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»**

**Выполнил**:

студентка группы 3821Б1ПМ2

Нагорная Е.А.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2022

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Метод решения 4](#_Toc26962563)

[Руководство пользователя 5](#_Toc26962564)

[Описание программной реализации 6](#_Toc26962565)

[Подтверждение корректности 7](#_Toc26962566)

[Результаты экспериментов 8](#_Toc26962567)

[Заключение 9](#_Toc26962568)

[Приложение 10](#_Toc26962569)

# Постановка задачи

Метод Гаусса – это наиболее универсальный метод решения систем линейный уравнений с помощью элементарных преобразований строк.

Задачей данной лабораторной работы была реализация метода Гаусса с выбором ведущего элемента. Программа должна работать с квадратными матрицами и выполнять проверку с помощью умножения полученного вектора на исходную матрицу. А также сообщать о возможных ошибках во время своей работы (попытка деления на 0, несовместная система линейных уравнений или разные размерности векторов исходной матрицы и полученного вектора).

# Метод решения

В основе программы лежит метод Гаусса с ведущим элементом. На вход программа получает расширенную матрицу. Затем в первом столбце находится максимальный по модулю элемент, и строки меняются местами так, чтоб этот элемент оказался на главной диагонали. С помощью элементарных преобразований (умножение строки на число и сложение строк) в столбце зануляются все элементы под максимальным. Далее во втором столбце находится максимальный, он ставится на главную диагональ, и элементы под ним зануляются. Так происходит с каждым следующим столбцом до конца матрицы. В результате всех преобразований матрица принимает верхнетреугольный вид. Затем с помощью обратного хода метода Гаусса вычисляются координаты нового вектора, который и будет решением системы линейных уравнений.

Если получается, что максимальный по модулю элемент в столбце равен нулю, то такая система линейных уравнений несовместна или имеет множество решений.

# Руководство пользователя

В начале работы программы пользователь видит строку “Введите размерность:”, после чего он должен ввести размер матрицы.

Затем программа предлагает выбрать заполнить матрицу самому, тогда пользователю следует ввести 0 и начать вводить матрицу по строкам, а потом вектор, или заполнить случайными числами, тогда необходимо ввести 1.

Далее, если размер матрицы не превышает 10, то программа выведет расширенную исходную матрицу. В независимости от размерности матрицы в результате работы программы пользователь сможет видеть на экране вектор Х (это результат решения системы линейных уравнений).

Также если размерность матрицы не будет больше 10, то пользователь сможет видеть на экране вектор В. Это значит, что программа выполнила проверку, и это вектор, который она получила при умножении исходной матрицы на вектор Х. Пользователь может самостоятельно визуально сравнить его с вектором в исходной матрице и убедиться в правильности работы программы. Независимо от размера матрицы программа также сама сравнивает эти векторы и выводит на экран фразу “Верно”, если вектора совпали, или “Неверно”, если погрешность превышает 10-10.

# Описание программной реализации

Программа состоит из одного файла, в который входят три класса: classVector, classMatrix, который наследуется от classVector<classVector<T>>, и SLAU.

classVector шаблонный и включает в себя поле size (размер вектора), конструктор по умолчанию (но может быть вызван и с параметрами), конструктор копирования, деструктор, метод get\_size (возвращает размер массива), операторы =, += и [ ], метод resize (позволяет менять размер вектора), метод swap (меняет местами два элемента в массиве), оператор \* для умножения вектора на число и для скалярного произведения и оператор +.

Class Matrix, наследуется от вектора, и в дополнение к его функциями содержит свой конструктор копирования и конструктор с параметрами, метод get\_size, оператор [ ], оператор \* для умножения матрицы на вектор, метод print (позволяющий вывести матрицу на экран).

Class SLAU имеет четыре поля: int size, classMatrix<T> A (исходная матрица), classVector<T> B (столбец свободных членов), classVector<T> X (вектор ответ). А также конструктор с параметрами, который позволяет в зависимости от выбора пользователя заполнить матрицу самостоятельно или случайными числами, деструктор, метод solve, в котором реализован метод Гаусса, метод prov (выполняет проверку) и метод print (выводит на экран расширенную матрицу).

Также вне классов выполнена перегрузка операторов >> и << для матрицы и вектора отдельно.

Функция main выполняет ввод размера матрицы и вызывает метод Гаусса, а также выводит на экран ошибку, которую нашла программа.

# Подтверждение корректности

Для подтверждения корректности в программе происходит умножение исходной матрицы на получившийся вектор. Затем полученный вектор сравнивается со столбцом свободных членов в расширенной матрица, если погрешность не превышает 10-10, то считается что вектора совпали и решение верно, иначе – пользователю сообщается об ошибке.

