**Лабораторная работа № 2**

**Тема:** Создание программ обработки массивов.

.

**Отчет**

Вариант 9

Выполнила ст. гр. 329198/9

Крупина Елена

**Цель работы:** изучить принципы описания и использования однородных структурированных данных в языке C#, получить практические навыки разработки программ по обработке массивов.

**Задание 1.** Вывести на экран одномерный массив, раскрасив в нем желтым цветом отрицательные элементы, если они четные, и зеленым цветом, если они нечетные.

**Математическая формулировка:**

1. Для выполнения задания был использован оператор if.
2. Для работы с массивами был использован цикл for
3. Для обеспечения отказоустойчивости был использован метод TryParse()

**Код программы:**

using System

public class MainClass

{

public static void Main()

{

//Размер массива и отказоустойчивость

Console.WriteLine("Введите размер массива (целое положительное число):");

int num;

string str;

str = Console.ReadLine();

while (!int.TryParse(str, out num) || num <= 0)

{

Console.WriteLine("Ошибка! Введите целое положительное число.");

str = Console.ReadLine();

}

//------------------------------------

Console.WriteLine("Одномерный массив: ");

Console.WriteLine();

int[] numbers = new int[num];

Random rnd = new Random();

//Заполнение массива

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

numbers[i]= rnd.Next(-100, 100);

Console.WriteLine("[ "+i+" ] "+numbers[i]);

}

//----------------------------------------------

Console.WriteLine("Одномерный массив (измененный): ");

//Вывод по заданию

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

if (numbers[i] % 2 == 0 && numbers[i] < 0)

{

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("[ " + i + " ] " + numbers[i]);

Console.ResetColor();

}

else if (numbers[i] % 2 != 0 && numbers[i] < 0)

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("[ " + i + " ] " + numbers[i]);

Console.ResetColor();

