**Тема 5:** Обработка одномерных массивов

**Цель работы:** овладеть основными приемами работы с таблицами для эффективной обработки массивов

***Задание 1:***

Разработать программу обработки одномерного массива по заданию своего варианта. Ввод исходных данных должен осуществляться пользователем с клавиатуры или случайно. Размеры таблицы должны меняться интерактивно в определённых пределах.

***Задание 2:***

Добавьте в проект проверку нажатых клавиш в процессе редактирования каждой ячейки таблицы. Запретить ввод недопустимых символов при редактировании в ручном режиме.

Добавьте в проект возможность загрузить исходные данные из текстового файла. При этом должно вызываться диалоговое окно openFileDialog.

*Указания:*

Таблица представляет собой специальный компонент, позволяющие отображать данные в виде строк и столбцов. Компонент **dataGridView** позволяет хранить и отображать текстовую и графическую информацию. Но хранение и отображение данных выполняется программистом. Есть возможность подключить источник данных, которые будут представляться в таблице.

**Вариант 2**

Заполнить одномерный массив целыми числами. Вычислить произведение элементов массива, кратных своим порядковым номерам, а затем удалить эти элементы из массива. Удаление элементов производиться путем сдвига вперёд. Если таких элементов нет, то вывести сообщение об этом.

Код:

using System;

using System.Windows.Forms;

using System.IO;

namespace WindowsFormsApp5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

Random r = new Random();

for (int i = 0; i < numberOfElements.Value; i++)

{

gridOfNumbers.Columns.Add(Convert.ToString(i), Convert.ToString(i));

if (gridOfNumbers.RowCount == 0)

gridOfNumbers.Rows.Add();

gridOfNumbers.Rows[0].Cells[i].Value = r.Next(-100, 100);

}

}

private void numberOfElements\_ValueChanged(object sender, EventArgs e)

{

if (gridOfNumbers.ColumnCount < numberOfElements.Value)

{

Random r = new Random();

for (int i = gridOfNumbers.ColumnCount; i < numberOfElements.Value; i++)

{

gridOfNumbers.Columns.Add(Convert.ToString(i), Convert.ToString(i));

if (gridOfNumbers.RowCount == 0)

gridOfNumbers.Rows.Add();

gridOfNumbers.Rows[0].Cells[i].Value = r.Next(-100, 100);

}

}

else

{

for (int i = gridOfNumbers.ColumnCount; i > numberOfElements.Value; i--)

gridOfNumbers.Columns.RemoveAt(i - 1);

}

}

private void numberOfElements\_KeyUp(object sender, KeyEventArgs e)

{

numberOfElements\_ValueChanged(sender, e);

}

private void gridOfNumbers\_EditingControlShowing(object sender, DataGridViewEditingControlShowingEventArgs e)

{

e.Control.KeyPress += new KeyPressEventHandler(gridOfNumbers\_KeyPress);

}

private void gridOfNumbers\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && e.KeyChar != '-')

{

e.Handled = true;

}

if (!string.IsNullOrEmpty((sender as TextBox).Text) && e.KeyChar == '-')

{

e.Handled = true;

}

}

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

{

int mul = 1;

int count = 0;

for (int i = 1; i < numberOfElements.Value - count; i++)

{

if (Convert.ToInt32(gridOfNumbers.Rows[0].Cells[i].Value) % (i + count) == 0)

{

count++;

mul \*= Convert.ToInt32(gridOfNumbers.Rows[0].Cells[i].Value);

gridOfNumbers.Columns.RemoveAt(i);

i--;

}

}

numberOfElements.Value = gridOfNumbers.ColumnCount;

if (count != 0)

numberOfChange.Text = Convert.ToString(mul);

else

numberOfChange.Text = "нет кратных чисел";

}

private void byHandsCheck\_CheckedChanged(object sender, EventArgs e)

{

if (byHandsCheck.Checked == false)

{

gridOfNumbers.ReadOnly = true;

Random r = new Random();

for (int i = 0; i < numberOfElements.Value; i++)

gridOfNumbers.Rows[0].Cells[i].Value = r.Next(-100, 100);

}

else

{

gridOfNumbers.ReadOnly = false;

}

}

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

{

OpenFileDialog dialog = new OpenFileDialog();

dialog.InitialDirectory = "c:\\";

dialog.Filter = "txt files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

dialog.FilterIndex = 2;

dialog.RestoreDirectory = true;

var fileContent = string.Empty;

if (dialog.ShowDialog() == DialogResult.OK)

{

var fileStream = dialog.OpenFile();

using (StreamReader reader = new StreamReader(fileStream))

{

fileContent = reader.ReadToEnd();

}

string[] grid = fileContent.Split(' ');