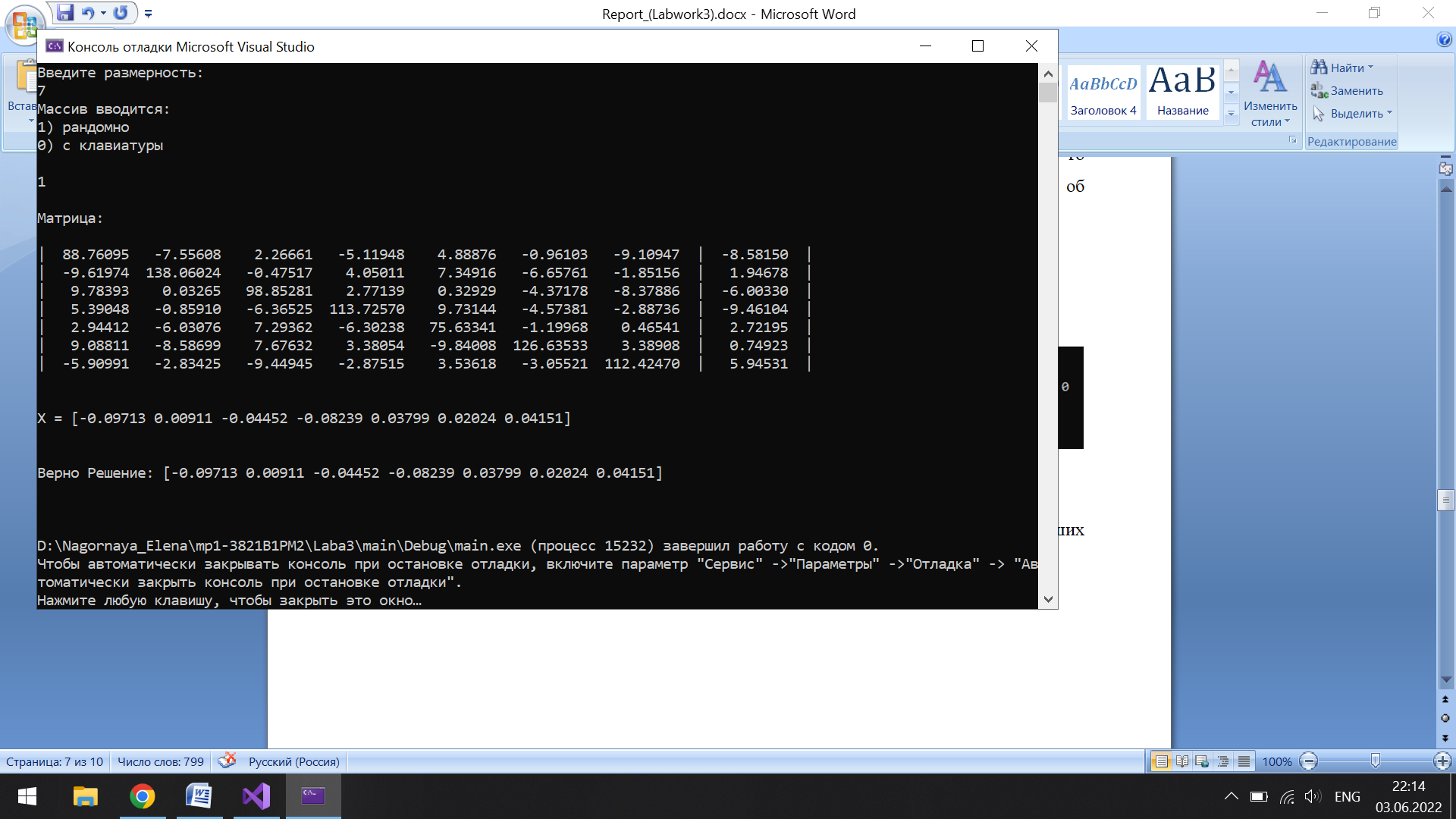


Рис. 1 Матрица 7x7

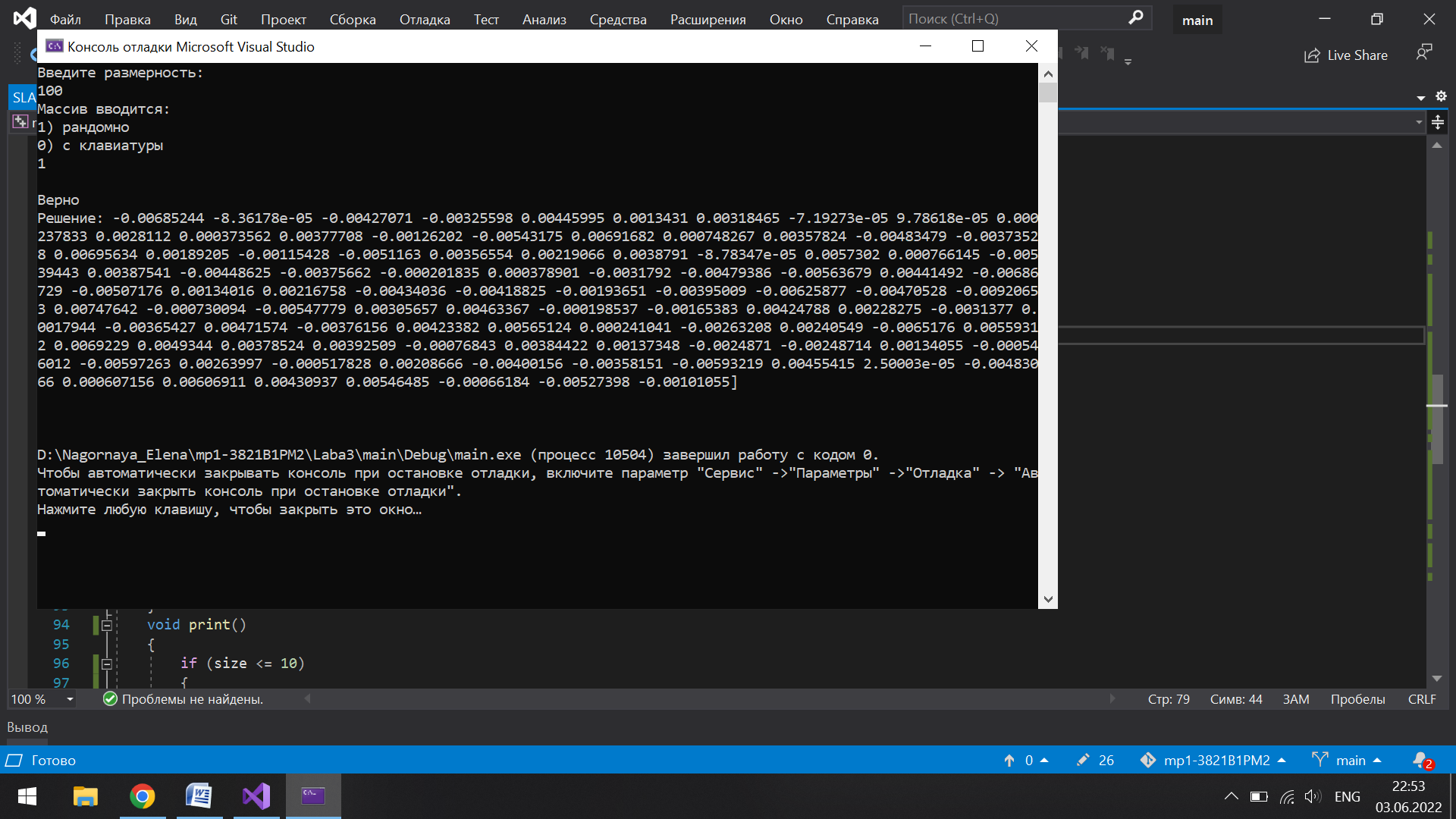


Рис. 2 Матрица 100x100

По рисункам 1 и 2 видно, что программа работает корректно и для небольших размерностей матриц, и для больших матриц.

