Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа №7

По дисциплине

“Объектно-ориентированное программирование”

# Тема “Разработка Windows-приложений”

Вариант 15

Выполнил работу

студент группы РИС-19-1б

Миннахметов Э.Ю.

Проверила

доцент кафедры ИТАС

Викентьева О.Л.

Работу выполнил:

Пермь 2020

**1 АНАЛИЗ ЗАДАЧИ**

**1.1 Постановка задачи**

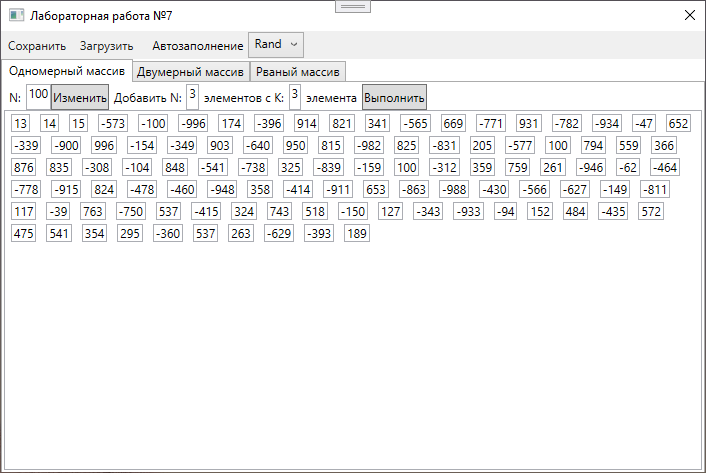
1. Разработать Windows – приложение, реализующее работу с одномерными, двумерными и рваными массивами.
2. Реализовать сохранение массивов в файле и загрузку ранее сохраненного массива из файла.

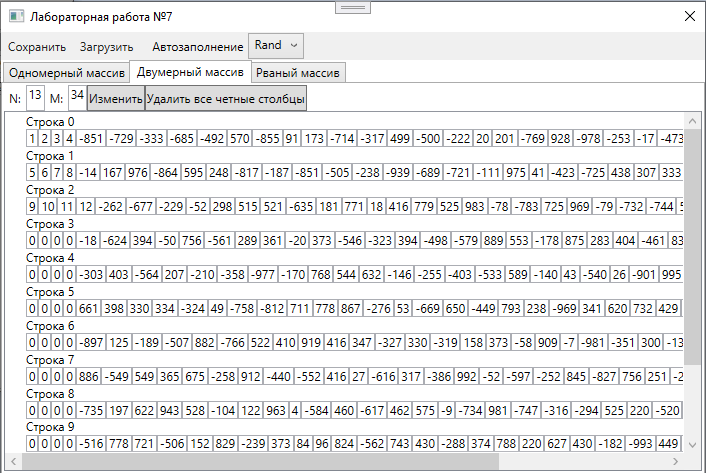
|  |  |  |
| --- | --- | --- |
| Одномерный массив | Двумерный массив | Рваный массив |
| Добавить N элементов, начиная с номера K | Удалить все четные строки | Добавить строку в конец массива |

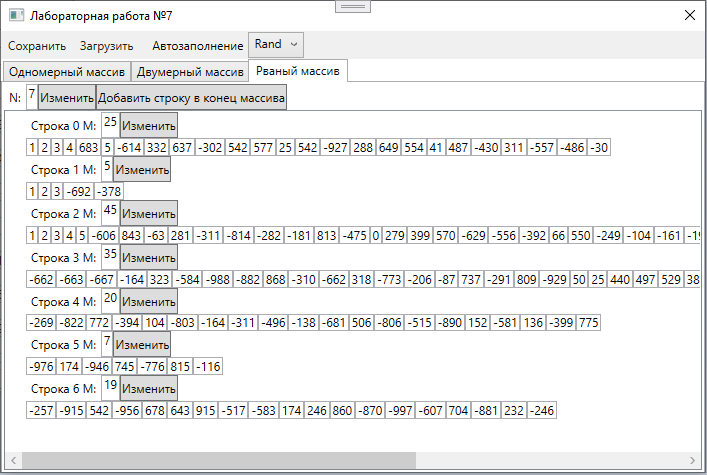
**1.2 Требуемые ннструменты**

Для реализации графического интерфейса будет использован Microsoft Windows Presentation Foundation (WPF), поскольку является в настоящее время более актуальной технологией.

