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**

**«Реализация метода Гаусса на языке C++**»

**Выполнил**:

студент группы 382003-1

Ягилев И.М.

**Проверил**:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Руководство пользователя 4](#_Toc26962564)

[Описание программной реализации 5](#_Toc26962565)

[Метод решения 8](#_Toc26962563)

[Подтверждение корректности 10](#_Toc26962566)

[Результаты экспериментов 11](#_Toc26962567)

[Заключение 12](#_Toc26962568)

[Приложение 13](#_Toc26962569)

# Постановка задачи

Необходимо реализовать метод Гаусса для действительной квадратной матрицы с выбором ведущего элемента на языке C++, пользуясь концепцией объектно-ориентированного программирования. А именно:

* Реализовать шаблонный класс вектор;
* Реализовать класс квадратная матрица, которая является шаблоном класса вектор векторов;
* Реализовать СЛАУ (класс-наследник квадратной матрицы), который включает в себя метод Гаусса.

# Руководство пользователя

* Сначала необходимо ввести размер квадратной матрицы;
* Затем выбрать способ задания элементов матрицы: случайными числами или вручную:
  1. При выборе первого варианта матрица заполнится случайными вещественными числами;
  2. Во втором случае необходимо ввести каждый элемент матрицы поэтапно;
* Далее следует поэтапный ввод вектора значений в соответствии с указанным размером матрицы;
* Если решение есть, вы увидите введённую расширенную матрицу, преобразованную треугольную расширенную матрицу, вектор, являющийся решением СЛАУ и проверку решения. Если решений нет, вы увидите уведомление с указанной причиной.

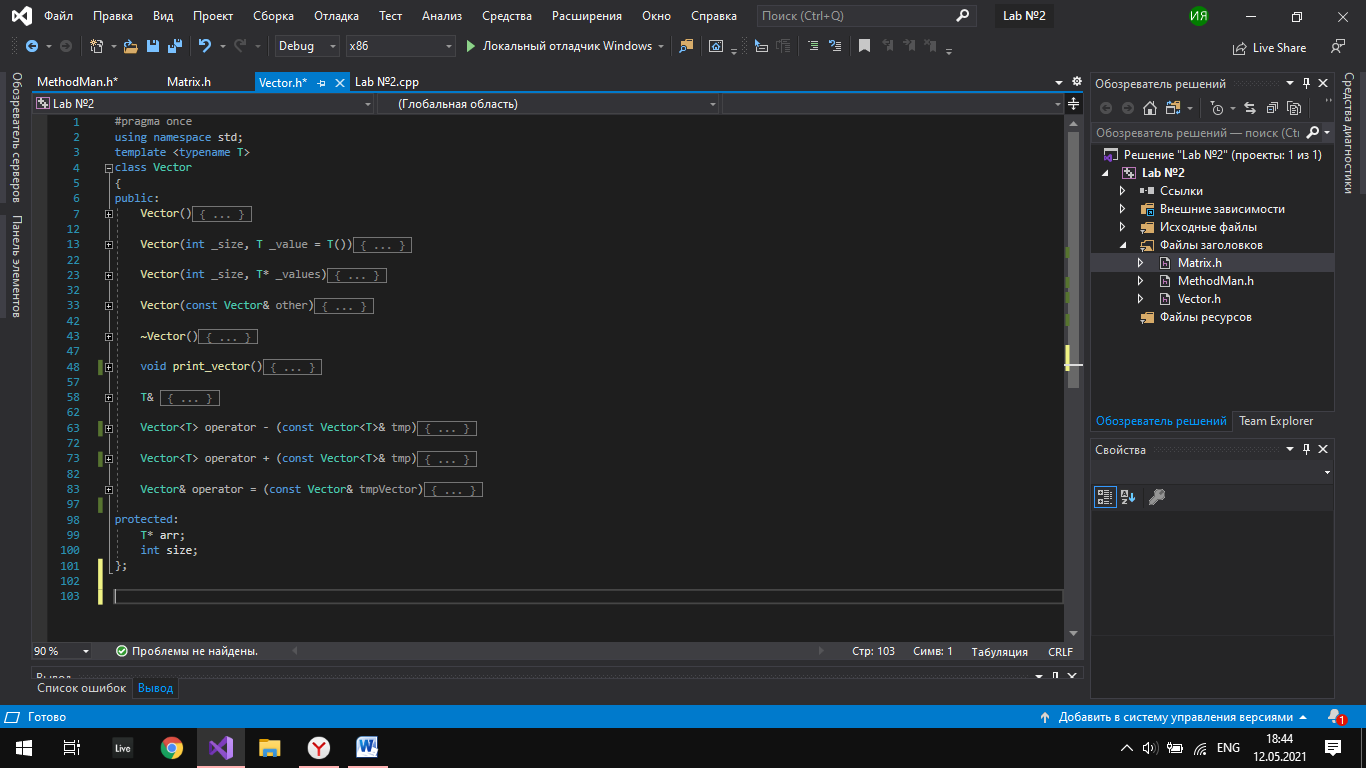
# Описание программной реализации

Проект содержит 4 файла:

* **Lab №2** – cpp файл, содержащий необходимые библиотеки, файлы заголовков, функцию main.
  + **Vector** – заголовочный файл, содержащий реализацию шаблонного класса Vector;
  + **Matrix** - заголовочный файл, содержащий реализацию шаблонного класса Matrix - наследника Vector;
  + **MethodMan** - заголовочный файл, содержащий реализацию шаблонного класса Method - наследника Matrix;

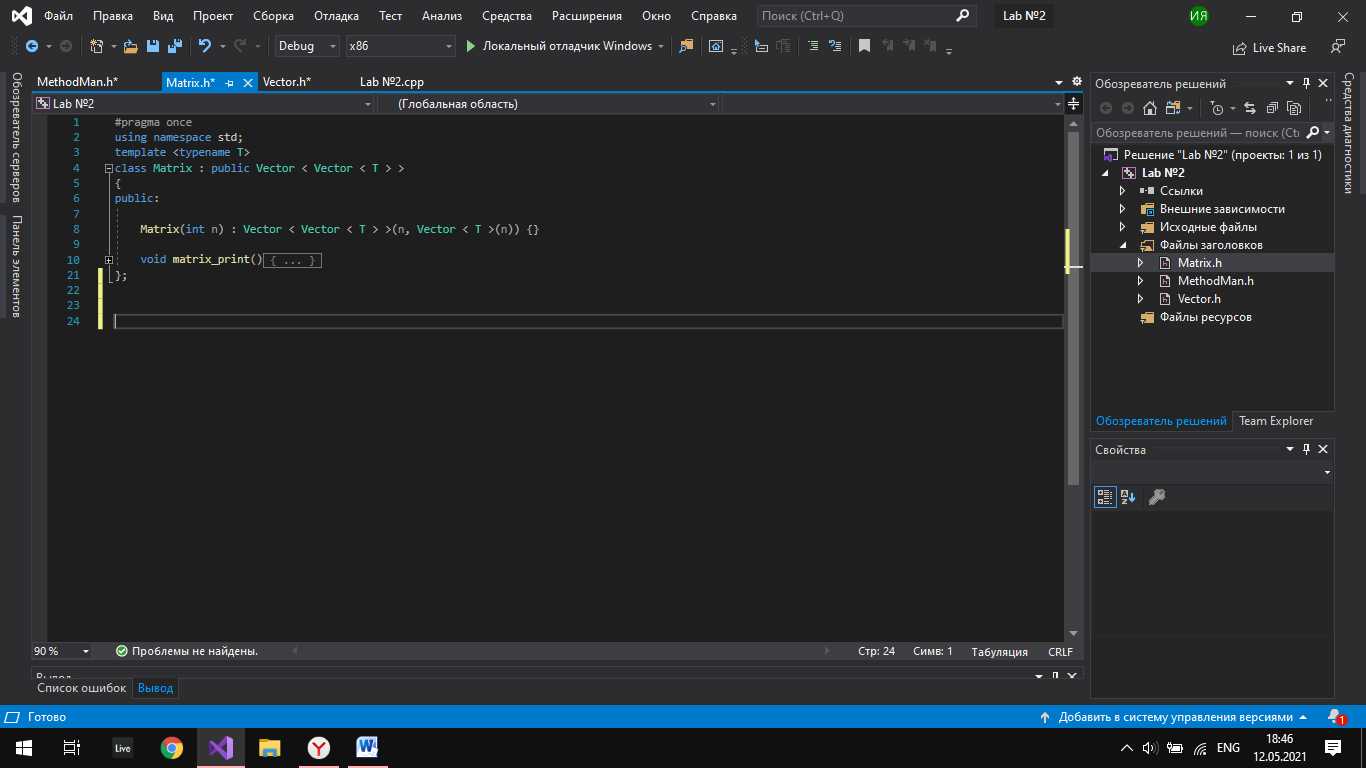
**Класс Vector:**

Имеет protected поля size и указатель на массив, конструктор по умолчанию, конструктор копирования, два конструктора, получающих на вход размер, значение элементов/ указатель на значения. Содержит перегрузку операторов: +, -, =, []; метод print\_vector и деструктор.