# Результаты экспериментов

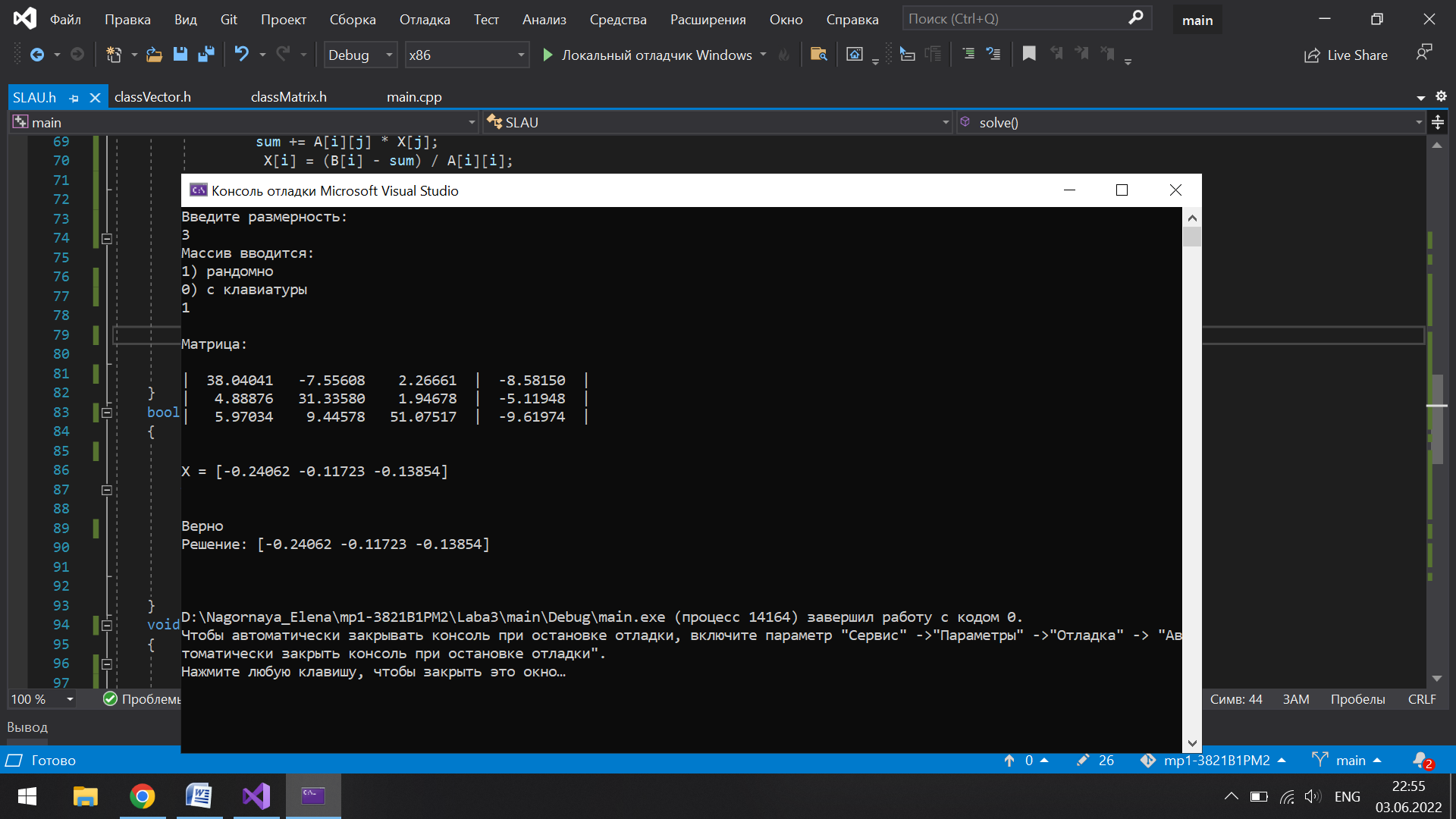


Рис. 3 СЛУ совместна, найдено решение

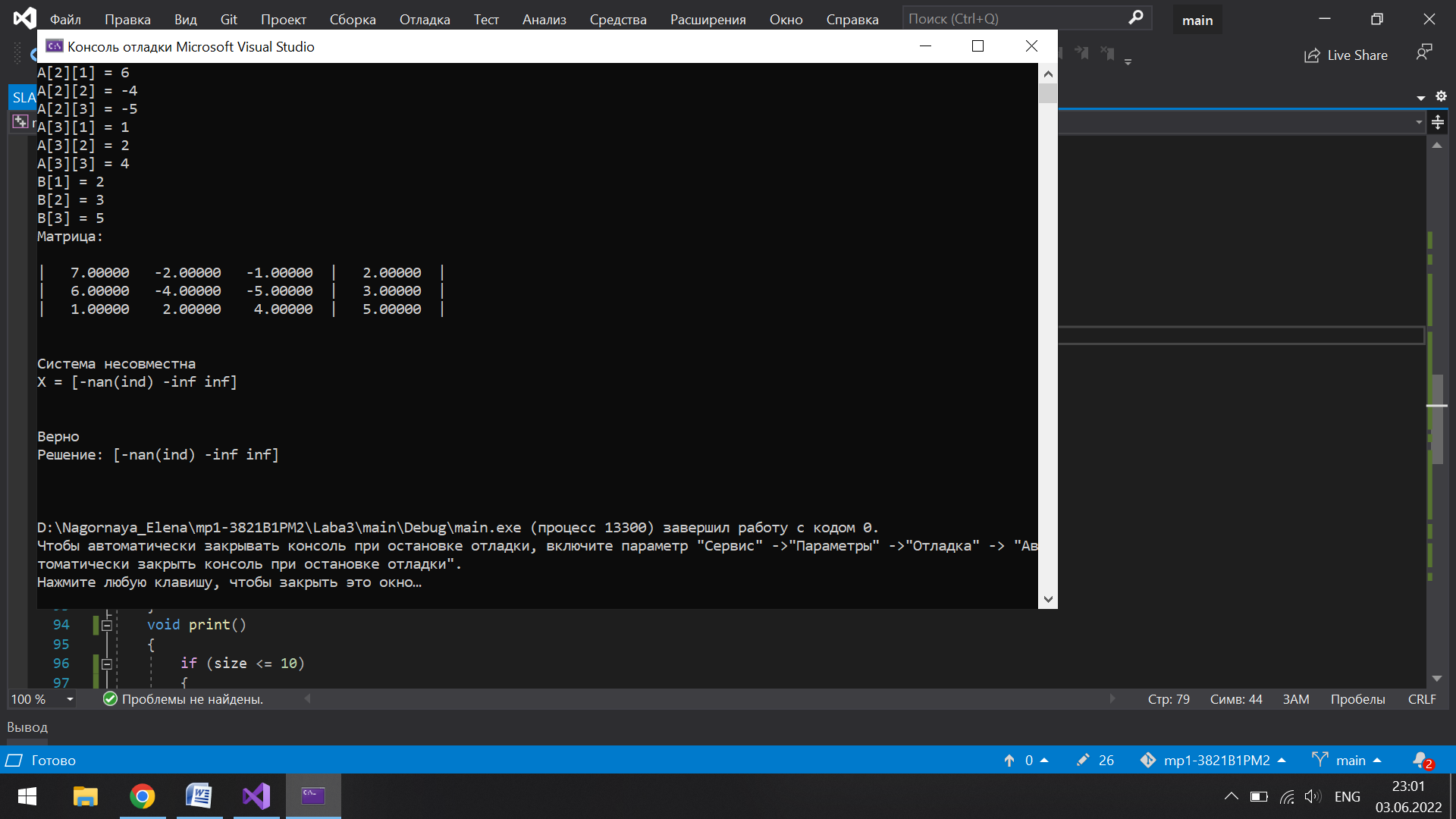


Рис. 4 Система несовместна

