МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ КОЗЬМЫ МИНИНА»

Факультет Информационных Технологий

Кафедра Информационных систем и цифровых сервисов в управлении

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль (специализация) Информационные системы и технологии

**К У Р С О В О Й    П Р О Е К Т**

на тему: Решение системы линейных уравнений методом Гаусса

СТУДЕНТ(КА)        \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   Гусева Е.А.

*(личная подпись)                                              (инициалы, фамилия)*

РУКОВОДИТЕЛЬ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   Тимофеева Е.А.

*(личная подпись)                             (ученая степень, звание, инициалы, фамилия)*

Нижний Новгород – 2025 г.

**Содержание**

[Введение 3](#_Toc199285966)

[1. Постановка задачи 5](#_Toc199285967)

[1.1. Математическая постановка задачи 5](#_Toc199285968)

[1.2. Основные этапы метода Гаусса 5](#_Toc199285969)

[1.3. Особенности и условия применимости 6](#_Toc199285970)

[2. Описание метода решения 8](#_Toc199285971)

[3. Алгоритм решения задачи 9](#_Toc199285972)

[4. Выбор языка программирования. 12](#_Toc199285973)

[5. Описание программного комплекса. 14](#_Toc199285974)

[6.Анализ результатов 17](#_Toc199285975)

[Заключение 19](#_Toc199285976)

[Список литературы 21](#_Toc199285977)

[Приложение 1 23](#_Toc199285978)

[Приложение 2 27](#_Toc199285979)

# **Введение**

Системы линейных уравнений встречаются в самых различных областях, включая физику, экономику, инженерию и многие другие дисциплины. Они представляют собой набор уравнений, в которых каждая переменная имеет степень, равную единице, и могут быть решены с помощью различных методов. Одним из наиболее известных и широко используемых методов является метод Гаусса, который позволяет эффективно находить решения для таких систем.

Актуальность данной работы обусловлена необходимостью разработки программных решений, которые могут автоматизировать процесс решения систем линейных уравнений. В условиях современного мира, где объем данных и сложность вычислений постоянно растут, использование алгоритмов, таких как метод Гаусса, становится особенно важным.

Целью работы является теоретическое изучение метода решения системы линейных уравнений методом Гаусса и разработка программы, реализующей данный метод.

Программирование на языке C# предоставляет мощные инструменты для реализации таких алгоритмов, что делает его идеальным выбором для данной задачи. В рамках данной курсовой работы будет разработана программа, которая реализует метод Гаусса, что позволит пользователям легко и быстро находить решения для различных систем линейных уравнений.

В работе будут освещены несколько ключевых тем. В первую очередь, будет представлен вводный раздел, посвященный теоретическим основам метода Гаусса. Далее будет описан алгоритм преобразования матрицы коэффициентов в верхнетреугольную форму, что является важным этапом в решении системы уравнений. Этот процесс включает в себя последовательное исключение переменных, что позволяет упростить систему и подготовить её к дальнейшему решению.

Кроме того, в работе будет представлена структура программы на C#, а также код, реализующий метод Гаусса. Это позволит не только ознакомиться с теоретическими аспектами, но и увидеть практическую реализацию алгоритма. Тестирование программы на различных примерах систем уравнений позволит оценить её эффективность и корректность работы.

В заключение, будут рассмотрены перспективы использования метода Гаусса в будущем, включая возможные улучшения и адаптации для решения более сложных задач. Таким образом, данная работа не только углубляет понимание метода Гаусса, но и демонстрирует его практическое применение в программировании на языке C#.

**Объект исследования:**

Методы, применяемые для решения СЛАУ.

**Предмет исследования:**

Применение метода решения системы линейных уравнений методом Гаусса.

# **1. Постановка задачи**

# **1.1. Математическая постановка задачи**

Система линейных алгебраических уравнений (СЛАУ) – это совокупность уравнений вида:

где - коэффициенты системы, – свободные члены, – неизвестные, n – количество неизвестных, m – количество уравнений.

Задача решения СЛАУ заключается в нахождении таких значений , которые удовлетворяют всем уравнениям системы одновременно. В случае, если система имеет единственное решение, она называется совместной и определенной. Если решений бесконечно много – совместной и неопределенной, а если решений нет – несовместной.

Метод Гаусса – один из наиболее распространенных численных методов решения СЛАУ. Он основан на последовательном исключении неизвестных и приведении системы к ступенчатому (треугольному) виду, после чего решение находится обратной подстановкой.

# **1.2. Основные этапы метода Гаусса**

Метод Гаусса состоит из двух основных этапов:

1. Прямой ход – приведение системы к ступенчатому виду с помощью элементарных преобразований:

* перестановка строк,
* умножение строки на ненулевое число,
* сложение строк с целью исключения переменных.

В результате получается матрица вида:

2. Обратный ход – нахождение значений неизвестных, начиная с последнего уравнения:

* из последнего уравнения выражается
* подставляется в предпоследнее уравнение для нахождения
* процесс продолжается до нахождения всех переменных.

# **1.3. Особенности и условия применимости**

Для успешного применения метода Гаусса необходимо, чтобы система была совместной (имела хотя бы одно решение). Кроме того, метод чувствителен к вычислительной погрешности, особенно если матрица системы близка к вырожденной (её определитель мал).

Особые случаи:

* Если в процессе прямого хода получается строка вида 0=, система несовместна.
* Если ступенчатая матрица содержит меньше ненулевых строк, чем переменных, система имеет бесконечное множество решений.

