**5.** **Массивы. Класс Array**

Задание №1. Найти минимальный элемент массива и его порядковый номер.

Листинг программы:

Console.Write("Введите кол-во элементов: ");

int n = Convert.ToInt32(Console.ReadLine());

int[] array = GetRandomMatrix(n);

WriteMatrix(array);

Console.WriteLine();

int indexMinNum = GetIndexMinElemInArray(array);

Console.WriteLine($"Найменьшое число в матрице {array[indexMinNum]} под индексом {indexMinNum}");

int[] GetRandomMatrix(int n)

{

int[] arr = new int[n];

Random random = new Random();

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

{

arr[i] = random.Next(1, 10);

}

return arr;

}

void WriteMatrix(int[] array)

{

for (int i = 0; i < array.GetUpperBound(0) + 1; i++)

{

Console.Write($"{array[i]}\t");

Console.WriteLine();

}

}

int GetIndexMinElemInArray(int[] array)

{

int[] arr2 = new int[array.Length];

Array.Copy(array,arr2,array.Length);

Array.Sort(arr2);

return Array.IndexOf(array, arr2[0]);

}

Таблица 3.1 – Входные и выходные данные программы задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | 3 под индексом 1 |

Анализ результатов:

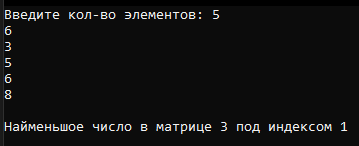


Рисунок 3.1 – Результат выполнения программы задание №1

Задание №2. Пусть даны натуральное число п, целые числа а 1 , ..., а n . Получите сумму положительных, число отрицательных и число нулевых членов последовательности а 1 , ..., а n .

Листинг программы:

Console.Write("Введите N: ");

int n = Convert.ToInt32(Console.ReadLine());

int[] arr = GetArray(n);

Console.WriteLine($"Массив: ");

WriteArray(arr);

Console.WriteLine($"Сумма положительных чисел в массиве: {GetSumArray(arr)}");

Console.WriteLine($"Кол-во отрицательных чисел: {GetCountNegativeNumInArr(arr)}");

Console.WriteLine($"Кол-во нулей: {GetCountZeroInArr(arr)}");

int[] GetArray(int n)

{

int[] arr = new int[n];

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

{

arr[i] = i;

}

return arr;

}

void WriteArray(int[] arr)

{

for (int i = 0; i < arr.Length; i++)

{

Console.Write($"{arr[i]}\t");

}

Console.WriteLine();

}

int GetSumArray(int[] arr)

{

int sum = 0;

for (int i = 0; i < arr.Length; i++)

{

if (arr[i] > 0)

{

sum += arr[i];

}

}

return sum;

}

int GetCountNegativeNumInArr(int[] arr)

{

int count = 0;

for (int i = 0; i < arr.Length; i++)

{

if (arr[i] < 0)

{

count += arr[i];

}

}

return count;

}

int GetCountZeroInArr(int[] arr)

{

int count = 0;

for (int i = 0; i < arr.Length; i++)

{

if (arr[i] == 0)

{

count += arr[i];

}

}

return count;

}

Таблица 3.2 – Входные и выходные данные программы задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | Сумма положительных чисел в массиве: 10  Кол-во отрицательных чисел: 0  Кол-во нулей: 0 |

Анализ результатов:

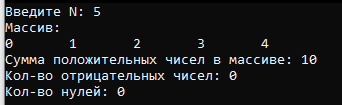


Рисунок 3.2 – Результат выполнения программы задание №2

Задание №3. В каждой задаче задается квадратная целочисленная матрица NxN(значение N вводится с клавиатуры, N<10). Программа должна заполнять

матрицу случайными числами из диапазона [a, b] (a, b вводятся с клавиатуры) и осуществлять вывод на экран исходной матрицы. Затем необходимо произвести необходимые действия и напечатать результаты.

Подсчитать количество чисел, меньших заданного D. Вычислить среднее арифметическое элементов каждого столбца.

