|  |  |  |  |
| --- | --- | --- | --- |
|  | | для прик эмбл |  |
| МИНОБРНАУКИ РОССИИ | | | |
| **Федеральное государственное бюджетное образовательное учреждение**  **высшего профессионального образования**  «Московский технологический университет» | | | |
| Институт информационных технологий  *(наименование института)* | | | |
| Математического обеспечения и стандартизации информационных технологий  *(наименование кафедры)* | | | |

|  |  |
| --- | --- |
| **КУРСОВАЯ РАБОТА** | |
| **по дисциплине** | |
| **«**Структуры и алгоритмы обработки данных**»**  *(наименование дисциплины)* | |
| **Тема «**Создание пользовательского типа данных на основе класса С++ и его использование в приложении.»  *(наименование темы)* | |
| Студент группы ИВБО-02-15  *(учебная группа)* | *Прохоров Андрей Валерьевич* |
| Руководитель курсового проекта  *должность, звание, ученая степень* | *Ст. преподаватель. каф. Скворцова Людмила Анатольевна* |
| Рецензент (*при наличии*)  *должность, звание, ученая степень* | *Фамилия И.О* |
|  |  |

|  |  |  |
| --- | --- | --- |
| Работа представлена к защите | «\_\_»\_\_\_\_\_\_\_2017 г. | *(подпись студента)* |
|  |  |  |
| «Допущен к защите» | «\_\_»\_\_\_\_\_\_\_2017 г. | *(подпись руководителя)* |

Москва 2017

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | | | |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  «Московский технологический университет»  МИРЭА | | | | |
| Институт информационных технологий  *(наименование института)* | | | | |
| Математического обеспечения и стандартизации информационных технологий  *(наименование кафедры)* | | | | |
|  | | **Утверждаю** | | |
|  | | Заведующий кафедрой МОСИТ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_*И.О. Фамилия* | | |
|  | | «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_2017\_\_\_ г. | | |
| **ЗАДАНИЕ** | | | | |
| **на выполнение курсового проекта (работы)** | | | | |
| **по** **дисциплине** «Структуры и алгоритмы обработки данных» | | | | |
|  | | | | |
| Студент *Прохоров Андрей Валерьевич* Группа *ИВБО-02-15* | | | | |
| 1. **Тема:** создание пользовательского типа данных на основе класса С++ и его использование в приложении. | | | | |
| 1. **Исходные данные:** класс – система линейных уравнений n-ой степени, должен обеспечивать вычисление значения системы уравнений методом Гаусса. | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
| * 1. **Перечень вопросов, подлежащих разработке, и обязательного графического материала:** | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
| 1. **Срок представления к защите курсового проекта (работы):** **до** «\_\_\_» \_\_\_\_\_\_\_2017 г. | | | | |
|  | | | | |
| Задание на курсовой  проект, (работу) выдал | «\_\_\_»\_\_\_\_\_\_2017 г. | | *Подпись руководителя*  *проекта* | *Скворцова Людмила Анатольевна* |
| Задание на курсовой  проект, (работу) получил | «\_\_\_»\_\_\_\_\_\_2017 г. | | *Подпись студента –*  *исполнителя проекта* | *Прохоров Андрей Валерьевич* |

Москва 2017

**Содержание**

[Реферат 4](#_Toc326102917)

[Введение 4](#_Toc326102918)

[1. Теоретическое введение 4](#_Toc326102919)

[1.1. СЛАУ и методы ее решения 4](#_Toc326102920)

[1.2. Метод Гаусса 5](#_Toc326102921)

[2. Проектная часть 6](#_Toc326102923)

[2.1. Постановка задачи 6](#_Toc326102924)

[2.2. Проектное решение 6](#_Toc326102925)

[2.2.1. Структуры данных 6](#_Toc326102926)

[2.2.2. Интерфейс программы 6](#_Toc326102927)

[3. Экспериментальная часть 7](#_Toc326102930)

[3.1. Подход к тестированию 7](#_Toc326102931)

[3.2. Протокол тестирования 8](#_Toc326102932)

[3.3. Проверка 10](#_Toc326102932)

[4. Список используемых информационных источников 11](#_Toc326102933)

5. [Приложение 1. Исходный код программы с комментариями 12](#_Toc326102934)

**Реферат**

Курсовая работа содержит 14 страниц, 7 рисунков, 3 раздела и 1 приложение.