Метод Гаусса применим как для квадратных, так и для прямоугольных систем. В случае переопределенных систем (когда уравнений больше, чем неизвестных) он позволяет найти решение в смысле метода наименьших квадратов.

Для программной реализации метода Гаусса часто используют языки программирования, поддерживающие матричные вычисления. Визуализация может включать вывод промежуточных шагов преобразования матрицы, что полезно для отладки и обучения.

# **2. Описание метода решения**

Метод Гаусса — классический алгоритм решения систем линейных уравнений, основанный на последовательном исключении неизвестных. Он состоит из двух ключевых этапов: прямого хода (приведение матрицы системы к ступенчатому виду) и обратного хода (нахождение значений переменных).

Прямой ход включает следующие шаги:

1. Исключение переменных:

* Выбор ведущего элемента (часто максимального по модулю в столбце для уменьшения погрешности).
* Перестановка строк, чтобы ведущий элемент оказался на диагонали.
* Нормирование строки: деление элементов строки на ведущий элемент.
* Вычитание текущей строки из последующих строк для обнуления элементов под ведущим.

1. Повторение для следующих переменных:  
   Процесс повторяется для подматрицы, исключая элементы из третьего уравнения.

Обратный ход:

* После приведения к верхнетреугольному виду, переменные вычисляются снизу вверх:
* Из последнего уравнения находится неизвестные.
* Подстановка одних неизвестных во второе уравнение даёт нахождение других неизвестных.
* Значения найденных неизвестных подставляются в первое уравнение для нахождения первого неизвестного.

1. **Алгоритм решения задачи**

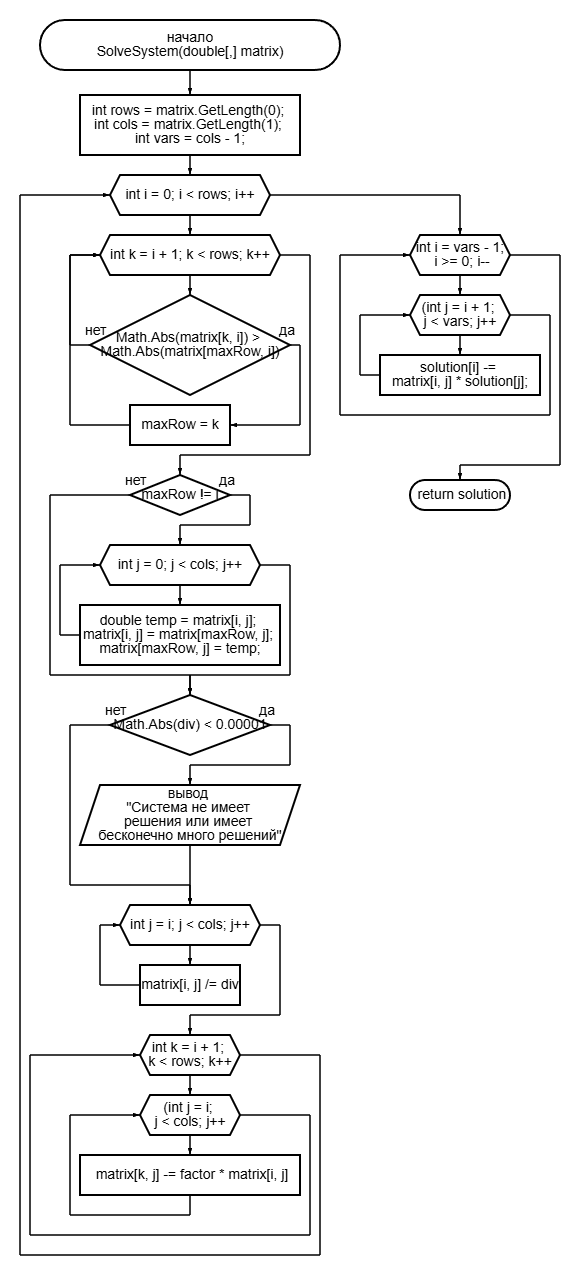


Рис.1. Блок-схема алгоритма решения СЛАУ методом Гаусса

Шаг 1. Объявляем переменные.

* rows: Число строк матрицы (количество уравнений).
* cols: Число столбцов матрицы (переменные + свободные члены).
* vars: Количество переменных (cols - 1, так как последний столбец — свободные члены).

Шаг 2. Цикл int i = 0; i < rows; i++ проходиться по всем элементам матрицы через второй цикл int k = i + 1; k < rows; k++ для поиска строки с максимальным элементом для устойчивости алгоритма.

Шаг 3. Проходя через условие Math.Abs(matrix[k, i]) > Math.Abs(matrix[maxRow, i]) обновляем индекс строки с максимальным элементом.

Шаг 4. При соблюдении условия maxRow != i через цикл int j = 0; j< cols; j++ меняем строки местами, чтобы избежать деления строки с максимальным элементом на ноль и уменьшить ошибки.

Шаг 5. Div – это ведущий элемент. Проходя через условие, если ведущий элемент близок к нулю, то выводим, что система не имеет решения. Если условие не соблюдается, делим всю строку на ведущий элемент.

Шаг 6. Обнуление элементов ниже диагонали. Цикл int k = i + 1; k < rows; k++ проходится по элементам ниже диагонали и через второй цикл int j = i; j < cols; j++ вычитает из нижней строки верхнюю строку, умноженную на множитель для обнуления.

