**Лабораторная работа № 2**

**Тема:** Создание программ обработки массивов.

**Цель работы:**

Изучить принципы описания и использования однородных структурированных данных в языке C#, получить практические навыки разработки программ по обработке массивов.

**Задача 1:**

*Условия:*

Вывести на экран одномерный массив, раскрасив в нем синим цветом элементы, расположенные между первым положительным элементом и последним положительным элементом.

*Входные данные:*

size – int, размер массива больше 1, элементы массива задаются случайным образом.

int mass[size] - массив

*Выходные данные:*

Массив целых чисел с их индексами с синими элементами от первого положительного элемента до последнего положительного элемента.

*Алгоритм:*



*Код:*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab2\_1

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random(); // Переменная генератора

int size; // Целая переменная размера массива

Console.WriteLine("Программа работы с массивом."); // Вывод информации о программе

Console.Write("Входные данные:\nВведите размер массива: "); // Подсказки для ввода размера массива

size = int.Parse(Console.ReadLine()); // Ввод размера массива

if (size > 1) // Проверка введеного размера

{

int[] mass = new int[size]; // Создание и выделение памяти под массив

int findFirst = -1; // Целая переменная для хранения позиции первого положительного числа

int findLast = -1; // Целая переменная для хранения позиции последней положительного числа

for (int i = 0; i < size; i++) // Цикл генерации массива

{

mass[i] = rnd.Next(-100, 100); // генерация элемента с помощью random

if (mass[i] > 0) // Проверка элемента больше 0

{

if (findFirst == -1) // Проверка нахождения позиции первого положительного элемента

{

findFirst = i; // Присваивание первой позиции

continue;

}

findLast = i; // Присваивание последней позиции

}

}

Console.WriteLine("Выходные данные:"); // Вывод данных

Console.WriteLine($"Сгенерированный массив с длинной {size}: "); // Подсказки

for (int i = 0; i < size; i++) // Цикл вывода массива

{

if (i > findFirst && i < findLast) // Проверка нахождения текущего элемента между первым и последним положительным элемнтом

{

Console.ForegroundColor = ConsoleColor.Blue; // Смена цвета текста на синий

}

else

{

Console.ResetColor(); // Сброс цвет на стандартный (белый)

}

Console.WriteLine($"mass[{i}] = {mass[i]}"); // Вывод элемента с его индексом

}

}

else

{

Console.WriteLine("Ошибка!!!"); // Подсказка о ошибке

Console.WriteLine("Неверные входные данные (Размер массива должен быть больше 1)."); // Вывод подсказки о неверных входных данных

}

Console.Write("Для завершения программы нажмите Enter..."); // Подсказка для завершения программы

Console.ReadLine(); // Ожидание ввода для завершения программы

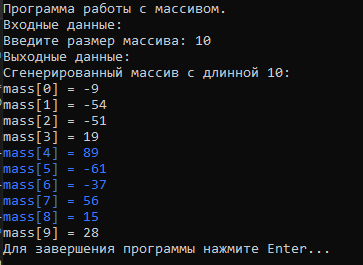
}

}

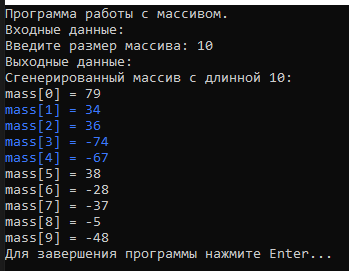
}

*Пример выполнения:*

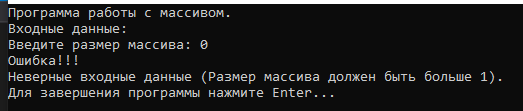
Сгенерированный массив.



Сгенерированный массив.



Ввод неверных данных.



**Задача 2:**

*Условия:*

Ввести двумерный массив произвольной размерности, найти первый положительный элемент каждой строки массива. Раскрасить при выводе малиновым цветом первый положительный элемент каждой строки массива.

*Входные данные:*

size\_i – int, количество строк больше 1.

size\_j – int, количество столбцов больше 1.

Int mass[size\_i,size\_j] – заполняется случайными числами.

*Выходные данные:*

Двумерный массив с закрашенным первым положительным элементом каждой строки.

