

**Колледж космического машиностроения и технологий**

**ОТЧЕТ**

**по лабораторной работе**

**по дисциплине "Основы алгоритмизации и программирования"**

**по теме "Заполнение двумерного массива по шаблону. Сортировка элементов массива. Поиск элементов по заданному критерию"**

Студент: Буланов ***Евгений Дмитриевич***

Специальность: ***09.02.04***

" Информационные системы (по отраслям)"

Группа ***Ис1-19***

Преподаватель: Соколова Н.В.

Оценка за выполнение работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка за оформление работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Королев, 2021 г.**

**Тема:**

Заполнение двумерного массива по шаблону. Сортировка элементов массива. Поиск элементов по заданному критерию.

**Текст задачи:**

Array39. Дан массив размера N. Найти количество его промежутков монотонности (то есть участков, на которых его элементы возрастают или убывают).

**Сценарий работы приложения:**

Программа должна найти количество промежутков монотонности, то есть участки возрастания и убывания.

**Краткие сведения по теме:**

Раздел задач **Array предоставляет задачи для создания, изменения, поиска и сортировки массивов и содержит подразделы, в каждом из которых находятся задачи определенного типа:**

Формирование массива и вывод его элементов

## Анализ элементов массива

## Работа с несколькими массивами

## Преобразование массива

## Серии целых чисел

## Множества точек на плоскости

## Данная задача относится к разделу преобразование массива. Для ввода и вывода массивов массива необходимо использовать таблицу DataGridView.

**Описание переменных, описание начальной формы в конструкторе**

int m = переменная, которой нужна для создание таблицы DataGridView. Так же обозначает количество столбцов.

int q = переменная, которой нужна для создание таблицы DataGridView.

int n = переменная, которой нужна для создание таблицы DataGridView. В начале ей присваивается значение 1.

int k = переменная, которая необходима для создания таблицы DataGridView.

Int G = счетчик, которая считает количество промежутков монотонности

int i = переменная, необходимая для заполнения Tab1. Обозначает порядковый номер элемента Tab1 .

int j переменная, необходимая для заполнения Tab1. Обозначает порядковый номер элемента Tab1 .

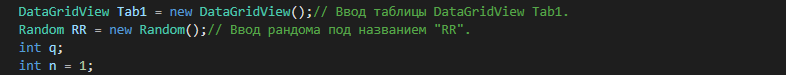
int[] Mas =переменная, необходимая для создания массива а также для заполнения Tab1.

bool f = переменная, которая проверяет элементы на возрастание и убывания

На начальной форме Form1.cs расположены: pictureBox1 для вывода картинки с задачей, label1 для надписи, button1 для старта выполнения программы.

**Описание алгоритма работы приложения**

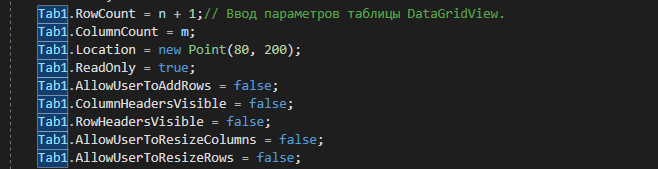
В начале работы вводим таблицу DataGridView под именем “Tab1”, вводим рандом под именем “RR”, а также вводим переменную q и n

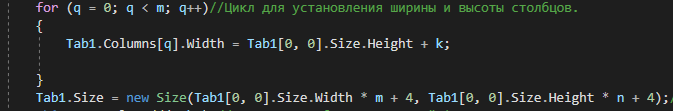


Придаём значение m



Вводим параметры для таблицы Tab1



Устанавливаем ширину и высоту колонок, а также устанавливаем размер колонок

Создаем таблицу по заданным параметрам



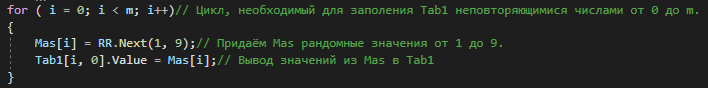
Вводим массив с названием “Mas” (Кол-во столбцов – m)