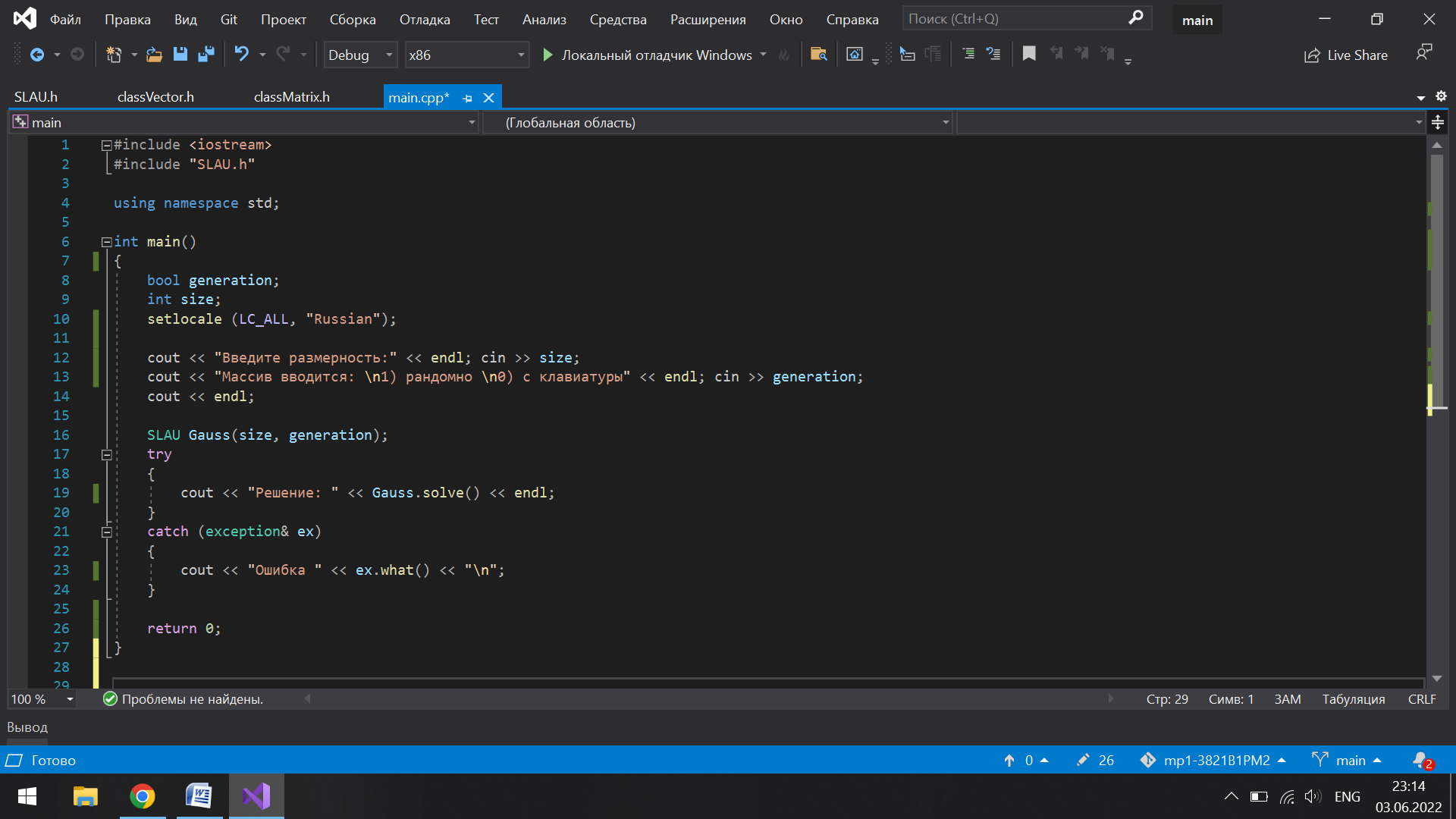
По данным экспериментов видно, что программа может как корректно решать системы линейных уравнений, так и сообщить об ошибке решения.

