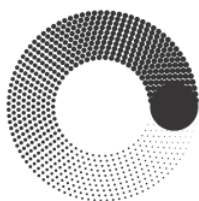


**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**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="JP3.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:JP3"
    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"/>
        </Grid>
    </Window>
```

```
<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>
```

```

        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;
    }
}
}
}

```

Калькулятор

46

$$12 + 6 = 18$$

$$18 - 5 = 13$$

$$13 * 3 = 39$$

$$39 + 7 = 46$$

1

2

3

4

5

6

+

-

7

8

9

\*

f-d

0

=

AC

/

l-d



Off



On



Отмена