**5 Массивы. Класс Array.**

Задание 1. В массиве из 15 целых чисел найти наибольший элемент и поменять его местами с первым элементом.

Листинг программы:

try

{

int[] arr = new int[15];

int jPosition = 0;

int max = 0;

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

{

Random r = new Random();

arr[i] = r.Next(1, 30);

}

Console.WriteLine("Начальный массив:");

foreach (int item in arr)

{

Console.Write(item + " ");

}

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

{

if (arr[i] > max)

{

max = arr[i];

jPosition = i;

}

}

Console.WriteLine($"\nmax = {max}");

arr[jPosition] = arr[0];

arr[0] = max;

Console.WriteLine("\nResult:");

foreach (int item in arr)

{

Console.Write(item + " ");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 5.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Result: 28 12 3 7 15 9 23 13 12 4 4 21 25 25 20 |

Анализ результатов:

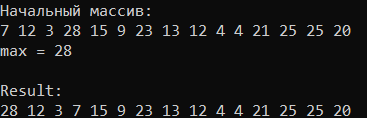


Рисунок 5.1 – Результаты работы программы

Источник: собственная разработка

Задание 2 Пусть дано 100 целых чисел. Распечатайте их в обратном порядке по 6 чисел в строке.

Листинг программы:

try

{

int[] arr = new int[100];

int temp;

int count = 0;

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

{

Random r = new Random();

arr[i] = r.Next(1, 1000);

}

Console.WriteLine("Начальный массив:");

foreach (int item in arr)

{

Console.Write(item + " ");

}

for (int i = 0; i < arr.Length - 1; i++)

{

for (int j = i + 1; j < arr.Length; j++)

{

if (arr[i] > arr[j])

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

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

foreach (int item in arr)

{

Console.Write(item + " ");

}

int n = arr.Length;

int center = n / 2;

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

{

temp = arr[i];

arr[i] = arr[n - i - 1];

arr[n - i - 1] = temp;

}

Console.WriteLine("\nИнверсия массива:");

foreach (int item in arr)

{

Console.Write(item + " ");

}

Console.WriteLine("\nИнверсия массива (вывод в строке 6 чисел):");

foreach (int item in arr)

{

Console.Write(item + " ");

count++;

if (count == 6)

{

count = 0;

Console.WriteLine();

}

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 5.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Инверсия массива (вывод в строке 6 чисел):  998 994 989 977 973 962  954 949 944 941 940 932  931 921 906 906 901 876  862 845 839 836 830 826  823 820 818 789 785 761  737 711 698 695 689 674  668 657 652 645 635 635  601 599 571 571 555 538  526 496 492 479 468 446  445 421 420 413 386 372  364 337 332 329 314 291  287 242 242 226 226 213  208 201 198 189 188 184  177 176 167 160 152 115  103 99 92 88 83 82  81 66 59 58 48 44  38 34 13 1 |

Анализ результатов:

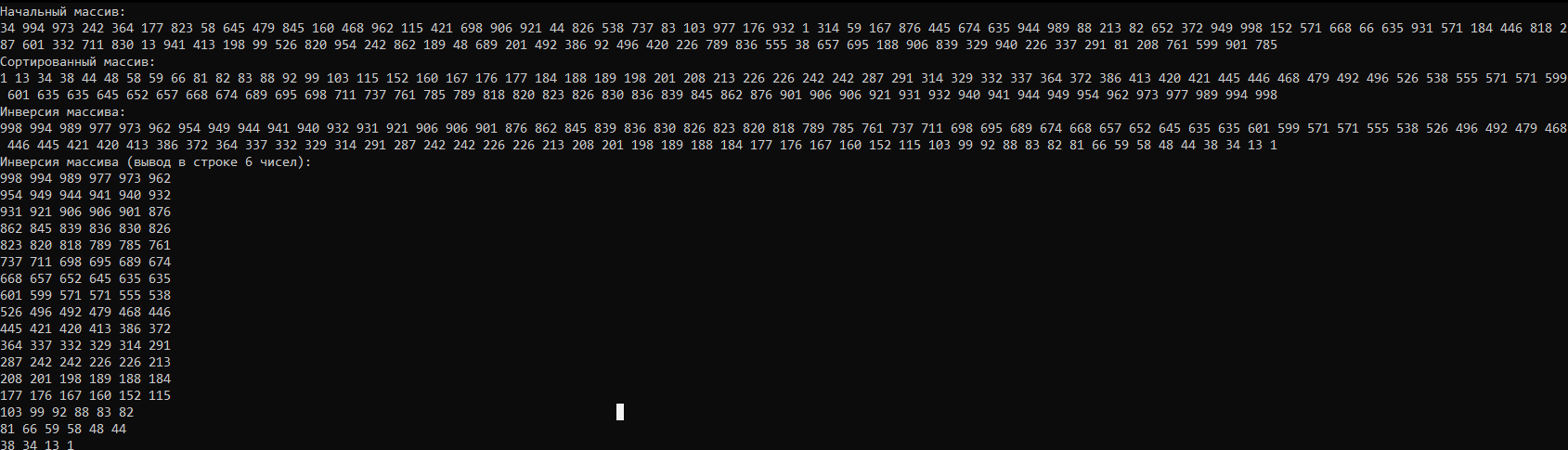


Рисунок 5.2 – Результаты работы программы

Источник: собственная разработка

Задание 3. Подсчитать количество положительных чисел. Вычислить сумму элементов каждой строки.