# Заключение

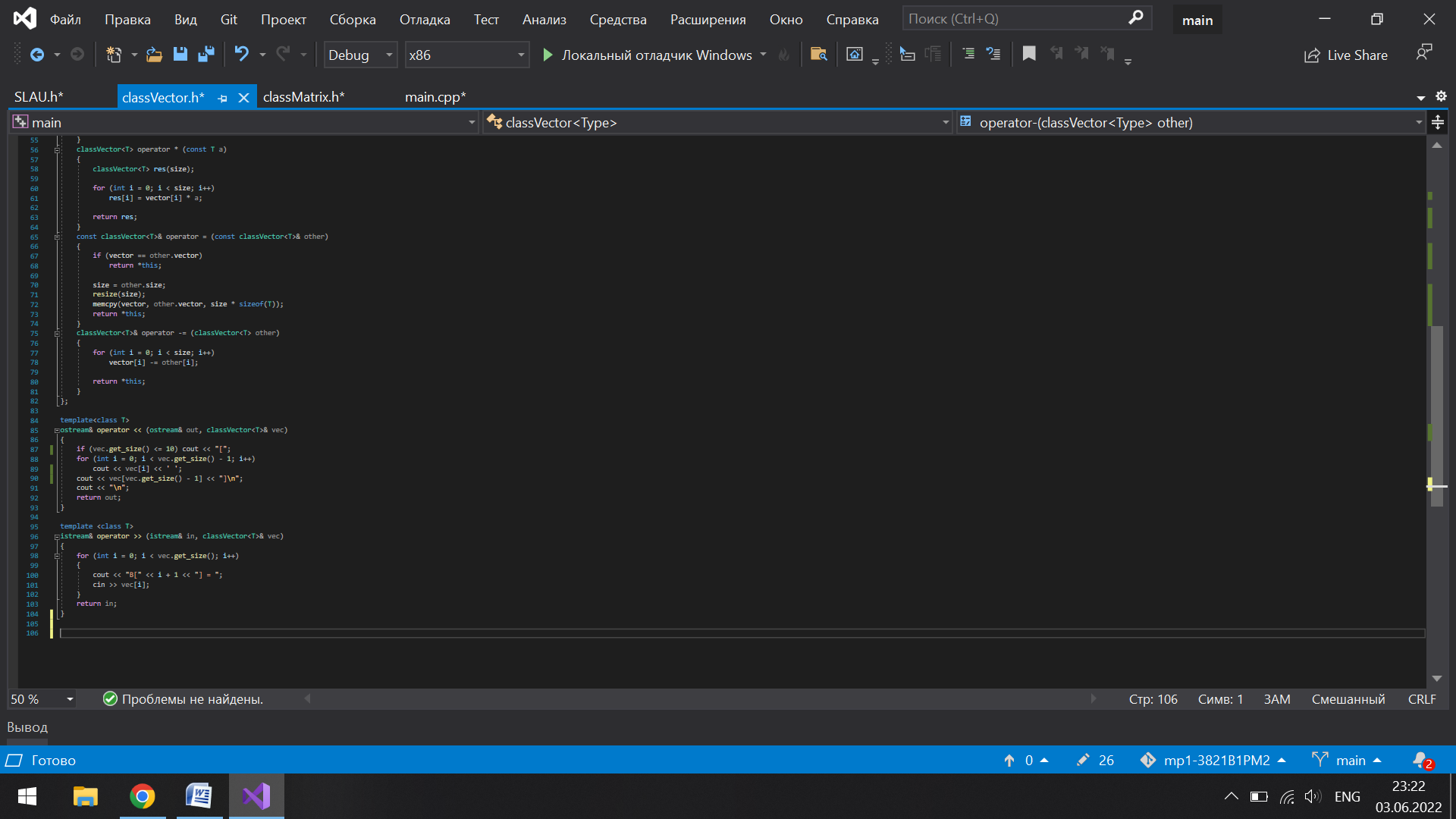
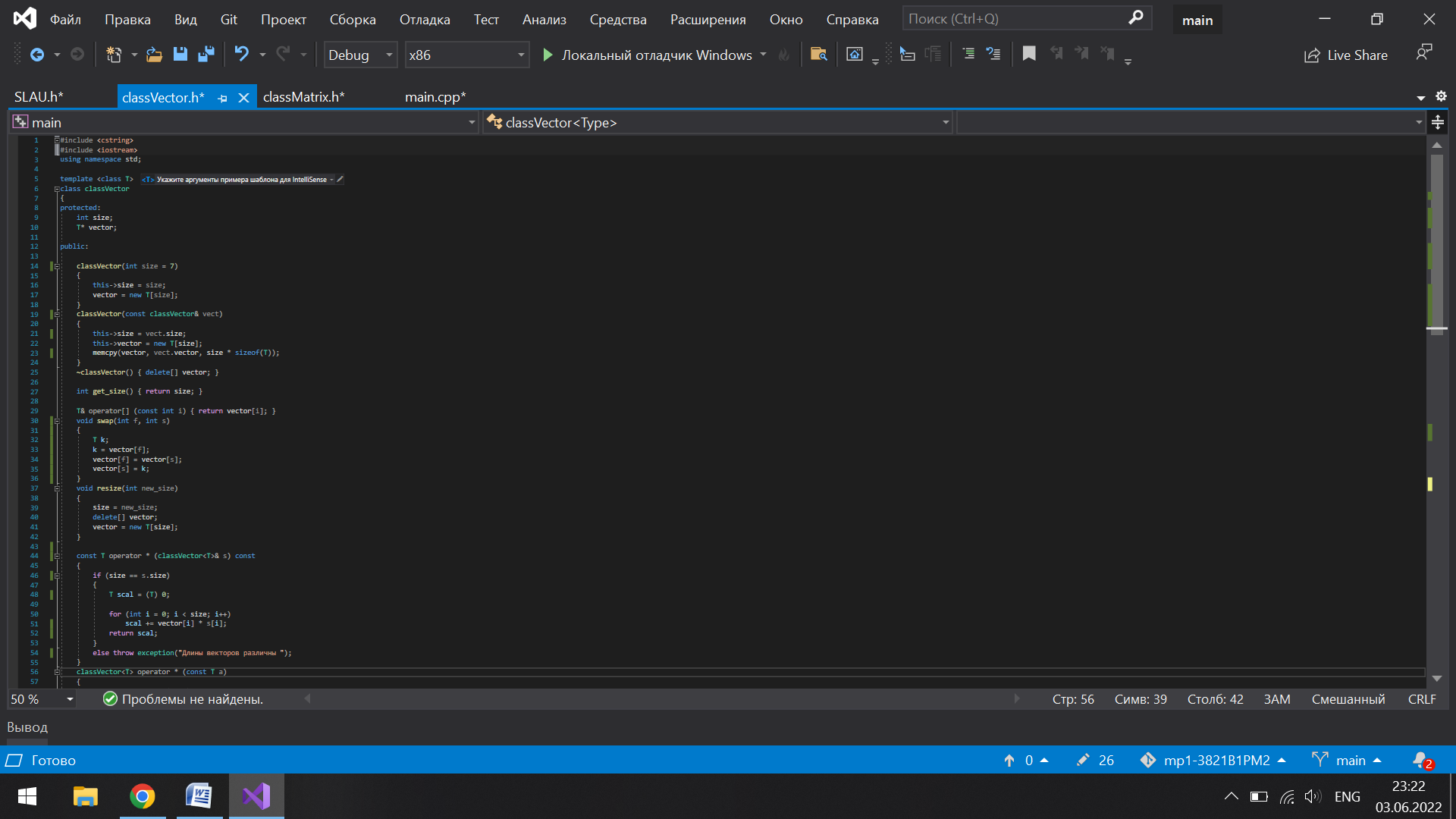
Таким образом, были выполнены основные задачи лабораторной работы. Реализован метод Гаусса с выбором ведущего элемента, В программе есть шаблон класса classVector, класс classMatrix, который наследуется от classVector<classVector<T>>, класс SLAU. Программа делает проверку своих подсчётов, по результатам которой можно сделать вывод, что она работает корректно. Также она выводит возможные ошибки.

