Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Воронежский государственный лесотехнический университет

имени Г.Ф. Морозова»

Кафедра Вычислительной техники и информационных систем

(название кафедры)

**ЗАДАНИЕ**

Курсовая работа

(вид работы)

Студенту Величко ВладиславАндреевич ИС2-191-ОБ

(фамилия и полные инициалы)

Информационные системы и технологии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование направления подготовки)

Срок представления работы к защите «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 202\_\_ г.

Исходные данные для проектирования:таблицы, содержащие данные об организациях, оказывающих услуги участки работ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перечень вопросов, подлежащих разработке: 1.Проектирование и создание пользовательских форм в среде программирования Rad studio \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перечень листов чертежей и плакатов графических документов формата А 1:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель доцент \_\_\_\_\_\_\_\_\_\_\_\_\_ Меерсон В.Э.

(ученая степень, ученое звание) (подпись) (инициалы и фамилия)

Задание принял студент \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Величко В.А.

(подпись) (число, месяц, год) (инициалы и фамилия)

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Воронежский государственный лесотехнический университет

имени Г.Ф. Морозова»

Кафедра Вычислительной техники и информационных систем

(название кафедры)

**Пояснительная записка**

Курсовая работа

(вид работы)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(тема)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование направления подготовки)

По дисциплине \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| Студент группыИС2-191-ОБ  (номер группы)  Руководитель, доцент  (ученая степень, ученое звание) | \_\_\_\_\_\_\_\_\_\_\_  (подпись)  \_\_\_\_\_\_\_\_\_\_  (подпись) | Величко А.В.  (инициалы и фамилия)  Меерсон В.Э.  (инициалы и фамилия) |

Воронеж 2020

(год)

Оглавление

[1 ВВЕДЕНИЕ 2](#_Toc41763361)

[2 Теоретическая часть 3](#_Toc41763362)

[2.1 Структурное программирование 3](#_Toc41763363)

[2.2 Переменные и типы данных 3](#_Toc41763364)

[2.3 Основные структуры алгоритма 4](#_Toc41763365)

[2.4 Элементы управления rad studio 5](#_Toc41763366)

[3 Постановка задачи 6](#_Toc41763367)

[4 Алгоритм 8](#_Toc41763368)

[4.1 Алгоритм Гаусса для решения систем алгебраических уравнений 8](#_Toc41763369)

[5 Заключение 10](#_Toc41763370)

[6 Список использованной литературы 11](#_Toc41763371)

[7 Листинг программы 12](#_Toc41763372)

# ВВЕДЕНИЕ

Решение систем линейных алгебраических уравнений – одна из основных задач вычислительной линейной алгебры. Хотя задача решения системы линейных уравнений сравнительно редко представляет самостоятельный интерес для приложений, от умения эффективно решать такие системы часто зависит сама возможность математического моделирования самых разнообразных процессов с применением ЭВМ. Значительная часть численных методов решения различных задач включает в себя решение систем линейных уравнений как элементарный шаг соответствующего алгоритма.

Одна из трудностей практического решения систем большой размерности связанна с ограниченностью оперативной памяти ЭВМ. Хотя объём оперативной памяти вновь создаваемых вычислительных машин растет очень быстро, тем не менее, еще быстрее возрастают потребности практики в решении задач все большей размерности. В значительной степени ограничения на размерность решаемых систем можно снять, если использовать для хранения матрицы внешние запоминающие устройства. Однако в этом случае многократно возрастают как затраты машинного времени, так и сложность соответствующих алгоритмов. Поэтому при создании вычислительных алгоритмов линейной алгебры большое внимание уделяют способам компактного размещения элементов матриц в памяти ЭВМ

Цель курсовой работы: Изучить решение систем линейных уравнений методом Гаусса.

Задача курсовой работы: Разработать алгоритм решения систем линейных уравнений методом Крамера и Гаусса.

# Теоретическая часть

## Структурное программирование

Структу́рное программи́рование — методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков. Предложена в 1970-х годах Э. Дейкстрой и др.