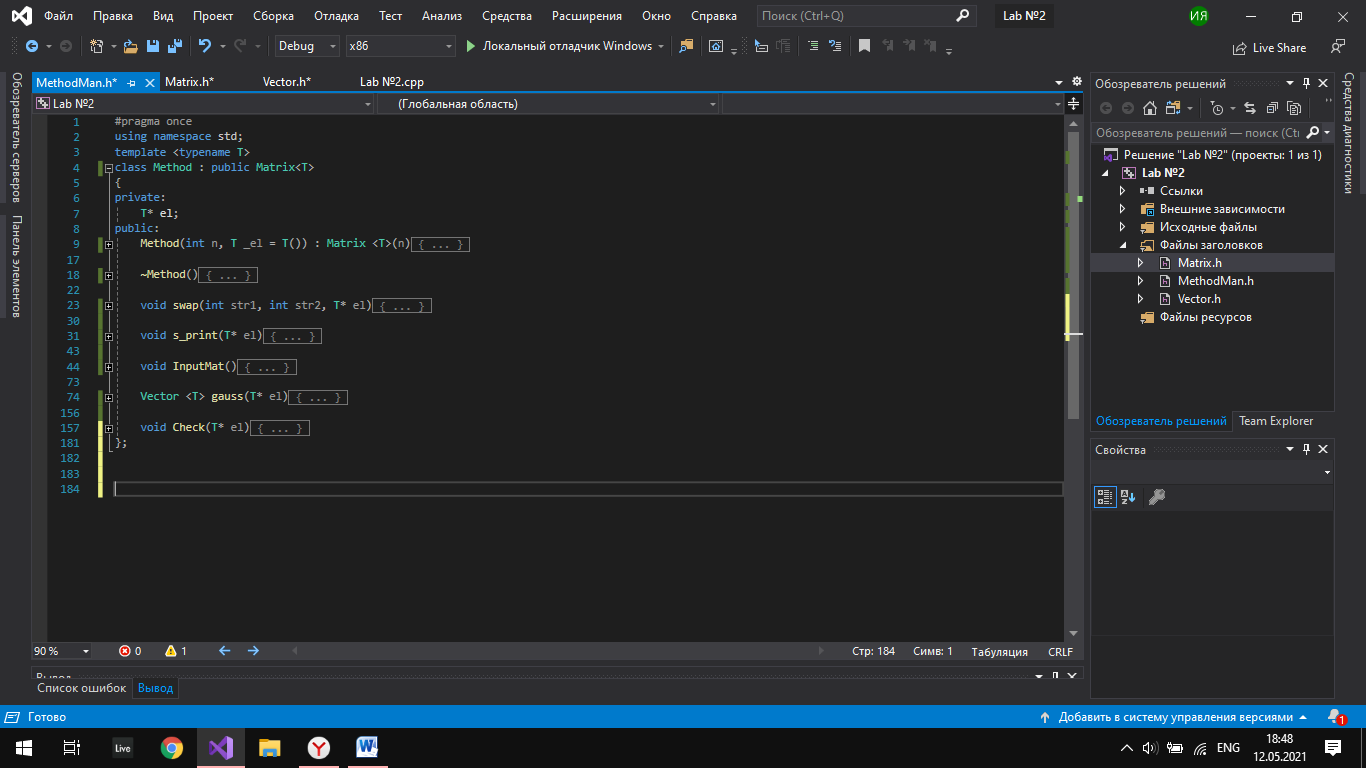


**Класс Matrix:**

Не имеет полей, содержит конструктор, использующий конструктор класса Vector и метод matrix\_print.