Шаг 7. Обратный ход метода. Цикл int i = vars - 1; i >= 0; i-- начинается с последней переменной, заканчивая первой. Далее берется начальное значение, равное свободному члену, и вычитаем все известные значения, умноженные на их коэффициенты.

Шаг 8. Возвращаем массив решений.

## **4. Выбор языка программирования.**

C# представляет собой современный язык программирования, который вобрал в себя множество особенностей, обеспечивающих его популярность среди разработчиков. Как и его предшественники C, C++ и Java, C# имеет синтаксис, понятный для опытных программистов, и тем не менее, обладает уникальными характеристиками, которые выделяют его на фоне других языков программирования.

Одной из заметных особенностей C# является его высокая производительность, что делает язык подходящим для разработки приложений различного назначения. Простота синтаксиса способствовала более лёгкому обучению и быстрому освоению языка начинающими программистами. Благодаря этому C# стал удобным инструментом для создания программ, решающих сложные алгоритмические задачи, такие как метод Гаусса.

Кроме того, объектно-ориентированное программирование в C# включает в себя такие принципы, как инкапсуляция, наследование и полиморфизм. Это позволяет разработчикам создавать гибкие, масштабируемые и поддерживаемые приложения. Одной из его сильных сторон является возможность обработки исключений и работа с ресурсами, что облегчает отладку программного обеспечения.

Тем не менее, существуют и недостатки. Высокий порог вхождения для некоторых разработчиков может сдерживать их от полного освоения языка, а также потребовать времени на изучение дополнительных концепций и библиотек. К тому же, C# имеет ограниченное применение в веб-разработке по сравнению с языками, такими как JavaScript или PHP, что может снизить его привлекательность для некоторых проектов.

Когда говорить о конкуренции с C++, можно отметить, что хотя C# и предлагает удобство, его производительность в некоторых сценариях может уступать результатам, полученным с использованием C++. Тем не менее, для большинства приложений, требующих динамического управления памятью и производительности, C# обеспечивает богатые возможности благодаря своей выразительности и большому количеству встроенных библиотек.

Необходимо учитывать, что выбор языка программирования зависит от конкретной задачи, требований проекта и предпочтений команды разработчиков. C# предоставляет все необходимые инструменты для эффективной разработки, особенно в среде Windows, и активно используется в корпоративных решениях и разработке игр на платформе Unity. Поэтому, несмотря на возможные ограничения, выбор C# в качестве языка для реализации алгоритмов, таких как метод Гаусса, вполне оправдан.

# **5. Описание программного комплекса.**

Программа представляет собой инструмент для решения систем линейный уравнений методом Гаусса.

Проект(программа) состоит из файлов:

• Program.cs: Файл служит точкой входа в приложение, отвечая за его запуск и подключение необходимых форм.

• Form1.cs: Данный файл содержит класс, отвечающий за управление компонентами формы. В него входит обработка ввода и вывода данных, а также реализация генерации ячеек матрицы и вызовов методов.