**2 ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА**







**3 ИСХОДНЫЙ КОД ПРОГРАММЫ**

**3.1 Exceptions.cs**

using System;

namespace Lab7

{

class IncorrectIndex : Exception { }

class IncorrectNewSize : Exception { }

class IncorrectValue : Exception { }

class AlreadySet : Exception { }

class EmptyArray : Exception { }

class IncorrectField : Exception

{

public int I { get; set; }

}

class IncorrectFields : Exception

{

public int N { get; set; }

public int M { get; set; }

}

}

**3.2 Array1D.cs**

namespace Lab7

{

class Array1D

{

private static int[] s\_iArray

= new int[] { 13, 14, 15 };

private int[] m\_iArray;

public Array1D()

=> m\_iArray = s\_iArray;

public Array1D(int size)

=> m\_iArray = new int[size];

public int Length

{

get => m\_iArray.Length;

}

public int this[int index]

{

get

{

if (index < 0 || index >= Length)

throw new IncorrectIndex();

return m\_iArray[index];

}

set

{

if (index < 0 || index >= Length)

throw new IncorrectIndex();

m\_iArray[index] = value;

}

}

public void Resize(int newSize, GetNumber GetNum)

{

if (newSize < 0)

throw new IncorrectNewSize();

int oldSize = Length;

if (oldSize == newSize)

throw new AlreadySet();

int[] temp = m\_iArray;

m\_iArray = new int[newSize];

if (oldSize > newSize)

for (int i = 0; i < newSize; ++i)

m\_iArray[i] = temp[i];

else

{

for (int i = 0; i < oldSize; ++i)

m\_iArray[i] = temp[i];

for(int i = oldSize; i < newSize; ++i)

GetNum(out m\_iArray[i]);

}

}

public void AddLine(int n, int k, GetNumber GetNum)

{

int oldSize = m\_iArray.Length;

int newSize = oldSize + n;

if (k < 0 || k > oldSize || n <= 0)

throw new IncorrectValue();

int[] temp = m\_iArray;

m\_iArray = new int[newSize];

for (int i = 0; i < k; ++i)

m\_iArray[i] = temp[i];

for (int i = k, j = k + n; i < j; ++i)

GetNum(out m\_iArray[i]);

for (int i = k + n; i < newSize; ++i)

m\_iArray[i] = temp[i - n];

}

}

}

**3.3 Array2D.cs**

namespace Lab7

{

class Array2D

{

private static int[,] s\_iArray = new int[3, 4] {

{ 1, 2, 3, 4 },

{ 5, 6, 7, 8 },

{ 9, 10, 11, 12 }

};

private int[,] m\_iArray;

public Array2D()

=> m\_iArray = s\_iArray;

public Array2D(int n, int m)

=> m\_iArray = new int[n, m];

public int N

{

get => m\_iArray.GetUpperBound(0) + 1;

}

public int M

{

get

{

if (N == 0)

return 0;

return m\_iArray.Length / N;

}

}

public int this[int i, int j]

{

get

{

if (i < 0 || j < 0 || i >= N || j >= M)

throw new IncorrectIndex();

return m\_iArray[i, j];

}

set

{

if (i < 0 || j < 0 || i >= N || j >= M)

throw new IncorrectIndex();

m\_iArray[i, j] = value;

}

}

public void Resize(int newN, int newM, GetNumber GetNum)

{

if (newN < 0 || newM < 0)

throw new IncorrectNewSize();

int oldN = N;

int oldM = M;

if (oldN == newN && oldM == newM)

throw new AlreadySet();

int[,] temp = m\_iArray;

m\_iArray = new int[newN, newM];

if (oldN > newN)

for (int i = 0; i < newN; i++)

ResizeRow(i, oldM, newM, temp, GetNum);

else

{

for (int i = 0; i < oldN; ++i)

ResizeRow(i, oldM, newM, temp, GetNum);

for (int i = oldN; i < newN; ++i)

for (int j = oldM; j < newM; ++j)