В разделе «Теоретическое введение» описаны СЛАУ и их решение методом Гаусса.

Раздел «Проектная часть» содержит описание интерфейса и реализации созданной программы.

«Экспериментальная часть» излагает результаты тестирования программы и заложенные в нее функции.

В приложении 1 приведен исходный код всех модулей программы.

**Введение**

Данная работа посвящена решению системы линейных уравнений, реализованной посредствам пользовательского класса С++. А результате выполнения работы написана и протестирована программа, создающая матрицу коэффициентов системы уравнений и решающее ее методом Гаусса. Программа использует консольный интерфейс Windows.

**1. Теоретическое введение**

1.1. СЛАУ и методы ее решения

Система линейных алгебраических уравнений (*линейная система*, также употребляются аббревиатуры *СЛАУ*, *СЛУ*) — система уравнений, каждое уравнение в которой является линейным — алгебраическим уравнением первой степени (см. рис. №1).

В классическом варианте коэффициенты при переменных, свободные члены и неизвестные считаются вещественными числами, но все методы и результаты сохраняются (либо естественным образом обобщаются) на случай любых полей, например, комплексных чисел.

Решение систем линейных алгебраических уравнений — одна из классических задач линейной алгебры, во многом определившая её объекты и методы. Кроме того, линейные алгебраические уравнения и методы их решения играют важную роль во многих прикладных направлениях, в том числе в линейном программировании, эконометрике.

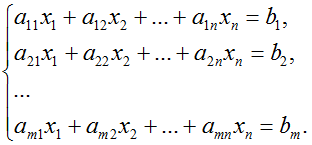


Рис. №1—общий вид СЛАУ

Система линейных алгебраических уравнений может быть представлена в матричной форме (см. рис. №2).

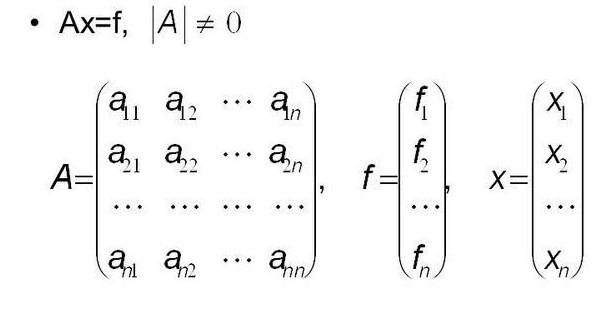


Рис. №2—матричное представление СЛАУ

На этом представлении и основаны основные методы решения систем уравнений. В этой работе для решения СЛАУ был выбран метод Гаусса.

1.2. Метод Гаусса

Метод Гаусса — классический метод решения системы линейных алгебраических уравнений (СЛАУ). Назван в честь немецкого математика Карла Фридриха Гаусса. Это метод последовательного исключения переменных, когда с помощью элементарных преобразований система уравнений приводится к равносильной системе треугольного вида (см. рис. №3), из которой последовательно, начиная с последних (по номеру), находятся все переменные системы.

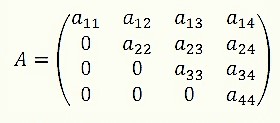


Рис. №3—верхний треугольный вид матрицы СЛАУ

**2. Проектная часть**

2.1. Постановка задачи

Программа должна осуществлять чтение коэффициентов уравнений, введенных с клавиатуры пользователем, создавать расширенную матрицу системы. Далее, сведением ее к треугольной матрице, находить значения неизвестных. Программа должна иметь консольный или оконный интерфейс Windows. Для облегчения импорта кода, содержимое программы должно быть представлено в виде пользовательского шаблонного класса и содержать конструктор.