Листинг программы:

try

{

Console.Write($"Введите N = ");

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

Console.Write($"Введите a = ");

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

Console.Write($"Введите b = ");

var b = int.Parse(Console.ReadLine());

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

int count = 0;

int summI = 0;

Random r = new Random();

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

{

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

{

matrix[i, j] = r.Next(a, b);

if (matrix[i, j] > 0)

{

count += 1;

}

}

}

Console.WriteLine("\nМатрица:");

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

{

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

{

Console.Write(matrix[i, j] + " ");

}

Console.WriteLine();

}

Console.WriteLine($"Положительные элементы: {count}");

Console.WriteLine("\nСумма каждой строки:");

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

{

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

{

summI += matrix[i, j];

}

Console.WriteLine($"Summ {i} строки = {summI}");

summI = 0;

}

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

}

catch (OverflowException ex)

{

Console.WriteLine(ex.Message);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 5.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| N = 4  a = -5  b = 5 | Матрица:  -3 0 0 2  -5 4 4 4  -1 -3 4 -1  -5 3 -2 3  Положительные элементы: 7  Сумма каждой строки:  Summ 0 строки = -1  Summ 1 строки = 7  Summ 2 строки = -1  Summ 3 строки = -1 |

Анализ результатов:

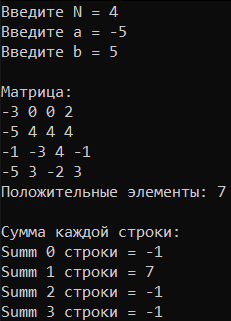


Рисунок 5.3 – Результаты работы программы

Источник: собственная разработка

Задание 4. Дан двумерный массив. Выяснить, является ли произведение элементов второго столбца массива трехзначным числом.

Листинг программы:

try

{

int n = 5;

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

int mul = 1;

Random r = new Random();

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

{

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

{

matrix[i, j] = r.Next(1, 20);

if (j == 1)

{

mul \*= matrix[i, j];

}

}

}

Console.WriteLine("Матрица:");

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

{

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

{

Console.Write(matrix[i, j] + " ");

}

Console.WriteLine();

}

if (mul / 100 >= 1)

{

Console.WriteLine("произведение элементов второго столбца массива является трехзначным числом.");

}

else

{

Console.WriteLine("произведение элементов второго столбца массива не является трехзначным числом.");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 5.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Матрица:  11 18 11 12 9  10 3 8 13 5  8 15 14 19 2  7 2 9 10 17  11 5 19 12 2  произведение элементов второго столбца массива является трехзначным числом. |

Анализ результатов:

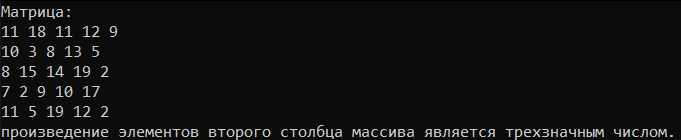


Рисунок 5.4 – Результаты работы программы

Источник: собственная разработка

Задание 5. Разработать рекурсивный алгоритм и программу решения задачи, в которой вычислить: f(n)=(n-2)!/(n-4)!. Исходные данные вводятся с клавиатуры.

Листинг программы:

static int Factorial(int f)

{

if (f == 0)

return 1;

else

return f \* Factorial(f - 1);

}

try

{

Console.Write("Введите n = ");

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

var f = Factorial(n - 2) / Factorial(n - 4);

Console.WriteLine($"result = {f}");

}

catch (FormatException e)

{

Console.WriteLine(e.Message);

}

catch (DivideByZeroException e)

{

Console.WriteLine(e.Message);

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Таблица 5.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| n = 5 | result = 6 |

Анализ результатов:



Рисунок 5.5 – Результаты работы программы

Источник: собственная разработка

Задание 6. Заполнить двумерный массив размером 7х7 числами 1, 2, …, 49, расположенными в нем по спирали.

Листинг программы:

try

{

const int n = 7;

const int m = 7;

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

int row = 0;

int col = 0;

int dx = 1;

int dy = 0;

int dirChanges = 0;

int visits = m;

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

{

matrix[row, col] = i + 1; // поочередное заполнение данными

if (--visits == 0)

{

visits = m \* (dirChanges % 2) + n \* ((dirChanges + 1) % 2) - (dirChanges / 2 - 1) - 2;

int temp = dx;

dx = -dy;

dy = temp;

dirChanges++;

}

col += dx;

row += dy;

}

for (int i = 0; i < n; i++) //вывод

{

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

{

Console.Write(matrix[i, j] + " ");

}

Console.WriteLine();

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 5.6 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | 1 2 3 4 5 6 7  24 25 26 27 28 29 8  23 40 41 42 43 30 9  22 39 48 49 44 31 10  21 38 47 46 45 32 11  20 37 36 35 34 33 12  19 18 17 16 15 14 13 |

Анализ результатов:

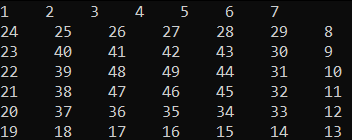


Рисунок 5.6 – Результаты работы программы

Источник: собственная разработка