GetNum(out m\_iArray[i, j]);

}

}

private void ResizeRow(int index, int oldM, int newM, int[,] old, GetNumber GetNum)

{

if (oldM > newM)

for (int i = 0; i < newM; i++)

m\_iArray[index, i] = old[index, i];

else

{

for (int i = 0; i < oldM; ++i)

m\_iArray[index, i] = old[index, i];

for (int i = oldM; i < newM; ++i)

GetNum(out m\_iArray[index, i]);

}

}

public void DropEven()

{

int[,] temp = m\_iArray;

int n = N;

int k = M;

if (n == 0 && k == 0)

throw new EmptyArray();

int m = n / 2;

m\_iArray = new int[m, k];

for (int i = 0; i < m; ++i)

for (int j = 0; j < k; ++j)

m\_iArray[i, j] = temp[i \* 2 + 1, j];

}

}

}

**3.4 Array2J.cs**

namespace Lab7

{

class Array2J

{

private static int[][] s\_iArray = new int[][] {

new int[] { 1, 2, 3, 4 },

new int[] { 1, 2, 3 },

new int[] { 1, 2, 3, 4, 5 }

};

private int[][] m\_iArray;

public Array2J()

=> m\_iArray = s\_iArray;

public Array2J(int size)

=> m\_iArray = new int[size][];

public int Length

{

get => m\_iArray.Length;

}

public int[] this[int index]

{

get

{

if (index < 0 || index >= Length)

throw new IncorrectIndex();

return m\_iArray[index];

}

set

{

if (index < 0 || index >= Length)

throw new IncorrectIndex();

m\_iArray[index] = value;

}

}

public int this[int i, int j]

{

get

{

if (i < 0 || j < 0 || i >= m\_iArray.Length || j >= m\_iArray[i].Length)

throw new IncorrectIndex();

return m\_iArray[i][j];

}

set

{

if (i < 0 || j < 0 || i >= m\_iArray.Length || j >= m\_iArray[i].Length)

throw new IncorrectIndex();

m\_iArray[i][j] = value;

}

}

public void Resize(int newSize)

{

if (newSize < 0)

throw new IncorrectNewSize();

int oldSize = Length;

if (oldSize == newSize)

throw new AlreadySet();

int[][] tempArray = m\_iArray;

m\_iArray = new int[newSize][];

if (oldSize > newSize)

for (int i = 0; i < newSize; ++i)

m\_iArray[i] = tempArray[i];

else

{

for (int i = 0; i < oldSize; ++i)

m\_iArray[i] = tempArray[i];

for (int i = oldSize; i < newSize; ++i)

m\_iArray[i] = new int[0];

}

}

public void ResizeByIndex(int index, int newSize, GetNumber GetNum)

{

if(newSize < 0)

throw new IncorrectField { I = index };

int oldSize = m\_iArray[index].Length;

if (oldSize == newSize)

throw new AlreadySet();

int[] tempArray = m\_iArray[index];

m\_iArray[index] = new int[newSize];

if (oldSize > newSize)

for (int i = 0; i < newSize; ++i)

m\_iArray[index][i] = tempArray[i];

else

{

for (int i = 0; i < oldSize; ++i)

m\_iArray[index][i] = tempArray[i];

for (int i = oldSize; i < newSize; ++i)

GetNum(out m\_iArray[index][i]);

}

}

public void AddLine()

{

int[][] temp = m\_iArray;

int n = temp.Length;

m\_iArray = new int[n + 1][];

for (int i = 0; i < n; ++i)

m\_iArray[i] = temp[i];

m\_iArray[n] = new int[0];

}

}

}

**3.5 Kernel.cs**

using System;

using System.IO;

namespace Lab7

{

delegate void GetNumber(out int num);

static class Kernel

{

private static Random s\_Rand = new Random();

public static GetNumber GetRand = (out int num) => num = s\_Rand.Next(-999, 999);

public static GetNumber GetNull = (out int num) => num = 0;

private static void SetByteArray(byte[] array, int pos, int num)

{

byte[] numA = BitConverter.GetBytes(num);

pos \*= 4;

for (int i = 0; i < 4; ++i)

array[pos + i] = numA[i];