2.2. Проектное решение

2.2.1. Структуры данных

Запись данных в приложении происходит в двумерный массив типа float—числа с плавающей запятой. Массив включает в себя матрицу коэффициентов при неизвестных Х и крайний правый столбец свободных членов.

2.2.2. Интерфейс программы

Программа использует консольный интерфейс Windows в связи с простотой и малым количеством вводимых данных. Его вид показан на рисунке №4.

Пользователь должен указать количество уравнений в системе и их старшую степень. В приложении реализована проверка ввода. Если количество уравнений и их степень являются целочисленными, то пользователю предлагается заполнить созданную расширенную матрицу. Работа проверки ввода представлена на рисунке №4.

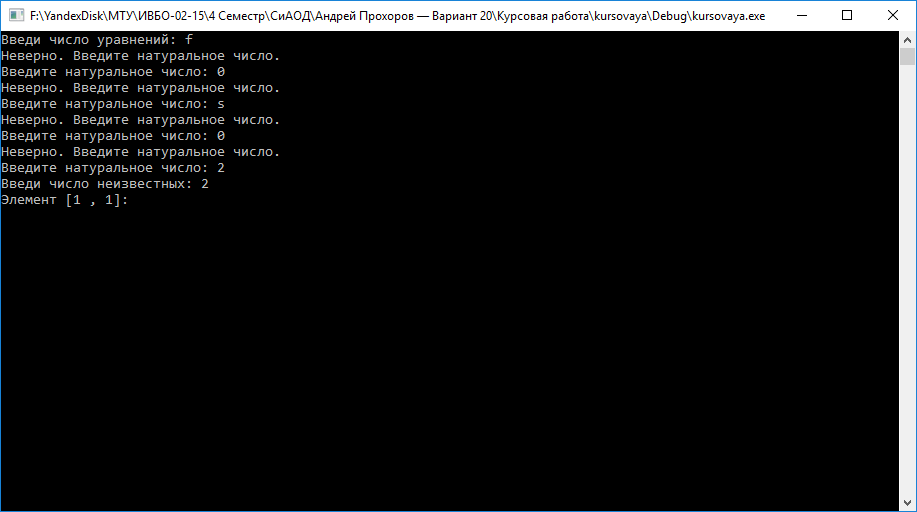


Рис. №4—проверка ввода.

После ее заполнения программа производит вычисления, необходимые для нахождения ответа и выводит его.

**3. Экспериментальная часть**

3.1. Подход к тестированию

Поставим задачей тестирования проверку того, как программа находит неизвестные и осуществляется ли проверка ввода данных с клавиатуры. Для этого создадим программу и введем данные, позволяющие найти действительное решение СЛАУ.

3.2. Протокол тестирования

Тест №1—проверка ввода

Запустим программу и введем данные, не удовлетворяющие требованиям программы. (см. рис. №5)

