Лабораторная работа №6

**Тема:** Разработка программ с использованием массивов.

**Цель:** Научится разрабатывать алгоритмы и реализовывать программы по обработке массивов с применением возможностей класса Array.

**Техническое оснащение:** ПК Win 10, MS Word, MS Visual Studio.

Выполнение работы

Вариант 7

**Пример 1.** Дан массив из n действительных чисел. Вычесть из каждого элемента массива среднее значение массива. Распечатать полученный массив.

Результат работы примера (рисунок 1).

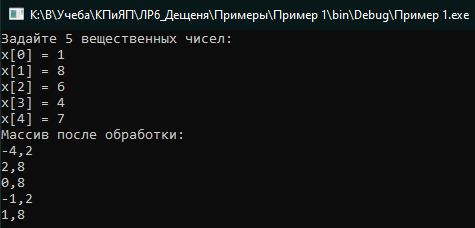


Рисунок 1 – Выполнение примера 1

**Пример 2.** Прономеровать матрицу действительных чисел A[2x3], т.е. каждый элемент матрицы поделить на максимальное по модулю число.

Результат работы примера (рисунок 2).

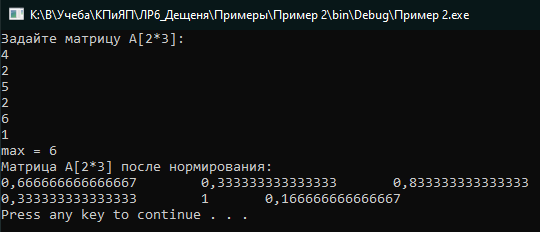


Рисунок 2 – Выполнение примера 2

**Пример 3.** Использование методов класса Array с одномерным массивом.

Результат работы примера (рисунок 3).

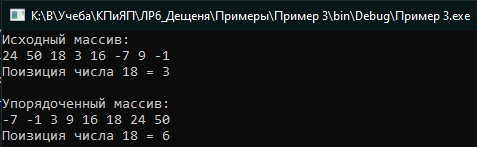


Рисунок 3 – Выполнение работы примера 3

**Пример 4.** Разработаем класс для работы с одномерным массивом. Создадим в нём индексатор, позволяющий обращаться к элементу массива по индексу.

Результат работы примера (рисунок 4).

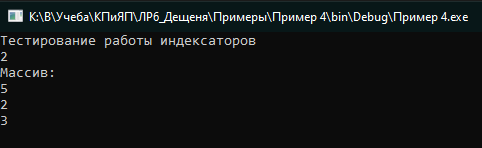


Рисунок 4 – Выполнение примера 4

**Пример 5.** Разработаем класс для работы с двумерным массивом (матрицей). Создадим в нём индексатор, позволяющий обращаться к элементу матрицы по индексу.

Результат работы примера (рисунок 5).

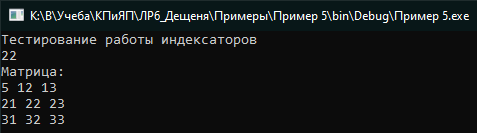


Рисунок 5 – Выполнение примера 5

**Пример 6.** Использование формы при решении задач: 1. В одномерном массиве нулевые элементы удалить, положительные элементы расставить по убыванию, отрицательные – по возрастанию. Получить зависимость затрат машинного времени от размера массива. 2. В матрице удалить строки с последними отрицательными элементами, а затем добавить строку из сумм элементов по столбцам.

Проверка работы с массивом (рисунок 6).

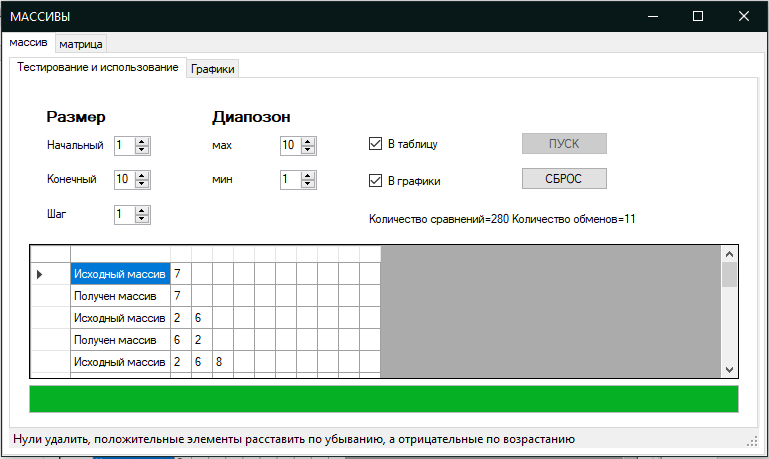


Рисунок 6 – Проверка работа с массивом

Проверка построения графиков (рисунок 7).

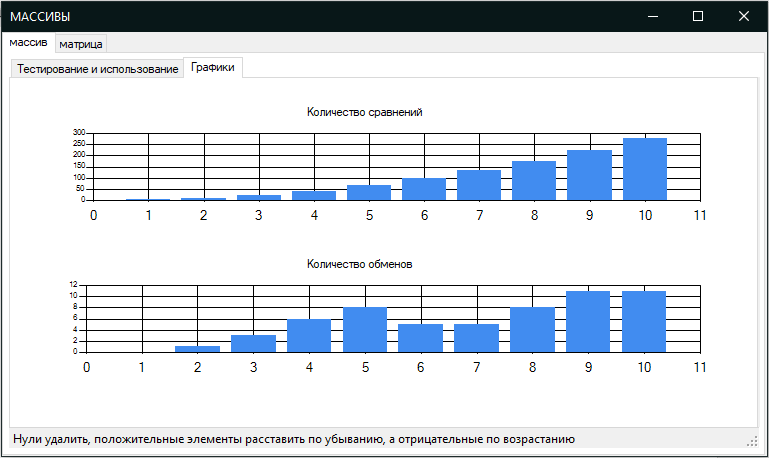


Рисунок 7 – Проверка построения графиков

Проверка работы с матрицами (рисунок 9).

