт%20по%20лабароторной%20работе1.docx#_bookmark1)

**Метод Гаусса** — классический метод решения [системы линейных алгебраических уравнений](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D1%8B%D1%85_%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D1%85_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B9) (СЛАУ). Назван в честь немецкого математика [Карла Фридриха Гаусса](https://ru.wikipedia.org/wiki/%D0%93%D0%B0%D1%83%D1%81%D1%81,_%D0%9A%D0%B0%D1%80%D0%BB_%D0%A4%D1%80%D0%B8%D0%B4%D1%80%D0%B8%D1%85). Это метод последовательного исключения [переменных](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B2%D0%B5%D0%BB%D0%B8%D1%87%D0%B8%D0%BD%D0%B0), когда с помощью элементарных преобразований система уравнений приводится к равносильной системе треугольного вида, из которой последовательно, начиная с последних (по номеру), находятся все переменные системы.

Алгоритм решения [СЛАУ](https://ru.wikipedia.org/wiki/%D0%A1%D0%9B%D0%90%D0%A3) методом Гаусса подразделяется на два этапа:

* На первом этапе осуществляется прямой ход, когда путём [элементарных преобразований](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%BD%D1%8B%D0%B5_%D0%BF%D1%80%D0%B5%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D1%8B) над строками систему приводят к ступенчатой  [форме](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0), либо устанавливают, что система несовместна. Для этого среди элементов первого столбца матрицы выбирают максимальный, перемещают содержащую его строку в крайнее верхнее положение, делая эту строку первой. Далее ненулевые элементы первого столбца всех нижележащих строк обнуляются путём [вычитания](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%87%D0%B8%D1%82%D0%B0%D0%BD%D0%B8%D0%B5) из каждой строки первой строки, до множенной на отношение первого элемента этих строк к первому элементу первой строки. После того, как указанные преобразования были совершены, переходят к следующей строке и столбцу. Если на какой-то из итераций среди элементов первого столбца не нашёлся ненулевой, то переходят к следующему столбцу и проделывают аналогичную операцию.
* На втором этапе осуществляется так называемый обратный ход, суть которого заключается в том, чтобы выразить все получившиеся базисные переменные через небазисные и построить [фундаментальную систему решений](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%B4%D0%B0%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9), либо, если все переменные являются базисными, то выразить в численном виде единственное решение системы линейных уравнений. Эта процедура начинается с последнего уравнения, из которого выражают соответствующую базисную переменную (а она там всего одна) и подставляют в предыдущие уравнения, и так далее, поднимаясь по «ступенькам» наверх. Каждой строчке соответствует ровно одна базисная переменная, поэтому на каждом шаге, кроме последнего (самого верхнего), ситуация в точности повторяет случай последней строки.

Метод Гаусса требует {\displaystyle O(n^{3})} О(n^3) арифметических операций.

# [Руководство пользователя](file:///C:\Users\syato\1_КУРС\ЯиМП\лабараторная%20работа%201\mp1-3821B1PM2\SyatovNikita\Lab_Sort_1\Отчёт%20по%20лабароторной%20работе1.docx#_bookmark2)

Программа запрашивает размер квадратной матрицы, который не может превышать 2147483647.После выводится случайно сгенерированная расширенная матрица указанного размером, затем выводится единичная расширенная матрица. После выводится результат проверки в формате: i строка матрицы А умноженная на вектор Х | разница между полученным произведением и i элементом вектора В | “Wrong” если разница превышает10-15 ,иначе – “Correct”.

# Описание программной реализации

## template class Container

private:

unsigned int length – переменная, хранящая длину выделенной памяти.

unsigned int last – переменная, хранящая индекс последнего записанного элемента.

T \*data – указатель на память , хранящею данные.

public:

**Container(unsigned int length) -** конструктор класса ,позволяющий задать длину. При вызове без параметров задается длинна равная 1.

* unsigned int length – длинна контейнера.

**Container(const Container &c)** – конструктор копирования.

**~Container()** - деструктор

**void append(const T &a)** - метод добавляет элемент типа Т в контейнер

* const T &a – константная ссылка на добавляемый элемент.

**void swap(unsigned int first, unsigned int second) -** меняет местами два элемента контейнера с указанными индексами. Если хотя бы один индекс больше “last” метод бросит исключение.