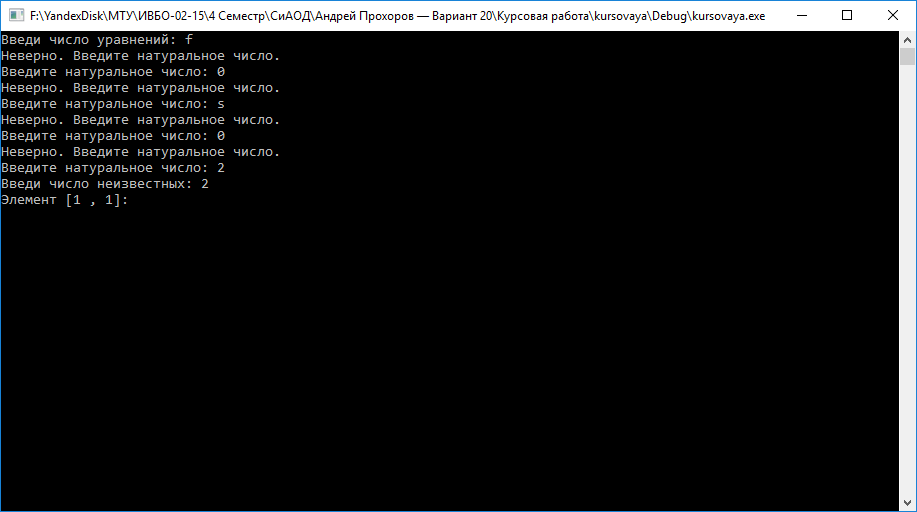
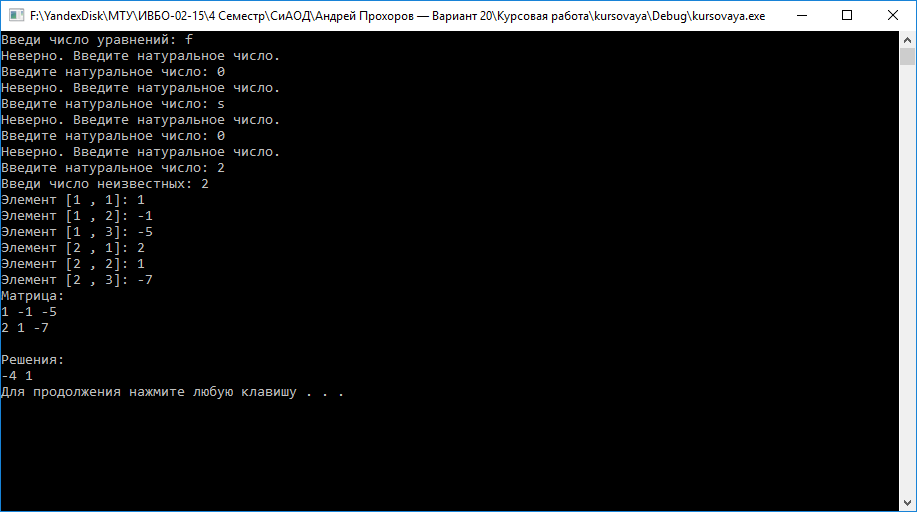


Рис. №5—запуск программы с последующим некорректным вводом

Тест №2—проверка вычислений

Теперь введем корректное количество уравнений и их старшую степень, заполним расширенную матрицу значениями, дающими действительные ответы. Результат показан на рисунке №6.

Рис. №6—вычисление корней системы уравнений.

3.3. Проверка

Сравним результаты расчетов программы с онлайн калькулятором, также использующим метод Гаусса. Проверка выполнялась на интернет-сайте https://math.semestr.ru/gauss/gauss.php (см. рис. №7).

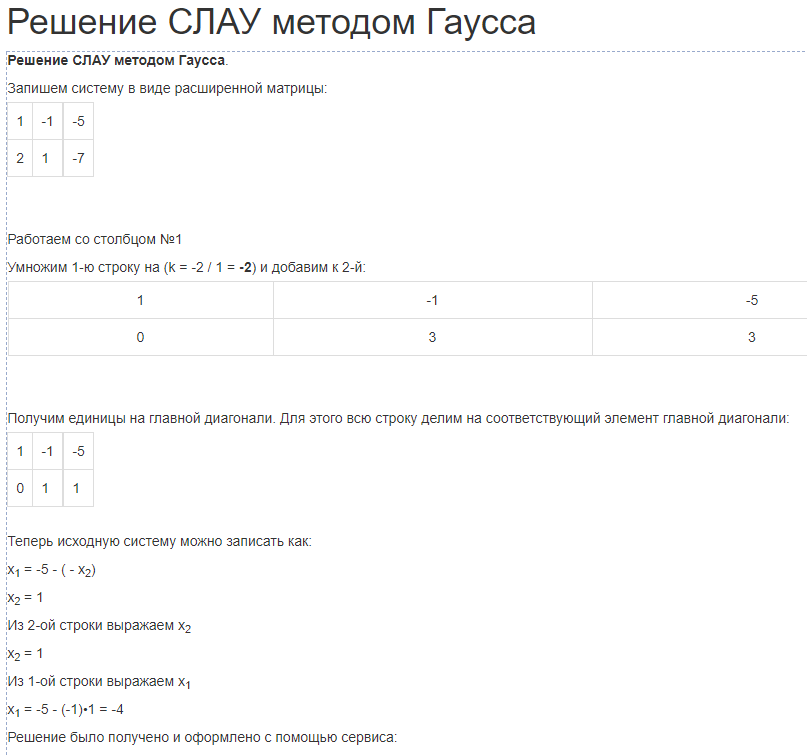


Рис. №7—проверка результатов работы программы

**4. Список используемых информационных источников**

1. Павловская Т.А. С/С++. Программирование на языке высокого уровня- Санкт-Петербург. Питер, 2006.