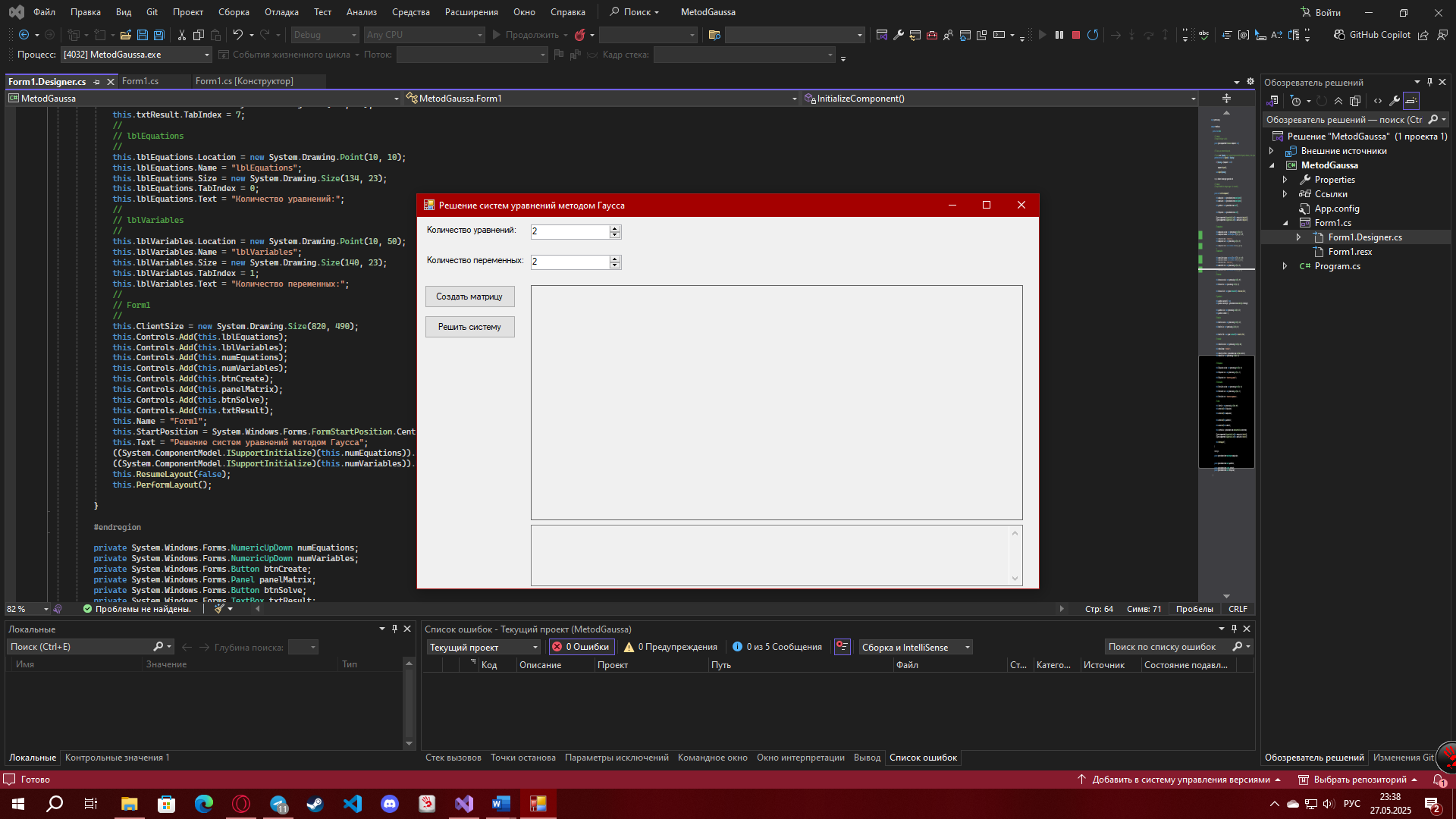


Рис.2. Окно разработанной программы.

Во время разработки программы были использованы следующие компоненты:

• System.Windows.Forms – классы для создания графического пользовательского интерфейса (GUI) в Windows Forms.Позволяет вводить и выводить данные в специальных контейнерах, использовать различные инструменты для оперирования с данными, выводить диалоговые окна.

• System.Drawing – Классы для работы с графикой.

• System.Text – Директива, позволяющая использовать типы из пространств имен без указания полного пространства имен этого типа.

Интерфейс программы – форма с размещенными в ней сгруппированными элементами (см. рис. 2):

* Слева сверху располагаются элементы для ввода пользователем количества уравнений и переменных.
* Под ними располагаются кнопки создания матрицы и решения системы.
* По середине располагается окно , где появится таблица с ячейками для ввода элементов матрицы.
* Под ней расположено окно для вывода решения заданной матрицы.

Алгоритм работы программы:

1. Ввести количество уравнений и переменных для генерации матрицы в виде таблицы.
2. Нажать кнопку «Создать матрицу» и по середине будет сгенерирована таблица.

