МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №1

по дисциплине: «Вычислительная математика»

тема: Метод Гаусса

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверил:

Бондаренко Татьяна Владимировна

Белгород 2023 г.

Цель работы: изучить прямой и обратный ход метода Гаусса для решения систем линейных алгебраических уравнений; изучить схему единственного деления с выбором максимального по модулю элемента; изучить применение метода Гаусса для вычисления определителя матрицы и обратной матрицы; получить практические навыки программной реализации метода Гаусса и решения поставленных задач методом Гаусса с помощью ЭВМ.

Задания:

1. Выполнить вручную действия над матрицами А и В из пункта 3 задания соответствующего варианта.

2. Выполнить следующие действия, **не используя** метод Гаусса: − решить вручную **систему** линейных алгебраических уравнений с тремя неизвестными из **пункта 1** задания соответствующего варианта; − найти вручную **определитель** матрицы А из **пункта 2** задания соответствующего варианта; − найти вручную матрицу **A^{-1}** обратную матрице А из **пункта 2** задания соответствующего варианта.

3. Решить вручную методом Гаусса систему линейных алгебраических уравнений с тремя неизвестными из пункта 1 задания соответствующего варианта.

4. Найти вручную с помощью метода Гаусса определитель матрицы А из пункта 2 задания соответствующего варианта.

5. Найти вручную с помощью метода Гаусса матрицу A^{-1} обратную матрице А из пункта 2 задания соответствующего варианта. Выполнить проверку полученной матрицы на соответствие условию: $A \cdot A^{-1} = E$, где Е — единичная матрица.

6. Создать модуль для работы с матрицами произвольного порядка, содержащий подпрограммы для умножения двух матриц, умножения числа на матрицу, сложения матриц, вычитания матриц, транспонирования матрицы, умножения матрицы на вектор, ввода и вывода матрицы.

7. Создать модуль, содержащий подпрограммы, реализующие прямой и обратный ход метода Гаусса для схемы единственного деления с выбором максимального по модулю элемента.

8. Создать программу для решения следующих задач:

− нахождение методом Гаусса решения системы линейных алгебраических уравнений с одной и той же матрицей коэффициентов при неизвестных и произвольным числом столбцах свободных членов. Замечание. В прямом ходе метода Гаусса выполняется приведение расширенной матрицы (коэффициенты при неизвестных и свободные члены) к треугольному виду, и одновременно изменяются все столбцы свободных членов. На этапе обратного хода выполняется вычисление решения системы для каждого столбца свободных членов, составляется матрица решений.

− вычисление определителя заданной матрицы методом Гаусса;

− нахождение для заданной матрицы обратной матрицы методом Гаусса.

9. Решить все задания соответствующего варианта с помощью составленной программы. Выполнить проверку правильности найденного решения системы линейных уравнений и матрицы обратной к заданной матрице с помощью составленной программы, сравнить значения, полученные при решении заданий с помощью метода Гаусса и с использованием произвольного метода решения.

Содержание отчета

1. Титульный лист.

2. Цель работы. Вариант задания.

3. Текст задания к работе.

4. Выполнение задания соответствующего варианта вручную полностью. Все действия выполняемые при решении задачи вручную расписывать подробно с указанием всех промежуточных операций. Указание только окончательного ответа не допускается.

5. Текст программы, включающий необходимые комментарии и спецификации подпрограмм.

6. Результаты работы программы.

Выполнение работы. Вариант 23:

23	$\begin{cases} 3x_1 - 2x_2 + 2x_3 = 10, \\ 2x_1 + x_2 - 4x_3 = -7, \\ 2x_1 + 2x_2 - 6x_3 = -12. \end{cases}$	$A = \begin{pmatrix} 3 & 5 & 1 \\ 1 & 0 & 2 \\ 2 & 2 & -1 \end{pmatrix}$	$\boldsymbol{(B^3 \cdot A^2) + A}$ $A = \begin{pmatrix} 1 & 3 & -2 \\ 6 & 5 & 2 \\ 7 & 0 & 4 \end{pmatrix}, B = \begin{pmatrix} 4 & 3 & 5 \\ 7 & 0 & 2 \\ 1 & 5 & 2 \end{pmatrix}$

