

Национальный исследовательский университет ИТМО

Кафедра программных систем

**Лабораторная работа 2**

Выполнил: Орел

Даниил Максимович

Группа № K3221

Проверил: Иванов С.Е.

Санкт-Петербург

2020

**Цель работы:**

* С помощью классов в методе нахождения обратной матрицы (задание 1) реализовать обработку исключения в случае операций деления на ноль и неособенной невырожденной, не сингулярной квадратной матрицы.

**Ход работы:**

1. В программу был добавлен класс исключения NonSquareMatrixException, предназначенный для обработки ситуации, когда программа пытается найти обратную матрицы для неквадратной матрицы:

class NonSquareMatrixException : Exception

{

public int Rows { get; set; }

public int Columns { get; set; }

public NonSquareMatrixException(string message, int rows, int columns) : base(message)

{

Rows = rows;

Columns = columns;

}

public override string ToString()

{

string message = $"{Message}\nКоличество строк: {Rows}\nКоличество столбцов: {Columns}\n";

return message;

}

}

1. Также был добавлен класс исключения DeterminantZeroException, срабатывающий в случае, когда определитель матрицы равен нулю:

class DeterminantZeroException : Exception

{

public DeterminantZeroException(string message) : base(message) { }

}

1. В классе Matrix был изменен метод CalculateInverseMatrix:

static public void CalculateInverseMatrix(int[,] matrix)

{

int rows = matrix.GetUpperBound(0) + 1;

int columns = matrix.GetUpperBound(1) + 1;

if (!isSquare(rows, columns))

{

throw new NonSquareMatrixException("Матрица не квадртная!", rows, columns);

}

int k = 1;

int[,] minor;

double[,] algebraic\_complement\_matrix;

int determinant = CalculateDeterminant(matrix, rows);

Console.WriteLine("Определитель = " + determinant);

if (determinant == 0)

{

throw new DeterminantZeroException("Определитель матрицы равен нулю. Матрица вырожденная и не имеет определителя!");

}

minor = new int[rows - 1, rows - 1];

matrix = FindTransparentMatrix(matrix);

algebraic\_complement\_matrix = new double[rows, rows];

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

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

{

algebraic\_complement\_matrix[i, j] = k \* CalculateDeterminant(GetMatr(matrix, minor, i, j, rows), rows - 1);

k = -k;

}

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

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

{

algebraic\_complement\_matrix[i, j] /= (double)determinant;

}

Console.WriteLine("Обратная матрица: ");

PrintMatrix(algebraic\_complement\_matrix);

}

1. Добавление обработчиков исключений в класс Program:

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Матрица A: ");

Console.Write("Введите количество строк в матрице: ");

int rows\_array1 = int.Parse(Console.ReadLine());

Console.Write("Введите количество столбцов в матрице: ");

int columns\_array1 = int.Parse(Console.ReadLine());

int[,] array1 = Matrix.CreateMatrix(rows\_array1, columns\_array1);

//Обработка исключений

try

{

Matrix.CalculateInverseMatrix(array1);

}

catch(DeterminantZeroException e)

{

Console.WriteLine(e.Message);

}

catch(NonSquareMatrixException e)

{

Console.WriteLine(e.ToString());

}

Console.ReadKey();

}

}

1. Результат работы программы:

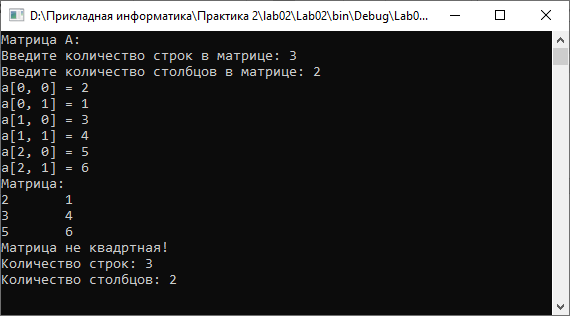


Рисунок 1 – Результат обработки исключения NonSquareMatrixException

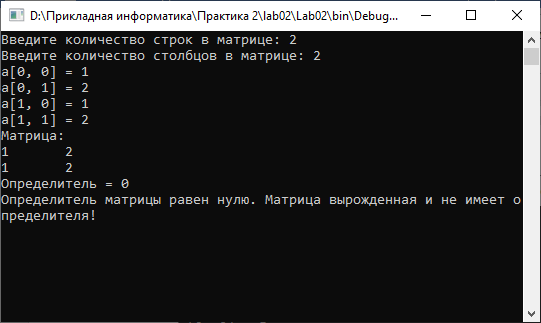


Рисунок 2 – Результат обработки исключения DeterminantZeroException

**Вывод:**

В ходе выполнения данной практической работы была созданы классы исключений DeterminantZeroException и NonSquareMatrixException, позволяющие обработать операцию деления на ноль и попытку поиска обратной матрицы для неквадратной матрицы.