**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **«СИНЕРГИЯ»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | 09.02.07 Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

.

**Отчет по лабораторной работе № 4**

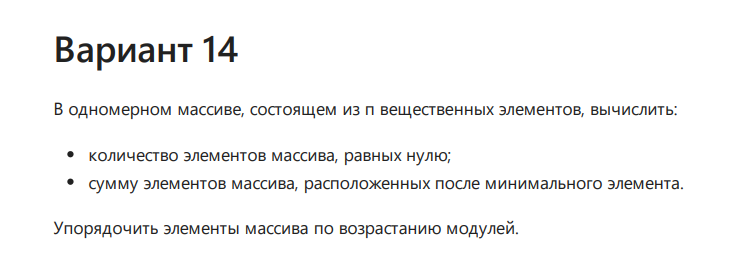
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** | |  | Использование операторов цикла | |
|  | | |  | (наименование темы) |
|  |  | |  | |
| **по дисциплине** | | | Разработка программных модулей | |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Обучающийся** |  | Касмынин Артем Егорович |  |
|  |  | (ФИО) |  |
| **Группа** |  | ДКИП 111 прог |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  |  |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва. 2025**

**Задание**



Код

using System;

using System.Linq;

class ArrayOperations

{

static void Main()

{

Console.Write("Введите количество элементов массива: ");

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

double[] array = new double[n];

Console.WriteLine($"Введите {n} вещественных чисел:");

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

{

Console.Write($"Элемент {i + 1}: ");

array[i] = double.Parse(Console.ReadLine());

}

Console.WriteLine("\nИсходный массив:");

PrintArray(array);

int zeroCount = array.Count(x => x == 0);

Console.WriteLine($"\nКоличество нулевых элементов: {zeroCount}");

double minValue = array.Min();

int minIndex = Array.IndexOf(array, minValue);

double sumAfterMin = array.Skip(minIndex + 1).Sum();

Console.WriteLine($"Минимальный элемент: {minValue} (индекс {minIndex})");

Console.WriteLine($"Сумма элементов после минимального: {sumAfterMin}");

Array.Sort(array, (a, b) => Math.Abs(a).CompareTo(Math.Abs(b)));

Console.WriteLine("\nМассив после сортировки по модулям:");

PrintArray(array);

}

static void PrintArray(double[] arr)