Задание 1. Выполнить вручную действия над матрицами А и В из пункта 3 задания соответствующего варианта.

$$A^2 = A \cdot A = \begin{pmatrix} 1 & 3 & -2 \\ 6 & 5 & 2 \\ 7 & 0 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 3 & -2 \\ 6 & 5 & 2 \\ 7 & 0 & 4 \end{pmatrix} = \begin{pmatrix} 5 & 18 & -4 \\ 50 & 43 & 6 \\ 35 & 21 & 2 \end{pmatrix}$$

$$B^2 = B \cdot B = \begin{pmatrix} 4 & 3 & 5 \\ 7 & 0 & 2 \\ 1 & 5 & 2 \end{pmatrix} \cdot \begin{pmatrix} 4 & 3 & 5 \\ 7 & 0 & 2 \\ 1 & 5 & 2 \end{pmatrix} = \begin{pmatrix} 42 & 37 & 36 \\ 30 & 31 & 39 \\ 41 & 13 & 19 \end{pmatrix}$$

$$B^3 = B \cdot B^2 = \begin{pmatrix} 4 & 3 & 5 \\ 7 & 0 & 2 \\ 1 & 5 & 2 \end{pmatrix} \cdot \begin{pmatrix} 42 & 37 & 36 \\ 30 & 31 & 39 \\ 41 & 13 & 19 \end{pmatrix} = \begin{pmatrix} 463 & 306 & 356 \\ 376 & 285 & 290 \\ 274 & 218 & 269 \end{pmatrix}$$

$$A^2 \cdot B^3 = \begin{pmatrix} 5 & 18 & -4 \\ 50 & 43 & 6 \\ 35 & 21 & 2 \end{pmatrix} \cdot \begin{pmatrix} 463 & 306 & 356 \\ 376 & 285 & 290 \\ 274 & 218 & 269 \end{pmatrix} = \begin{pmatrix} 7987 & 5788 & 5924 \\ 40962 & 28863 & 31884 \\ 24649 & 17131 & 19088 \end{pmatrix}$$

$$A^2 \cdot B^3 + A = \begin{pmatrix} 7987 & 5788 & 5924 \\ 40962 & 28863 & 31884 \\ 24649 & 17131 & 19088 \end{pmatrix} + \begin{pmatrix} 1 & 3 & -2 \\ 6 & 5 & 2 \\ 7 & 0 & 4 \end{pmatrix} = \begin{pmatrix} 7988 & 5791 & 5922 \\ 40968 & 28668 & 31886 \\ 24656 & 17131 & 19092 \end{pmatrix}$$

Задание 2. Выполнить следующие действия, **не используя** метод Гаусса: − решить вручную **систему** линейных алгебраических уравнений с тремя неизвестными из **пункта 1** задания соответствующего варианта; − найти вручную **определитель** матрицы А из **пункта 2** задания соответствующего варианта; − найти вручную матрицу $\mathbf{A}^{-1}$ обратную матрице А из **пункта 2** задания соответствующего варианта.

Решение системы из пункта 1:

$$\begin{cases} 3x_1-2x_2+2x_3=10 \\ 2x_1+x_2-4x_3=-7 \\ 2x_1+2x_2-6x_3=-12 \end{cases}; \quad \begin{cases} x_1=\dfrac{-12-2x_2+6x_3}{2} \\ 2x_1+x_2-4x_3=-7 \\ 2x_1+2x_2-6x_3=-12 \end{cases};$$

$$\begin{cases} x_1=\dfrac{-12-2x_2+6x_3}{2} \\ 2\left(\dfrac{10+2x_2-2x_3}{3}\right)+x_2-4x_3=-7 \\ 2x_1+2x_2-6x_3=-12 \end{cases}; \quad \begin{cases} x_1=\dfrac{-12-2x_2+6x_3}{2} \\ x_2=\dfrac{5-2x_3}{-1} \\ 2x_1+2x_2-6x_3=-12 \end{cases};$$