* unsigned int first – индекс первого элемента.
* unsigned int second – индекс второго элемента.

**void addData(unsigned int first, unsigned int second, double k) –** метод прибавляет строку с индексом “second” умноженную на коэффициент “к” к строке с индексом “first”. Если хотя бы один индекс больше “last” метод бросит исключение.

**unsigned int getLength() const –** метод возвращает“last”.

Перегруженные операторы:

**T const &operator [](int ind) const**

**T &operator [](int ind)**

**void operator +=(const Container<T> &c)**

**Container<T> &operator \*=(const T c)**

**T operator \*(const Container<T> &c) const**

**Container operator =(const Container &c)**

## template **class Matrix**

public:

**unsigned int getColumnMaxInd(unsigned int columnInd) const –** метод возвращает индекс максимального по модулю элемента в столбце. Если столбец нулевой, то метод вернет -1;

**void append(const Container<T>& c) –** метод добавляет контейнер типа Т в матрицу. Если длинна добавляемого контейнера не соответствует длине первого ,то метод выдаст исключение.

**unsigned int getHeight() const -** метод возвращает высоту матрицы(количество контейнеров в матрице)

**unsigned int getWidth() const -** метод возвращает длину контейнера в матрице(количество элементов в каждом контейнере из матрицы)

Перегруженные операторы:

**Container<T> operator \*(const Container<T> &m) const**

**Container<T> const &operator [](int ind) const**

**Container<T> &operator [](int ind**

**const Matrix<T> &operator =(const Matrix<T> &m)**

## class SLAE

private:

Matrix<double> A – исходная матрица системы.

Matrix<double> A1 – копия матрицы А, с ней происходят все преобразования.

Container<double> B – вектор В.

Container<double> X – вектор Х.

**void addStr(unsigned int first, unsigned int second, double k) –** метод прибавляет строку с индексом “second” умноженную на коэффициент “к” к строке с индексом “first”. Если хотя бы один индекс выходит за границу одного из контейнеров ,то происходит бросок исключения.

**void swap(unsigned int first, unsigned int second) -** метод прибавляет строку с индексом “second” умноженную на коэффициент “к” к строке с индексом “first”. Если хотя бы один индекс выходит за границу одного из контейнеров ,то происходит бросок исключения.

public:

**SLAE(){}**

**SLAE(Matrix<double> A, Container<double> B)**

* Matrix<double> A
* Container<double> B

**void solveByGauss() –** метод решит СЛАУ методом Гаусса. Если СЛАУ не былаинициализирована ,то метод бросит ошибку.

**void printChek() –** метод печатает результат проверки в формате:

i строка матрицы А умноженная на вектор Х | разница между полученным произведением и i элементом вектора В | “Wrong” если разница превышает10-15 ,иначе – “Correct”.Если СЛАУ не была решена, произойдёте бросок исключения.

**void setRandomSyst(unsigned int size) –** создает квадратную матрицу с диагональным преобладанием размера “size” и вектор “В” длинный “size” ,заполненные случайными числами. Максимальный размер матрицы -2147483647 (если хватит памяти на устройстве).

template<typename T> ostream& operator<<(std::ostream& strm, const Container<T> &c) – перегруженный оператор вывода для вектора.

template<typename T> ostream& operator <<(std::ostream& strm, const Matrix<T> &c) – перегруженный оператор вывода для матрицы. Если матрица превышает размер 20 на 20 ,то она напечатана не будет.

# Подтверждение корректности

Подтверждение корректности происходит через метод void printChek() в классе SLAE.

# [Заключение](file:///C:\Users\syato\1_КУРС\ЯиМП\лабараторная%20работа%201\mp1-3821B1PM2\SyatovNikita\Lab_Sort_1\Отчёт%20по%20лабароторной%20работе1.docx#_bookmark6)

В ходе лабораторной работы используя язык программирования Си++ были реализованы: шаблон класса вектор, шаблон класса матрица и СЛАУ. Класс вектор хранит в себе данные, позволяет добавлять их, менять местами и производить арифметические операции. Класс матрица наследуется от класса вектор и расширяет его функционал. Класс СЛАУ включает в себя класс вектор и класс матрица. Слау содержит в себе метод ,решающий СЛАУ методом Гаусса и метод производящий проверку.