*Алгоритм:*



*Код:*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab2\_2

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random(); // Переменная генератора

int size\_i; // Целая переменная количества строк

int size\_j; // Целая переменная количества столбцов

Console.WriteLine("Программа работы с двумерным массивом."); // Вывод информации о программе

Console.Write("Входные данные:\nВведите количество строк: "); // Подсказки для ввода количества строк

size\_i = int.Parse(Console.ReadLine()); // Ввод количества строк

Console.Write("Введите количество столбцов: "); // Подсказки для ввода количества столбцов

size\_j = int.Parse(Console.ReadLine()); // Ввод количества столбцов

if (size\_i > 1 && size\_j > 1) // Проверка правильности введённых данных

{

int[,] mass = new int[size\_i, size\_j]; // Создание и выделение памяти для двумерного массива

for (int i = 0; i < size\_i; i++) // Двойной цикл для заполнения двумерного массива случайными числами

{

for (int j = 0; j < size\_j; j++)

{

mass[i, j] = rnd.Next(-100, 100); // Заполнение элемента массива

}

}

bool firstNum = false; // Булевая переменна для нахождения первого положительного элемента строки

Console.WriteLine("Выходные данные:"); // Вывод данных

Console.WriteLine($"Сгенерированный массив с количеством строк {size\_i} и количеством столбцов {size\_j}: "); // Подсказки

for (int i = 0; i < size\_i; i++) // Двойной цикл для вывода двумерного массива

{

for (int j = 0; j < size\_j; j++)

{

if (!firstNum && mass[i, j] > 0) // Проверка положительности элемента и нахождения первого элемента

{

Console.ForegroundColor = ConsoleColor.Red; // Смена цвета консоли

Console.Write($"{mass[i, j]}\t"); // Вывод элемента

Console.ResetColor(); // Сброс цвета консоли

firstNum = true; // Нахождение первого положительного элемента

continue; // Продолжение цикла

}

Console.Write($"{mass[i, j]}\t"); // Вывод элемента

}

Console.WriteLine(); // Перевод строки

firstNum = false; // Сброс переменной нахождения первого положительного элемента строки

}

}

else

{

Console.WriteLine("Ошибка!!!"); // Подсказка о ошибке

Console.WriteLine("Неверные входные данные (Количество строк и количество столбцов должно быть больше 1)."); // Вывод подсказки о неверных входных данных

}

Console.Write("Для завершения программы нажмите Enter..."); // Подсказка для завершения программы

Console.ReadLine(); // Ожидание ввода для завершения программы

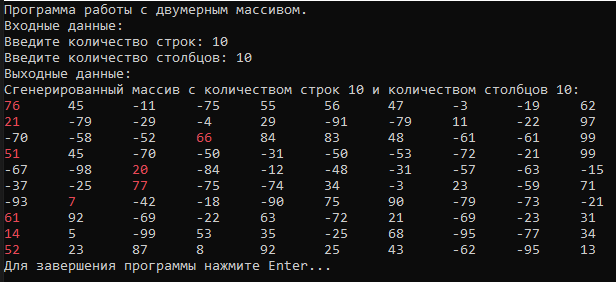
}

}

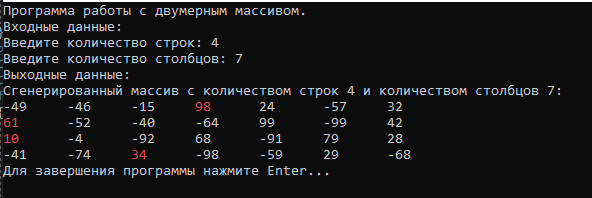
}

*Пример выполнения:*

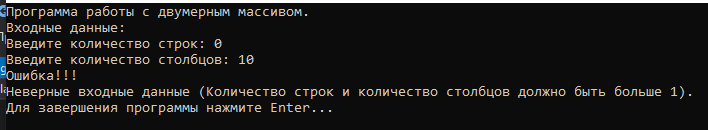
Успешное выполнение программы с массивом 10x10.



Успешное выполнение программы с массивом 4x7.



Неверные введенные данные - количество строк.



Неверные введенные данные - количество столбцов.