$$\begin{cases} x_1=\dfrac{-12-2x_2+6x_3}{2} \\ x_2=\dfrac{5-2x_3}{-1} \\ 2x_1+2x_2-6x_3=-12 \end{cases}; \quad \begin{cases} x_1=\dfrac{-12-2x_2+6x_3}{2} \\ x_2=\dfrac{5-2x_3}{-1} \\ 2x_1+2x_2-6x_3=-12 \end{cases};$$

$$\begin{cases} x_1=\dfrac{-12-2x_2+6x_3}{2} \\ x_2=\dfrac{5-2x_3}{-1} \\ x_3=3 \end{cases}; \quad \begin{cases} x_1=2 \\ x_2=1 \\ x_3=3 \end{cases}$$

Определитель матрицы А из пункта 2:

$$\Delta A=\begin{pmatrix} 3 & 5 & 1 \\ 1 & 0 & 2 \\ 2 & 2 & -1 \end{pmatrix}=3\cdot0\cdot(-1)+5\cdot2\cdot2+1\cdot1\cdot2-1\cdot0\cdot2-3\cdot2\cdot2-5\cdot1\cdot(-1)=0+20+2-0-12+5=15$$

Обратная матрица для матрицы A:

$$A = \begin{pmatrix} 3 & 5 & 1 \\ 1 & 0 & 2 \\ 2 & 2 & -1 \end{pmatrix}$$

$$A^{*} = \begin{pmatrix} -4 & -5 & 0 \\ -7 & -5 & -4 \\ 9 & 5 & -5 \end{pmatrix} \qquad A^{*T} = \begin{pmatrix} -4 & -7 & 9 \\ -5 & -5 & 5 \\ 0 & -4 & -5 \end{pmatrix}$$

$$\frac{A^{*T}}{|A|} = \frac{1}{15}\begin{pmatrix} -4 & -7 & 9 \\ -5 & -5 & 5 \\ 0 & -4 & -5 \end{pmatrix} = \begin{pmatrix} -\frac{4}{15} & \frac{7}{15} & \frac{2}{3} \\ \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\ \frac{2}{15} & \frac{4}{15} & -\frac{1}{3} \end{pmatrix}$$

$$A \cdot A^{-1} = \begin{pmatrix} 3 & 5 & 1 \\ 1 & 0 & 2 \\ 2 & 2 & -1 \end{pmatrix} \begin{pmatrix} -\frac{4}{15} & \frac{7}{15} & \frac{2}{3} \\ \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\ \frac{2}{15} & \frac{4}{15} & -\frac{1}{3} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Задание 3. Решить вручную методом Гаусса систему линейных алгебраических уравнений с тремя неизвестными из пункта 1 задания соответствующего варианта.

$$\left(\begin{array}{ccc|c} 3 & -2 & 2 & 10 \\ 2 & 1 & -4 & -7 \\ 2 & 2 & -6 & -12 \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & -\frac{2}{3} & \frac{2}{3} & \frac{10}{3} \\ 2 & 1 & -4 & -7 \\ 2 & 2 & -6 & -12 \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & -\frac{2}{3} & \frac{2}{3} & \frac{10}{3} \\ 0 & \frac{7}{3} & -\frac{16}{3} & -\frac{41}{3} \\ 0 & \frac{10}{3} & -\frac{22}{3} & -\frac{56}{3} \end{array}\right) \sim$$

$$\sim \left(\begin{array}{ccc|c} 1 & -\frac{2}{3} & \frac{2}{3} & \frac{10}{3} \\ 0 & 1 & -\frac{16}{7} & -\frac{41}{7} \\ 0 & \frac{10}{3} & -\frac{22}{3} & -\frac{56}{3} \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & 0 & -\frac{6}{7} & -\frac{4}{7} \\ 0 & 1 & \frac{16}{7} & -\frac{41}{7} \\ 0 & 0 & \frac{2}{7} & \frac{6}{7} \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & 0 & -\frac{6}{7} & -\frac{4}{7} \\ 0 & 1 & \frac{16}{7} & -\frac{41}{7} \\ 0 & 0 & 1 & 3 \end{array}\right) \sim$$

$$\sim \left(\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 3 \end{array}\right) \Rightarrow \begin{cases} x_1 = 2 \\ x_2 = 1 \\ x_3 = 3 \end{cases}$$

