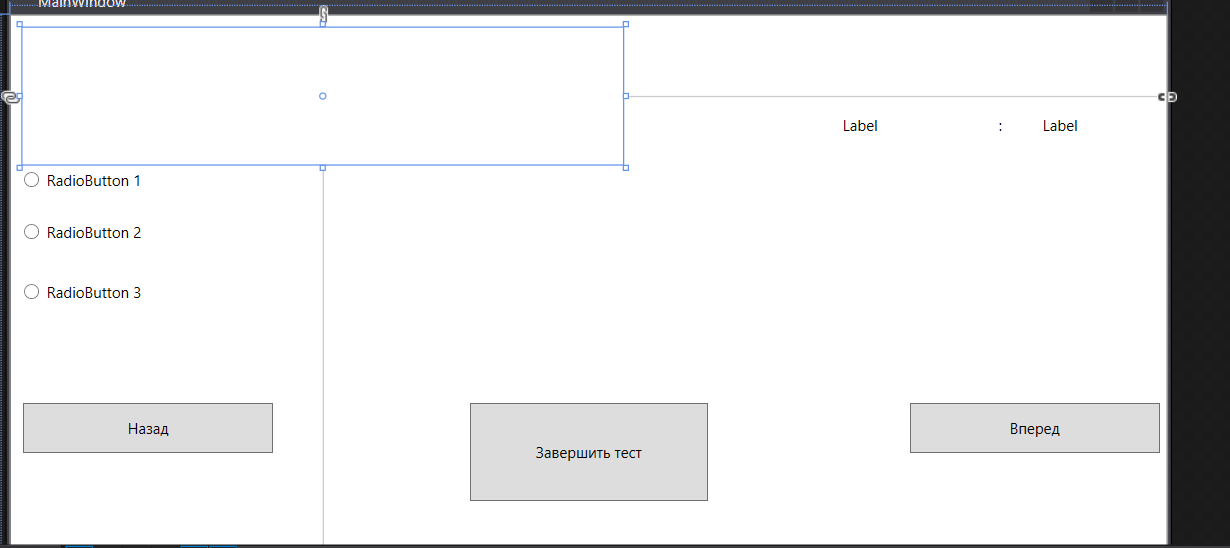
Адигамов Даниль 09-922

**Задание №2 WPF**

Задача:

Написание и реализация WPF приложения, которое сможет считывать вопросы из xml файла, выводить их на экран с возможностью выбора ответов на эти вопросы. После ответа на все вопросы, должен выводиться определенный результат. Также, приложение должно содержать таймер, по окончанию которого приложение должно завершаться и выводить результат.

Структура:



Приложение содержит:

* TextBox для вывода основного вопроса
* 3 RadioButton для выбора вариантов ответов
* 3 Button кнопки для переключения между вопросами и завершения теста
* 3 label для реализации таймера и вывода его на экран

Конструктор проекта:

<Window x:Class="WPF\_2.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"

xmlns:local="clr-namespace:WPF\_2"

mc:Ignorable="d"

Title="MainWindow" Height="488" Width="931">

<Grid Margin="0,0,0,-6" x:Name="mainRoot">

<TextBlock x:Name="text1" HorizontalAlignment="Left" Margin="10,10,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="110" Width="480"/>

<StackPanel x:Name="panel" Margin="10,125,611,221">

<RadioButton x:Name="radio1" Content="RadioButton 1" Height="42"/>

<RadioButton x:Name="radio2" Content="RadioButton 2" Height="48"/>

<RadioButton x:Name="radio3" Content="RadioButton 3" Height="44"/>

</StackPanel>

<Button x:Name="button2" Content="Вперед" HorizontalAlignment="Left" Margin="719,310,0,0" VerticalAlignment="Top" Height="40" Width="200" Click="buttonNext\_click"/>

<Button x:Name="button1" Content="Назад" HorizontalAlignment="Left" Margin="10,310,0,0" VerticalAlignment="Top" Height="40" Width="200" Click="buttonPrev\_Click"/>

<Button x:Name="button\_end" Content="Завершить тест" HorizontalAlignment="Center" Margin="0,310,0,0" VerticalAlignment="Top" Height="78" Width="190" Click="button\_end\_click"/>

<Label x:Name="lab1" Content="Label" HorizontalAlignment="Left" Margin="661,75,0,0" VerticalAlignment="Top" Width="100" Height="45" RenderTransformOrigin="1.521,0.452"/>

<Label x:Name="lab2" Content="Label" HorizontalAlignment="Left" Margin="821,75,0,0" VerticalAlignment="Top" Height="45" Width="100"/>

<Label Content=":" HorizontalAlignment="Left" Margin="786,75,0,0" VerticalAlignment="Top" Height="45"/>

</Grid>

</Window>

Вспомогательный класс:

Класс Data потребуется нам для структурирования вопросов из xml файла

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WPF\_2

{

class Data

{

private string question, answ1, answ2, answ3;

private int rightAnsw;

public string Question { get { return question; } set { question = value; } }

public string Answ1 { get { return answ1; } set { answ1 = value; } }

public string Answ2 { get { return answ2; } set { answ2 = value; } }

public string Answ3 { get { return answ3; } set { answ3 = value; } }

public int RightAnsw { get { return rightAnsw; } set { rightAnsw = value; } }

public Data() { }

public Data(string question, string answ1, string answ2, string answ3, int rightAnsw)

{

this.question = question;

this.answ1 = answ1;

this.answ2 = answ2;

this.answ3 = answ3;

