федеральное государственное автономное образовательное учреждение высшего образования

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)
(Факультет информационных технологий)

(Институт Принтмедиа и информационных технологий)
Кафедра Информатики и информационных технологий

направление подготовки
09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 3

Дисциплина: Технология прикладного программирования

Тема: Калькулятор улучшенный

Выполнил(а): студент(ка) группы 221-377
Ежов Тимофей Алексеевич
(Фамилия И.О.)

Дата, подпись 14.02.2023
Проверил:_________________ ____________
(Фамилия И.О., степень, звание) **(Оценка)**

Дата, подпись _________________ ____________
(Дата) (Подпись)

Замечания: ____

____

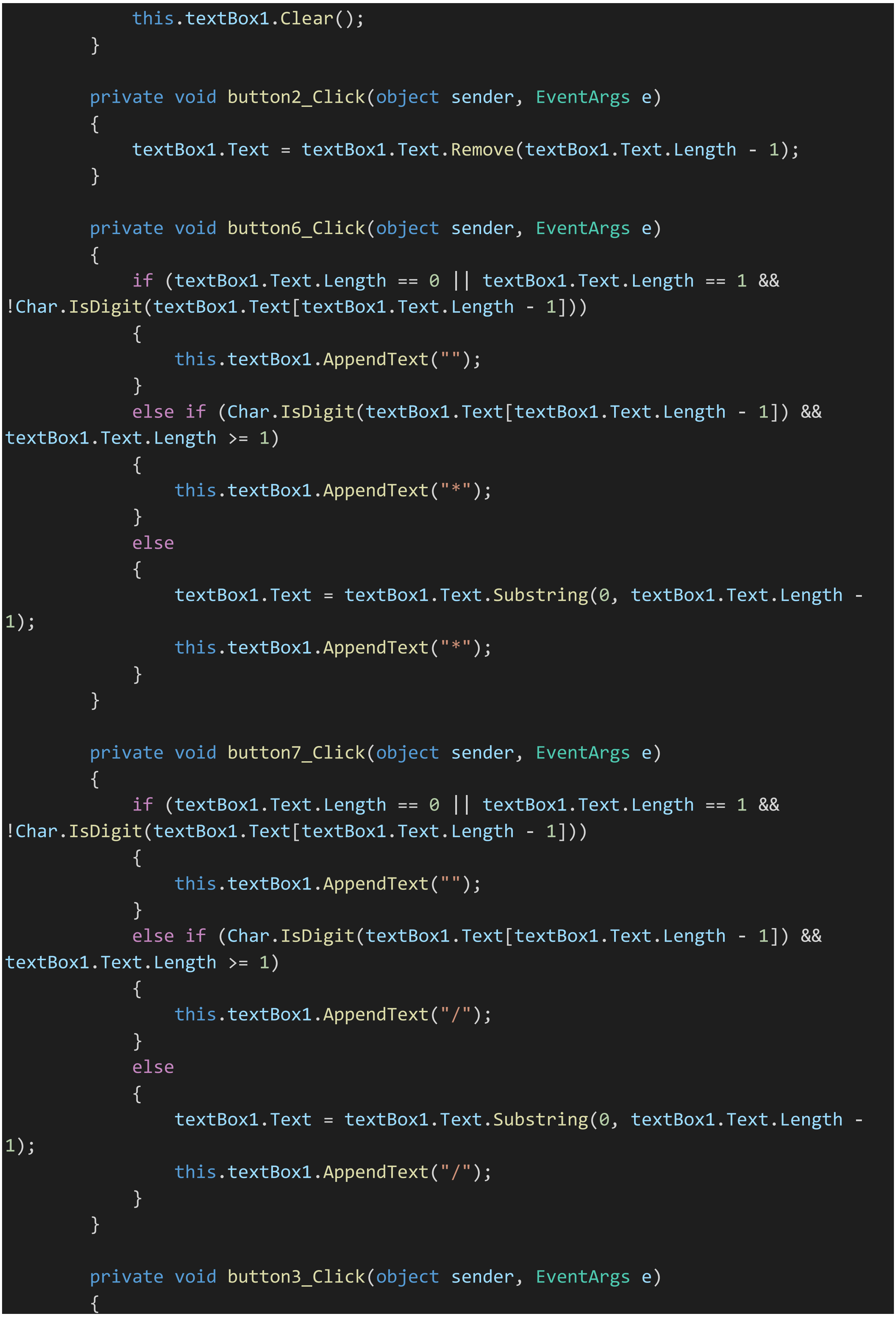
____

Москва 2023

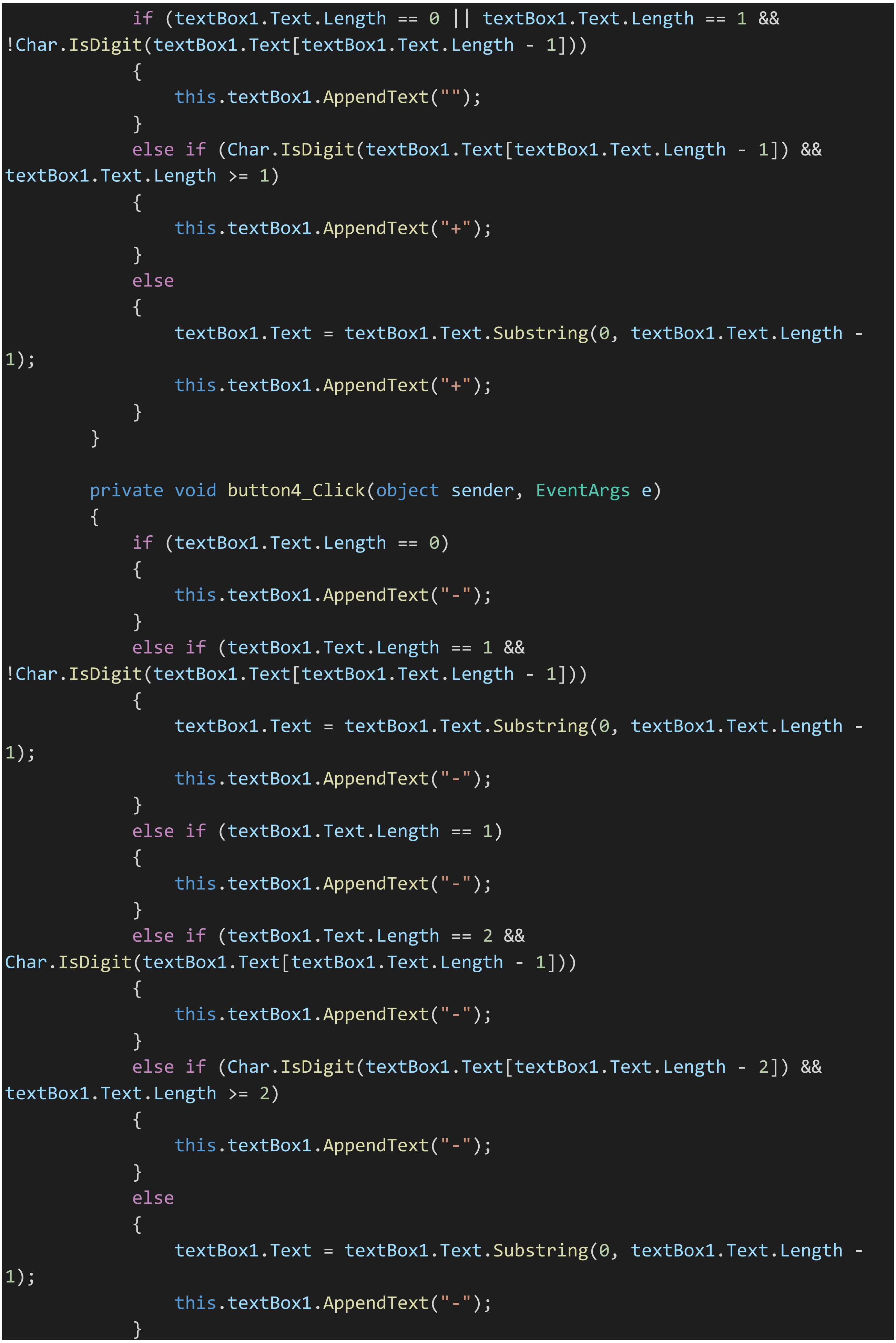
Лабораторная работа №3
Листинг:

```
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 LAB_2
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            textBox1.Clear();
            radioButton1.IsChecked = true;
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void button15_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("1");
        }

        private void button18_Click(object sender, EventArgs e)
        {
            Random rnd = new Random();
            foreach (Button item in grid1.Children.OfType<Button>())
            {
                item.Background = new
SolidColorBrush(Color.FromArgb(Convert.ToByte(rnd.Next(1, 255)),
Convert.ToByte(rnd.Next(1, 255)), Convert.ToByte(rnd.Next(1, 255)),
Convert.ToByte(rnd.Next(1, 255))));
```

```
            }

        }

        private void button16_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("2");
        }

        private void button17_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("3");
        }

        private void button12_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("4");
        }

        private void button13_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("5");
        }

        private void button14_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("6");
        }

        private void button8_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("7");
        }

        private void button10_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("8");
        }

        private void button11_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("9");
        }

        private void button9_Click(object sender, EventArgs e)
        {
            this.textBox1.AppendText("0");
        }

        private void button1_Click(object sender, EventArgs e)
        {
```