2. Интернет-ресурс: http://cppstudio.com/post/439/

3. Интернет-ресурс: http://cppstudio.com/post/5188/

4. Интернет-ресурс: http://mathprofi.ru/metod\_gaussa\_dlya\_chainikov.html

5. Интернет ресурс: https://math.semestr.ru/gauss/gauss.php

6. Скворцова Л.А. «Системы и алгоритмы обработки данных», лекции, МТУ МИРЭА, 2017.

7. Сыромятников В.П. «Программирование», лекции, МТУ МИРЭА, 2016/2017.

**5. Приложение**

5.1. Исходный код программы с комментариями

#include<iostream>

#include <iomanip>

#include <math.h>

#include <cstdlib>

using namespace std;

template <typename T>

class Gauss

{

private:

int equationNumber;

int equationRate;

public:

Gauss(); //конструктор

void setMatrix(); //определение размера матрицы

void calculate(); //инициализация матрицы и решение

};

//конструктор

template <typename T>

Gauss<T>::Gauss() {

setMatrix();

}

//определение размера матрицы

template <typename T>

void Gauss<T>::setMatrix() {

int n;

int m;

//создаем массив

cout << "Введи число уравнений: ";

cin >> n;

if (n<1 || cin.fail())

cout << "Неверно. Введите натуральное число." << endl;

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail());

while (n<1) {

cout << "Введите натуральное число: ";

cin >> n;

if (n<1 || cin.fail())

cout << "Неверно. Введите натуральное число." << endl;

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail());

}

equationNumber = n; //инициализируем количество уравнений

cout << "Введи число неизвестных: ";

cin >> m;

if (m<1 || cin.fail())

cout << "Неверно. Введите натуральное число." << endl;

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail());

while (m<1) {

cout << "Введите натуральное число: ";

cin >> m;

if (m<1 || cin.fail())

cout << "Неверно. Введите натуральное число." << endl;

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail());

}

m += 1;

equationRate = m; //инициализируем старшую степень уравнений

}

//инициализация матрицы и решение

template <typename T>

void Gauss<T>::calculate() {

float \*\*matrix = new float \*[equationNumber];

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

matrix[i] = new float[equationRate];

//инициализируем

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

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

{

cout << "Элемент " << "[" << i + 1 << " , " << j + 1 << "]: ";

cin >> matrix[i][j];

}

//выводим массив

cout << "Матрица: " << endl;

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

{

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

cout << matrix[i][j] << " ";

cout << endl;

}

cout << endl;

//Прямой ход, приведение к верхнетреугольному виду

float temp;

float \*xTable = new float[equationRate];

int k;

for (int i = 0; i<equationNumber; i++) {

temp = matrix[i][i];

for (int j = equationNumber; j >= i; j--)

matrix[i][j] /= temp;

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

{

temp = matrix[j][i];

for (int k = equationNumber; k >= i; k--)

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

}

}

xTable[equationNumber - 1] = matrix[equationNumber - 1][equationNumber];

for (int i = equationNumber - 2; i >= 0; i--) {

xTable[i] = matrix[i][equationNumber];

for (int j = i + 1; j < equationNumber; j++) {

xTable[i] -= matrix[i][j] \* xTable[j];

}

}

//Выводим решения

cout << "Решения: " << endl;

for (int i = 0; i < equationNumber; i++) {

cout << xTable[i] << " ";

}

cout << endl;

}

int main()

{

setlocale(LC\_ALL, "Russian");

Gauss <float> myGauss;

myGauss.calculate();

system("Pause");

return 0;

}