Государственное автономное профессиональное образовательное учреждение

«Краснокаменский горно-промышленный техникум»

**Специальность 09.02.07 «Информационные системы и программирование»**

**КУРСОВАЯ РАБОТА**

По ОП.04 «Основы алгоритмизации и программирование»

Тема: «Разработка приложения для преобразования систем чисел матрицы от минимальных до максимальных»

Вариант №17

Студента 2 курса 203 группы

Небытова Ярослава Алексеевича

Преподаватель Воробьёва Наталья Анатольевна

дата 15.05.23 роспись

**Краснокаменск 2023**

**СОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| ВВЕДЕНИЕ………………………………………………………..…… | | 3 |
| 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ……………………………….………..… | | 5 |
| 1.1 | Постановка задачи………………………………………………. | 5 |
| 1.2 | Требования к программному средству…………………….…… | 7 |
| 1.3 | Требования к программной документации………………..….... | 9 |
| 1.4 | Стадии и этапы разработки…………………………………..….. | 12 |
| 2 ПРАКТИЧЕСКАЯ ЧАСТЬ………………………………………….…. | | 13 |
| 2.1 | Разработка алгоритма решения задачи…………………………. | 13 |

ЗАКЛЮЧЕНИЕ…………………………………………………………….. 25

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ………………………. 26

ПРИЛОЖЕНИЕ…………………………………………………………….. 27

**ВВЕДЕНИЕ**

Данная курсовая работа посвящена разработке матрицы, которая использует двухмерный массив для вычисления минимального и максимального числа. Создание такого консольного приложения позволит существенно упростить и автоматизировать процесс обработки данных.

C# – это язык программирования, разработанный компанией Microsoft. Он используется для создания приложений для Windows, веб-приложений и игр. C# является объектно-ориентированным языком программирования и имеет синтаксис, похожий на языки Java и C++.

Двумерный массив - это одномерный массив, элементами которого являются одномерные массивы. Другими словами, это набор однотипных данных, имеющий общее имя, доступ к элементам которого осуществляется по двум индексам.

Целью данной задачи является разработка консольного приложения на языке C#, которое позволит работать с массивами и выполнять необходимые операции для вычисления.

**Задачи курсовой работы**:

1. Изучить основы массивов в C#.
2. Разработать консольное приложение.
3. Протестировать и отладить программный код.

Определившись с целью и задачами работы, можно приступить к выполнению курсовой работы.

Результатом данной работы будет функционирующая консольное приложение, которое сможет вычислять максимальное и минимальное число в матрице используя при этом двухмерный массив.

# **1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.1 Постановка задачи**

Создать консольное приложения с матрицей, которая использует двухмерный массив для вычисления минимального и максимального числа.

Что бы создать данный проект, нужно:

1. Создать консольное приложение в Visual Studio.
2. Создать двухмерный массив «,», который будет заполнятся случайными числами.
3. Прописываем формулу по которой массив будет вычислять своё максимальное и минимальное число.
4. Вводим размер матрицы.
5. Матрица по своему заданному размеру, начинает заполнять себя случайными числами.
6. Выводим максимальное и минимальное число матрицы.
7. Завершаем программу.

**1.2** **Математическая модель**

Дан двумерный массив со случайными числами. Двумерный массив нужно заполнить случайными числами:

matrix[i, j] = rnd.Next(100);

Console.Write("{0,3} ", matrix[i, j]);

По заданию нужно найти максимальное и минимальное число этого двумерного массива

Программа перебирает все числа массива и по данному коду находит минимальное и максимальное число:

if (matrix[i, j] > max)

{

max = matrix[i, j];

max\_i = i;

max\_j = j;

}

if (matrix[i, j] < min)

{

min = matrix[i, j];

min\_i = i;

min\_j = j;

}

**1.3 Разработка работы алгоритма приложения**

Блок-схема – распространённый тип схем, описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединённых между собой линиями, указывающими направление последовательности.

**Плюсы использования блок-схем:**

Блок-схемы проясняют каждую деталь, что может помочь командам быстро разобраться в поставленных задачах, когда и кем они должны выполняться. Эта информация может привести к сокращению числа совещаний, уменьшению возникающих вопросов и увеличению времени для выполнения работы. Блок-схемы также полезны для информационных руководств или даже для ориентации новых сотрудников.

