|  |  |
| --- | --- |
|  | /storage/emulated/0/.polarisOffice5/polarisTemp/image1.png |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования «Московский технологический университет»  МИРЭА | | |

Институт Информационных Технологий

Кафедра Корпоративных Информационных Систем

**ОТЧЕТ**

по Лабораторной Работе №4

на тему

«Тип данных - массив»

по дисциплине

«Процедурное Программирование»

Выполнил студент группы ИКБО-07-17 Акжигитов Р. Р.

Принял Асс. Каф. КИС Алдобаева В. Н.

Выполнено «13» ноября 2017 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_ 2017 г.

Москва, 2017

**Задание**

Разработать программу, обрабатывающую статический и динамический массив (по индивидуальному заданию). Сделать заполнение поэлементно путем ввода чисел через клавиатуру.

**Вариант 1**

1) Вычислить сумму элементов массива

2) Вычислить сумму 3-х максимальных элементов массива

3) Вычислить произведение элементов с четными индексами

4) Вычислить сумму элементов, больших среднего

**Теоретическая часть**

Массивы бывают нескольких видов: статический (использует фиксированное кол-во памяти (ячеек для хранения информации)) и динамический (имеет возможность после объявления длины, изменять длину массива).

В C# практически все массивы динамические, можно использовать статические, но только в небезопасном коде.

Массивы – простейшая составная структура данных, для хранения большого количества элементов, одинаковых по своему типу.

Простейшая операция над массивом – доступ к его элементу по индексу. Для доступа к ячейке памяти программе нужно знать ее адрес. В динамических массивах адресс начала массива хранится в специальносй переменной–указателе.

Основное преимущество массивом над набором переменных – возможность организовать цикличную обработку его элементов (например, через цикл for или foreach).

Объявление массива: **Type[] name = new Type[length]**

У динамического массива можно изменять длину с помощью **Array<T>.Resize()**.

В конкретно этой программе мы используем для статического массива фиксированную длину и никак ее не изменяем (10 элементов), а для динамического массива: создаем новый массив, длинной на 1 ячейку больше, копируем старый массив в новый (+1 длины), добавляем в новую ячейку нового массива новый элемент.

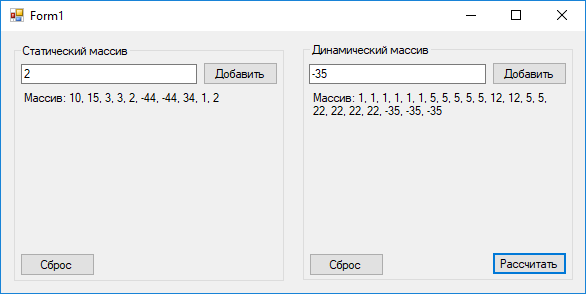
Одновременно с вводом данных в массив, выводим их в метку с помощью того же цикла for или же с помощью **String.Join(delim, array).**

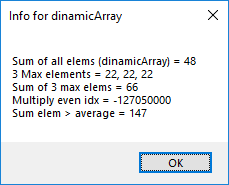
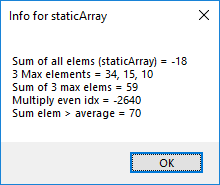
После переполнения статического массива можно осуществить полный сброс всех элементов и заполнять все с нуля, либо перезаполнять первые элементы, причем последние оставлять в сохранности.

**Алгоритм решения задачи**

1. Создать форму и создать разметку в конструкторе (все компоненты).
2. Создать статический и динамический массивы.
3. Написать циклы для обработки массивов в соответствии с индивидуальным вариантом.
4. Вывести результат обработки.

**Тестирование**





**Заключение**

Реализовав подобное приложение, я лучше разобрался в том, как работать с массивом: объявлять массив, заполнять его, вычислять суммы, средние, искать максимумы и т.д.

**Исходный код**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace laba4

{

public partial class Form1 : Form

{

int[] staticArray = new int[10];

int[] dinamicArray = { };

int number = 0;

int cxS = 0;

public Form1()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

if (int.TryParse(textBox2.Text, out number))

{

int[] tempArray = new int[dinamicArray.Length + 1];

dinamicArray.CopyTo(tempArray, 0); // Копируем старый массив в новый

dinamicArray = tempArray;

dinamicArray[dinamicArray.Length - 1] = number;

}

label2.Text = "Массив: " + string.Join(", ", dinamicArray);

}

private void button1\_Click(object sender, EventArgs e)

{

if (int.TryParse(textBox1.Text, out number))

{

if (cxS >= 10)

{

staticArray.Select(x => 0);

cxS = 0;

}

else

{

staticArray[cxS] = number;

cxS++;

}

}

label1.Text = "Массив: " + string.Join(", ", staticArray);

}

private void button3\_Click(object sender, EventArgs e)

{

label1.Text = "Массив: пуст";

staticArray = new int[10];

cxS = 0;

}

private void button4\_Click(object sender, EventArgs e)

{

label2.Text = "Массив: пуст";

dinamicArray = new int[0];

//Array.Clear(dinamicArray, 0, dinamicArray.Length);

}

public void showArrayInfo(int[] arr, string name)

{

MessageBox.Show(String.Format("Sum of all elems ({5}) = {0}\n3 Max elements = {1}\nSum of 3 max elems = {2}\nMultiply even idx = {3}\nSum elem > average = {4}",

arr.Sum(),

String.Join(", ", arr.OrderByDescending(x => x).Take(3)),

arr.OrderByDescending(x => x).Take(3).Sum(),

arr.Where((x, ind) => ind % 2 == 0).Aggregate(1, (acc, x) => acc \* x),

arr.Where(x => x > arr.Average()).Sum(),

name), String.Format("Info for {0}", name));

}

private void button5\_Click(object sender, EventArgs e)

{

showArrayInfo(staticArray, "staticArray");

showArrayInfo(dinamicArray, "dinamicArray");

}

}

}