**Министерство образования и науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Владимирский государственный университет**

**имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(ВлГУ)**

**Колледж инновационных технологий и предпринимательства**

**Лабораторная работа №3**

По дисциплине:

«Прикладное программирование»

Выполнил:

Ст. гр. ПКсп-115

Сорокин И.А.

Принял:

Преподаватель Куприянов А.А.

Шишкина М.В.

**Владимир 2018 г**

**Цель работы:** Закрепить навыки работы с одномерными массивами на языке С#, создание, инициализация, обращение к элементам, циклический перебор элементов. Закрепление знаний и совершенствование навыков создания приложений на языке C#, работы с компонентами отображения текстовой информации, управляющими компонентами.

**Теоретическая часть**

Массивом называют упорядоченную совокупность элементов одного типа. Каждый элемент массива имеет индексы, определяющие порядок элементов. Индексы задаются целочисленным типом. Число индексов характеризует размерность массива. Если конечное значение задано константным выражением, то число элементов массива известно в момент его объявления и ему может быть выделена память ещё на этапе трансляции. Такие массивы называются статическими. Если же конечное значение зависят от переменной, то такой массив называют динамическим, поскольку память ему может быть отведена только динамически в процессе выполнения программы, когда становятся известными значения соответствующих переменных. Массиву, как правило, выделяется непрерывная область памяти.

В языке C# каждый индекс изменяется в диапазоне от 0 до некоторого конечного значения. Массивы в языке C# являются настоящими динамическими массивами. Как следствие этого, массивы относятся к ссылочным типам, память им отводится динамически в "куче". Массивы могут быть одномерными и многомерными.

Объявление одномерного массива выглядит следующим образом:

<тип>[] <объявители>;

Как и в случае объявления простых переменных, каждый объявитель может быть именем или именем с инициализацией. В первом случае речь идёт об отложенной инициализации. Нужно понимать, что при объявлении с отложенной инициализацией сам массив не формируется, а создаётся только ссылка на массив, имеющая неопределённое значение Null. Поэтому пока массив не будет реально создан и его элементы инициализированы, использовать его в вычислениях нельзя!!! Вот пример объявления трёх массивов с отложенной инициализацией:

int[] a, b, c;

Чаще всего при объявлении массива используется имя с инициализацией. И опять-таки, как и в случае простых переменных, могут быть два варианта инициализации. В первом случае инициализация является явной и задаётся константным массивом. Вот пример:

double[] x = {5.5, 6.6, 7.7};

Следуя синтаксису, элементы константного массива следует заключать в фигурные скобки. Если массив инициализируется константным массивом, то в динамической памяти создаётся константный массив с заданными значениями, с которым и связывается ссылка.

Во втором случае создание массива выполняется с помощью операции new. Вот пример:

int[] d = new int[5];

Здесь объявлен динамический целочисленный массив, в котором будут храниться 5 целых чисел. Массив создаётся в динамической памяти, его элементы получают начальные нулевые значения, и ссылка связывается с этим массивом.

**Практическая часть**

**Задание 1**

В массив A[N] занесены натуральные числа. Найти сумму тех элементов, которые кратны данному К (Рис. 1).

**Задание 2**

Дана последовательность целых чисел а1, а2,..., аn. Выяснить, какое число встречается раньше — положительное или отрицательное (Рис. 2).

**Задание 3**

Дана последовательность действительных чисел а1, а2,..., аn. Выяснить, будет ли она возрастающей (Рис. 3).

**Задание 4**

Дана последовательность натуральных чисел а1, а2,..., аn. Вычислить сумму четных чисел этой последовательности (не индексов). Если таких чисел нет, то вывести сообщение об этом факте (Рис. 4).