Листинг программы:

Console.Write("Введите N: ");

int n = Convert.ToInt32(Console.ReadLine());

while ( n >= 10)

{

Console.WriteLine("n должно быть меньше 10");

Console.Write("Введите N: ");

n = Convert.ToInt32(Console.ReadLine());

}

Console.Write("Введите a: ");

int a = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите b: ");

int b = Convert.ToInt32(Console.ReadLine());

int[,] matrix = GetRandomMatrix(n, a, b);

WriteMatrix(matrix);

Console.Write("Введите D: ");

int d = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"Кол-во чисел меньше {d} равно {GetCountElemLessNum(d, matrix)}");

Console.WriteLine("Среднее арифметическое элементов каждого столбца: ");

double[] arrAripheticColumMatrix = GetAripheticColumMatrix(matrix);

for(int i = 0; i < arrAripheticColumMatrix.Length; i++)

{

Console.WriteLine($"Столбец {i} арифметическое {arrAripheticColumMatrix[i]}");

}

int[,] GetRandomMatrix(int n, int a, int b)

{

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

Random random = new Random();

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

{

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

{

matrix[i, j] = random.Next(a, b + 1);

}

}

return matrix;

}

void WriteMatrix(int[,] matrix)

{

for (int i = 0; i < matrix.GetUpperBound(0) + 1; i++)

{

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

{

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

}

Console.WriteLine();

}

}

int GetCountElemLessNum(int num, int[,] matrix)

{

int count = 0;

for (int i = 0; i < matrix.GetUpperBound(0) + 1; i++)

{

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

{

if(num > matrix[i, j])

{

count++;

}

}

}

return count;

}

double[] GetAripheticColumMatrix(int[,] matrix)

{

double[] arrReselt = new double[matrix.GetUpperBound(0) + 1];

for (int i = 0; i < matrix.GetUpperBound(1) + 1; i++)

{

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

{

arrReselt[i] += matrix[j, i];

}

arrReselt[i] /= matrix.GetUpperBound(1) + 1;

}

return arrReselt;

}

Таблица 3.3 – Входные и выходные данные программы задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| n:4  a:1  a:9 | Кол-во чисел меньше 3 равно 4  Среднее арифметическое элементов каждого столбца:  Столбец 0 арифметическое 4  Столбец 1 арифметическое 5,5  Столбец 2 арифметическое 4,5  Столбец 3 арифметическое 4,25 |

Анализ результатов:

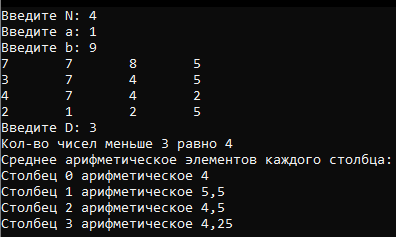


Рисунок 3.3 – Результат выполнения программы задание №3

Задание №4. зрительном зале 23 ряда, в каждом из которых 40 мест (кресел). Информация о проданных билетах хранится в двумерном массиве, номера строк которых соответствуют номерам рядов, а номера столбцов – номерам мест. Если билет на то или иное место продан, то соответствующий элемент массива имеет значение 1, в противном случае – 0. Определить, имеются ли свободные места в первом ряду.

Листинг программы:

int[,] places = GetRandomMatrix(23,40);

WriteMatrix(places);

Console.WriteLine("Свободные места в первом ряду: ");

int[] arrFreeSeats = FindFreeSeats(places,0);

foreach(int num in arrFreeSeats)

{

Console.Write($"{num + 1} ");

}

Console.WriteLine();

int[,] GetRandomMatrix(int n, int m)

{

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

Random random = new Random();

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

{

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

{

matrix[i, j] = random.Next(0, 2);

}

}

return matrix;

}

void WriteMatrix(int[,] matrix)

{

for (int i = 0; i < matrix.GetUpperBound(0) + 1; i++)

{

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

{

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

}

Console.WriteLine();

}

}

int[] FindFreeSeats(int[,] matrix, int rowNum)

{

List<int> list = new List<int>();

for(int i = 0; i < matrix.GetUpperBound(0) + 1; i++)

{

if (matrix[rowNum, i] == 1)

{

list.Add(i);

}

}

return list.ToArray();

}

Таблица 3.4 – Входные и выходные данные программы задание №4

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Свободные места в первом ряду:  3 4 6 8 10 11 15 19 20 21 22 23 |

Анализ результатов:



Рисунок 3.4 – Результат выполнения программы задание №4

Задание №5. Разработать рекурсивный алгоритм и программу решения задачи, в которой вычислить: f(n)=(n-1)!/(n-2)!. Исходные данные вводятся с клавиатуры.

Листинг программы:

Console.Write($"Введите n: ");

int n = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(Factorial(n - 1) / Factorial(n - 2));

int[] arr = new int[5];

int Factorial(int num)

{

if(num == 0)

{

return 1;

}

else

{

return num \* Factorial(num - 1);

}

}

Таблица 3.5 – Входные и выходные данные программы задание №5

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | 4 |

Анализ результатов:



Рисунок 3.5 – Результат выполнения программы задание №5