**Минусы использования блок-схем:**

Минусы: трудоемкость, громоздкость. Линейный – алгоритм, в котором блоки выполняются последовательно друг за другом в порядке заданном блок-схемой. Разветвляющийся – алгоритм, в котором в зависимости от выполнения некоторого логического условия вычислительный процесс осуществляется по одной или другой ветви.

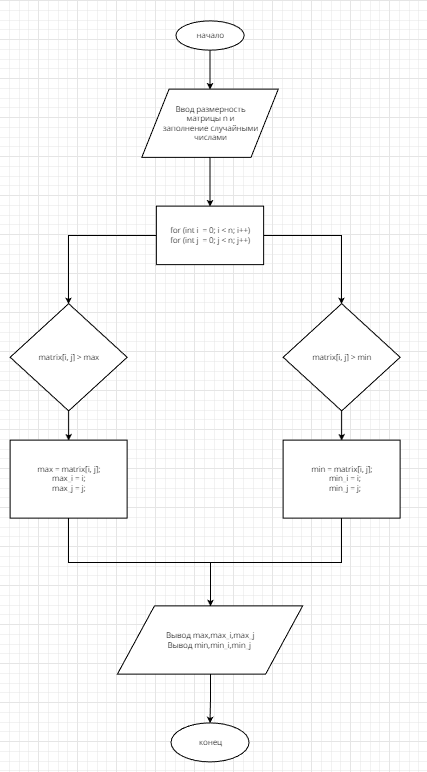
****

Рисунок 1 – Блок-схема алгоритма

В данной блок-схеме показан алгоритм выполнения задачи. Начинается блок-схема с «Начало» далее идёт ввод размерности матрицы n и её заполнения случайными числами. После этого начинается процесс перебора элементов матрицы, дальше происходит вычисление и присваивание элементов, после выполнения всех операций и перебора элементов происходит вывод переменный max,max\_i,max\_j и min,min\_i,min\_j.

**2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

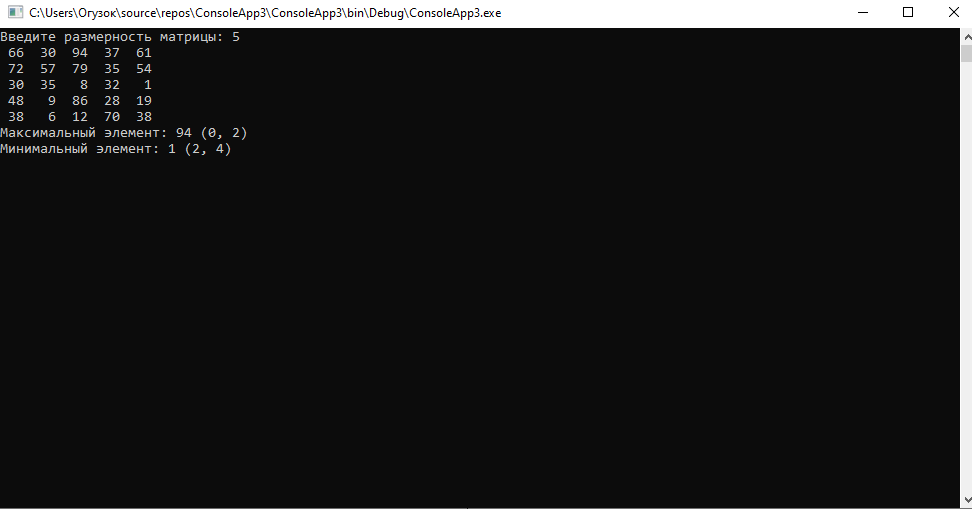
**2.1 Разработка интерфейса приложения**

Интерфейс – это ряд средств, предназначенных для взаимодействия человека и программы.

Иными словами это посредник в обмене данными.

Интерфейс нужен для общения программой и комуникациирования для выполне поставленой задачи перед пользавтелем.

В даном случае представленна довольная старая версия интерфейса в виде **Командной строки,** она считается одним их наиболее надежных типов пользовательского интерфейса.

  
Рисунок1- интерфейс матрицы

(На рисунке1) показан интерфейс матрицы

## **2.****2 Разработка программного кода**

Данный программный код написан на языке программирование C#

консольное предложение в коде был добавлен массив, матрица

и код счисления минимальных и максимальных чисел.