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

{

gridOfNumbers.Columns.Add(Convert.ToString(i), Convert.ToString(i));

if (gridOfNumbers.RowCount == 0)

gridOfNumbers.Rows.Add();

gridOfNumbers.Rows[0].Cells[i].Value = grid[i];

}

numberOfElements.Value = grid.Length;

}

}

}

}

Вид формы (рис. 5.1):

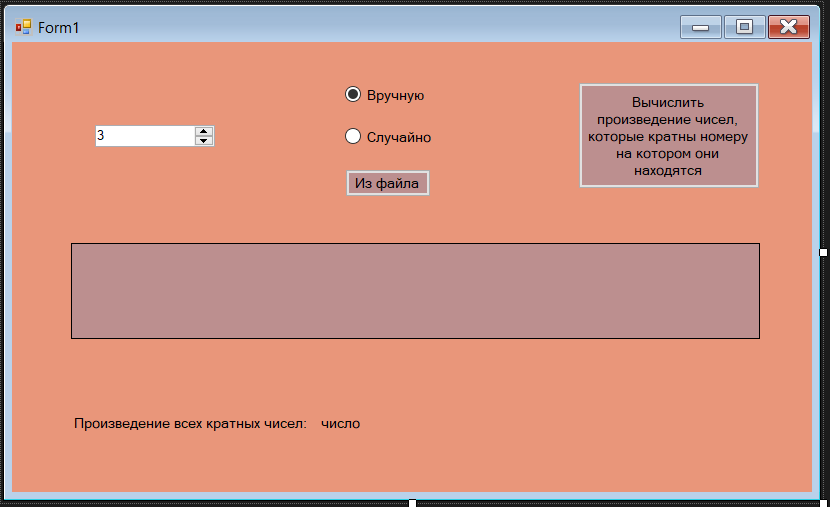


Рисунок 5.1 – Форма

Значение свойств (табл. 1):

|  |  |  |
| --- | --- | --- |
| **Название компонента** | **Свойства** | **Значения** |
| Form1 | Text | «» |
| BackColor | DarkSalmon |
| Font | Гарнитура: Microsoft Sans Serif |
| Размер: 7, 8 |
| Начертание: обычный |
| Label1 | Text | «Произведение всех кратных чисел::» |
| numberOfChange | Font | Цвет: DarkSalmon |
| Text | «число» |
| byFileButton | Click | byFileButton\_Click |
| Text | «Из файла» |
| numberOfElements | ValueChanged | numberOfElements\_ValueChanged |
| KeyUp | numberOfElements\_KeyUp |
| byHandsCheck | Text | «Вручную» |
| CheckedChanged | byHandsCheck\_\_CheckedChanged |
| byRandomCheck | Text | «Случайно» |
| Enabled | True |
| gridOfNumbers | Text | «» |
| countButton | Text | «Вычислить произведение чисел, которые кратны номеру на котором они находятся» |
| Click | countButton\_Click |

Таблица 1

Протокол испытаний (табл. 2):

|  |  |  |  |
| --- | --- | --- | --- |
| **Проверяемые требования** | **Сообщения программы и вводимые значения** | **Ожидаемые результаты** | **Фактические результаты** |
| Способность программы обеспечить ввод исходных данных вручную. | Ввести в поля ввода массив (в каждое поле вещественное число без знака с разделителем запятая или точка) | В поле ввода можно ввести целое или вещественное число без знака, можно стирать символы в поле ввода, используя клавишу BackSpace. | Соответствуют ожиданиям.  Рисунок 5.2 |
| Способность программы обеспечить ввод исходных данных случайно. | Ввести в поля ввода массив (в каждое поле вещественное число без знака с разделителем запятая или точка) | В поле ввода можно ввести целое или вещественное число без знака, можно стирать символы в поле ввода, используя клавишу BackSpace. | Соответствуют ожиданиям.  Рисунок 5.3 |
| Способность программы обеспечить ввод исходных данных из файла | Ввести в поля ввода массив (в каждое поле вещественное число без знака с разделителем запятая или точка) | В поле ввода можно ввести целое или вещественное число без знака, можно стирать символы в поле ввода, используя клавишу BackSpace. | Соответствуют ожиданиям.  Рисунок 5.4 |
| Проверить способность программы вычислять произведение чисел, которые кратны номеру на котором они находятся. | Ввести в поля ввода массив числе (в каждое поле вещественное число без знака с разделителем запятая или точка) | Вывод количества смены знаков | Сообщение выводится.  (см. Рисунок 5.4) |