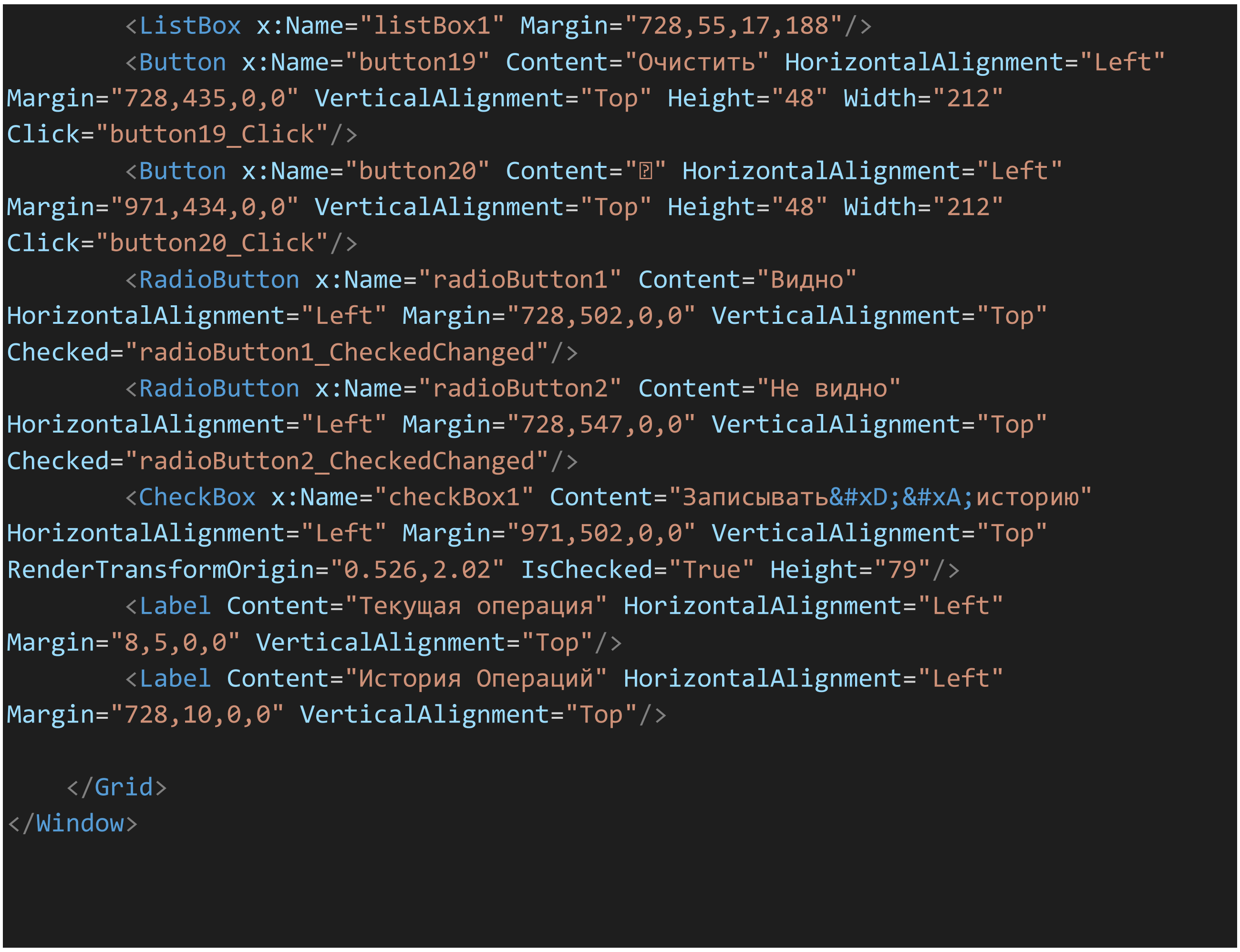
```
            this.textBox1.Clear();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1);
        }

        private void button6_Click(object sender, EventArgs e)
        {
            if (textBox1.Text.Length == 0 || textBox1.Text.Length == 1 &&
!Char.IsDigit(textBox1.Text[textBox1.Text.Length - 1]))
            {
                this.textBox1.AppendText("");
            }
            else if (Char.IsDigit(textBox1.Text[textBox1.Text.Length - 1]) &&
textBox1.Text.Length >= 1)
            {
                this.textBox1.AppendText("*");
            }
            else
            {
                textBox1.Text = textBox1.Text.Substring(0, textBox1.Text.Length -
1);
                this.textBox1.AppendText("*");
            }
        }

        private void button7_Click(object sender, EventArgs e)
        {
            if (textBox1.Text.Length == 0 || textBox1.Text.Length == 1 &&
!Char.IsDigit(textBox1.Text[textBox1.Text.Length - 1]))
            {
                this.textBox1.AppendText("");
            }
            else if (Char.IsDigit(textBox1.Text[textBox1.Text.Length - 1]) &&
textBox1.Text.Length >= 1)
            {
                this.textBox1.AppendText("/");
            }
            else
            {
                textBox1.Text = textBox1.Text.Substring(0, textBox1.Text.Length -
1);
                this.textBox1.AppendText("/");
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
```



```
            if (textBox1.Text.Length == 0 || textBox1.Text.Length == 1 &&
!Char.IsDigit(textBox1.Text[textBox1.Text.Length - 1]))
            {
                this.textBox1.AppendText("");
            }
            else if (Char.IsDigit(textBox1.Text[textBox1.Text.Length - 1]) &&
textBox1.Text.Length >= 1)
            {
                this.textBox1.AppendText("+");
            }
            else
            {
                textBox1.Text = textBox1.Text.Substring(0, textBox1.Text.Length -
1);
                this.textBox1.AppendText("+");
            }
        }

        private void button4_Click(object sender, EventArgs e)
        {
            if (textBox1.Text.Length == 0)
            {
                this.textBox1.AppendText("-");
            }
            else if (textBox1.Text.Length == 1 &&
!Char.IsDigit(textBox1.Text[textBox1.Text.Length - 1]))
            {
                textBox1.Text = textBox1.Text.Substring(0, textBox1.Text.Length -
1);
                this.textBox1.AppendText("-");
            }
            else if (textBox1.Text.Length == 1)
            {
                this.textBox1.AppendText("-");
            }
            else if (textBox1.Text.Length == 2 &&
Char.IsDigit(textBox1.Text[textBox1.Text.Length - 1]))
            {
                this.textBox1.AppendText("-");