}

private static int IntFromByteArray(byte[] array, int pos)

=> BitConverter.ToInt32(array, pos \* 4);

public static void SaveToFile(string file, Array1D array1D, Array2D array2D, Array2J array2J)

{

using (FileStream fstream = new FileStream(file, FileMode.OpenOrCreate))

{

int Sum = 16,

n1d = array1D.Length,

n2d = array2D.N,

m2d = array2D.M,

n2j = array2J.Length;

int[] m2j = new int[n2j];

for (int i = 0; i < n2j; ++i)

{

m2j[i] = array2J[i].Length;

Sum += m2j[i] \* 4;

}

Sum += (n1d + n2d + m2d + n2j) \* 4;

byte[] A = new byte[Sum];

SetByteArray(A, 0, n1d);

SetByteArray(A, 1, n2d);

SetByteArray(A, 2, m2d);

SetByteArray(A, 3, n2j);

for (int i = 0; i < n2j; ++i)

SetByteArray(A, 4 + i, m2j[i]);

for (int i = 0; i < n1d; ++i)

SetByteArray(A, 4 + n2j + i, array1D[i]);

for (int i = 0; i < n2d; ++i)

for (int j = 0; j < m2d; ++j)

SetByteArray(A, 4 + n2j + n1d + i + j, array2D[i, j]);

for (int i = 0; i < n2j; ++i)

for (int j = 0; j < m2j[i]; ++j)

SetByteArray(A, 4 + n2j + n1d + n2d + m2d + i + j, array2J[i][j]);

fstream.Write(A, 0, Sum);

fstream.Close();

}

}

public static void LoadFromFile(string file, out Array1D array1D, out Array2D array2D, out Array2J array2J)

{

using (FileStream fstream = new FileStream(file, FileMode.OpenOrCreate))

{

byte[] A = new byte[fstream.Length];

fstream.Read(A, 0, A.Length);

int n1d = IntFromByteArray(A, 0);

int n2d = IntFromByteArray(A, 1);

int m2d = IntFromByteArray(A, 2);

int n2j = IntFromByteArray(A, 3);

int[] m2j = new int[n2j];

for (int i = 0; i < n2j; ++i)

m2j[i] = IntFromByteArray(A, 4 + i);

array1D = new Array1D(n1d);

array2D = new Array2D(n2d, m2d);

array2J = new Array2J(n2j);

for (int i = 0; i < n1d; ++i)

array1D[i] = IntFromByteArray(A, 4 + n2j + i);

for (int i = 0; i < n2d; ++i)

for (int j = 0; j < m2d; ++j)

array2D[i, j] = IntFromByteArray(A, 4 + n2j + n1d + i + j);

for (int i = 0; i < n2j; ++i)

{

array2J[i] = new int[m2j[i]];

for (int j = 0; j < m2j[i]; ++j)

array2J[i, j] = IntFromByteArray(A, 4 + n2j + n1d + n2d + m2d + i + j);

}

fstream.Close();

}

}

}

}

**3.6 MainWindow.xaml.cs**

using System.Windows;

using Microsoft.Win32;

using System.Windows.Media;

using System.Windows.Controls;

namespace Lab7

{

public partial class MainWindow : Window

{

private Array1D m\_Array1D;

private Array2D m\_Array2D;

private Array2J m\_Array2J;

private GetNumber GetNum;

public MainWindow()

{

InitializeComponent();

m\_Array1D = new Array1D();

m\_Array2D = new Array2D();

m\_Array2J = new Array2J();

GetNum = Kernel.GetNull;

SetArray1D();

SetArray2D();

SetArray2J();

}

private void ComboBoxNullSelected(object sender, RoutedEventArgs e)

=> GetNum = Kernel.GetNull;

private void ComboBoxRandSelected(object sender, RoutedEventArgs e)

=> GetNum = Kernel.GetRand;

private void ResizeArray1D(object sender, RoutedEventArgs e)

{

try

{

FixArray1D();

if (int.TryParse(Array1DSize.Text, out int size))

m\_Array1D.Resize(size, GetNum);

else

throw new IncorrectNewSize();

SetArray1D();