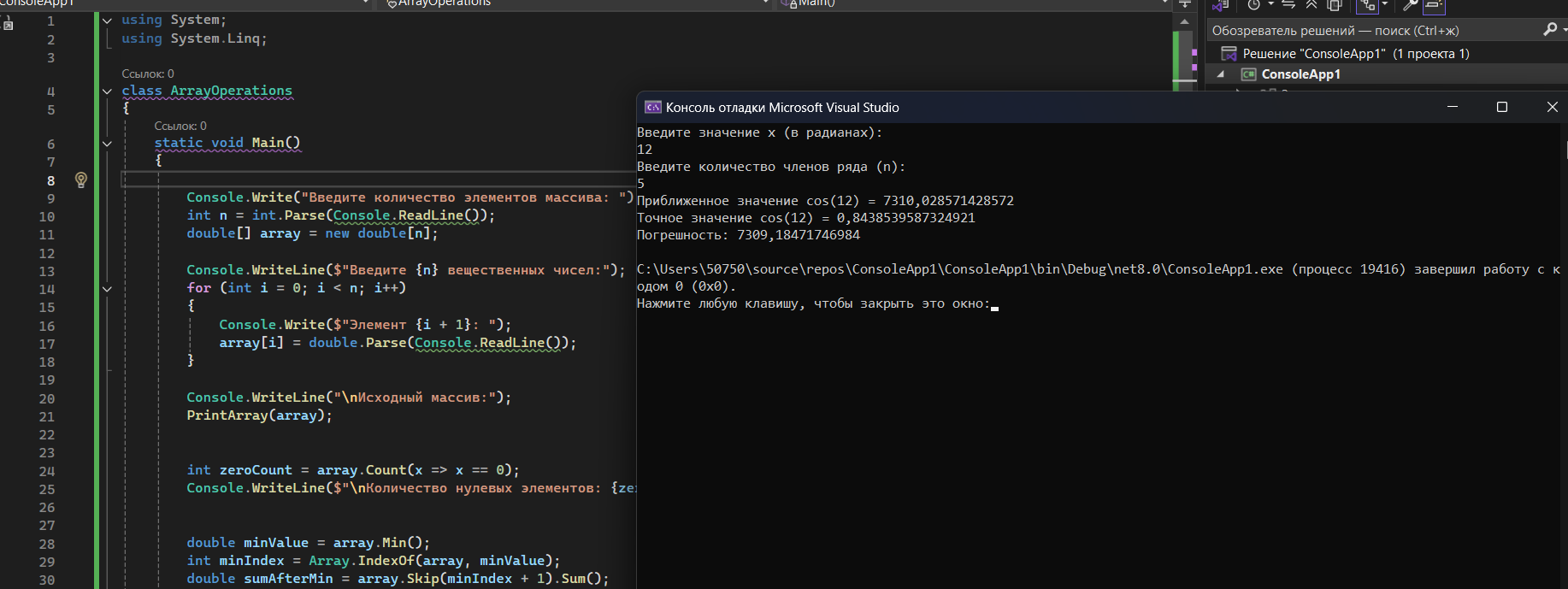
{

Console.WriteLine(string.Join(" ", arr));

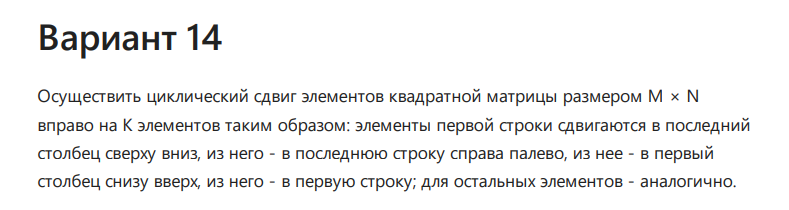
}

}

Скрин



Задание



Код

using System;

class MatrixShift

{

static void Main()

{

Console.Write("Введите количество строк (M): ");

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

Console.Write("Введите количество столбцов (N): ");

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

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

Console.WriteLine("Введите элементы матрицы:");

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

{

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

{

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

matrix[i, j] = int.Parse(Console.ReadLine());

}

}

Console.Write("Введите количество сдвигов (K): ");

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

Console.WriteLine("\nИсходная матрица:");

PrintMatrix(matrix);

for (int shift = 0; shift < k; shift++)

{

matrix = ShiftMatrix(matrix);

}

Console.WriteLine($"\nМатрица после {k} сдвигов:");

PrintMatrix(matrix);

}

static int[,] ShiftMatrix(int[,] matrix)

{

int m = matrix.GetLength(0);

int n = matrix.GetLength(1);

int[,] newMatrix = new int[m, n];

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

{

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

{

if (i == 0 || j == n - 1 || i == m - 1 || j == 0)

{

int newI, newJ;

if (i == 0 && j < n - 1)

{

newI = j + 1;

newJ = n - 1;

}

else if (j == n - 1 && i < m - 1)

{

newI = m - 1;

newJ = n - 2 - i;

}

else if (i == m - 1 && j > 0)

{

newI = m - 1 - j;

newJ = 0;

}

else

{

newI = 0;

newJ = i - 1;

}

newMatrix[newI, newJ] = matrix[i, j];

}

else

{

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

}

}

}

return newMatrix;

}

static void PrintMatrix(int[,] matrix)

{

int m = matrix.GetLength(0);

int n = matrix.GetLength(1);

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

{

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

{

Console.Write($"{matrix[i, j],4}");