Задание 4. Найти вручную с помощью метода Гаусса определитель матрицы A из пункта 2 задания соответствующего варианта.

$$A=\begin{pmatrix}3&5&1\\1&0&2\\2&2&-1\end{pmatrix}\sim\begin{pmatrix}3&5&1\\1&0&2\\0&2&-5\end{pmatrix}\sim\begin{pmatrix}3&5&1\\0&-\frac{5}{3}&\frac{5}{3}\\0&0&-3\end{pmatrix}\Rightarrow \Delta A=-3\cdot\left(-\frac{5}{3}\right)\cdot 3=15$$

Задание 5. Найти вручную с помощью метода Гаусса матрицу A^{-1} обратную матрице A из пункта 2 задания соответствующего варианта. Выполнить проверку полученной матрицы на соответствие условию: $A\cdot A^{-1} = E$, где E — единичная матрица.

$$A^{-1}=\left(\begin{array}{ccc|ccc}3&5&1&1&0&0\\1&0&2&0&1&0\\2&2&-1&0&0&1\end{array}\right)\sim\left(\begin{array}{ccc|ccc}1&1\frac{2}{3}&\frac{1}{3}&\frac{1}{3}&0&0\\1&0&2&0&1&0\\2&2&-1&0&0&1\end{array}\right)\sim\left(\begin{array}{ccc|ccc}1&1\frac{2}{3}&\frac{1}{3}&\frac{1}{3}&0&0\\0&-1\frac{2}{3}&1\frac{2}{3}&-\frac{1}{3}&1&0\\0&-1\frac{1}{3}&-1\frac{2}{3}&-\frac{2}{3}&0&1\end{array}\right)\sim$$

$$\sim\left(\begin{array}{ccc|ccc}1&1\frac{2}{3}&\frac{1}{3}&\frac{1}{3}&0&0\\0&1&-1&0{,}2&-0{,}6&0\\0&-1\frac{1}{3}&-1\frac{2}{3}&-\frac{2}{3}&0&1\end{array}\right)\sim\left(\begin{array}{ccc|ccc}1&0&2&0&1&0\\0&1&-1&0{,}2&-0{,}6&0\\0&0&-3&-0{,}4&-0{,}8&1\end{array}\right)\sim$$

$$\sim\left(\begin{array}{ccc|ccc}1&0&2&0&1&0\\0&1&-1&0{,}2&-0{,}6&0\\0&0&1&\frac{2}{15}&\frac{4}{15}&-\frac{1}{3}\end{array}\right)\sim\left(\begin{array}{ccc|ccc}1&0&0&-\frac{4}{15}&\frac{7}{15}&\frac{2}{3}\\0&1&0&\frac{1}{3}&-\frac{1}{3}&-\frac{1}{3}\\0&0&1&\frac{2}{15}&\frac{4}{15}&-\frac{1}{3}\end{array}\right)$$

$$A^{-1}=\begin{pmatrix}-\frac{4}{15}&\frac{7}{15}&\frac{2}{3}\\\frac{1}{3}&-\frac{1}{3}&-\frac{1}{3}\\\frac{2}{15}&\frac{4}{15}&-\frac{1}{3}\end{pmatrix}$$

Задание 6. Создать модуль для работы с матрицами произвольного порядка, содержащий подпрограммы для умножения двух матриц, умножения числа на матрицу, сложения матриц, вычитания матриц, транспонирования матрицы, умножения матрицы на вектор, ввода и вывода матрицы.