            }
            else if (Char.IsDigit(textBox1.Text[textBox1.Text.Length - 2]) &&
textBox1.Text.Length >= 2)
            {
                this.textBox1.AppendText("-");
            }
            else
            {
                textBox1.Text = textBox1.Text.Substring(0, textBox1.Text.Length -
1);
                this.textBox1.AppendText("-");
            }
        }

        private void button5_Click(object sender, EventArgs e)
        {
            string first = "";
            string second = "";
            char x = '0';
            bool flag = true;
            string res = textBox1.Text;
            bool flag2 = true;
            foreach (char ch in textBox1.Text)
            {
                if ((Char.IsDigit(ch) || ch == '-' && flag2) && flag)
                {
                    first += ch;
                    flag2 = false;
                }
                else if (!Char.IsDigit(ch) && flag)
                {
                    x = ch;
                    flag = false;
                }
                else
                {
                    second += ch;
                }
            }
            if (x == '+')
            {
                textBox1.Text = Convert.ToString(Convert.ToInt32(first) +
Convert.ToInt32(second));
            }
            else if (x == '-')
            {
                textBox1.Text = Convert.ToString(Convert.ToInt32(first) -
Convert.ToInt32(second));
            }
            else if (x == '/')
            {
                textBox1.Text = Convert.ToString(Convert.ToInt32(first) /
Convert.ToInt32(second));
            }
            else if (x == '*')
            {
                textBox1.Text = Convert.ToString(Convert.ToInt32(first) *
Convert.ToInt32(second));
            }
            res += " = " + textBox1.Text + '\n';
            if (Convert.ToBoolean(checkBox1.IsChecked)) listBox1.Items.Add(res);
        }
        private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {
            button19.Visibility = Visibility.Visible;
            button20.Visibility = Visibility.Visible;
            checkBox1.Visibility = Visibility.Visible;
            listBox1.Visibility = Visibility.Visible;
        }

        private void radioButton2_CheckedChanged(object sender, EventArgs e)
        {
            button19.Visibility = Visibility.Hidden;
            button20.Visibility = Visibility.Hidden;
            checkBox1.Visibility = Visibility.Hidden;
            listBox1.Visibility = Visibility.Hidden;
        }

        private void button20_Click(object sender, EventArgs e)
        {
            listBox1.Items.RemoveAt(listBox1.Items.Count - 1);
        }

        private void button19_Click(object sender, EventArgs e)
        {
            listBox1.Items.Clear();
        }

    }

}
```

