**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

****

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет информационных технологий**

**Кафедра Информатики и информационных технологий**

**направление подготовки 09.03.02 «Информационные системы и технологии»,**

**профиль «Цифровая трансформация»**

**ЛАБОРАТОРНАЯ РАБОТА №3**

**Дисциплина:** Технологии прикладного программирования

**Выполнил:** студент группы 231-337

Сильченко Александр Алексеевич

**Дата, подпись** 11/02/2024  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверила:** Полубояринова А.С.\_\_\_\_\_\_\_\_\_\_\_

**(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2024**

Текст задачи

В рамках данной лабораторной работы необходимо дополнить проект калькулятор следующими элементами:

1. ListBox/TextBox (на выбор). Данный элемент должен использоваться для сохранения всех операций калькулятора. В случае TextBox указываются свойства TextWrapping="Wrap" AcceptsReturn="True"

2. CheckBox. Данный элемент отвечает за включение/выключения функции записи операций калькулятора в ListBox/TextBox

3. Button. Кнопка для полной очистки ListBox/TextBox

4. Button. Кнопка для удаления последней операции из ListBox/TextBox

5. Radiobutton (2 шт.). Для выбора режима видимости/невидимости элементов из пп. 1-4.

Код вёрстки окон и код программной логики

<Window x:Class="ЛР3.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:ЛР3"

mc:Ignorable="d"

Title="Калькулятор" Height="700" Width="480">

<Grid>

<Grid ShowGridLines="False">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Button x:Name="one" Grid.Column="0" Grid.Row="1" Content="1" FontSize="24" Click="Numbers"/>

<Button x:Name="two" Grid.Column="1" Grid.Row="1" Content="2" FontSize="24" Click="Numbers"/>

<Button x:Name="three" Grid.Column="2" Grid.Row="1" Content="3" FontSize="24" Click="Numbers"/>

<Button x:Name="four" Grid.Column="0" Grid.Row="2" Content="4" FontSize="24" Click="Numbers"/>

<Button x:Name="five" Grid.Column="1" Grid.Row="2" Content="5" FontSize="24" Click="Numbers"/>

<Button x:Name="six" Grid.Column="2" Grid.Row="2" Content="6" FontSize="24" Click="Numbers"/>

<Button x:Name="seven" Grid.Column="0" Grid.Row="3" Content="7" FontSize="24" Click="Numbers"/>

<Button x:Name="eight" Grid.Column="1" Grid.Row="3" Content="8" FontSize="24" Click="Numbers"/>

<Button x:Name="nine" Grid.Column="2" Grid.Row="3" Content="9" FontSize="24" Click="Numbers"/>

<Button x:Name="zero" Grid.Column="0" Grid.Row="4" Content="0" FontSize="24" Click="Numbers"/>

<Button x:Name="equals" Grid.Column="1" Grid.Row="4" Content="=" FontSize="24" Click="Result"/>

<Button x:Name="AC" Grid.Column="2" Grid.Row="4" Content="AC" FontSize="24" Click="del"/>

<Button x:Name="plus" Grid.Column="3" Grid.Row="2" Content="+" FontSize="24" Click="Action"/>

<Button x:Name="minus" Grid.Column="4" Grid.Row="2" Content="-" FontSize="24" Click="Action"/>

<Button x:Name="multi" Grid.Column="3" Grid.Row="3" Content="\*" FontSize="24" Click="Action"/>

<Button x:Name="division" Grid.Column="3" Grid.Row="4" Content="/" FontSize="24" Click="Action"/>

<Button x:Name="full\_del" Grid.Column="4" Grid.Row="3" Content="f-d" FontSize="24" Click="Full\_delete"/>

<Button x:Name="last\_del" Grid.Column="4" Grid.Row="4" Content="l-d" FontSize="24" Click="Last\_delete"/>

<RadioButton x:Name="On" Grid.Column="1" Grid.Row="5" Content="On" FontSize="24" Click="Activation"/>

<RadioButton x:Name="Off" Grid.Column="0" Grid.Row="5" Content="Off" FontSize="24" Click="Activation"/>

<CheckBox x:Name="Check\_box" Grid.Column="2" Grid.Row="5" Content="Отмена" FontSize="20"/>

<TextBlock x:Name="Input\_block" Grid.Column="0" Grid.ColumnSpan="3" Text="0" FontSize="26" TextWrapping="Wrap"/>