Содержимое заголовочного файла:

```cpp
#ifndef CODE_MATRIX_H
#define CODE_MATRIX_H

#include <iostream>
#include <utility>
#include "vector"

using namespace std;
```

```
using matrixRow = vector<double>;
using matrix = vector<matrixRow>;

class Matrix {
 public:
 matrix data;
 int nRows;
 int nColumns;

 explicit Matrix(int nRows, int nColumns) {
  this->nRows = nRows;
  this->nColumns = nColumns;
  for (int i = 0; i < nColumns; i++) {
   matrixRow r(nRows);
   data.push_back(r);
  }
 }

 explicit Matrix(matrix m) {
  this->nRows = m[0].size();
  this->nColumns = m.size();
  data = std::move(m);
 }

 void Output();
 void Input();

 static Matrix Multiplication(Matrix &m1, Matrix &m2);
 static Matrix Multiplication(Matrix &m, int x);
 static Matrix Multiplication(Matrix &m1, matrixRow &m2);

 static Matrix Addition(Matrix &m1, Matrix &m2);

 static Matrix Difference(Matrix &m1, Matrix &m2);

 static Matrix Transposition(Matrix &m);
 void Transposition();

};

#endif //CODE_MATRIX_H
```

Содержимое исполняемого файла:

```
#include "Matrix.h"

void Matrix::Output() {
 for (const auto &i: data) {
  for (auto j: i)
   cout << j << ' ';
  cout << '\n';
 }
}

void Matrix::Input() {
```

```
  for (int i = 0; i < nRows; i++) {
    for (int j = 0; j < nColumns; j++)
      cin >> this->data[i][j];
  }
}

Matrix Matrix::Multiplication(Matrix &m1, Matrix &m2) {
  if (m1.nColumns != m2.nRows) {
    cerr << "The number of columns of the first matrix is not equal "
            "to the number of rows of the second matrix";
    Matrix res(0, 0);
    return res;
  }

  Matrix res(m1.nRows, m2.nColumns);
  for (int i = 0; i < m1.nRows; i++) {
    for (int j = 0; j < m2.nColumns; j++) {
      res.data[i][j] = 0;
      for (int k = 0; k < m1.nColumns; ++k)
        res.data[i][j] += m1.data[i][k] * m2.data[k][j];
    }
  }
  return res;
}

Matrix Matrix::Multiplication(Matrix &m1, matrixRow &m2) {
  if (m1.nColumns != m2.size()) {
    cerr << "The number of columns of the first matrix is not equal "
            "to the number of rows of the second matrix";
    Matrix res(0, 0);
    return res;
  }

  Matrix res(m1.nRows, m1.nColumns);
  for (int i = 0; i < m1.nRows; i++) {
    for (int j = 0; j < m1.nColumns; j++) {
      res.data[i][j] = m1.data[i][j] * m2[j];
    }
  }
  return res;
}

Matrix Matrix::Multiplication(Matrix &m, int x) {
  Matrix res(m.nRows, m.nColumns);
  for (int i = 0; i < m.nRows; i++) {
    for (int j = 0; j < m.nColumns; j++) {
      res.data[i][j] = m.data[i][j] * x;
    }
  }
  return res;
}

Matrix Matrix::Addition(Matrix &m1, Matrix &m2) {
  if (m1.nRows != m2.nRows && m1.nColumns != m2.nColumns) {
    cerr << "Different number of rows or columns for matrices";
    Matrix res(0, 0);
    return res;
  }
```

```
  Matrix res(m1.nRows, m1.nColumns);
  for (int i = 0; i < m1.nColumns; i++)
    for (int j = 0; j < m1.nRows; j++)
      res.data[i][j] = m1.data[i][j] + m2.data[i][j];
  return res;
}

Matrix Matrix::Difference(Matrix &m1, Matrix &m2) {
  if (m1.nRows != m2.nRows && m1.nColumns != m2.nColumns) {
    cerr << "Different number of rows or columns for matrices";
    Matrix res(0, 0);
    return res;
  }

  Matrix res(m1.nRows, m1.nColumns);
  for (int i = 0; i < m1.nColumns; i++)
    for (int j = 0; j < m1.nRows; j++)
      res.data[i][j] = m1.data[i][j] - m2.data[i][j];
  return res;
}

Matrix Matrix::Transposition(Matrix &m) {
  Matrix res(m.nColumns, m.nRows);
  for (int i = 0; i < m.nRows; i++)
    for (int j = 0; j < m.nColumns; j++)
      res.data[j][i] = m.data[i][j];
  return res;
}


void Matrix::Transposition() {
  for (int i = 0; i < this->nRows; i++)
    for (int j = 0; j < this->nColumns; j++)
      this->data [j][i] = this->data[i][j];
}
```