Для удобства написал комментарии в коде программы, для быстрого поиска нужного элемента.

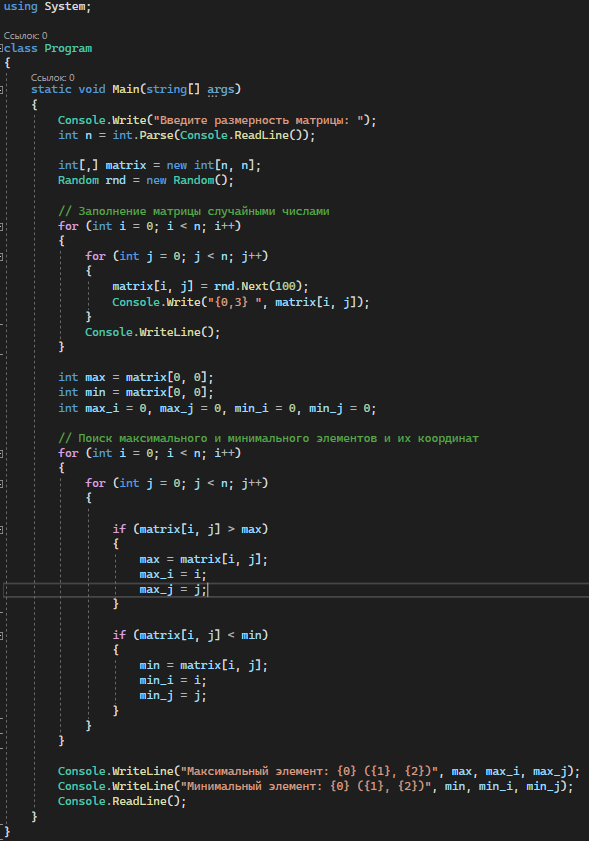


Рисунок2- Код программы на C#

(На рисунке2)- представлен код для нахождения случайных чисел в матрице минимального и максимального значения.

## 2.3**Тестирование и отладка**

Тестирование проводится для проверки программы на работа-способность и на правильность работы программы. Без тестирования не проходит ни одна программа.

На данном рисунке выполнено тестирование программы.