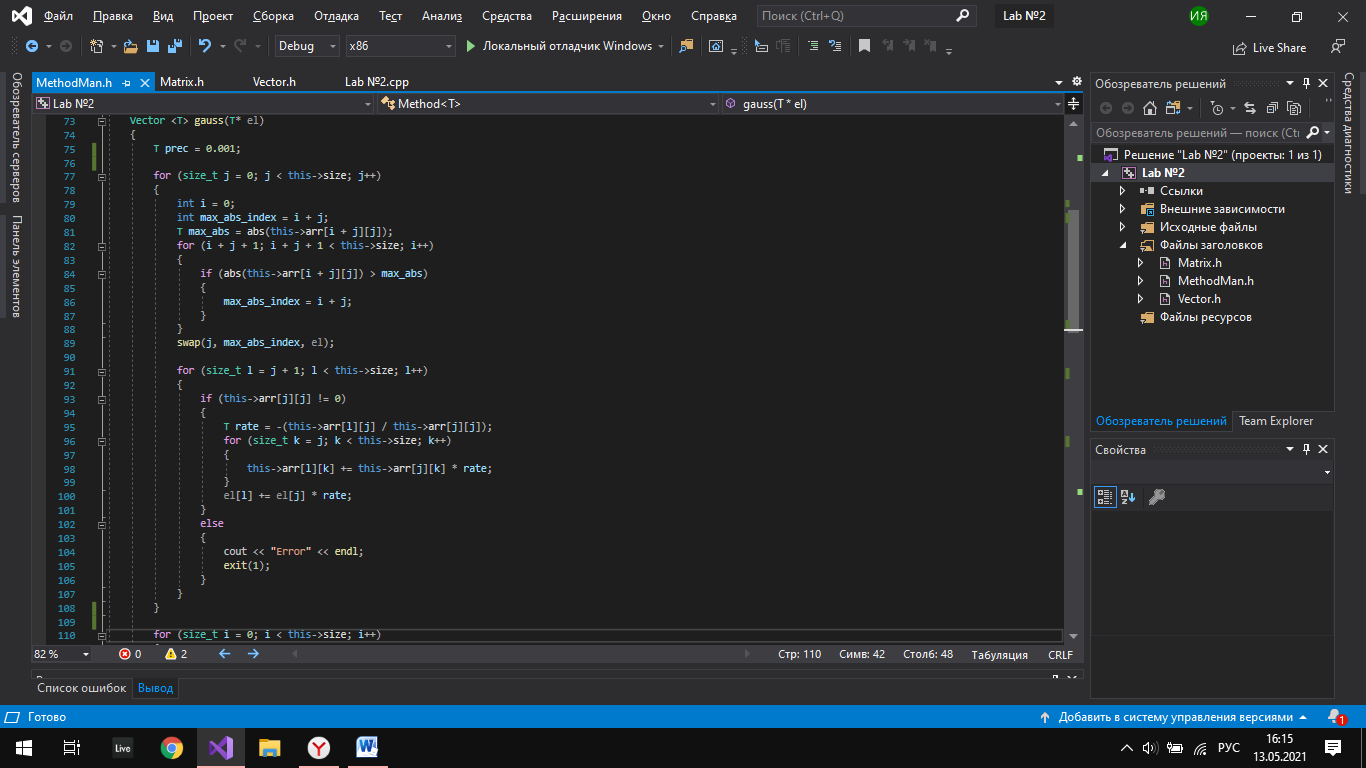
**Класс MethodMan:**

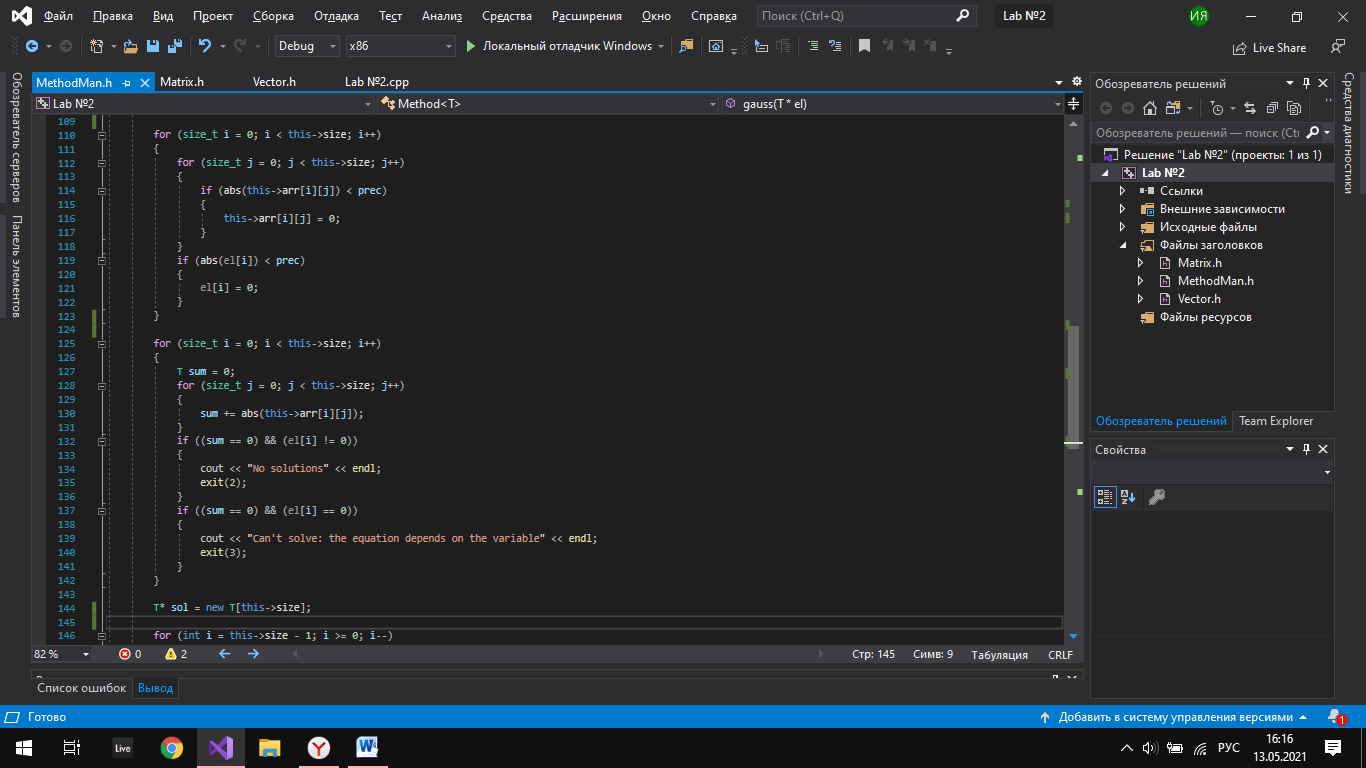
Имеет privat-поле – указатель на массив элементов, конструктор, использующий конструктор матрицы, метод swap, меняющий строки и элементы местами, метод s\_print, выводящий всю СЛАУ, метод InputMat, наполняющий систему значениями, метод Гаусса, производящий основные вычисления, метод Check, проверяющий корректность вычислений и деструктор класса.