Задание №1

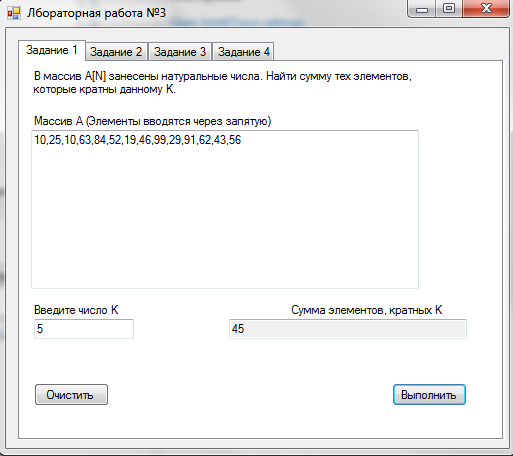


Рисунок 1 – Задание 1

Фрагмент кода:

privatevoid button1\_Click(object sender, EventArgs e)

{

txtbxResK.Clear();

string[] splitMas = txtbxMasA.Text.Split(',');

int[] nMas = newint[splitMas.Length];

int i = 0, sumOnK = 0, K;

foreach (string item in splitMas)

{

if (!int.TryParse(item, out nMas[i]))

{

txtbxResK.Text = "Ошибкаданных!";

return;

}

if (nMas[i] < 0)

{

txtbxResK.Text = "Встречено не натуральное число!";

return;

}

i++;

}

if (int.TryParse(txtbxK.Text, out K))

{

foreach (int item in nMas)

{

if (item % K == 0) sumOnK += item;

}

txtbxResK.Text = sumOnK.ToString();

}

else

{

txtbxResK.Text = "Ошибка данных!";

}

}

Задание №2

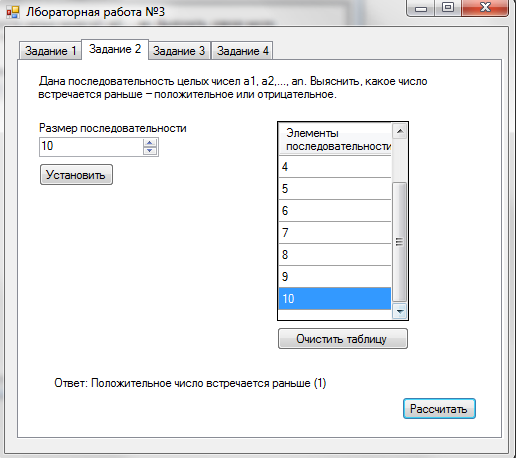


Рисунок 2 – Задание 2

Фрагмент кода:

privatevoid buttonRs\_Click(object sender, EventArgs e)

{

dataGrMasAN.RowCount = Convert.ToInt32(nmrcRowCount.Value);

}

privatevoid buttonRes2\_Click(object sender, EventArgs e)

{

lbRes2.Text = "Ответ: ";

int[] t2mas = newint[Convert.ToInt32(dataGrMasAN.Rows.Count)];

bool flag = true, uno = true; int i = 0;

for (i = 0; i < t2mas.Length; i++)

{

if (dataGrMasAN.Rows[i].Cells[0].Value == null)

{

lbRes2.Text = "Существует пустая строка!";

return;

}

if (int.TryParse(dataGrMasAN.Rows[i].Cells[0].Value.ToString(), out t2mas[i]))

{

if(t2mas[i]!=0)

{

uno = false;

}

if (t2mas[i] < 0 && t2mas[i] != 0)

{

flag = false;

break;

}

if (t2mas[i] > 0 && t2mas[i] != 0)

{

flag = true;

break;

}

}

else

{

lbRes2.Text = "Ошибкаданных!";

return;

}

}

if (flag && !uno)

{

lbRes2.Text += "Положительноечисловстречаетсяраньше (" + t2mas[i].ToString() + ")";

}

if(!flag && !uno)

{

lbRes2.Text += "Отрицательноечисловстречаетсяраньше (" + t2mas[i].ToString() + ")";

}

if(uno)

{

lbRes2.Text += "Вся последовательность состоит из нулей";