Задание 7. Создать модуль, содержащий подпрограммы, реализующие прямой и обратный ход метода Гаусса для схемы единственного деления с выбором максимального по модулю элемента.

```
void Matrix::forwardGauss() {
  for (int i = 0; i < nRows - 1; i++) {
    for (int j = i + 1; j < nRows; j++)
      if (data[j][i] > data[i][i]) {
        swap(data[j], data[i]);
        nSwap++;
      }

    for (int j = i + 1; j < nRows; j++)
      if (abs(data[i][i]) >= EPS) {
        double dif = data[j][i] / data[i][i];
        for (int k = i; k < nColumns; k++)
```

```
      data[j][k] -= data[i][k] * dif;
    }
  }
}

void Matrix::reverseGauss() {
  for (int i = nRows - 1; i >= 0; i--) {
    if (abs(data[i][i]) >= EPS) {
      double dif = data[i][i];
      for (int j = i; j < nColumns; j++)
        data[i][j] /= dif;
    }

    for (int j = i - 1; j >= 0; j--) {
      data[j][nColumns - 1] -= data[j][i] * data[i][nColumns - 1];
      data[j][i] = 0;
    }
  }
}
```

Задание 8. Создать программу для решения следующих задач:

– нахождение методом Гаусса решения системы линейных алгебраических уравнений с одной и той же матрицей коэффициентов при неизвестных и произвольным числом столбцах свободных членов. Замечание. В прямом ходе метода Гаусса выполняется приведение расширенной матрицы (коэффициенты при неизвестных и свободные члены) к треугольному виду, и одновременно изменяются все столбцы свободных членов. На этапе обратного хода выполняется вычисление решения системы для каждого столбца свободных членов, составляется матрица решений.

```
bool Matrix::isFindZeroRow() {
  for (int i = 0; i < nRows; i++) {
    bool allZero = true;
    for (int j = 0; j < nRows; j++)
      if (abs(data[i][j]) >= EPS) {
        allZero = false;
        break;
      }
    if (allZero)
      return true;
  }
  return false;
}

void Matrix::linearEquation() {
  this->forwardGauss();
  this->reverseGauss();
  if (isFindZeroRow()) {
    cerr << "No solution because null string was found";
    return;
  }
```

```
  for (int i = nRows - 1; i >= 0; i--) {
    cout << "x" << nRows - i << " = " << data[i][nColumns - 1] << ";\n";
  }
}
```

– вычисление определителя заданной матрицы методом Гаусса;

```
void Matrix::deleteZeroRows() {
  int n = nRows;
  for (int i = 0; i < n; i++) {
    bool allZero = true;
    for (int j = 0; j < n; j++) {
      if (abs(data[i][j]) > EPS) {
        allZero = false;
        break;
      }
    }
    if (allZero) {
      data[i].clear();
      data.erase(data.begin());
      nRows--;
    }
  }
}

double Matrix::determinant() {
  this->forwardGauss();
  this->deleteZeroRows();
  double det = data[0][0];
  for (int i = 1; i < nRows; i++) {
    det *= data[i][i];
  }
  if (nSwap % 2 == 1)
    det *= -1;
  return det;
}
```

– нахождение для заданной матрицы обратной матрицы методом Гаусса:

```
void Matrix::inverse() {
  Matrix res(this->data);
  Matrix E(nRows, nColumns);
  for (int i = 0; i < nRows; i++)
    E.data[i][i] = 1;
  for (int i = 0; i < nRows; i++)
    for (auto j: E.data[i])
      res.data[i].push_back(j);
  res.nColumns *= 2;

  res.forwardGauss();
  res.reverseGauss();
  for (int i = 0; i < nRows; i++)
    for (int j = 0; j < nColumns; j++)
      data[i][j] = res.data[i][j + nColumns];
}
```

Задание 9. Решить все задания соответствующего варианта с помощью составленной программы. Выполнить проверку правильности найденного решения системы линейных уравнений и матрицы обратной к заданной матрице с помощью составленной программы, сравнить значения, полученные при решении заданий с помощью метода Гаусса и с использованием произвольного метода решения.