В соответствии с данной методологией любая программа строится без использования оператора goto из трёх базовых управляющих структур: последовательность, ветвление, цикл; кроме того, используются подпрограммы. При этом разработка программы ведётся пошагово, методом «сверху вниз».

Метод пошаговой детализации (или метод разработки «сверху - вниз»). При этом методе первоначально продумывается и фиксируется множество данных и результатов алгоритма без детальной проработки отдельных частей. Задачу разбивают на автономные части, каждая из которых существенно проще. Может оказаться необходимым повторять процесс детализации многократно, но это определяется только сложностью решаемых задач. Конечным уровнем детализации алгоритма можно считать такой, при котором в алгоритме нет действий более крупных, чем: обращение к готовому алгоритму; вычисление арифметического выражения и присваивание значения переменной; сравнение арифметических выражений (или переменных); ввод (вывод) данных и т.п.

## Переменные и типы данных

Переменная в императивном программировании — поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным. Данные, находящиеся в переменной (то есть по данному адресу памяти), называются значением этой переменной. В других парадигмах программирования, например, в функциональной и логической, понятие переменной оказывается несколько иным. В таких языках переменная определяется как имя, с которым может быть связано значение, или даже как место (location) для хранения значения.

Тип данных характеризует одновременно:

* множество допустимых значений, которые могут принимать данные, принадлежащие к этому типу;
* набор операций, которые можно осуществлять над данными, принадлежащими к этому типу.

Первое свойство можно рассматривать как теоретико-множественное определение понятия типа; второе — как процедурное (или поведенческое) определение. Кроме этого, в программировании используется низкоуровневое определение типа — как заданных размерных и структурных характеристик ячейки памяти, в которую можно поместить некое значение, соответствующее этим характеристикам. Такое определение является частным случаем теоретико-множественного. На практике, с ним связан ряд важных свойств (обусловленных особенностями организации памяти копьютера, требующих отдельного рассмотрения.

|  |  |  |
| --- | --- | --- |
| **Тип C++** | **Номинальное выделение памяти** | **Диапазон значений** |
| [Integer](https://docs.microsoft.com/ru-ru/dotnet/visual-basic/language-reference/data-types/integer-data-type) | 4 байта | от-2 147 483 648 до 2 147 483 647 (подписано) |
| [Float](https://docs.microsoft.com/ru-ru/dotnet/visual-basic/language-reference/data-types/long-data-type) | Зависит от значения n  *(*от **1** до **53***)* | - 1,79E+308 — -2,23E-308, 0 и 2,23E-308 — 1,79E+308 |

## Основные структуры алгоритма

Алгоритм – точное предписание исполнителю совершить определенную последовательность действий для достижения поставленной цели за конечное число шагов.

Согласно основным принципам структурного программирования в алгоритме решения поставленной задачи используются только три базовых структуры: последовательность, ветвление, цикл.

Ветвление – структура алгоритма, обеспечивающая выполнение команды или набора команд в зависимости от истинности переменной или выражения.

Цикл — разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций. Также циклом может называться любая многократно исполняемая последовательность инструкций, организованная любым способом.

Подпрограмма — поименованная или иным образом идентифицированная часть компьютерной программы, содержащая описание определённого набора действий. В данном алгоритме подпрограммы вызываются для перевода чисел из одной системы счисления в другую.

## Элементы управления rad studio

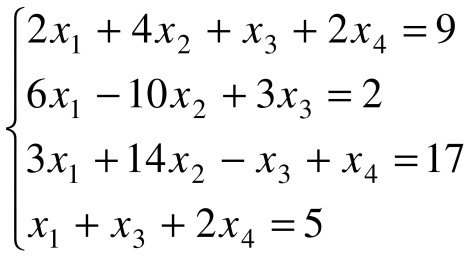
Rad studio обладает встроенным набором элементов управления. Используя этот набор и редактор форм не трудно создать любой пользовательский интерфейс, который будет удовлетворять всем требованиям, предъявляемым к интерфейсу в среде Windows. В данной главе дан обзор встроенных элементов управления Rad studio. Элементы управления являются объектами. Поэтому, как любые объекты, они обладают свойствами, методами и событиями. Элементы управления создаются при помощи панели инструментов Элементы управления. На этой панели представлены кнопки, позволяющие конструировать элементы управления, а также кнопки вызова окна свойств, перехода в режим конструктора и редактор кода.

|  |  |  |
| --- | --- | --- |
| Элемент управления | Имя | Кнопка для создания элемента |
| Поле | Edit |  |
| Надпись | Label |  |
| Кнопка | Button |  |
| Рисунок | Image |  |

В данной курсовой работе использовались такие элементы управления как Кнопки, Надписи и Поля для текста.