}

}

Задание №3

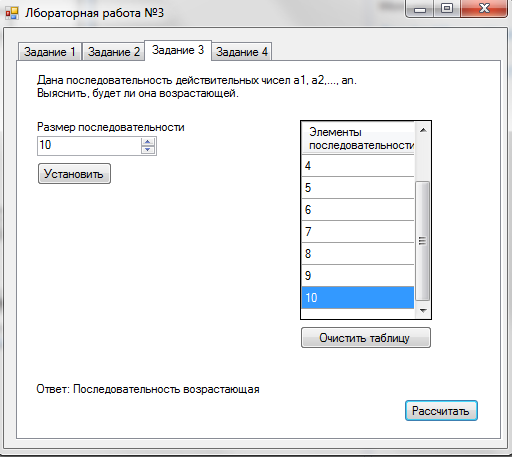


Рисунок 3 – Задание 3

Фрагмент кода:

privatevoid buttonRes3\_Click(object sender, EventArgs e)

{

lbRes3.Text = "Ответ: ";

double[] t3mas = newdouble[Convert.ToInt32(dataGrMasANT3.Rows.Count)];

bool flag = true;

for (int i = 0; i < t3mas.Length; i++)

{

if (dataGrMasANT3.Rows[i].Cells[0].Value == null)

{

lbRes3.Text = "Существует пустая строка!";

return;

}

if (!(double.TryParse(dataGrMasANT3.Rows[i].Cells[0].Value.ToString(), out t3mas[i]))) //t3mas[i] = Convert.ToDouble(dataGrMasANT3.Rows[i].Cells[0].Value);

{

lbRes3.Text = "Ошибкаданных!";

return;

}

}

for (int i = 0; i < t3mas.Length - 1; i++)

{

if (t3mas[i] > t3mas[i + 1])

{

flag = false;

break;

}

}

if (flag)

{

lbRes3.Text += "Последовательность возрастающая";

}

else

{

lbRes3.Text += "Последовательность не является возрастающей";

}

}

Задание №4

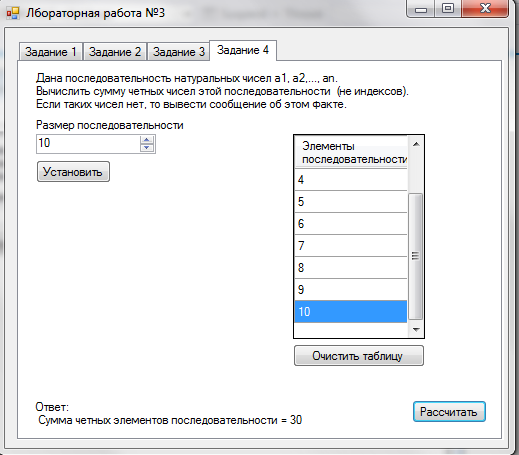


Рисунок 4 – Задание 4

Фрагмент кода:

privatevoid buttonRes4\_Click(object sender, EventArgs e)

{

lbRes4.Text = "Ответ: ";

int[] t4mas = newint[Convert.ToInt32(dataGrMasANT4.Rows.Count)];

int sumT4=0;

for (int i = 0; i < t4mas.Length; i++)

{

if (dataGrMasANT4.Rows[i].Cells[0].Value == null)

{

lbRes4.Text = "Существует пустая строка!";

return;

}

if (!(int.TryParse(dataGrMasANT4.Rows[i].Cells[0].Value.ToString(), out t4mas[i])))

{

lbRes4.Text = "Ошибкаданных!";

return;

}

if (t4mas[i] < 0)

{

lbRes4.Text = "Ошибкаданных!";

return;

}

if(t4mas[i]%2==0)

{

sumT4 += t4mas[i];

}

}

if(sumT4==0)

{

lbRes4.Text = "Четных чисел в последовательности нет!";

}

else

{

lbRes4.Text += "\n Сумма четных элементов последовательности = " + sumT4.ToString();

}

}

**Вывод:** В результате выполнения лабораторной работы были приобретены навыки работы с оконными приложениями Windows на языке C#. Были изучены элементы управления, позволяющие отображать и редактировать данные, создавать группированные списки и такие элементы, как кнопки и группы элементов с выбором. Также были рассмотрены свойства, возвращающие или задающие значения элементам управления и методы, преобразующие одни типы данных в другие. Были написаны и отлажены все представленные программы.