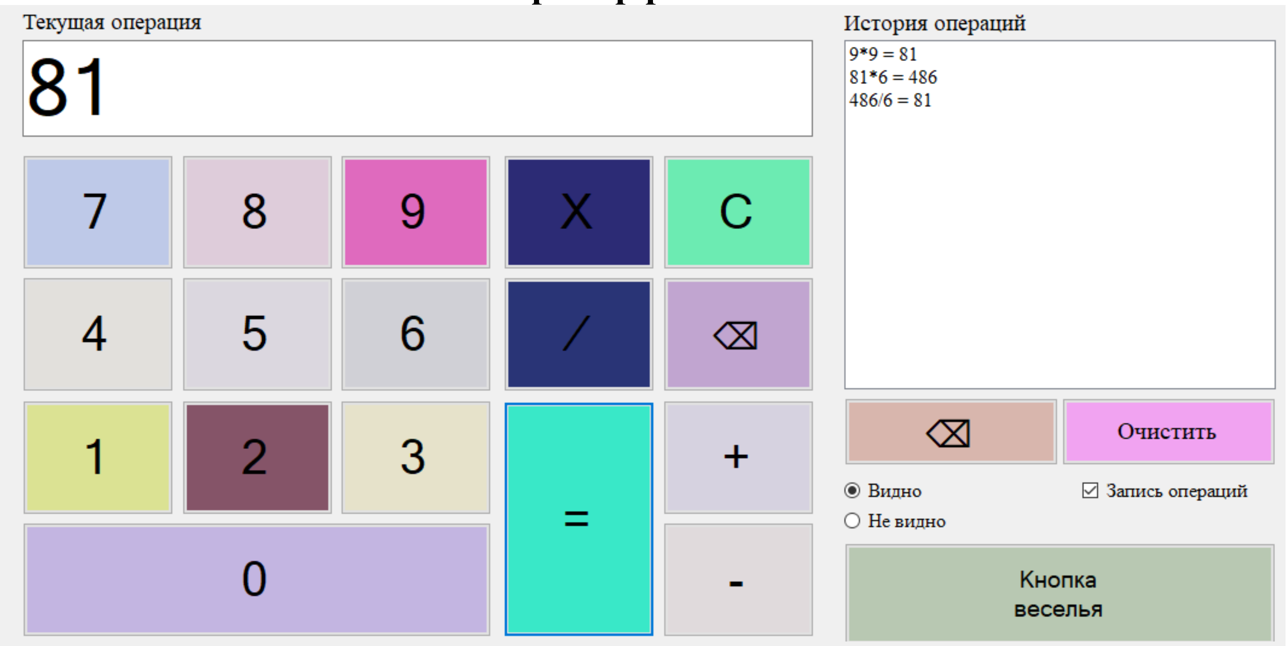
```
<Window x:Class="LAB_2.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:LAB_2"
        mc:Ignorable="d"

        Title="Калькулятор KGBT++" Height="631" Width="1198"
Icon="/cover1__w820.jpg" FontSize="26">
    <Grid x:Name="grid1" Height="610" VerticalAlignment="Top">
        <Button x:Name="button1" Content="C" HorizontalAlignment="Left"
Margin="580,149,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button1_Click"/>
        <Button x:Name="button2" Content="" HorizontalAlignment="Left"
Margin="580,262,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button2_Click"/>
```

```
        <Button x:Name="button3" Content="+" HorizontalAlignment="Left"
Margin="580,375,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button3_Click"/>
        <Button x:Name="button4" Content="-" HorizontalAlignment="Left"
Margin="580,488,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button4_Click"/>
        <TextBox x:Name="textBox1" HorizontalAlignment="Left" Margin="8,55,0,0"
Text="TextBox" TextWrapping="Wrap" VerticalAlignment="Top" Width="697"
Height="73"/>
        <Button x:Name="button5" Content="=" HorizontalAlignment="Left"
Margin="439,488,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button5_Click"/>
        <Button x:Name="button6" Content="/" HorizontalAlignment="Left"
Margin="439,149,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button6_Click"/>
        <Button x:Name="button7" Content="X" HorizontalAlignment="Left"
Margin="439,262,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button7_Click"/>
        <Button x:Name="button18" Content="Кнопка&#xD;&#xA;Веселья"
HorizontalAlignment="Left" Margin="439,375,0,0" VerticalAlignment="Top" Height="95"
Width="125" Click="button18_Click"/>
        <Button x:Name="button8" Content="7" HorizontalAlignment="Left"
Margin="8,149,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button8_Click"/>
        <Button x:Name="button9" Content="0" HorizontalAlignment="Left"
Margin="8,488,0,0" VerticalAlignment="Top" Height="95" Width="404"
Click="button9_Click"/>
        <Button x:Name="button10" Content="8" HorizontalAlignment="Left"
Margin="148,149,0,0" VerticalAlignment="Top" Height="95" Width="124"
Click="button10_Click"/>
        <Button x:Name="button11" Content="9" HorizontalAlignment="Left"