}

catch (IncorrectNewSize)

{

MessageBox.Show($"Некорректное значение поля ввода размера!");

}

catch (AlreadySet)

{

MessageBox.Show($"Заданная размерность уже установлена.");

}

}

private void ResizeArray2D(object sender, RoutedEventArgs e)

{

try

{

FixArray2D();

if (int.TryParse(Array2SizeN.Text, out int n) &&

int.TryParse(Array2SizeM.Text, out int m))

m\_Array2D.Resize(n, m, GetNum);

else

throw new IncorrectNewSize();

SetArray2D();

}

catch (IncorrectNewSize)

{

MessageBox.Show($"Некорректные значения полей ввода размера!");

}

catch (AlreadySet)

{

MessageBox.Show($"Заданная размерность уже установлена.");

}

}

private void ResizeArray2J(object sender, RoutedEventArgs e)

{

try

{

FixArray2J();

if (int.TryParse(Array2JSize.Text, out int n))

m\_Array2J.Resize(n);

else

throw new IncorrectNewSize();

SetArray2J();

}

catch (IncorrectNewSize)

{

MessageBox.Show($"Некорректное значение поля ввода размера!");

}

catch (AlreadySet)

{

MessageBox.Show($"Заданная размерность уже установлена.");

}

}

private void ResizeArray2JByIndex(object sender, RoutedEventArgs e)

{

try

{

FixArray2J();

string sIndex = (sender as Button).Name.Remove(0, 1);

int.TryParse(sIndex, out int n);

TextBox tempTextBox = FindChild<TextBox>(Application.Current.MainWindow, "t" + n);

if (int.TryParse(tempTextBox.Text, out int m))

m\_Array2J.ResizeByIndex(n, m, GetNum);

else

throw new IncorrectField { I = n };

SetArray2J();

}

catch (IncorrectField ex)

{

MessageBox.Show($"Некорректное значение поля [{ex.I}]!");

}

catch (AlreadySet)

{

MessageBox.Show($"Заданная размерность уже установлена.");

}

}

private void ChangeArray1DField(object sender, RoutedEventArgs e)

=> FixArray1D();

private void ChangeArray2DField(object sender, RoutedEventArgs e)

=> FixArray2D();

private void ChangeArray2JField(object sender, RoutedEventArgs e)

=> FixArray2J();

private void FixArray1D()

{

try

{

for (int i = 0, n = m\_Array1D.Length; i < n; ++i)

{

TextBox tempTextBox = FindChild<TextBox>(Application.Current.MainWindow, "A1D" + i);

if (int.TryParse(tempTextBox.Text, out int num))

m\_Array1D[i] = num;

else

throw new IncorrectField { I = i };

}

}

catch (IncorrectField ex)

{

MessageBox.Show($"Некорректное значение поля [{ex.I}]!");

}

}

private void FixArray2D()

{

try

{

int n = m\_Array2D.N;

if (n == 0)

return;

int m = m\_Array2D.M;

for (int i = 0; i < n; ++i)

for (int j = 0; j < m; ++j)

{

TextBox tempTextBox = FindChild<TextBox>(Application.Current.MainWindow, "A2D" + i + "\_" + j);

if (int.TryParse(tempTextBox.Text, out int num))

m\_Array2D[i, j] = num;

else

throw new IncorrectFields { N = i, M = j };

}

}

catch (IncorrectFields ex)

{

MessageBox.Show($"Некорректное значение поля [{ex.N}, {ex.M}]!");

}

}

private void FixArray2J()

{

try

{

for (int i = 0, n = m\_Array2J.Length; i < n; ++i)

for (int j = 0, m = m\_Array2J[i].Length; j < m; ++j)

{

TextBox tempTextBox = FindChild<TextBox>(Application.Current.MainWindow, "AJG" + i + "\_" + j);

if (int.TryParse(tempTextBox.Text, out int num))

m\_Array2J[i][j] = num;

else

throw new IncorrectFields { N = i, M = j };

}

}

catch (IncorrectFields ex)

{

MessageBox.Show($"Некорректное значение поля [{ex.N}][{ex.M}]!");

}

}

private void AddLineArray1D(object sender, RoutedEventArgs e)