Результат работы программы при нахождении линейного уравнения:

```cpp
#include <iostream>
#include "Matrix/Matrix.h"

int main() {
    //matrix m = {{1, 1,2,3, 5}, {1, 4, 5, 6, 6}, {2, 7, 8, 9, 7}, {2, 4, 1, 4, 4}};
    matrix m = {{3, -2, 2, 10}, {2, 1, -4, -7}, {2, 2, -6, -12}};
    //matrix m = {{3, 5, 1}, {1, 0, 2}, {2, 2, -1}};

    Matrix M1( m: m);
    M1.linearEquation();
    M1.Output();

    return 0;
}
```

main

Code

```
D:\BGTU\VicMat\Lab1\Code\cmake-build-debug\Code.exe
x1 = 3;
x2 = 1;
x3 = 2;
1,      0,      0,      2,
0,      1,      0,      1,
0,      0,      1,      3,


Process finished with exit code 0
```

Результат работы программы при нахождении определителя:

```
#include <iostream>
#include "Matrix/Matrix.h"

int main() {
  //matrix m = {{1, 1,2,3, 5}, {1, 4, 5, 6, 6}, {2, 7, 8, 9, 7}, {2, 4, 1, 4, 4}};
  //matrix m = {{3, -2, 2, 10}, {2, 1, -4, -7}, {2, 2, -6, -12}};
  matrix m = {{3, 5, 1}, {1, 0, 2}, {2, 2, -1}};

  Matrix M1( m: m);
  cout << M1.determinant();

  return 0;
}
```

```
main
Code
D:\BGTU\VicMat\Lab1\Code\cmake-build-debug\Code.exe
15
Process finished with exit code 0
```

Результат работы программы при нахождении обратной матрицы:

```
#include <iostream>
#include "Matrix/Matrix.h"

int main() {
  //matrix m = {{1, 1,2,3, 5}, {1, 4, 5, 6, 6}, {2, 7, 8, 9, 7}, {2, 4, 1, 4, 4}};
  //matrix m = {{3, -2, 2, 10}, {2, 1, -4, -7}, {2, 2, -6, -12}};
  matrix m = {{3, 5, 1}, {1, 0, 2}, {2, 2, -1}};

  Matrix M1( m: m);
  M1.inverse();
  M1.Output();

  return 0;
}
```

```
main
Code
D:\BGTU\VicMat\Lab1\Code\cmake-build-debug\Code.exe
-0.267, 0.467,  0.667,
0.333,  -0.333, -0.333,
0.133,  0.267,  -0.333,


Process finished with exit code 0
```

Проверка обратной матрицы (Некоторые значения получились неравные нулю, но очень близкие к нему. Связано это с тем, что используется тип данных double):

```cpp
#include <iostream>
#include "Matrix/Matrix.h"

int main() {
    //matrix m = {{1, 1,2,3, 5}, {1, 4, 5, 6, 6}, {2, 7, 8, 9, 7}, {2, 4, 1, 4, 4}};
    //matrix m = {{3, -2, 2, 10}, {2, 1, -4, -7}, {2, 2, -6, -12}};
    matrix m = {{3, 5, 1}, {1, 0, 2}, {2, 2, -1}};

    Matrix M1( m: m);
    Matrix M2( m: m);
    M1.inverse();

    Matrix res = Matrix::Multiplication( &: M1,  &: M2);
    res.Output();
```

main

Code

```
D:\BGTU\VicMat\Lab1\Code\cmake-build-debug\Code.exe
1,       4.44e-16,       1.11e-16,
-1.11e-16,       1,       -5.55e-17,
0,       2.22e-16,       1,


Process finished with exit code 0
```

Вывод: изучили прямой и обратный ход метода Гаусса для решения систем линейных алгебраических уравнений; изучили схему единственного деления с выбором максимального по модулю элемента; изучили применение метода Гаусса для вычисления определителя матрицы и обратной матрицы; получили практические навыки программной реализации метода Гаусса и решения поставленных задач методом Гаусса с помощью ЭВМ.