}

else { Console.WriteLine("[ " + i + " ] " + numbers[i]); }

}

//----------------------------------------------------------

}

}

**Блок-схема:**





**Результат:**

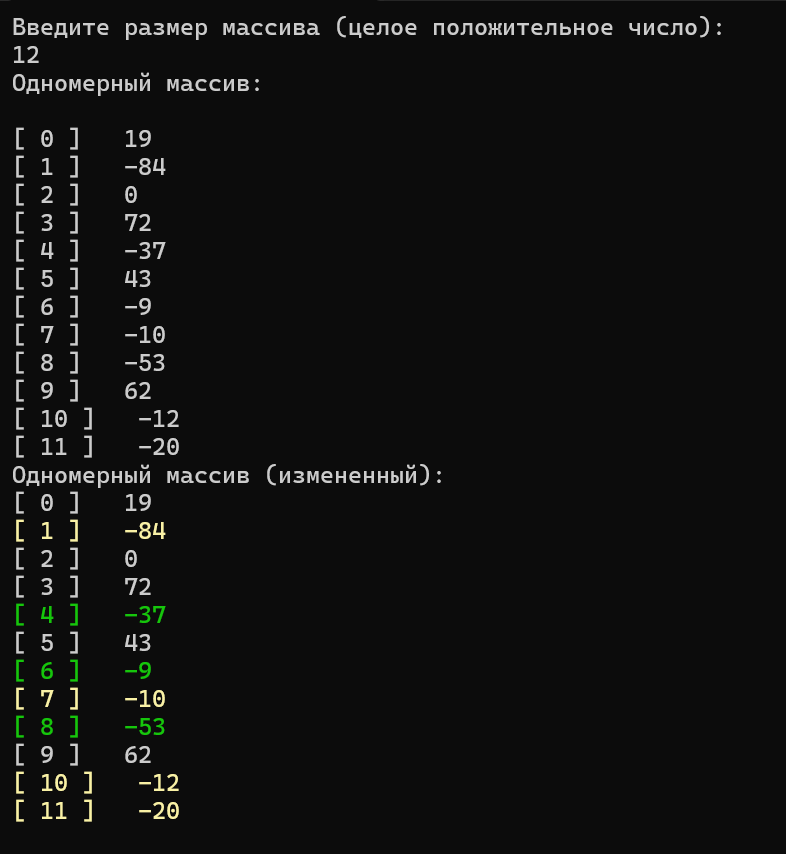
****

Рис.1 – Результат выполнения программы

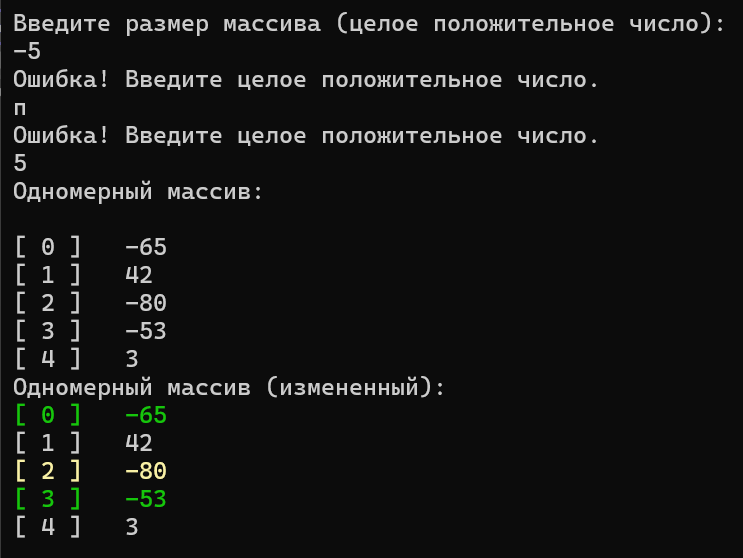
****

Рис.2 - Отказоустойчивость

**Задание 2.** Ввести двумерный массив произвольной размерности, найти минимальный элемент каждого столбца. Раскрасить в массиве при выводе розовым цветом минимальные элементы каждого столбца.

**Математическая формулировка:** Для реализации окрашивания минимальных элементов каждого столбца было создано два дополнительных массива: для хранения минимального элемента и для хранения индексов минимальных элементов.

**Код программы:**

using System;

public class MainClass

{

public static void Main()

{

bool proccess = false;

while (!proccess)

{

try

{

int col;

int row;

string str;

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

str = Console.ReadLine();

while (!int.TryParse(str, out row) || row <= 0)

{

Console.WriteLine("Ошибка! Введите целое положительное число.");

str = Console.ReadLine();

}

row = Convert.ToInt32(str);

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

str = Console.ReadLine();

while (!int.TryParse(str, out col) || col <= 0)

{

Console.WriteLine("Ошибка! Введите целое положительное число.");

str = Console.ReadLine();

}

col = Convert.ToInt32(str);

int[,] numbers = new int[row, col];

Random rnd = new Random();

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

{

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

{

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

}

}

// 1. Вывод исходной матрицы

Console.WriteLine("Двумерный массив: ");

Console.WriteLine();

Console.Write(" ");

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

{

Console.Write("[ " + i + " ]\t");

}

Console.WriteLine();

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

{

Console.Write("[ " + i + " ]\t");

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

{

Console.Write(numbers[i, j] + "\t");

}

Console.WriteLine();

}

Console.WriteLine();

Console.WriteLine();

// 2. Вычисление минимумов в каждом столбце и окрашивание

Console.WriteLine();

Console.WriteLine("Двумерный массив с минимальными элементами:");

Console.WriteLine();

Console.Write(" ");

int[,] indexMin = new int[col, 2]; //Массив для хранения индексов минимальных элементов

int[] min = new int[col]; //Массив для хранения минимальных элементов

//Заполнение массива минимумов

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

{

min[i] = 101;

}

//Заполнение массива индексов и массива минимальных элементов

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

{

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

{

if (numbers[i, j] < min[j])

{

min[j] = numbers[i, j];

indexMin[j, 0] = i;

indexMin[j, 1] = j;

}

}

}

//Собственно окрашивание

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

{

Console.Write("[ " + i + " ]\t");

}

Console.WriteLine();

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

{

Console.Write("[ " + i + " ]\t");

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

{

if ((i == indexMin[j, 0]) && (j == indexMin[j, 1]))

{

Console.ForegroundColor = ConsoleColor.Magenta;

Console.Write(numbers[i, j] + "\t");

Console.ResetColor();

}

else

{

Console.Write(numbers[i, j] + "\t");

}

}

Console.WriteLine();

proccess = true;