{

try

{

if (int.TryParse(Array1DN.Text, out int n) && int.TryParse(Array1DK.Text, out int k))

m\_Array1D.AddLine(n, k, GetNum);

else

throw new IncorrectValue();

SetArray1D();

}

catch (IncorrectValue)

{

MessageBox.Show($"Некорректное значение поля!");

}

}

private void DropEvenArray2D(object sender, RoutedEventArgs e)

{

try

{

m\_Array2D.DropEven();

SetArray2D();

}

catch (EmptyArray)

{

MessageBox.Show($"Массив уже пуст.");

}

}

private void AddLineArray2J(object sender, RoutedEventArgs e)

{

m\_Array2J.AddLine();

SetArray2J();

}

private void SaveToFile(object sender, RoutedEventArgs e)

{

SaveFileDialog dialog = new SaveFileDialog();

dialog.Filter = "Документ ARR (\*.arr)|\*.arr";

if (dialog.ShowDialog() == true)

Kernel.SaveToFile(dialog.FileName, m\_Array1D, m\_Array2D, m\_Array2J);

}

private void LoadFromFile(object sender, RoutedEventArgs e)

{

OpenFileDialog dialog = new OpenFileDialog();

dialog.Filter = "Документ ARR (\*.arr)|\*.arr";

if (dialog.ShowDialog() == true)

{

Kernel.LoadFromFile(dialog.FileName, out m\_Array1D, out m\_Array2D, out m\_Array2J);

SetArray1D();

SetArray2D();

SetArray2J();

}

}

private void SetArray1D()

{

ListBoxArray1D.Items.Clear();

for (int i = 0, n = m\_Array1D.Length; i < n; ++i)

{

TextBox textBox = new TextBox { Name = "A1D" + i, Text = m\_Array1D[i].ToString() };

textBox.TextChanged += ChangeArray1DField;

ListBoxArray1D.Items.Add(new ListBoxItem { Content = textBox });

}

Array1DSize.Text = m\_Array1D.Length.ToString();

}

private void SetArray2D()

{

TreeViewArray2D.Items.Clear();

int n = m\_Array2D.N;

int m = m\_Array2D.M;

for (int i = 0; i < n; ++i)

{

WrapPanel tempWrapPanel = new WrapPanel();

TreeViewItem tempItem = new TreeViewItem { Header = "Строка " + i };

TreeViewArray2D.Items.Add(tempItem);

TreeViewArray2D.Items.Add(tempWrapPanel);

for (int j = 0; j < m; ++j)

{

string tempText = m\_Array2D[i, j].ToString();

TextBox tempTextBox = new TextBox { Name = "A2D" + i + "\_" + j, Text = tempText };

tempTextBox.TextChanged += ChangeArray2DField;

tempWrapPanel.Children.Add(tempTextBox);

}

}

Array2SizeN.Text = n.ToString();

Array2SizeM.Text = m.ToString();

}

private void SetArray2J()

{

TreeViewArray2J.Items.Clear();

int n = m\_Array2J.Length;

for (int i = 0; i < n; ++i)

{

int m = m\_Array2J[i].Length;

WrapPanel tempWrapPanel = new WrapPanel();

Label tempLabel = new Label { Content = $"Строка {i} M:" };

TextBox tempTextBox = new TextBox { Name = "t" + i, Text = m.ToString() };

Button tempButton = new Button { Name = "b" + i, Content = "Изменить" };

StackPanel tempStackPanel = new StackPanel { Orientation = Orientation.Horizontal };

TreeViewItem tempItem = new TreeViewItem { Name = "i" + i, Header = tempStackPanel };

tempButton.Click += ResizeArray2JByIndex;

tempStackPanel.Children.Add(tempLabel);

tempStackPanel.Children.Add(tempTextBox);

tempStackPanel.Children.Add(tempButton);

TreeViewArray2J.Items.Add(tempItem);

TreeViewArray2J.Items.Add(tempWrapPanel);

for (int j = 0; j < m; ++j)

{

string tempText = m\_Array2J[i][j].ToString();