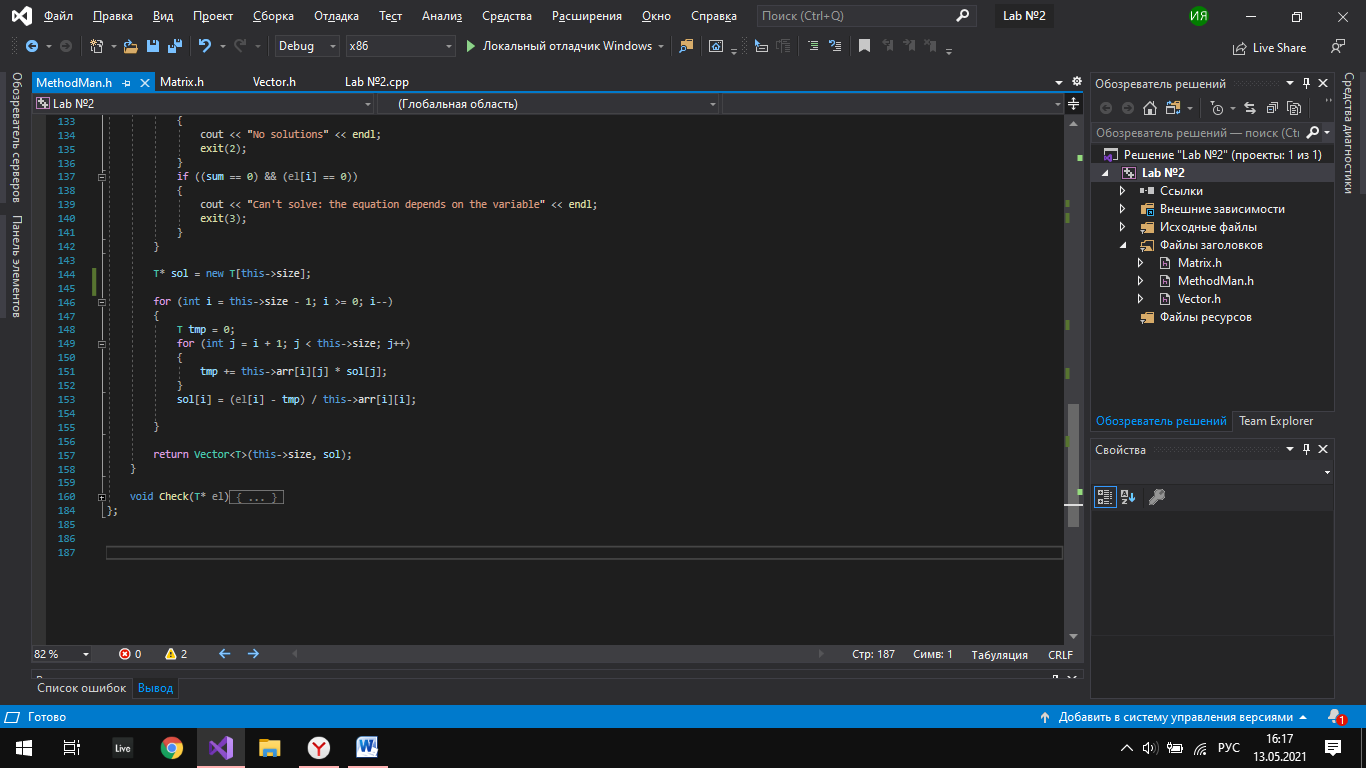
# Метод решения

Подробно описывать конструкторы, деструктор, перегрузку операторов и метод print классов Vector и Matrix смысла нет, так как их реализация достаточно шаблонная. Подробнее хотелось бы остановиться