## Условие задачи

Написать программу для решения системы линейных уравнений методом Гаусса

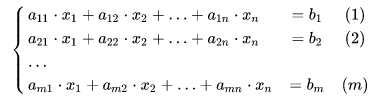


# Постановка задачи

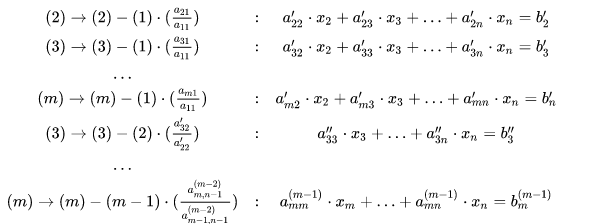
Одним из самых распространенных методов решения систем линейных уравнений является метод Гаусса. Этот метод известен в различных вариантах уже более 2000 лет.

Вычисления с помощью метода Гаусса заключаются в последовательном исключении неизвестных из системы для преобразования ее к эквивалентной системе с верхней треугольной матрицей. Вычисления значений неизвестных производят на этапе обратного хода.

В простейшем случае алгоритм выглядит так:

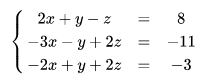


**Прямой ход:**

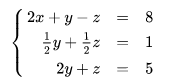


**Обратный ход.** Из последнего ненулевого уравнения выражаем базисную переменную через небазисные и подставляем в предыдущие уравнения. Повторяя эту процедуру для всех базисных переменных, получаем фундаментальное решение.

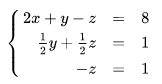
**Пример:** Покажем, как методом Гаусса можно решить следующую систему:



Обнулим коэффициенты при x во второй и третьей строчках. Для этого прибавим к ним первую строчку, умноженную на 3/2 и 1, соответственно:



Теперь обнулим коэффициент при y в третьей строке, вычтя из неё вторую строку, умноженную на 4:



В результате мы привели исходную систему к треугольному виду, тем самым закончив первый этап алгоритма.

На втором этапе разрешим полученные уравнения в обратном порядке. Имеем:

z = -1из третьего;

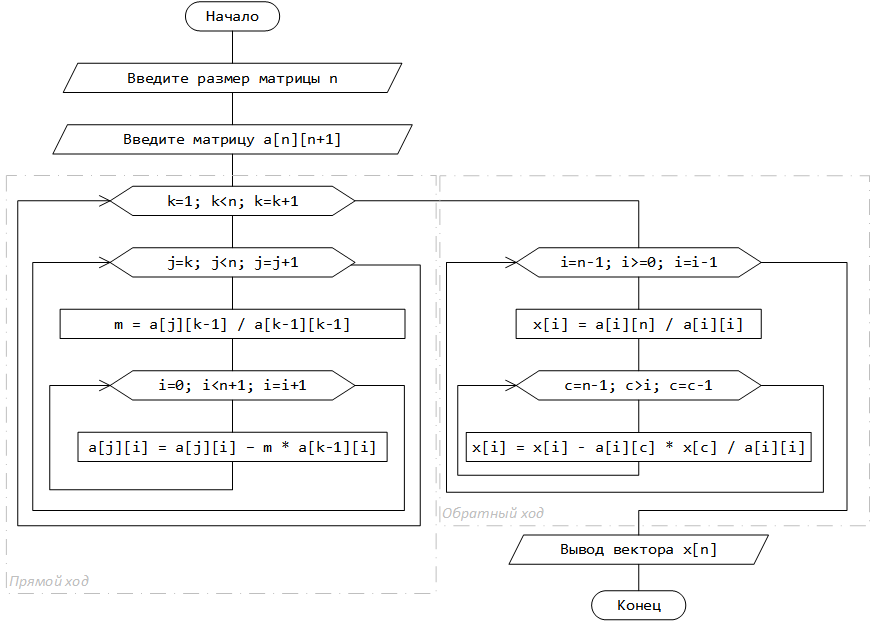
y = 3 из второго, подставив полученное z;

x = 2 из первого, подставив полученные y и z;

Таким образом исходная система решена.

