Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно-графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Алгоритмизация и программирование*** |
|  |  |
| на тему | Разработка программы |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020-РГР-02.03.03-№ 4-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Скиба Данила Сергеевича | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | МО-**231** | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | 02.03.03 | | |
|  | | | Математическое обеспечение и администрирование информационных систем | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | | 02.01.2024 | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | | | |
|  |  |  |  | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

Омск 2023

Содержание

[1. ВВЕДЕНИЕ 3](#_Toc147832292)

[2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc147832293)

[3. РАЗРАБОТКА КОДА 6](#_Toc147832294)

[4. ПРИМЕРЫ РАБОТЫ](#_Toc147832295) 8

[5. ЗАКЛЮЧЕНИЕ 9](#_Toc147832296)

6. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ [10](#_Toc147832296)

**ВВЕДЕНИЕ**

В данной расчётно-графической работе показан процесс реализации алгоритма вычисления определителя матрицы. В ходе работы будут представлены необходимая теоретическая часть для построения алгоритма, код алгоритма и примеры его работы.

**ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

Любой квадратной матрице n-ого порядка

можно поставить в соответствие выражение, которое называется ***определителем*** (детерминантом) матрицы , и обозначается так:

или или .

Определитель **2-го** порядка задаётся равенством:

(1.0)

Таким образом, определитель 2-го порядка есть сумма 2-х слагаемых, каждое из которых представляет собой произведение 2-х сомножителей – элементов матрицы , по одному из каждой строки и каждого столбца. Одно из слагаемых берётся со знаком «+», другое – со знаком «-».

Определитель **3-го** порядка раскладывается с помощью определителей 2-го порядка. Разложение определителя 3-го порядка по первой строке:

.

При таком способе вычисления определителя каждый из трёх элементов первой строки умножается на определитель 2-го порядка, составленный из элементов матрицы , оставшиеся после вычёркивания 1-й строки и *j*-го столбца. При этом слагаемое с множителем умножается на число .

Аналогично раскладываются любые определители n-го порядка по первой строке:

(1.1)

Определителем 1-го порядка квадратной матрицы называется значение (1.2)

***Дополнительным минором*** к элементу квадратной матрицы называется минор, составленный из элементов , оставшихся после вычёркивания *i*-й строки и *j*-го столбца.

Например, в матрице минором является определитель, составленный из элементов матрицы, оставшихся после вычёркивания 2-ой строки и 1-го столбца.

**РАЗРАБОТКА КОДА**

Для написания алгоритма вычисления определителей n-го порядка будем использовать язык программирования C#. В ходе реализации алгоритма для описания действий над матрицами и определителями будем использовать двумерные массивы. В коде реализованы 4 основных метода, в двух из которых “Minor” и “Det” описан основной алгоритм вычисления определителя. Метод “Det” определяет размерность “size” матрицы и в зависимости от неё выбирает способ подсчёта определителя: при размерности 1 определитель 1-го порядка вычисляется по формуле (1.2), при размерности 2 – по формуле (1.0). В случае матрицы с размерностью 3 и больше определитель раскладывается в цикле по формуле (1.1), где в каждом слагаемом отдельно считается дополнительный минор в методе “Det”. Данный метод в результате возвращает значение определителя, попавшей в него матрицы. Метод “Minor” возвращает дополнительный минор к любому элементу из исходного определителя, используя попадаемые в него индексы определённого элемента. Методы “Zapolnenie” и “Print” используются для заполнения пользователем матрицы и её вывода соответственно.

class Matrix

{

public static void Main()

{

int a = 0;

while (a!=1)

{

Console.WriteLine("\tЗаполнение квадратной матрицы");

Console.Write("Введите размерность матрицы: ");

int n = int.Parse(Console.ReadLine());

double[,] matrix = new double[n, n];

Zapolnenie(matrix);

Print(matrix);

Console.WriteLine($"Определитель матрицы: {Det(matrix)}");

Console.WriteLine();

Console.Write("0-продолжить/ 1-остановить: ");

a = int.Parse(Console.ReadLine());

}

}

static void Zapolnenie(double[,] matrix)

{

for (int i = 0; i < matrix.GetLength(0); i++)

{

Console.Write($"{i + 1}-ая строка(через пробел): ");

string[] s = Console.ReadLine().Split(" ");

for (int j = 0; j < matrix.GetLength(1); j++)

{

matrix[i, j] = double.Parse(s[j]);

}

}

}

static void Print(double[,] matrix)

{

Console.WriteLine("Вывод квадратной матрицы: ");

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

Console.Write($" {matrix[i, j]} ");

}

Console.WriteLine();

}

}

static double Det(double[,] matrix)

{

double det = 0;

double size = matrix.GetLength(0);

if (size == 1)

det = matrix[0, 0];

if (size == 2)

det = (matrix[0, 0] \* matrix[1, 1]) - (matrix[0, 1] \* matrix[1, 0]);

if (size >= 3)

{

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

{

det += Math.Pow(-1, i) \* matrix[0, i] \* Det(Minor(matrix, 0, i));

}

}

return det;

}

static double[,] Minor(double[,] matrix, int str, int st)

{

double[,] minor = new double[matrix.GetLength(0) - 1, matrix.GetLength(1) - 1];

for(int i =0; i<matrix.GetLength(0); i++)

{

for(int j = 0; j<matrix.GetLength(1); j++)

{

if (i > str && j > st)

minor[i - 1, j - 1] = matrix[i, j];

if (i > str && j < st)

minor[i-1, j] = matrix[i, j];

if (i < str && j > st)

minor[i, j-1] = matrix[i, j];

if (i < str && j < st)

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

}

}

return minor;

}

}

**ПРИМЕРЫ РАБОТЫ**

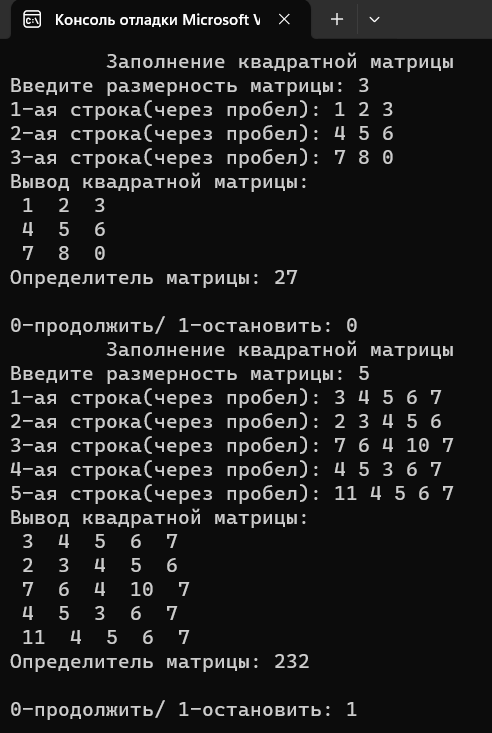
Результаты работы программы представлены на Рисунке 1.

Рисунок 1 – примеры работы программы

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения расчётно-графической работы был реализован и описан алгоритм вычисления определителя матрицы. Были использованы дополнительные справочные материалы для полного освоения темы.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Лунгу, К. Н. Сборник задач по высшей математике. 1 курс. Изд. – Москва: Айрис-пресс, 2008. стр. 18-21.