на реализации метода gauss класса MethodMan.

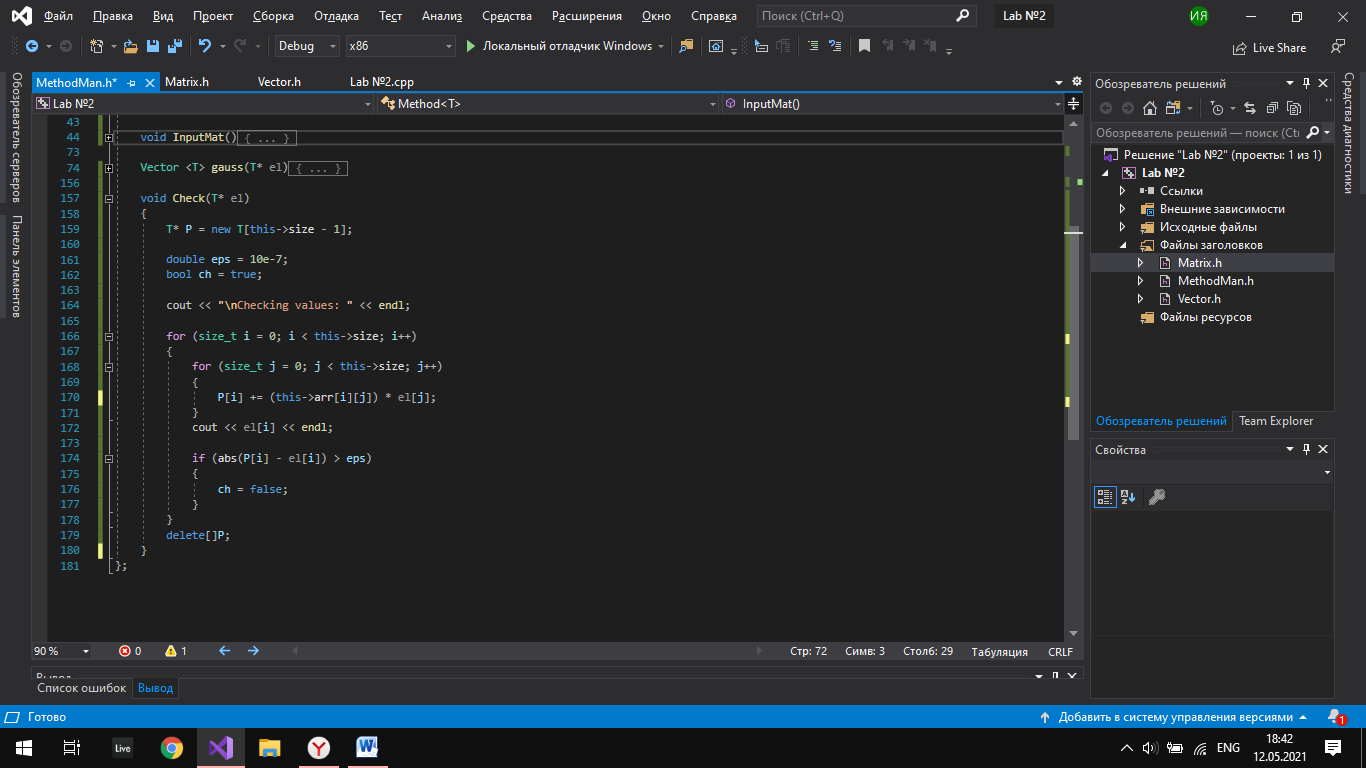
gauss – метод - реализация метода Гаусса, который принимает указатель на массив и возвращает вектор значений, который является решением СЛАУ. Подробное описание кода:

1. Объявление и определение prec – переменной, с помощью которой определяется погрешность вычисления;
2. Цикл, содержащий два цикла.1 - вычисляет максимальный элемент в столбце, помещает его на главную диагональ с перемещением всей строки. 2 – «зануляет» элементы, лежащие ниже главной диагонали, выдаёт ошибку, если значение элемента главной диагонали равно нулю;
3. Цикл, зануляющий элементы расширенной матрицы, если после преобразования их модуль меньше prec;
4. Цикл, проверяющий наличие решений у треугольной матрицы;
5. Объявление вектора решений;
6. Цикл, наполняющий вектор решений необходимыми значениями;
7. Возвращение вектора.

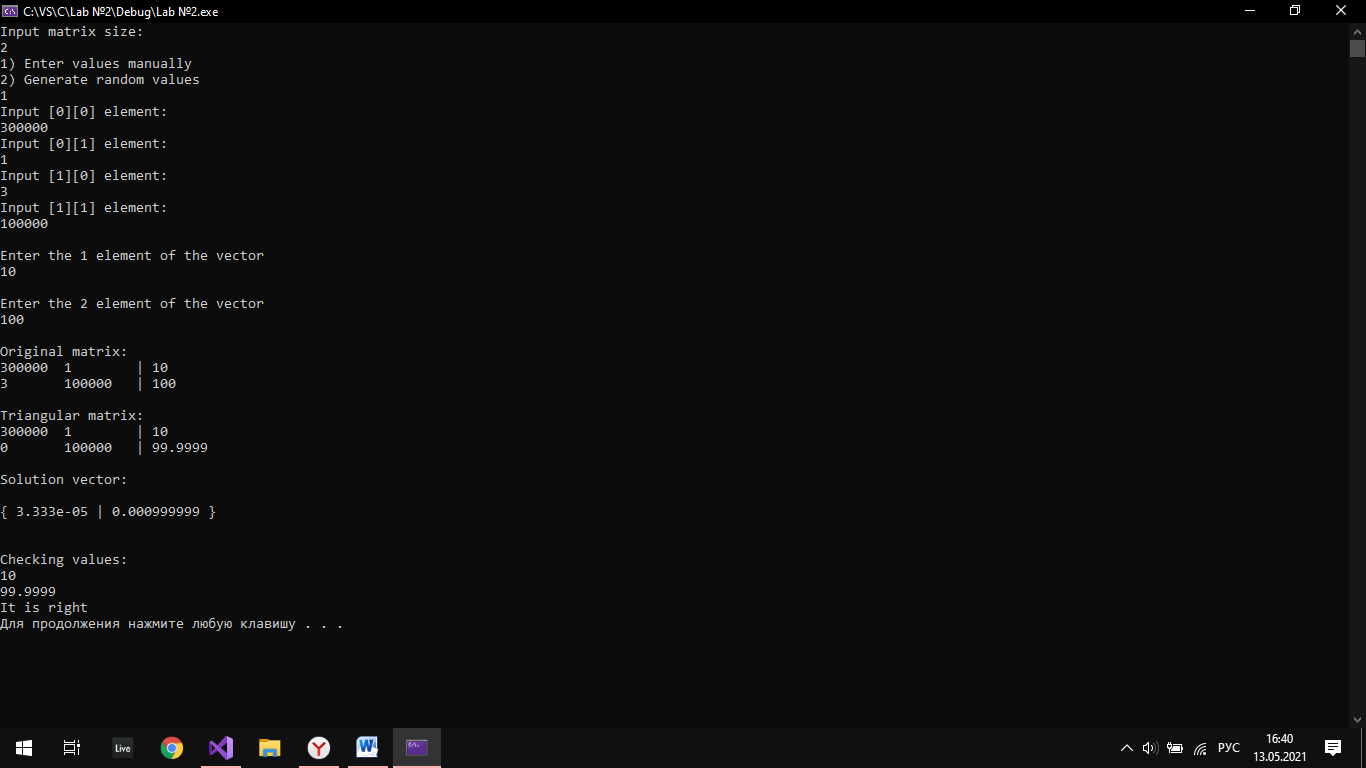
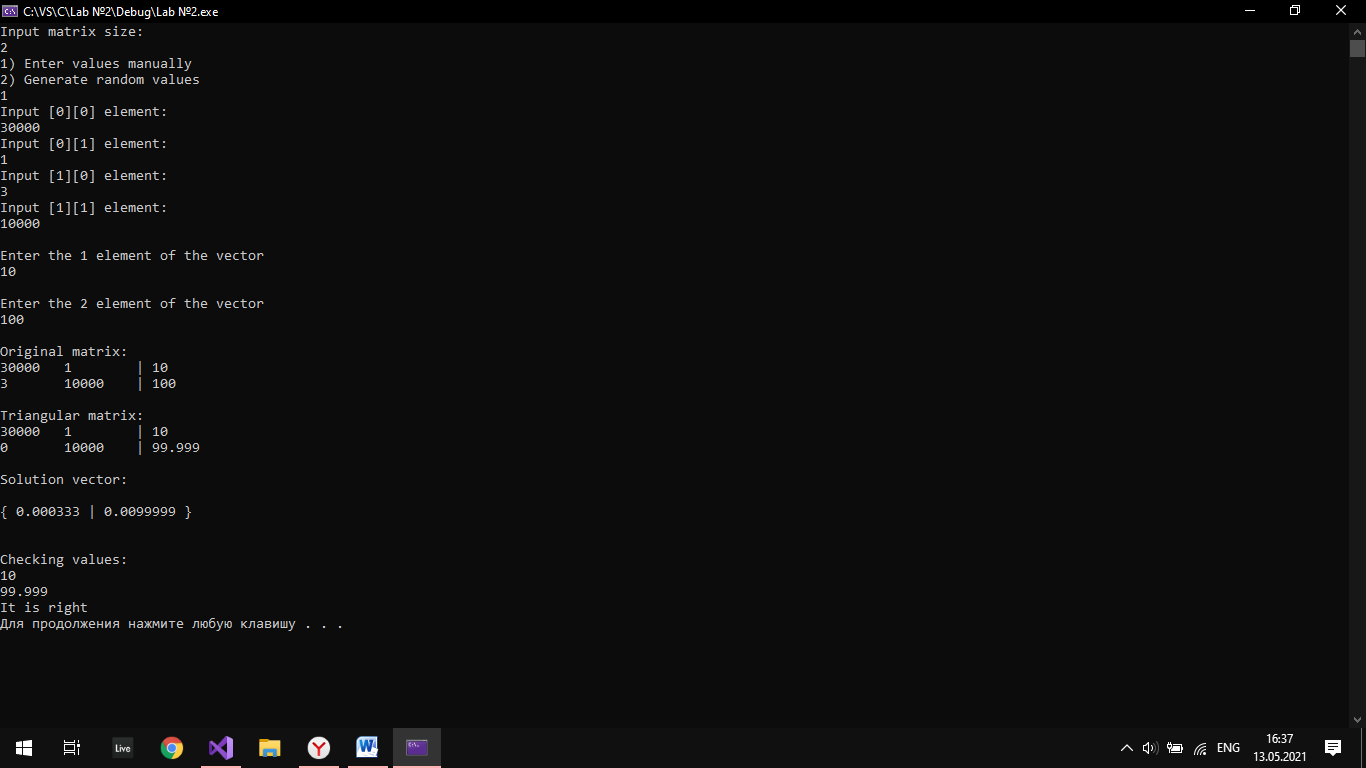




# Подтверждение корректности

Для подтверждения корректности вычислений в программе предусмотрен метод Check класса Method, сравнивающий (по всей системе) разность получившейся левой и правой части уравнения с эпсилон (10e-7). Таким образом происходит проверка наличия влияния накопившейся погрешности вычисления на результат.

# Результаты экспериментов

По данным экспериментов видно, что, в пределах заданной точности, программа выполняет вычисления корректно, но при большем различии в порядках точность ухудшается.

# Заключение

Я реализовал метод Гаусса, корректно выполняющий вычисления, но точность для типа double, например, очень далека от 100%. Для увеличения точности можно установить наибольшее количество знаков после запятой, но даже это не приблизит к идеальной точности вычисления.

# Приложение