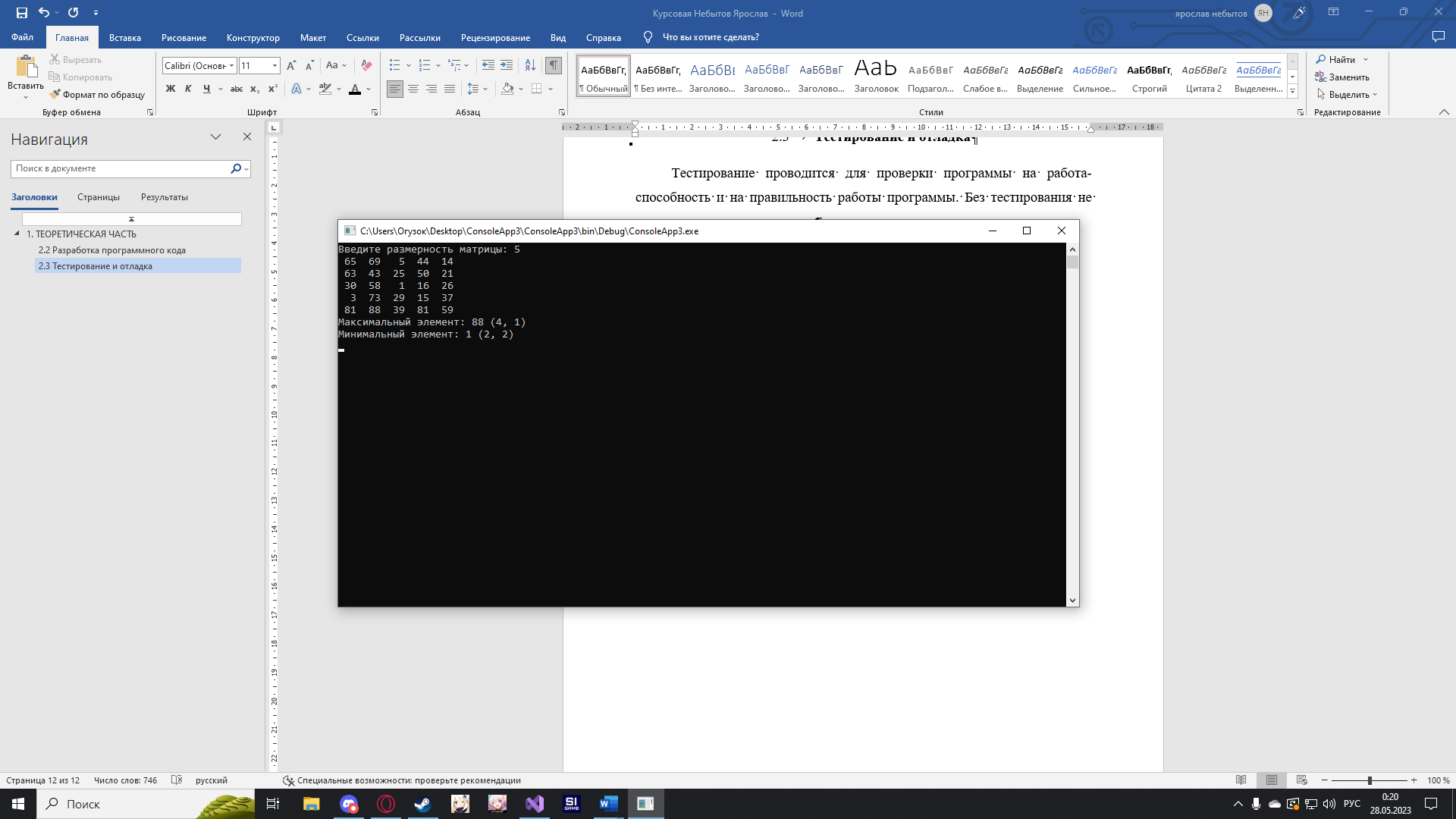


Рисунок3- Тест ввода случайных чисел в матрице

(На рисунке3) Программа автоматически при вводе случайных чисел с клавиатуры заполняет матрицу случайными числами и выводит матрицу на интерфейс программы.

И как видно на рисунке3 никаких проблем программа не выдаёт.

2. Проведём дополнительное тестирование уже с другим числом матрицы, дополнительная проверка никогда не повредит программе, а наоборот может выявить новые проблемы, которые в последствии надо решить.