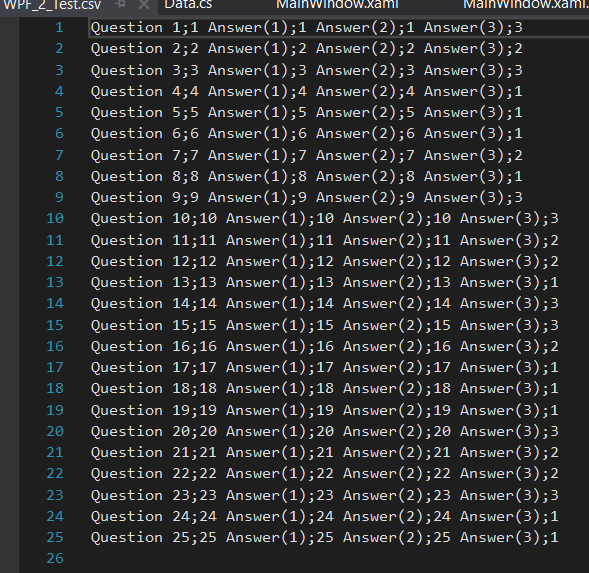
this.rightAnsw = rightAnsw;

}

}

}

Xml файл мы переведем в csv для упрощения парсинга. Он будет содержать:



Main класс:

using Microsoft.VisualBasic.FileIO;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Windows.Threading;

namespace WPF\_2

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

//**Для начала нам потребуется dataList(Который будет содержать все наши данные с вопросами).**

**//currPos будет показывать на каком вопросе мы находимся**

**//массив choosenAnsw будет хранить выбранные пользователем варианты ответов**

**//time время таймера соответственно**

private List<Data> dataList;

internal List<Data> DataList { get => dataList; set => dataList = value; }

public int currPos;

public int[] choosenAnsw;

public int time = 120;

DispatcherTimer timer = new DispatcherTimer();

public MainWindow()

{

InitializeComponent();

**//Парсим файл и добавляем данные в dataList**

List<Data> list = new List<Data>();

using (TextFieldParser parser = new TextFieldParser(@"C:\Users\Даниль\source\repos\WPF\_2\WPF\_2\WPF\_2\_Test.csv"))

{

parser.TextFieldType = FieldType.Delimited;

parser.SetDelimiters(";");

while (!parser.EndOfData)

{

string[] rows = parser.ReadFields();

list.Add(new Data(rows[0], rows[1], rows[2], rows[3], int.Parse(rows[4])));

}

}

DataList = list;

currPos = 0;

choosenAnsw = new int[list.Count];

setQuestion(0);

timer.Start();

timer.Tick += new EventHandler(timer\_tick);

timer.Interval = new TimeSpan(0, 0, 1);

}

**//Метод для тика таймера и его вывода на экрарн**

private void timer\_tick(object sender, EventArgs e)

{

time--;

lab1.Content = (time/60).ToString();

lab2.Content = (time - time/60\*60).ToString();

if (time == 0)

{

getCounter();

}

}

**//Метод, который сохраняет выбранный ответ и показывает предыдущий вопрос**

private void buttonPrev\_Click(object sender, RoutedEventArgs e)

{

choosenAnsw[currPos] = radioChecker();

if (currPos != 0)

{

currPos--;

setQuestion(currPos);

}

}

**//Аналогичный метод, только показывает следующий вопрос**

private void buttonNext\_click(object sender, RoutedEventArgs e)

{

choosenAnsw[currPos] = radioChecker();

if (currPos != DataList.Count -1)

{

currPos++;

setQuestion(currPos);

}

}

**//Завершает приложение и показывает результат**

private void button\_end\_click(object sender, RoutedEventArgs e)

{

getCounter();

}

private void getCounter()

{

choosenAnsw[currPos] = radioChecker();

int count = 0;

for (int i = 0; i < DataList.Count; i++)

{

if (DataList[i].RightAnsw == choosenAnsw[i])

{

count++;

}

}

MessageBox.Show("Правильных ответов: " + count + " из " + DataList.Count);

}

**//Проверяет radioButtons и возвращает число от 1 до 3 в зависимости от выбранного варианта**

private int radioChecker()

{

int numb = 0;

if (radio1.IsChecked == true)

{

numb = 1;

}

if (radio2.IsChecked == true)

{

numb = 2;

}

if (radio3.IsChecked == true)

{

numb = 3;

}

return numb;

}

**//Метод выводит вопрос и варианты ответа на экран, используя данные из DataList**

private void setQuestion(int pos)

{

text1.Text = DataList[pos].Question;

radio1.Content = DataList[pos].Answ1;

radio2.Content = DataList[pos].Answ2;

radio3.Content = DataList[pos].Answ3;

if (choosenAnsw[pos] != 0)

{

switch(choosenAnsw[pos])

{

case 1:

radio1.IsChecked = true;

radio2.IsChecked = false;

radio3.IsChecked = false;

break;

case 2:

radio2.IsChecked = true;

radio1.IsChecked= false;

radio3.IsChecked= false;

break;

case 3:

radio3.IsChecked = true;

radio1.IsChecked = false;

radio2.IsChecked= false;

break;

}

}

else

{

radio1.IsChecked = false;

radio2.IsChecked = false;

radio3.IsChecked = false;

}

}

}

}

Вывод:

Нам удалось реализовать приложение используя WPF.Net. Таким образом мы научились парсить данные из файла, применять их в приложении, анализировать полученный результат, также использовать таймер.