Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Казанский (Приволжский) федеральный университет»

*Институт вычислительной математики и информационных технологий*

**ОТЧЕТ**

**По Лабораторной работе**

Обучающийся Уваров А.В. 09-913

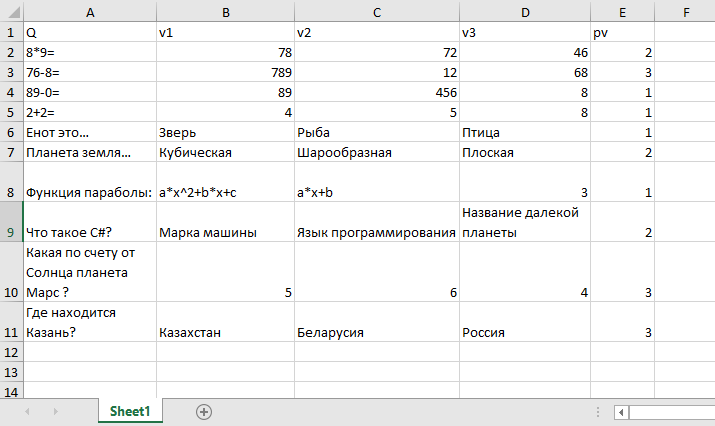
Дата сдачи отчета 21.03.2022

Казань, 2020

Цель: создать тест на время с помощью языка программирования C# и технологии WPF.

Алгоритм решения:

1. Создать Excel-файл, в котором располагаются вопросы с вариантами ответов:



1. Design

<Window x:Class="ExcelForm2.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:ExcelForm2"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" Initialized="Window\_Initialized">

<ScrollViewer VerticalScrollBarVisibility="Visible"

HorizontalScrollBarVisibility="Disabled"

HorizontalAlignment="Right" Width="780">

<StackPanel Name="MainPanel" Orientation="Vertical">

<StackPanel>

<Label x:Name="lblTime" FontSize="30" Content="Осталось времени:"

Width="288" HorizontalAlignment="Left"/>

<Label x:Name="Time" FontSize="30" Width="68" HorizontalAlignment="Left"/>

</StackPanel>

</StackPanel>

</ScrollViewer>

</Window>

1. Code

using System;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Threading;

using System.Windows.Media;

namespace ExcelForm2

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{//В конструкторе задаем таймер

public MainWindow()

{

InitializeComponent();

DispatcherTimer timer = new DispatcherTimer();

timer.Interval = TimeSpan.FromSeconds(1);

timer.Tick += timer\_Tick;

timer.Start();

Time.Content = "150";

}

static int countQ;

StackPanel[] Panels;

Object[,] Question;

int[] answers;

int[] userAnswers;

int ticks;

void timer\_Tick(object sender, EventArgs e)

{

ticks = Int32.Parse(Time.Content.ToString());

ticks = ticks - 1;

Time.Content = ticks.ToString();

if(ticks<=0)

{

BeforeCheckedMethod();

CheckedMethod();

}

}

private void Window\_Initialized(object sender, EventArgs e)

{

Initial();

}

private void Initial()

{

Microsoft.Office.Interop.Excel.Application excelApp

= new Microsoft.Office.Interop.Excel.Application();

Microsoft.Office.Interop.Excel.Workbook excelBook =

excelApp.Workbooks.Open(@"C:\Users\tolya\source\repos\ExcelForm2\InputForTest.xlsx", 0, true, 5, "", "", true,

Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "\t", false,

false, 0, true, 1, 0);

Microsoft.Office.Interop.Excel.Worksheet excelSheet =

(Microsoft.Office.Interop.Excel.Worksheet)excelBook.Worksheets.get\_Item(1);

Microsoft.Office.Interop.Excel.Range excelRange = excelSheet.UsedRange;

countQ = excelRange.Rows.Count - 1;

Panels = new StackPanel[countQ];

Question = new object[countQ, 4];

answers = new int[countQ];

userAnswers = new int[countQ];

for(var i=0;i<countQ;i++)

{

Panels[i] = new StackPanel();

Panels[i].Orientation = Orientation.Vertical;

Panels[i].Background = new SolidColorBrush(Colors.CadetBlue);

}

for(var i=0;i<countQ;i++)

{

Question[i, 0] = new Label();

(Question[i, 0] as Label).Content = Convert.ToString((excelRange.Cells[i + 2, 1] as Microsoft.Office.Interop.Excel.Range).Value2);

(Question[i, 0] as Label).FontSize = 20;

Panels[i].Children.Add(Question[i, 0] as Label);

for(var j=1;j<4;j++)

{

Question[i, j] = new RadioButton();

(Question[i,j] as RadioButton).Content=Convert.ToString((excelRange.Cells[i + 2, j+ 1] as Microsoft.Office.Interop.Excel.Range).Value2);

(Question[i, j] as RadioButton).FontSize = 20;

Panels[i].Children.Add(Question[i, j] as RadioButton);

}

answers[i]=Convert.ToInt32((excelRange.Cells[i + 2, 5] as Microsoft.Office.Interop.Excel.Range).Value2);

Panels[i].Margin = new Thickness(3, 5, 0, 0);

MainPanel.Children.Add(Panels[i]);

}

Button btn = new Button();

btn.Content = "Ok";

btn.Width = 85;

btn.Margin = new Thickness(10, 10, 10, 10);

btn.HorizontalAlignment = HorizontalAlignment.Right;

btn.Click += Button\_Click;

MainPanel.Children.Add(btn);

excelBook.Close(true, null, null);

excelApp.Quit();

}

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

{

BeforeCheckedMethod();

CheckedMethod();

}

private void BeforeCheckedMethod()

{

for(var i=0;i<countQ;i++)

{

for (var j = 1; j < 4; j++)

if ((Question[i, j] as RadioButton).IsChecked == true)

userAnswers[i] = j;

}

for (var i = 0; i < countQ; i++)

for (var j = 1; j < 4; j++)

(Question[i, j] as RadioButton).IsEnabled = false;

}

private void CheckedMethod()

{

int k = 0;

bool everyQIsAnswered = true;

for(var i=0;i<countQ;i++)

{

if(userAnswers[i]!=0)

{

if(userAnswers[i]==answers[i])

{

(Question[i, userAnswers[i]] as RadioButton).Foreground =

new SolidColorBrush(Colors.Green);

k++;

}

else

{

(Question[i, userAnswers[i]] as RadioButton).Foreground =

new SolidColorBrush(Colors.Red);

}

}

else

{

Panels[i].Background = new SolidColorBrush(Colors.Yellow);

everyQIsAnswered = false;

//(Question[i, userAnswers[i]] as RadioButton).Foreground =

// new SolidColorBrush(Colors.Green);

}

}

if(!everyQIsAnswered)

MessageBox.Show("You didn't answer on every question");

MessageBox.Show($"Correct answers: {k.ToString()}, time spent: {150-ticks} seconds");

}

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

{

BeforeCheckedMethod();

CheckedMethod();

}

}

}