В случае, если число уравнений в совместной системе получилось меньше числа неизвестных, то тогда ответ будет записываться в виде [фундаментальной системы 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).

# Алгоритм Гаусса для решения систем алгебраических уравнений



# Ход рещения задачи (2-3 листа)

Для выполнения задачи была использована программа Rad Studio C++ Builder 10 Seattle

На первой стадии разработки был разработан пользованельский интерфейс (рис. 1)

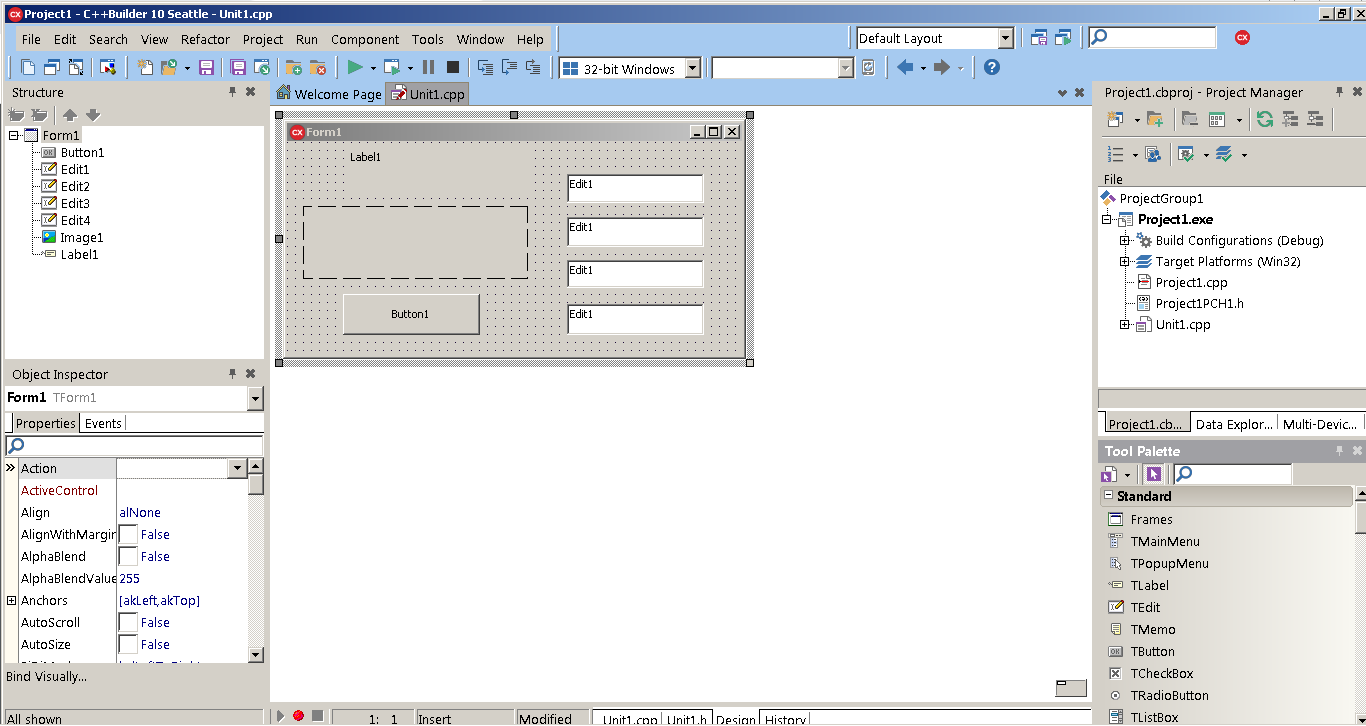


Рисунок - 1

В ходе разработки алгоритма использовались математические функции. В следствии чего при написании программы возникла потребность в использовании библиотек. (рис. 2)