<ListBox x:Name="Input\_block\_1" Grid.Column="3" Grid.ColumnSpan="2" Grid.RowSpan="2" FontSize="26" />

</Grid>

</Grid>

</Window>

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;

namespace ЛР3

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

public float branches(string arg)

{

switch (arg.Substring(0, Math.Min(1, arg.Length)))

{

case "+":

meaning = meaning + meaning\_2;

break;

case "-":

meaning = meaning - meaning\_2;

break;

case "\*":

meaning = meaning \* meaning\_2;

break;

case "/":

if (meaning\_2 == 0)

{

break;

}

else

{

meaning = meaning / meaning\_2;

}

break;

}

return meaning;

}

float meaning = 0; float meaning\_2 = 0;

int count = 0;

bool view = true; bool view\_1 = true;

string action;

private void Numbers(object sender, RoutedEventArgs e)

{

if (count == 346784)

{

Input\_block.Text = "0"; count = 0;

}

// обнуление после '='

var data = sender as Button;

if (Input\_block.Text != "0" && view == true)

{

Input\_block.Text = Input\_block.Text + data.Content.ToString();

}

else

{

if (meaning != 0 && view\_1 == true)

{

Input\_block.Text = meaning.ToString() + " " + action + " " + data.Content.ToString();

view\_1 = false;

}

else if (meaning != 0 && view\_1 == false)

{

Input\_block.Text = Input\_block.Text + data.Content.ToString();

}

else

{

Input\_block.Text = data.Content.ToString();

view = true;

}

}

// работа с выводом

if (count == 0)

{

if (meaning == 0)

{

meaning = float.Parse(data.Content.ToString());

}

else

{

meaning = float.Parse(meaning.ToString() + data.Content.ToString());

}

}

else

{

if (meaning\_2 == 0)

{

meaning\_2 = float.Parse(data.Content.ToString());

}

else

{

meaning\_2 = float.Parse(meaning\_2.ToString() + data.Content.ToString());

}

}

// работа с переменными

}

private void del(object sender, RoutedEventArgs e)

{

Input\_block.Text = "0";

Input\_block\_1.Items.Clear();

meaning = 0; meaning\_2 = 0; count = 0;

}

private void Result(object sender, RoutedEventArgs e)

{

if (meaning\_2 == 0 && action == "/")

{

Input\_block.Text = "Здесь вам не пределы!";

}

else

{

if (Check\_box.IsChecked == false)

{

Input\_block\_1.Items.Add(meaning.ToString() + " " + action + " " + meaning\_2.ToString() + " = "

+ branches(action).ToString());

Input\_block.Text = meaning.ToString();

}

else

{

branches(action);

Input\_block.Text = meaning.ToString();

}

}

count = 346784;

meaning = 0; meaning\_2 = 0;

}

private void Action(object sender, RoutedEventArgs e)

{

var data = sender as Button;

if (count != 0)

{

if (meaning\_2 == 0 && action == "/")

{

Input\_block.Text = "Здесь вам не пределы!";

}

else

{

if (Check\_box.IsChecked == false)

{

Input\_block\_1.Items.Add(meaning.ToString() + " " + action + " " + meaning\_2.ToString() + " = "

+ branches(action).ToString());

Input\_block.Text = meaning.ToString();

}

else

{

branches(action);

Input\_block.Text = meaning.ToString();

}

}

meaning\_2 = 0;

view = false;

view\_1 = true;

}

else

{

Input\_block.Text = Input\_block.Text + " " + data.Content.ToString() + " ";

}

action = data.Content.ToString();

count++;

}

private void Full\_delete(object sender, RoutedEventArgs e)

{

Input\_block\_1.Items.Clear();

}

private void Last\_delete(object sender, RoutedEventArgs e)

{

int last = Input\_block\_1.Items.Count - 1;

if (last > 0)

{

Input\_block\_1.Items.RemoveAt(last);

}

}

private void Activation(object sender, RoutedEventArgs e)

{

RadioButton radioButton = sender as RadioButton;

if (radioButton.Content.ToString() == "Off")

{

Input\_block\_1.Visibility = Visibility.Hidden;

full\_del.Visibility = Visibility.Hidden; last\_del.Visibility = Visibility.Hidden;

Check\_box.Visibility = Visibility.Hidden;

}

else

{

Input\_block\_1.Visibility = Visibility.Visible;

full\_del.Visibility = Visibility.Visible; last\_del.Visibility = Visibility.Visible;

Check\_box.Visibility = Visibility.Visible;

}

}

}

}