Margin="288,149,0,0" VerticalAlignment="Top" Height="95" Width="124"
Click="button11_Click" />
        <Button x:Name="button12" Content="4" HorizontalAlignment="Left"
Margin="8,262,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button12_Click"/>
        <Button x:Name="button13" Content="5" HorizontalAlignment="Left"
Margin="148,262,0,0" VerticalAlignment="Top" Height="95" Width="124"
Click="button13_Click"/>
        <Button x:Name="button14" Content="6" HorizontalAlignment="Left"
Margin="288,262,0,0" VerticalAlignment="Top" Height="95" Width="124"
Click="button14_Click" />
        <Button x:Name="button15" Content="1" HorizontalAlignment="Left"
Margin="8,375,0,0" VerticalAlignment="Top" Height="95" Width="125"
Click="button15_Click"/>
        <Button x:Name="button16" Content="2" HorizontalAlignment="Left"
Margin="148,375,0,0" VerticalAlignment="Top" Height="95" Width="124"
Click="button16_Click"/>
        <Button x:Name="button17" Content="3" HorizontalAlignment="Left"
Margin="288,375,0,0" VerticalAlignment="Top" Height="95" Width="124"
Click="button17_Click"/>
```



```
        <ListBox x:Name="listBox1" Margin="728,55,17,188"/>
        <Button x:Name="button19" Content="Очистить" HorizontalAlignment="Left"
Margin="728,435,0,0" VerticalAlignment="Top" Height="48" Width="212"
Click="button19_Click"/>
        <Button x:Name="button20" Content="⍰" HorizontalAlignment="Left"
Margin="971,434,0,0" VerticalAlignment="Top" Height="48" Width="212"
Click="button20_Click"/>
        <RadioButton x:Name="radioButton1" Content="Видно"
HorizontalAlignment="Left" Margin="728,502,0,0" VerticalAlignment="Top"
Checked="radioButton1_CheckedChanged"/>
        <RadioButton x:Name="radioButton2" Content="Не видно"
HorizontalAlignment="Left" Margin="728,547,0,0" VerticalAlignment="Top"
Checked="radioButton2_CheckedChanged"/>
        <CheckBox x:Name="checkBox1" Content="Записывать&#xD;&#xA;историю"
HorizontalAlignment="Left" Margin="971,502,0,0" VerticalAlignment="Top"
RenderTransformOrigin="0.526,2.02" IsChecked="True" Height="79"/>
        <Label Content="Текущая операция" HorizontalAlignment="Left"
Margin="8,5,0,0" VerticalAlignment="Top"/>
        <Label Content="История Операций" HorizontalAlignment="Left"
Margin="728,10,0,0" VerticalAlignment="Top"/>

    </Grid>
</Window>
```