Рисунок – 2

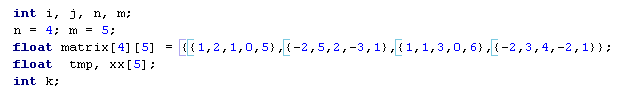


Рисунок - 3 Объявление переменных

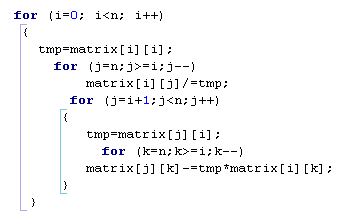


Рисунок – 4 Прямой ход

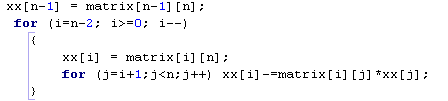


Рисунок – 5 Обратный ход

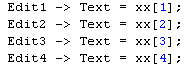
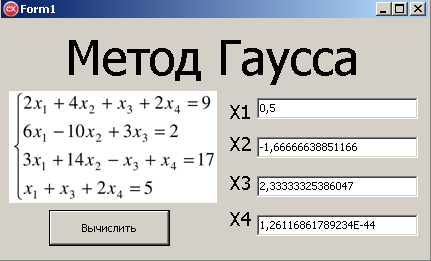


Рисунок – 6 Вывод решения

Тестовый пример



Проанализировав результаты работы программы можно утверждать что программа работает корректно и обладает достаточной точностью вычислений.

# Заключение

В данной курсовой работе реализован алгоритм решения систем линейных уравнений методом Гаусса. На основе рассмотренных теоретических вопросов создана действующая программа, алгоритм, которой приводится в курсовой работе.

Данная программа значительно упрощает решение систем линейных алгебраических уравнений матричными методами и может быть использована в различных областях, где требуется решение систем линейных уравнений.

# Список использованной литературы

1. Абрамовица М. Справочник по специальным формулам и функциям /М. Абрамовица, И. Стиган.: Наука, 2010
2. Бахвалов Н.С., Жидков Н.П., Кобельников Г.М. Численные методы. –М.: Наука, 1987. –445с
3. Васильков Ю.В., Василькова Н.Н. Компьютерные технологии вычислений в математическом моделировании: Учебное пособие. –М.: Финансы и статистика, 2002. –256 с.
4. Воробьева Г.Н., Данилова А.Н. Практикум по вычислительной математике. –М.: Высш.шк., 1990, –184 с.
5. Д.Мак–Кракен, У.Дорн. Численные методы и программирование на Фортране. –М.: Мир, 1977. –584 с.
6. Плис А.И., Сливина Н.А. Лабораторный практикум по высшей математике. –М.: Высшая школа, 1994. –416 с.
7. Самарский А.А., Гулин А.В. Численные методы: Учебное пособие для ВУЗов. М.: Наука, 1989. –432 с.

# Листинг программы

#include <math.h>

#include <iostream>

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

int i, j, n, m;

n = 4; m = 5;

float matrix[4][5] = {{1,2,1,0,5},{-2,5,2,-3,1},{1,1,3,0,6},{-2,3,4,-2,1}};

float tmp, xx[5];

int k;

for (i=0; i<n; i++)

{

tmp=matrix[i][i];

for (j=n;j>=i;j--)

matrix[i][j]/=tmp;

for (j=i+1;j<n;j++)

{

tmp=matrix[j][i];

for (k=n;k>=i;k--)

matrix[j][k]-=tmp\*matrix[i][k];

}

}

xx[n-1] = matrix[n-1][n];

for (i=n-2; i>=0; i--)

{

xx[i] = matrix[i][n];

for (j=i+1;j<n;j++) xx[i]-=matrix[i][j]\*xx[j];

}

Edit1 -> Text = xx[1];

Edit2 -> Text = xx[2];

Edit3 -> Text = xx[3];

Edit4 -> Text = xx[4];

}