}

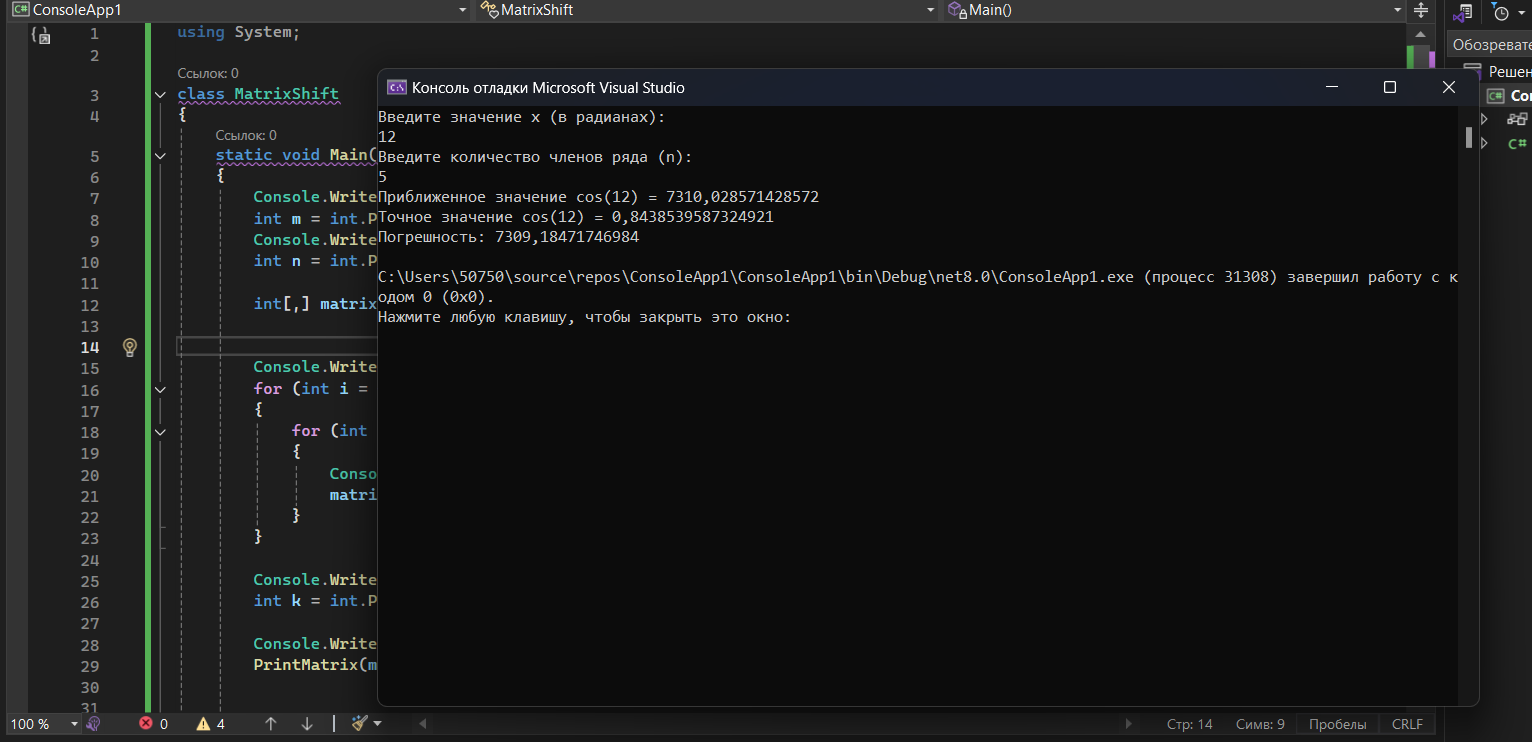
Console.WriteLine();

}

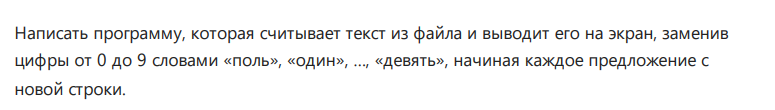
}

}

Скрин



Задание



Код

using System;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

class Program

{

static void Main()

{

string[] digitWords = {

"ноль", "один", "два", "три", "четыре",

"пять", "шесть", "семь", "восемь", "девять"

};

Console.Write("Введите путь к файлу: ");

string filePath = Console.ReadLine();

try

{

string text = File.ReadAllText(filePath, Encoding.UTF8);

string processedText = Regex.Replace(text, @"\d", match =>

{

int digit = int.Parse(match.Value);

return digitWords[digit];

});

string[] sentences = Regex.Split(processedText, @"(?<=[.!?])\s+");

Console.WriteLine("\nРезультат обработки:");

foreach (string sentence in sentences)

{

if (!string.IsNullOrWhiteSpace(sentence))

{

Console.WriteLine(sentence.Trim());

}

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка: {ex.Message}");

}

}

}

Скрин