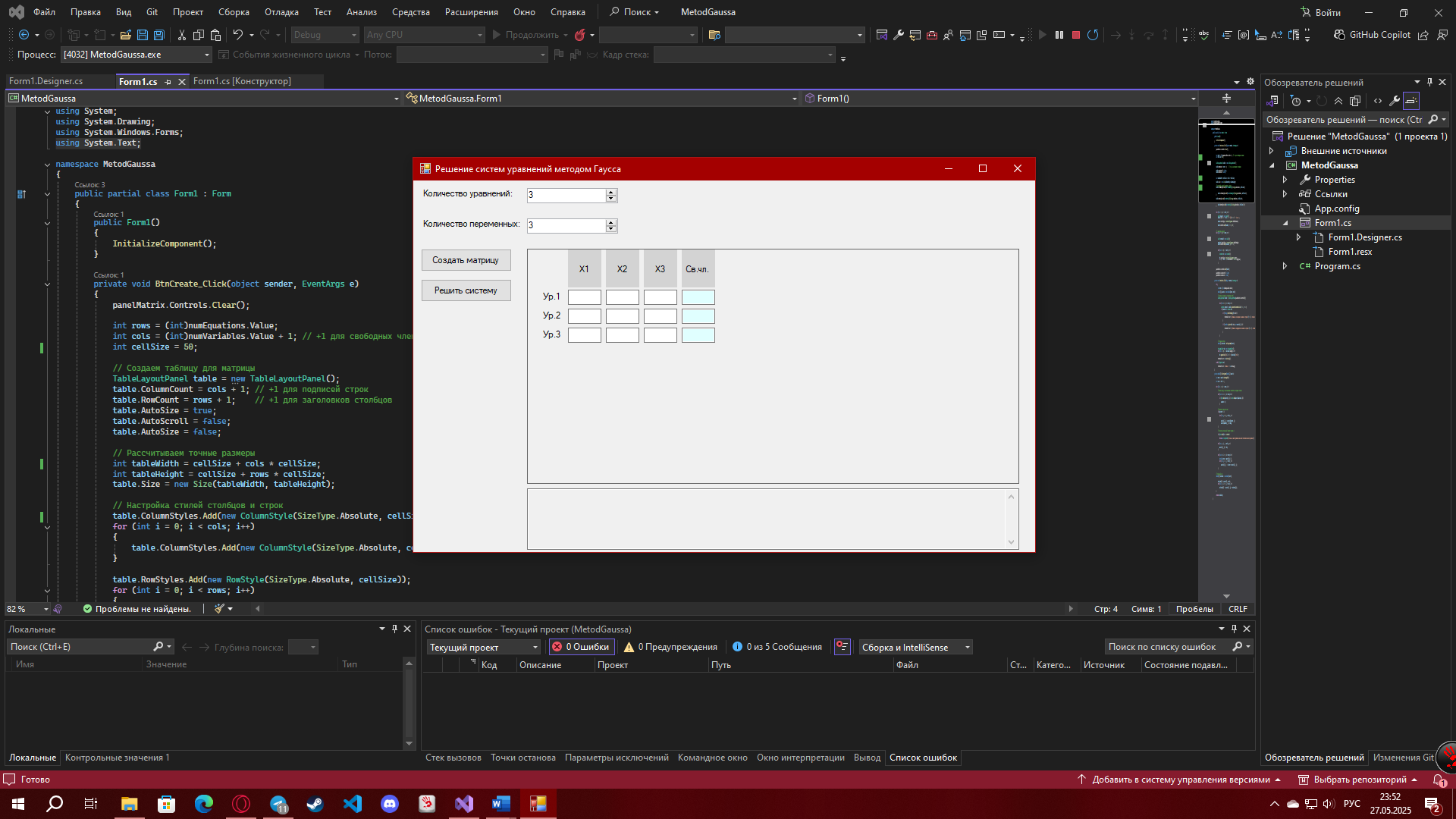


Рис. 3. Окно с сгенерированной таблицей.

1. При нажатии на кнопку «Решить систему», при условии, что данные введены, в нижней части программы выведется решение задачи.

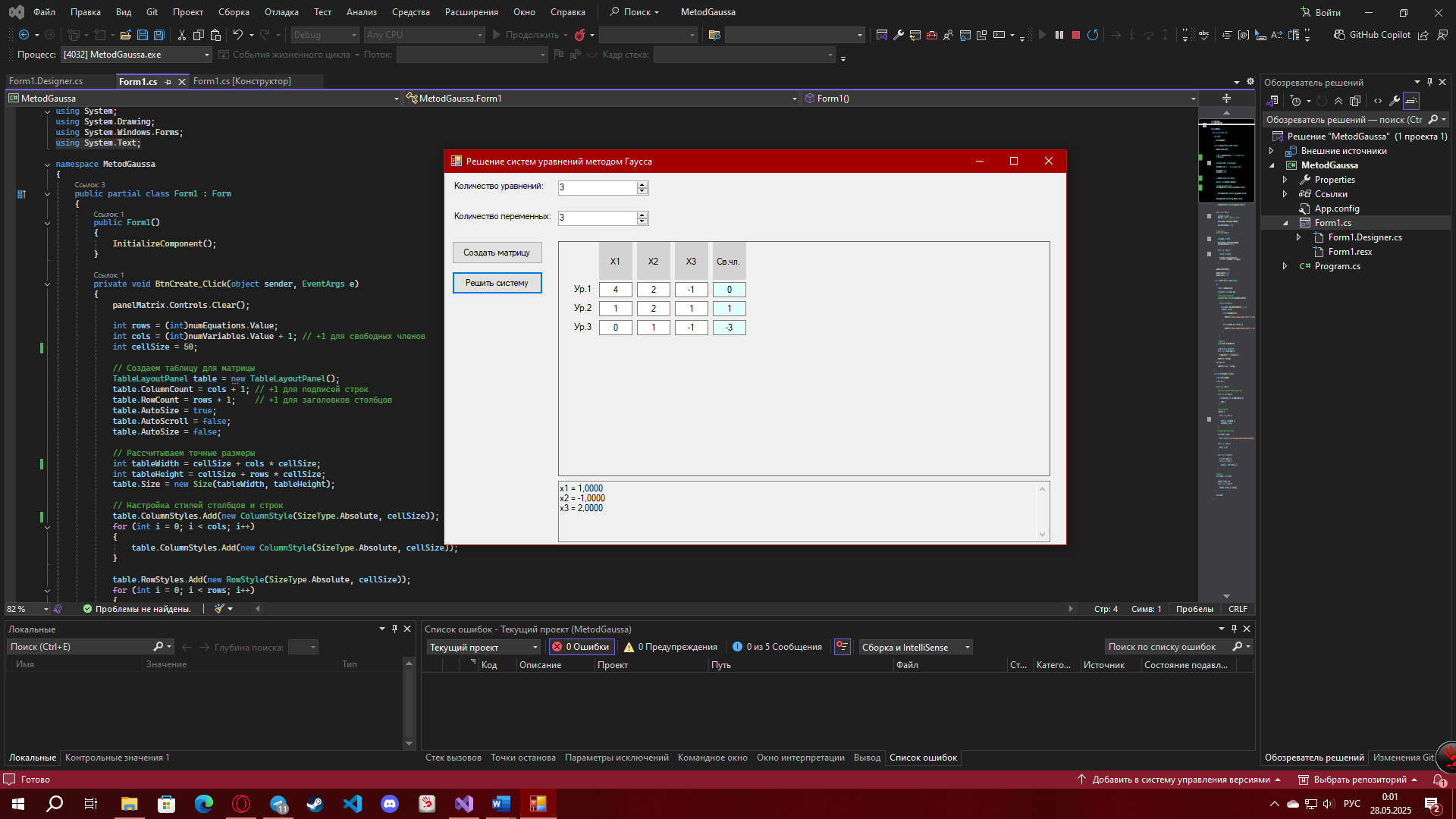


Рис. 4. Результат работы программы.