# Приложение

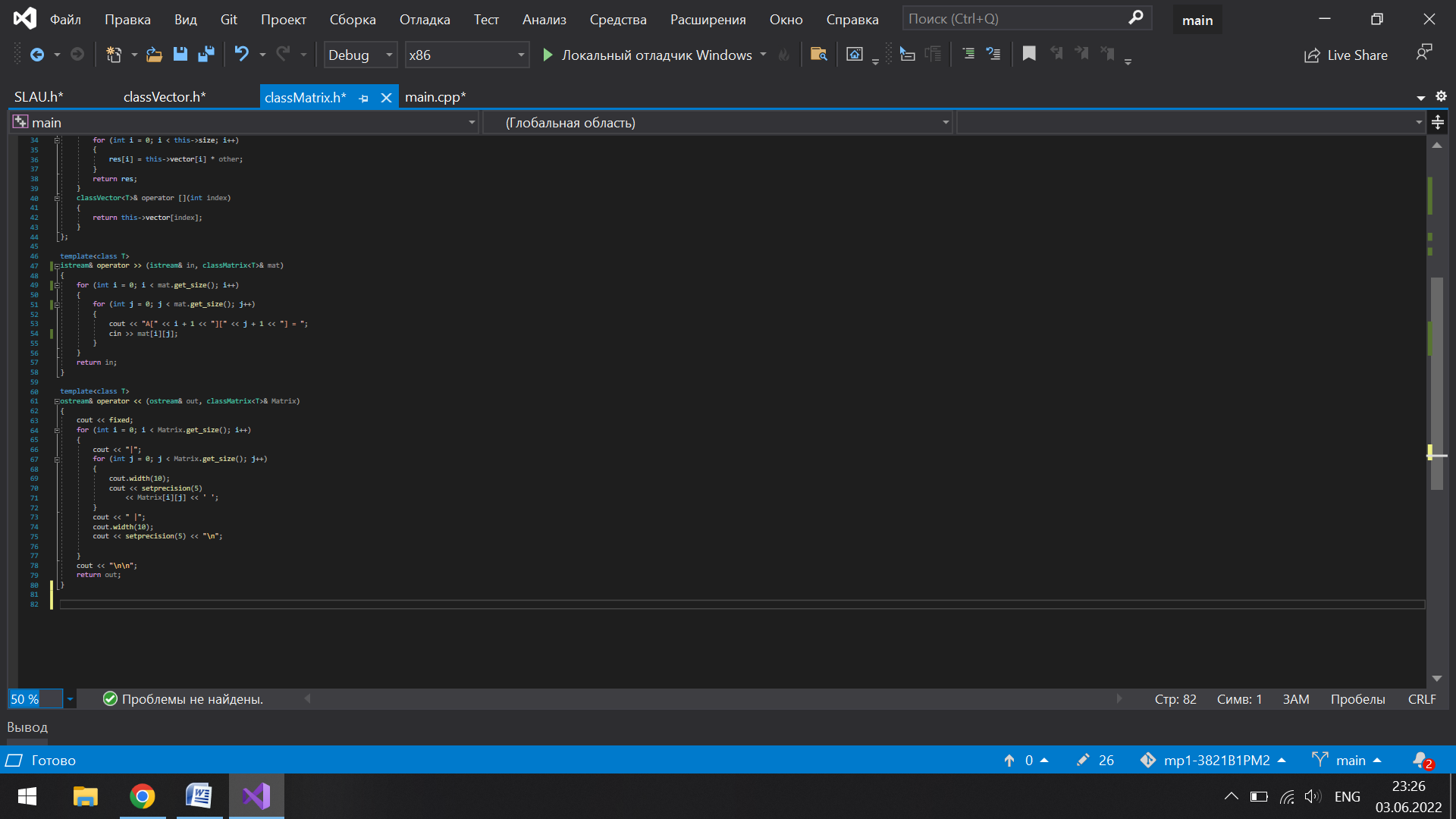
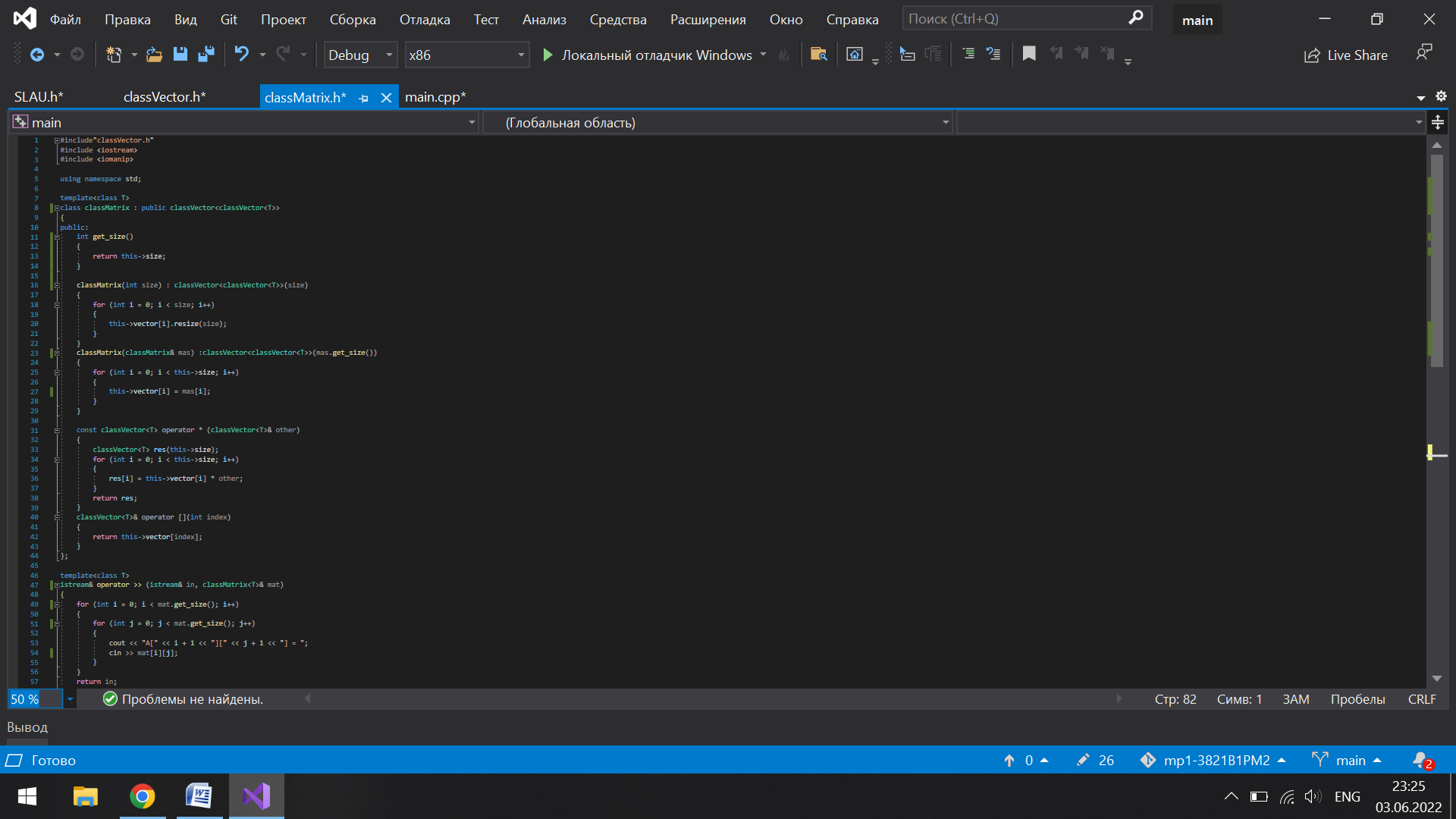
main.cpp



classVector.h



classMatrix.h



SLOU.h