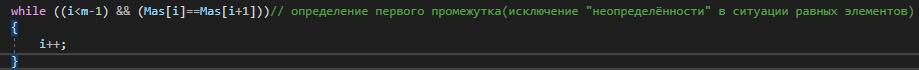
Вводим значения



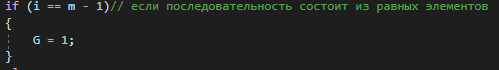
Заполняем таблицу



Определяем первый промежуток



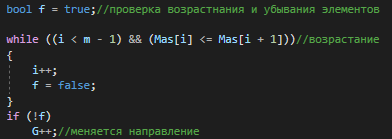
Вводим проверку, если в последовательности появляются одинаковы элементы



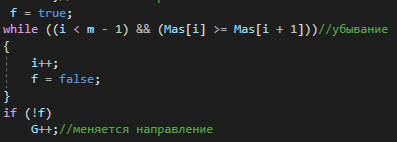
Вводим проверку выхода границы массива



Вводим проверку возрастания элементов



Вводим проверку убывания элементов



Выводим количество промежутков на Form1 через label1



**Код программы**

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 array39

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Button1\_Click(object sender, EventArgs e)

{

DataGridView Tab1 = new DataGridView();// Ввод таблицы DataGridView Tab1.

Random RR = new Random();// Ввод рандома под названием "RR".

int q;

int n = 1;

int m = RR.Next(5,10);// Ввод случайного количества столбцов

int k = 0;

Tab1.RowCount = n + 1;// Ввод параметров таблицы DataGridView.

Tab1.ColumnCount = m;

Tab1.Location = new Point(80, 200);

Tab1.ReadOnly = true;

Tab1.AllowUserToAddRows = false;

Tab1.ColumnHeadersVisible = false;

Tab1.RowHeadersVisible = false;

Tab1.AllowUserToResizeColumns = false;

Tab1.AllowUserToResizeRows = false;

for (q = 0; q < m; q++)//Цикл для установления ширины и высоты столбцов.

{

Tab1.Columns[q].Width = Tab1[0, 0].Size.Height + k;

}

Tab1.Size = new Size(Tab1[0, 0].Size.Width \* m + 4, Tab1[0, 0].Size.Height \* n + 4);// Установка размера рамки для таблицы.

this.Controls.Add(Tab1);// Создание таблицы по данной форме.

int[] Mas = new int[m];// Ввод массива под именем "Mas" с кол-вом столбцов m.

int G = 0;//введение счетчика промежутков

int i = 0;//введение счетчика столбцов

for ( i = 0; i < m; i++)// Цикл, необходимый для заполения Tab1 неповторяющимися числами от 0 до m.

{

Mas[i] = RR.Next(1, 9);// Придаём Mas рандомные значения от 1 до 9.

Tab1[i, 0].Value = Mas[i];// Вывод значений из Mas в Tab1

}

i = 0;

while ((i<m-1) && (Mas[i]==Mas[i+1]))// определение первого промежутка(исключение "неопределённости" в ситуации равных элементов)

{

i++;

}

if (i == m - 1)// если последовательность состоит из равных элементов

{

G = 1;

}

else

{

while (i < m - 1)// пока не вышли за границы массива

{

bool f = true;//проверка возрастнания и убывания элементов

while ((i < m - 1) && (Mas[i] <= Mas[i + 1]))//возрастание

{

i++;

f = false;

}

if (!f)

G++;//меняется направление

f = true;

while ((i < m - 1) && (Mas[i] >= Mas[i + 1]))//убывание

{

i++;

f = false;

}

if (!f)

G++;//меняется направление

}

}

label1.Text = "Колличество промежутков= " + G.ToString();

}

}

}