}

}

catch { Console.WriteLine("Массив оказался слишком большим"); }

}

}

}

**Блок-схема:**









**Результат выполнения:**

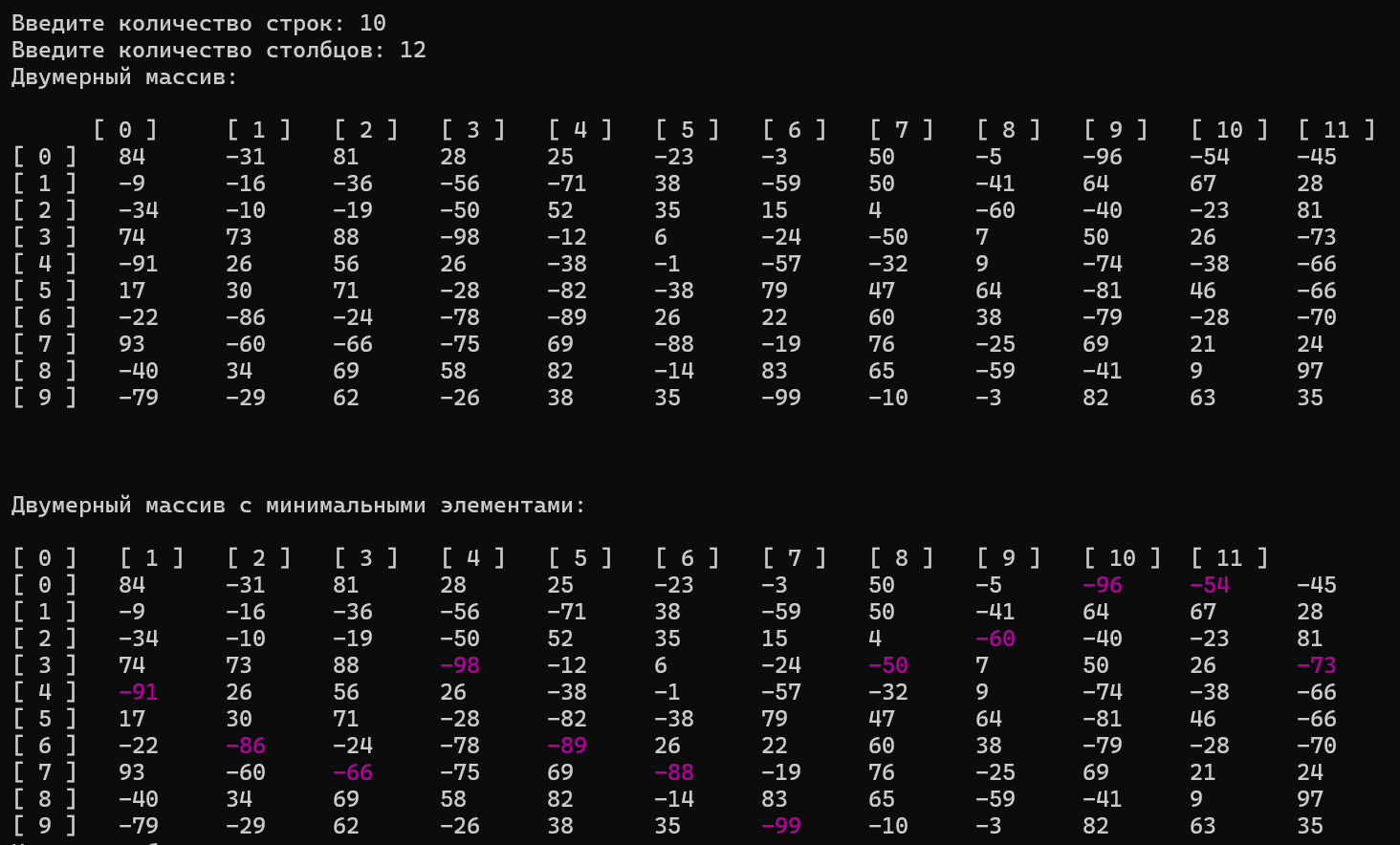
****

Рис.3 – Результат выполнения программы

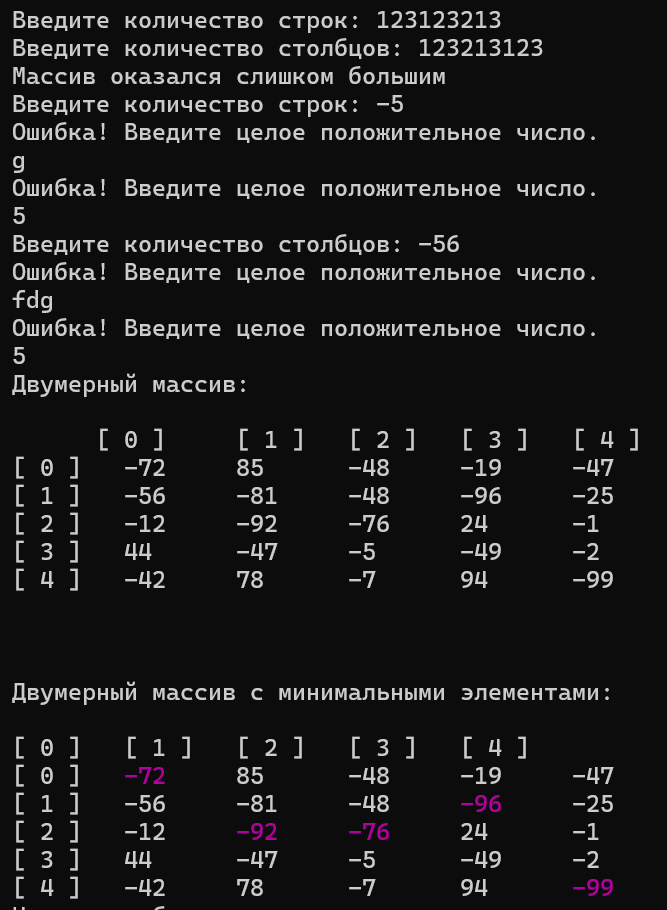
****

Рис.4 – Отказоустойчивость