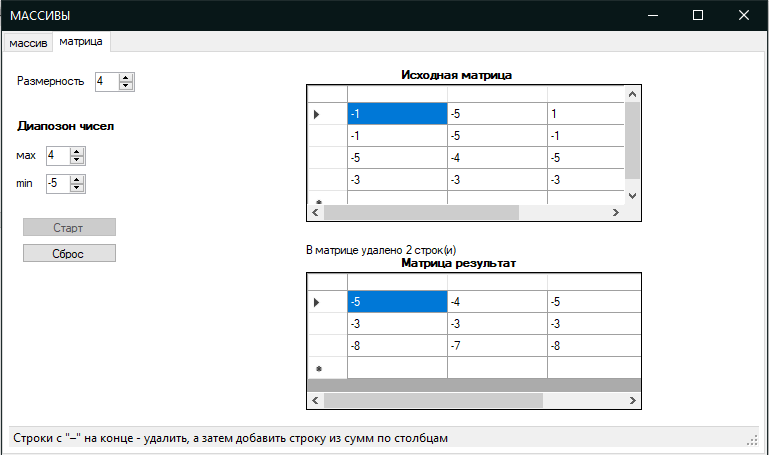


Рисунок 9 – Проверка работы с матрицами

**Задание 1.** Дан массив ненулевых целых чисел размера N. Проверить, чередуются ли в нем четные и нечетные числа. Если чередуются, то вывести 0, если нет, то вывести номер первого элемента, нарушающего закономерность.

**Листинг программы**

|  |
| --- |
| using Microsoft.VisualBasic;  using System;  namespace Zadanie1  {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("Введите длину массива ");  int n = int.Parse(Console.ReadLine());  int[] arr = new int[n];  Console.WriteLine("Массив");  Random rnd = new Random();  for (int i = 0; i < arr.Length; ++i)  {  arr[i] = rnd.Next(10);    }  Console.WriteLine(String.Join(" ", arr));  int j = 1;  while (j < arr.Length && (arr[j] - arr[j - 1]) % 2 != 0) ++j;  Console.WriteLine("\n{0}", j == arr.Length ? 0 : j + 1); // Слова не нужны. Только 0 или номер элемента.  //j + 1 потому, что порядковый номер (нумерация с 1, а индексация с 0)  Console.ReadKey(true);  }  }  } |

Проверка работы программы (рисунок 1).

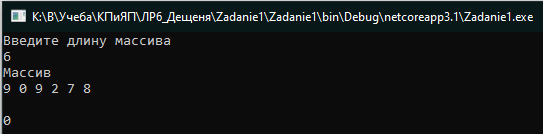


Рисунок 10 – Проверка работы программы

**Задание 2.** В матрице удалить строки, последние элементы которых отрицательны, а затем в качестве первой добавить строкуиз элементов заданного массива.

**Листинг программы**

|  |
| --- |
| using System;  using System.Globalization;  namespace Zadanie2  {  class Program  {    static void Main(string[] args)  {  Console.Write("Введите количество строк ");  int n = int.Parse(Console.ReadLine());  Console.Write("Введите количество столбцов ");  int m = int.Parse(Console.ReadLine());  int[,] x = new int [n,m];  int[,] y = new int[n, m];  int tempN = 0;    Random ran = new Random();    for (int i = 0; i < n; i++)  {  for (int j = 0; j < m; j++)  {  Console.ForegroundColor = ConsoleColor.Blue;  x[i, j] = ran.Next(-10, 5);  Console.Write("{0}\t", x[i, j]);  }  Console.WriteLine();  }  Console.ResetColor();  for (int i = 0; i < n; i++)  {  if (x[i, m - 1] > 0)  {  tempN++;  for (int j = 0; j < n; j++)  {  y[tempN, j] = x[i, j];  }    }    }  for (int j = 0; j < m; j++)  {  y[0, j] = y[ran.Next(1, tempN), j];  }  Console.ForegroundColor = ConsoleColor.Yellow;  Console.WriteLine("Измененная матрица");  for (int i = 0; i < tempN+1; i++)  {    for (int j = 0; j < m; j++)  {  Console.Write("{0}\t", y[i, j]);  }  Console.WriteLine();    }  Console.Write("\n");  Console.ResetColor();  Console.BackgroundColor = ConsoleColor.Blue;  Console.Write(" \n" +  " \n");  Console.ResetColor();  Console.BackgroundColor = ConsoleColor.Yellow;  Console.Write(" \n" +  " \n");  Console.ResetColor();  }  }  } |

Проверка работы программы (рисунок 2).

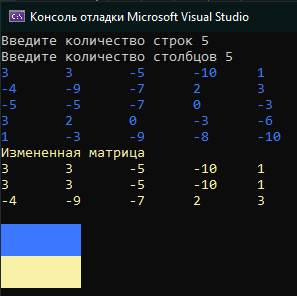


Рисунок 11 – Выполнение работы программы

**Вывод:** в результате выполнения лабораторной работы были получены практические навыки по работе с классом данных Array, его методами и свойствами.