# **6.Анализ результатов**

Проведем сравнение вычисления неизвестных переменных ручным способом (с использованием метода Гаусса) с вычисленным значением в приложении:

Решим систему методом Гаусса:

Запишем систему в виде расширенной матрицы:

Для удобства вычислений поменяем строки местами:

Преобразуем матрицу к нижнеступенчатому виду:

4(2)-(3)

6(1)-(2)

Теперь исходную систему можно записать так:

Итого, у нас получается:

,

После решения данной системы линейных алгебраических уравнений ручным способом и выполненное решение программы сошлись. Метод Гаусса в программе использует выбор ведущего элемента, что уменьшает ошибки округления. Но при ручном решении человек может допустить арифметические ошибки, особенно в больших системах. Ручное решение занимает много времени, особенно для систем с большим количеством уравнений. Программа же обрабатывает данные мгновенно. Программа предоставляет графический интерфейс, что упрощает ввод данных и интерпретацию результатов. В ручном режиме легко запутаться, особенно с множеством переменных. Также программа проверяет корректность ввода, что снижает вероятность ошибок из-за неверных данных.

# **Заключение**

В заключение данной курсовой работы можно подвести итоги, касающиеся реализации метода Гаусса для решения систем линейных уравнений на языке C#. Метод Гаусса, как один из наиболее распространенных и эффективных алгоритмов, позволяет решать системы уравнений различной сложности, что делает его незаменимым инструментом в математике и программировании.

Разработка алгоритма для реализации метода Гаусса на языке C# была выполнена с учетом всех необходимых этапов, включая преобразование матрицы коэффициентов в верхнетреугольную форму и последующий обратный ход для нахождения значений переменных. Важно отметить, что использование языка C# для реализации данного метода позволило создать интуитивно понятный интерфейс, который упрощает процесс ввода данных и делает программу доступной для пользователей с различным уровнем подготовки. Это особенно актуально в образовательных целях, когда студенты могут самостоятельно экспериментировать с различными системами уравнений и наблюдать за результатами.

Тестирование программы на примерах системы уравнений подтвердило ее работоспособность и эффективность. Результаты тестирования показали, что метод Гаусса демонстрирует высокую скорость решения. Это делает его предпочтительным выбором для решения больших систем уравнений, где важна не только точность, но и скорость вычислений.

Перспективы использования метода Гаусса в программировании выглядят многообещающими. С учетом постоянного роста объемов данных и сложности задач, связанных с решением систем уравнений, метод Гаусса может быть адаптирован и улучшен с использованием современных технологий, таких как параллельные вычисления и оптимизация алгоритмов. Это открывает новые горизонты для его применения в различных областях, включая инженерные расчеты, экономические модели и научные исследования.

Таким образом, данная курсовая работа не только продемонстрировала практическую реализацию метода Гаусса на языке C#, но и подчеркнула его значимость и актуальность в современном программировании.

# **Список литературы**

1. Заика Ирина Викторовна СОЗДАНИЕ ОКОННЫХ ПРИЛОЖЕНИЙ СРЕДСТВАМИ VISUAL STUDIO С++ // Вестник Таганрогского института имени А. П. Чехова. 2022. №2. URL: <https://cyberleninka.ru/article/n/sozdanie-okonnyh-prilozheniy-sredstvami-visual-studio-s>.
2. М Х. Раупова, Б Э. Хаитматова ДВА РАЗНЫХ СПОСОБА РЕШЕНИЯ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ, ИХ ПРЕИМУЩЕСТВА И НЕДОСТАТКИ // Academic research in educational sciences. 2023. №CSPU Conference 1. URL: <https://cyberleninka.ru/article/n/dva-raznyh-sposoba-resheniya-sistemy-lineynyh-uravneniy-ih-preimuschestva-i-nedostatki>.
3. Иламанов Б. Б., Хайдарова О. ОСНОВЫ И ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ НА C++: ОТ НАЧАЛЬНЫХ ШАГОВ ДО ПРОДВИНУТЫХ КОНЦЕПЦИЙ // Вестник науки. 2024. №1 (70). URL: <https://cyberleninka.ru/article/n/osnovy-i-printsipy-programmirovaniya-na-c-ot-nachalnyh-shagov-do-prodvinutyh-kontseptsiy>.
4. Горячкин Борис Сергеевич, Бакланов Никита Владимирович, Попов Максим Алексеевич ПОНЯТНЫЙ КОД // E-Scio. 2023. №2 (77). URL: <https://cyberleninka.ru/article/n/ponyatnyy-kod>.
5. Базеева Наталья Алексеевна, Харитонов Виталий Игоревич, Малкина Анастасия Анатольевна РАЗРАБОТКА ЧЕРЕЗ ТЕСТИРОВАНИЕ (TDD) // E-Scio. 2023. №4 (79). URL: <https://cyberleninka.ru/article/n/razrabotka-cherez-testirovanie-tdd>.
6. Овечкина Анастасия Игоревна СРАВНЕНИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ С++ И С# // IN SITU. 2023. №4. URL: <https://cyberleninka.ru/article/n/sravnenie-yazykov-programmirovaniya-s-i-s>.
7. Перцев С. А., Островский В. В., Чернова С. В. ЯЗЫК ПРОГРАММИИРОВАНИЯ С#. ДОСТОИНСТВА И НЕДОСТАТКИ, ОСНОВНЫЕ ВОЗМОЖНОСТИ. СОПОСТАВЛЕНИЕ С ДРУГИМИ ЯЗЫКАМИ ПРОГРАММИРОВАНИЯ // Экономика и социум. 2016. №3 (22). URL: <https://cyberleninka.ru/article/n/yazyk-programmiirovaniya-s-dostoinstva-i-nedostatki-osnovnye-vozmozhnosti-sopostavlenie-s-drugimi-yazykami-programmirovaniya>.