TextBox tempEdit = new TextBox { Name = "AJG" + i + "\_" + j, Text = tempText };

tempEdit.TextChanged += ResizeArray2JByIndex;

tempWrapPanel.Children.Add(tempEdit);

}

}

Array2JSize.Text = n.ToString();

}

private static T FindChild<T>(DependencyObject parent, string childName)

where T : DependencyObject

{

if (parent == null) return null;

T foundChild = null;

int childrenCount = VisualTreeHelper.GetChildrenCount(parent);

for (int i = 0; i < childrenCount; i++)

{

var child = VisualTreeHelper.GetChild(parent, i);

T childType = child as T;

if (childType == null)

{

foundChild = FindChild<T>(child, childName);

if (foundChild != null) break;

}

else if (!string.IsNullOrEmpty(childName))

{

var frameworkElement = child as FrameworkElement;

if (frameworkElement != null && frameworkElement.Name == childName)

{

foundChild = (T)child;

break;

}

}

else

{

foundChild = (T)child;

break;

}

}

return foundChild;

}

}

}

**3.7 MainWindow.xaml**

<Window x:Class="Lab7.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

Title="Лабораторная работа №7"

Height="480" Width="720"

ResizeMode="NoResize">

<StackPanel>

<Menu>

<MenuItem Click="SaveToFile" Header="Сохранить" />

<MenuItem Click="LoadFromFile" Header="Загрузить" />

<MenuItem>

<MenuItem.Header>

<StackPanel Orientation="Horizontal">

<Label>Автозаполнение</Label>

<ComboBox>

<ComboBoxItem Selected="ComboBoxNullSelected" IsSelected="True">Null</ComboBoxItem>

<ComboBoxItem Selected="ComboBoxRandSelected">Rand</ComboBoxItem>

</ComboBox>

</StackPanel>

</MenuItem.Header>

</MenuItem>

</Menu>

<TabControl>

<TabItem Header="Одномерный массив">

<StackPanel>

<StackPanel Orientation="Horizontal">

<Label>N:</Label>

<TextBox Name="Array1DSize"/>

<Button Click="ResizeArray1D">Изменить</Button>

<Label>Добавить N:</Label>

<TextBox Name="Array1DN" Text="3" />

<Label>элементов с K:</Label>

<TextBox Name="Array1DK" Text="3" />

<Label>элемента</Label>

<Button Click="AddLineArray1D">Выполнить</Button>

</StackPanel>

<ListBox Name="ListBoxArray1D"

ScrollViewer.HorizontalScrollBarVisibility="Disabled"

Height="360">

<ListBox.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel />

</ItemsPanelTemplate>

</ListBox.ItemsPanel>

</ListBox>

</StackPanel>

</TabItem>

<TabItem Header="Двумерный массив">

<StackPanel>

<StackPanel Orientation="Horizontal">

<Label>N:</Label>

<TextBox Name="Array2SizeN" />

<Label>M:</Label>

<TextBox Name="Array2SizeM" />

<Button Click="ResizeArray2D">Изменить</Button>

<Button Click="DropEvenArray2D">Удалить все четные столбцы</Button>

</StackPanel>

<TreeView Name="TreeViewArray2D"

ScrollViewer.HorizontalScrollBarVisibility="Auto"

ScrollViewer.VerticalScrollBarVisibility="Auto"

ScrollViewer.CanContentScroll="True"

Height="360" />

</StackPanel>

</TabItem>

<TabItem Header="Рваный массив">

<StackPanel>

<StackPanel Orientation="Horizontal">

<Label>N:</Label>

<TextBox Name="Array2JSize" />

<Button Name="ChangeArray2J" Click="ResizeArray2J">Изменить</Button>

<Button Click="AddLineArray2J">Добавить строку в конец массива</Button>

</StackPanel>

<TreeView Name="TreeViewArray2J"

ScrollViewer.HorizontalScrollBarVisibility="Auto"

ScrollViewer.VerticalScrollBarVisibility="Auto"

ScrollViewer.CanContentScroll="True"

Height="360" />

</StackPanel>

</TabItem>

</TabControl>

</StackPanel>

</Window>

**4 ТЕСТИРОВАНИЕ**

Тесты эквивалентны тестам 6-ой лабораторной работы.