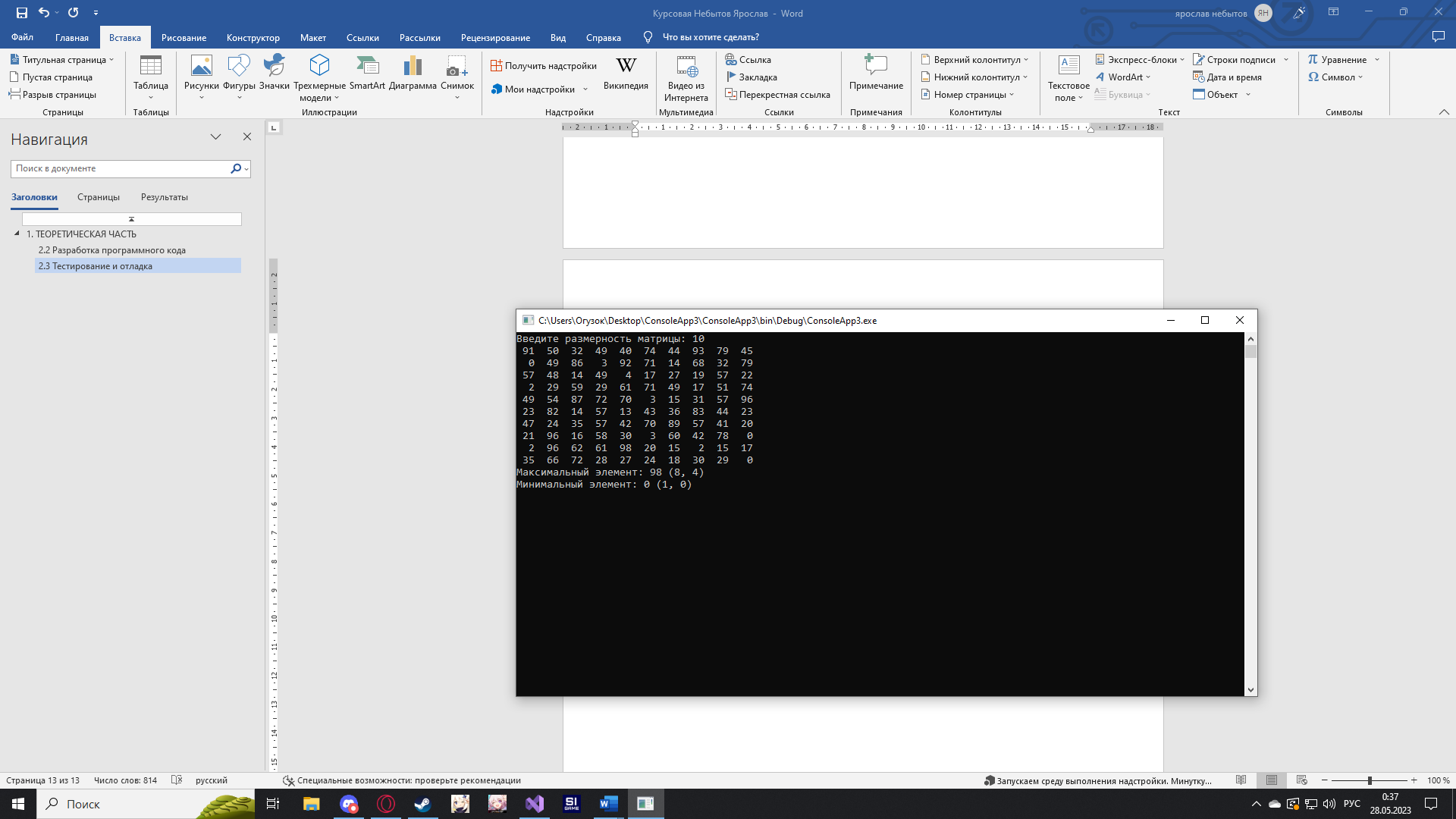


Рисунок4- Тест ввода случайных чисел в матрицу.

И как видим программа и в этот раз не выдала никаких проблем.

2.4 **Листинг программы**

using System;

class Program

{

static void Main(string[] args)

{

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

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

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

Random rnd = new Random();

// Заполнение матрицы случайными числами

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

{

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

{

matrix[i, j] = rnd.Next(100);

Console.Write("{0,3} ", matrix[i, j]);

}

Console.WriteLine();

}

int max = matrix[0, 0];

int min = matrix[0, 0];

int max\_i = 0, max\_j = 0, min\_i = 0, min\_j = 0;

// Поиск максимального и минимального элементов и их координат

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

{

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

{

if (matrix[i, j] > max)

{

max = matrix[i, j];

max\_i = i;

max\_j = j;

}

if (matrix[i, j] < min)

{

min = matrix[i, j];

min\_i = i;

min\_j = j;

}

}

}

Console.WriteLine("Максимальный элемент: {0} ({1}, {2})", max, max\_i, max\_j);

Console.WriteLine("Минимальный элемент: {0} ({1}, {2})", min, min\_i, min\_j);

Console.ReadLine();

}

}

**Заключение**

Как мы видим в данной работе были описаны все виды работы с программным кодом и реализации всего в курсовую работу с четкими пометка каждых действий непосредственно на языке программирования C# в консольном приложении.

Благодаря этому исследованию я понял, как работают матрицы и массивы, а также с помощью блок-схемы было понятно наглядное построение кода в программе.

Всё это было реализовано на языке программирования C# в Консольном приложении.

Были выполнены все поставленные задачи

1.Изучить основы массивов в C#.

2.Разработать консольное приложение.

3.Протестировать и отладить программный код.

**Литература**

## Книги и учебники:

## 1.Троелсен и Джепикс. Язык программирования C# 7 и платформы .NET и .NET Core.

## 2. Албахари. C# 7.0. Справочник. Полное описание языка.

## 3. Мюллер Д. П. C# для чайников.

## 4.Прайс. C# 7 и .NET Core. Кроссплатформенная разработка для профессионалов.

Интернет-ресурсы:

Интернет-ресурс по матрицам на языке C# Ссылка- <https://habr.com/ru/articles/494586/>

Интернет-ресурс по массивам на языке C# Ссылка- <https://metanit.com/sharp/tutorial/2.4.php>

Интернет-ресурс по всему языку C# Ссылка- <https://metanit.com/sharp/tutorial/>