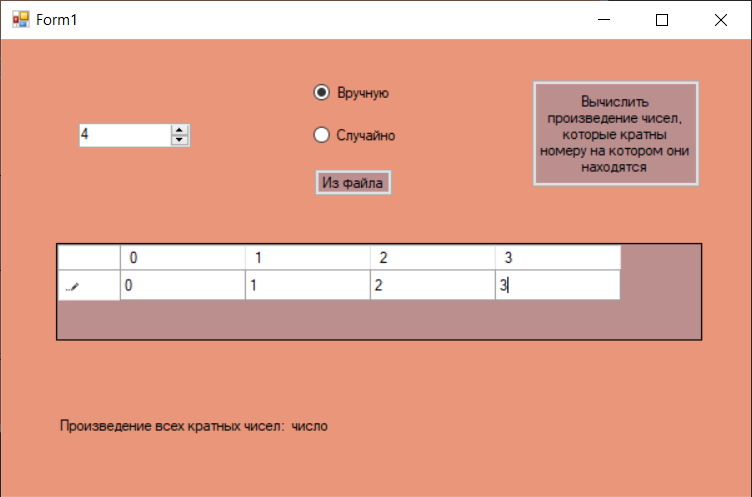


Рисунок 5.2 – Вручную

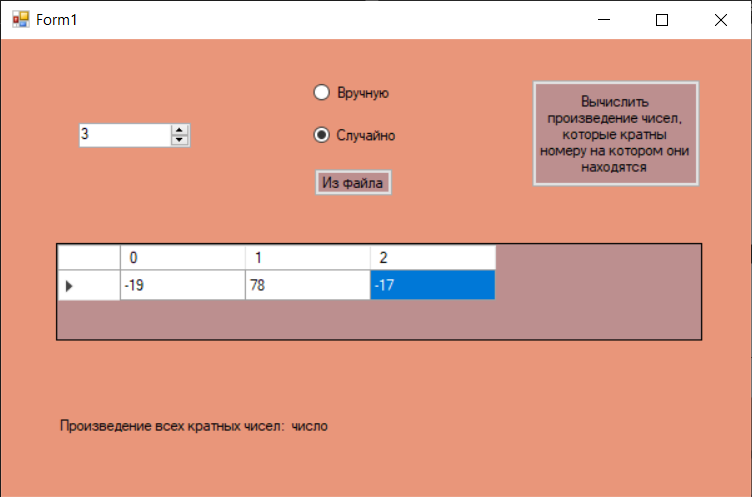


Рисунок 5.3 – Случайно

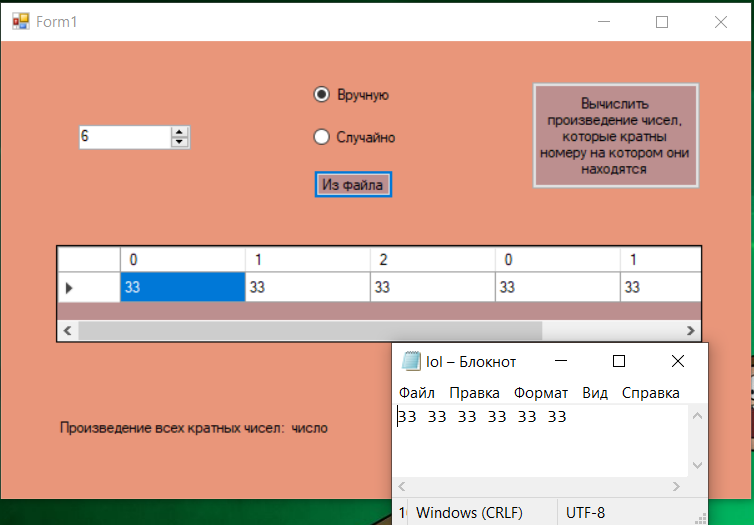


Рисунок 5.4 – Из файла

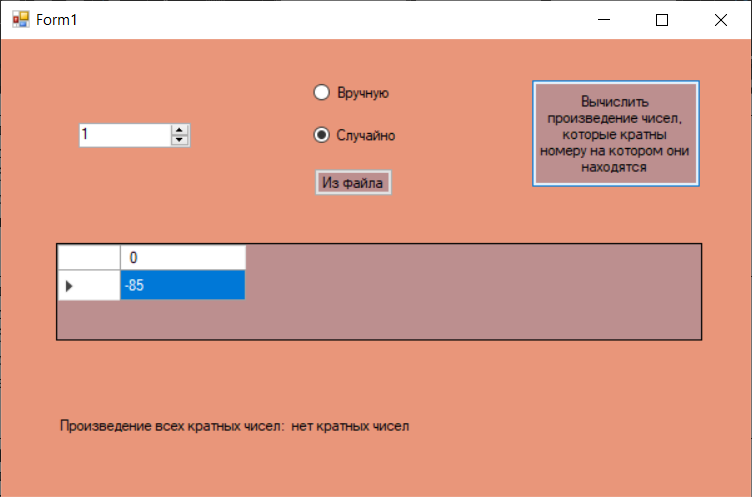


Рисунок 5.4 – Проверка