# **Приложение 1**

Код Form1.cs

using System;

using System.Drawing;

using System.Windows.Forms;

using System.Text;

namespace MetodGaussa

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void BtnCreate\_Click(object sender, EventArgs e)

{

panelMatrix.Controls.Clear();

int rows = (int)numEquations.Value;

int cols = (int)numVariables.Value + 1; // +1 для свободных членов

int cellSize = 50;

// Создаем таблицу для матрицы

TableLayoutPanel table = new TableLayoutPanel();

table.ColumnCount = cols + 1; // +1 для подписей строк

table.RowCount = rows + 1; // +1 для заголовков столбцов

table.AutoSize = true;

table.AutoScroll = false;

table.AutoSize = false;

// Рассчитываем точные размеры

int tableWidth = cellSize + cols \* cellSize;

int tableHeight = cellSize + rows \* cellSize;

table.Size = new Size(tableWidth, tableHeight);

// Настройка стилей столбцов и строк

table.ColumnStyles.Add(new ColumnStyle(SizeType.Absolute, cellSize));

for (int i = 0; i < cols; i++)

{

table.ColumnStyles.Add(new ColumnStyle(SizeType.Absolute, cellSize));

}

table.RowStyles.Add(new RowStyle(SizeType.Absolute, cellSize));

for (int i = 0; i < rows; i++)

{

table.RowStyles.Add(new RowStyle(SizeType.Absolute, cellSize/2));

}

// Заголовки столбцов

for (int c = 0; c < cols; c++)

{

Label header = new Label();

header.Text = c < cols - 1 ? $"X{c + 1}" : "Св.чл.";

header.Size = new Size(cellSize, cellSize);

header.TextAlign = ContentAlignment.MiddleCenter;

header.BackColor = Color.LightGray;

table.Controls.Add(header, c + 1, 0);

}

// Заполняем матрицу

for (int r = 0; r < rows; r++)

{

// Подпись строки

Label rowLabel = new Label();

rowLabel.Text = $"Ур.{r + 1}";

rowLabel.TextAlign = ContentAlignment.MiddleRight;

table.Controls.Add(rowLabel, 0, r + 1);

// Поля ввода

for (int c = 0; c < cols; c++)

{

TextBox tb = new TextBox();

tb.Size = new Size(cellSize, cellSize);

tb.TextAlign = HorizontalAlignment.Center;

if (c == cols - 1) tb.BackColor = Color.LightCyan;

table.Controls.Add(tb, c + 1, r + 1);

}

}

panelMatrix.Controls.Add(table);

panelMatrix.AutoScroll = false;

panelMatrix.AutoSize = false;

}

private void BtnSolve\_Click(object sender, EventArgs e)

{

try

{

int rows = (int)numEquations.Value;

int cols = (int)numVariables.Value + 1;

double[,] matrix = new double[rows, cols];

// Получаем данные из текстовых полей

TableLayoutPanel table = (TableLayoutPanel)panelMatrix.Controls[0];

for (int r = 0; r < rows; r++)

{

for (int c = 0; c < cols; c++)

{

Control control = table.GetControlFromPosition(c + 1, r + 1);

if (control is TextBox tb)

{

if (string.IsNullOrEmpty(tb.Text))

{

txtResult.Text = $"Ошибка: не введено значение в строке {r + 1}, столбце {c + 1}";

return;

}

if (!double.TryParse(tb.Text, out matrix[r, c]))

{

txtResult.Text = $"Ошибка: некорректное значение в строке {r + 1}, столбце {c + 1}";

return;

}

}

}

}

// Решаем систему

double[] solution = SolveSystem(matrix);

// Выводим результат

StringBuilder sb = new StringBuilder();

for (int i = 0; i < solution.Length; i++)

{

sb.AppendLine($"x{i + 1} = {solution[i]:F4}");

}

txtResult.Text = sb.ToString();

}

catch (Exception ex)

{

txtResult.Text = "Ошибка: " + ex.Message;

}

}

private double[] SolveSystem(double[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

int vars = cols - 1;

// Прямой ход метода Гаусса

for (int i = 0; i < rows; i++)

{

// Ищем строку с максимальным элементом в текущем столбце

int maxRow = i;

for (int k = i + 1; k < rows; k++)

{

if (Math.Abs(matrix[k, i]) > Math.Abs(matrix[maxRow, i]))

{

maxRow = k;

}

}

// Меняем строки местами

if (maxRow != i)

{

for (int j = 0; j < cols; j++)

{

double temp = matrix[i, j];

matrix[i, j] = matrix[maxRow, j];

matrix[maxRow, j] = temp;

}

}

// Делаем диагональный элемент равным 1

double div = matrix[i, i];

if (Math.Abs(div) < 0.00001)

{

throw new Exception("Система не имеет решения или имеет бесконечно много решений");

}

for (int j = i; j < cols; j++)

{

matrix[i, j] /= div;

}

// Обнуляем элементы ниже

for (int k = i + 1; k < rows; k++)

{

double factor = matrix[k, i];

for (int j = i; j < cols; j++)

{

matrix[k, j] -= factor \* matrix[i, j];

}

}

}

// Обратный ход

double[] solution = new double[vars];

for (int i = vars - 1; i >= 0; i--)

{

solution[i] = matrix[i, vars];

for (int j = i + 1; j < vars; j++)

{

solution[i] -= matrix[i, j] \* solution[j];

}

}

return solution;

}

}

}

# **Приложение 2**

Код Form1.Designer.cs

using System.Drawing;

using System;

namespace MetodGaussa

{

partial class Form1

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.numEquations = new System.Windows.Forms.NumericUpDown();

this.numVariables = new System.Windows.Forms.NumericUpDown();

this.btnCreate = new System.Windows.Forms.Button();

this.panelMatrix = new System.Windows.Forms.Panel();

this.btnSolve = new System.Windows.Forms.Button();

this.txtResult = new System.Windows.Forms.TextBox();

this.lblEquations = new System.Windows.Forms.Label();

this.lblVariables = new System.Windows.Forms.Label();

((System.ComponentModel.ISupportInitialize)(this.numEquations)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.numVariables)).BeginInit();

this.SuspendLayout();

//

// numEquations

//

this.numEquations.Location = new System.Drawing.Point(150, 10);

this.numEquations.Maximum = new decimal(new int[] {10, 0, 0, 0});

this.numEquations.Minimum = new decimal(new int[] {1, 0, 0, 0});

this.numEquations.Name = "numEquations";

this.numEquations.Size = new System.Drawing.Size(120, 20);

this.numEquations.TabIndex = 2;

this.numEquations.Value = new decimal(new int[] {2, 0, 0, 0});

//

// numVariables

//

this.numVariables.Location = new System.Drawing.Point(150, 50);

this.numVariables.Maximum = new decimal(new int[] {10, 0, 0, 0});

this.numVariables.Minimum = new decimal(new int[] {1, 0, 0, 0});

this.numVariables.Name = "numVariables";

this.numVariables.Size = new System.Drawing.Size(120, 20);

this.numVariables.TabIndex = 3;

this.numVariables.Value = new decimal(new int[] {2, 0, 0, 0});

//

// btnCreate

//

this.btnCreate.Location = new System.Drawing.Point(10, 90);

this.btnCreate.Name = "btnCreate";

this.btnCreate.Size = new System.Drawing.Size(120, 30);

this.btnCreate.TabIndex = 4;

this.btnCreate.Text = "Создать матрицу";

this.btnCreate.Click += new System.EventHandler(this.BtnCreate\_Click);

//

// panelMatrix

//

this.panelMatrix.AutoScroll = true;

this.panelMatrix.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

this.panelMatrix.Location = new System.Drawing.Point(150, 90);

this.panelMatrix.Name = "panelMatrix";

this.panelMatrix.Size = new System.Drawing.Size(649, 310);

this.panelMatrix.TabIndex = 5;

//

// btnSolve

//

this.btnSolve.Location = new System.Drawing.Point(10, 130);

this.btnSolve.Name = "btnSolve";

this.btnSolve.Size = new System.Drawing.Size(120, 30);

this.btnSolve.TabIndex = 6;

this.btnSolve.Text = "Решить систему";

this.btnSolve.Click += new System.EventHandler(this.BtnSolve\_Click);

//

// txtResult

//

this.txtResult.Location = new System.Drawing.Point(150, 406);

this.txtResult.Multiline = true;

this.txtResult.Name = "txtResult";

this.txtResult.ReadOnly = true;

this.txtResult.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;

this.txtResult.Size = new System.Drawing.Size(649, 81);

this.txtResult.TabIndex = 7;

//

// lblEquations

//

this.lblEquations.Location = new System.Drawing.Point(10, 10);

this.lblEquations.Name = "lblEquations";

this.lblEquations.Size = new System.Drawing.Size(134, 23);

this.lblEquations.TabIndex = 0;

this.lblEquations.Text = "Количество уравнений:";

//

// lblVariables

//

this.lblVariables.Location = new System.Drawing.Point(10, 50);

this.lblVariables.Name = "lblVariables";

this.lblVariables.Size = new System.Drawing.Size(140, 23);

this.lblVariables.TabIndex = 1;

this.lblVariables.Text = "Количество переменных:";

//

// Form1

//

this.ClientSize = new System.Drawing.Size(820, 490);

this.Controls.Add(this.lblEquations);

this.Controls.Add(this.lblVariables);

this.Controls.Add(this.numEquations);

this.Controls.Add(this.numVariables);

this.Controls.Add(this.btnCreate);

this.Controls.Add(this.panelMatrix);

this.Controls.Add(this.btnSolve);

this.Controls.Add(this.txtResult);

this.Name = "Form1";

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

this.Text = "Решение систем уравнений методом Гаусса";

((System.ComponentModel.ISupportInitialize)(this.numEquations)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.numVariables)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.NumericUpDown numEquations;

private System.Windows.Forms.NumericUpDown numVariables;

private System.Windows.Forms.Button btnCreate;

private System.Windows.Forms.Panel panelMatrix;

private System.Windows.Forms.Button btnSolve;

private System.Windows.Forms.TextBox txtResult;

private System.Windows.Forms.Label lblEquations;

private System.Windows.Forms.Label lblVariables;

}

}