Пример работы:



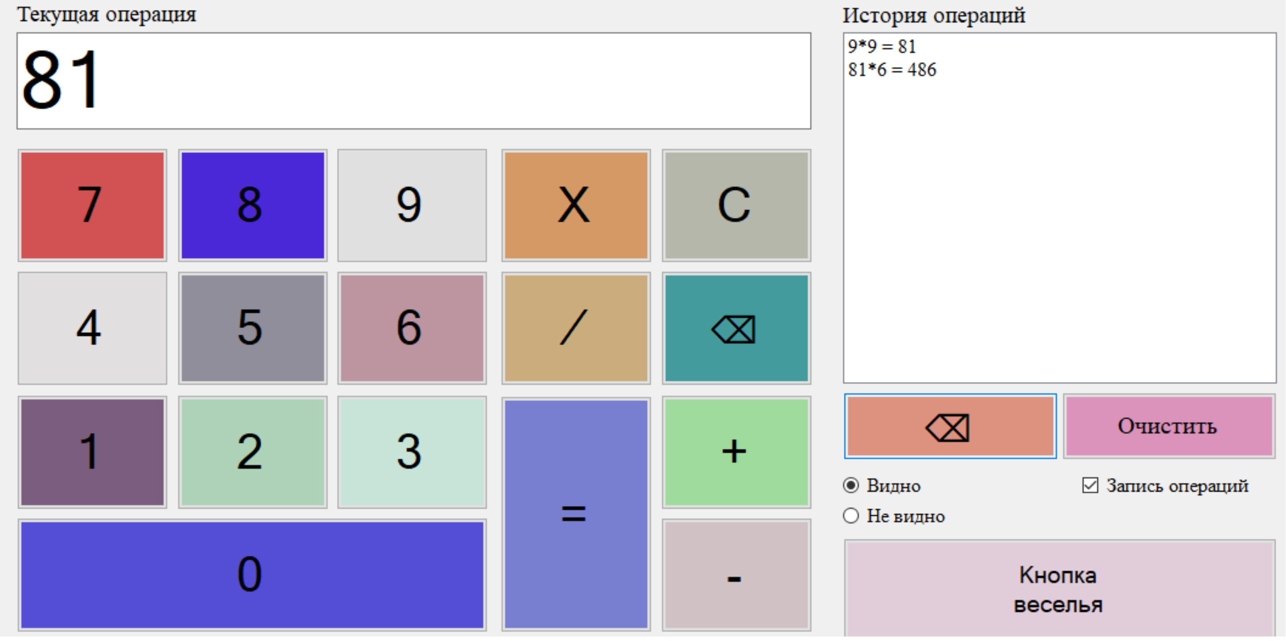
Текущая операция

81

История операций

9*9 = 81
81*6 = 486
486/6 = 81

● Видно
○ Не видно
☑ Запись операций



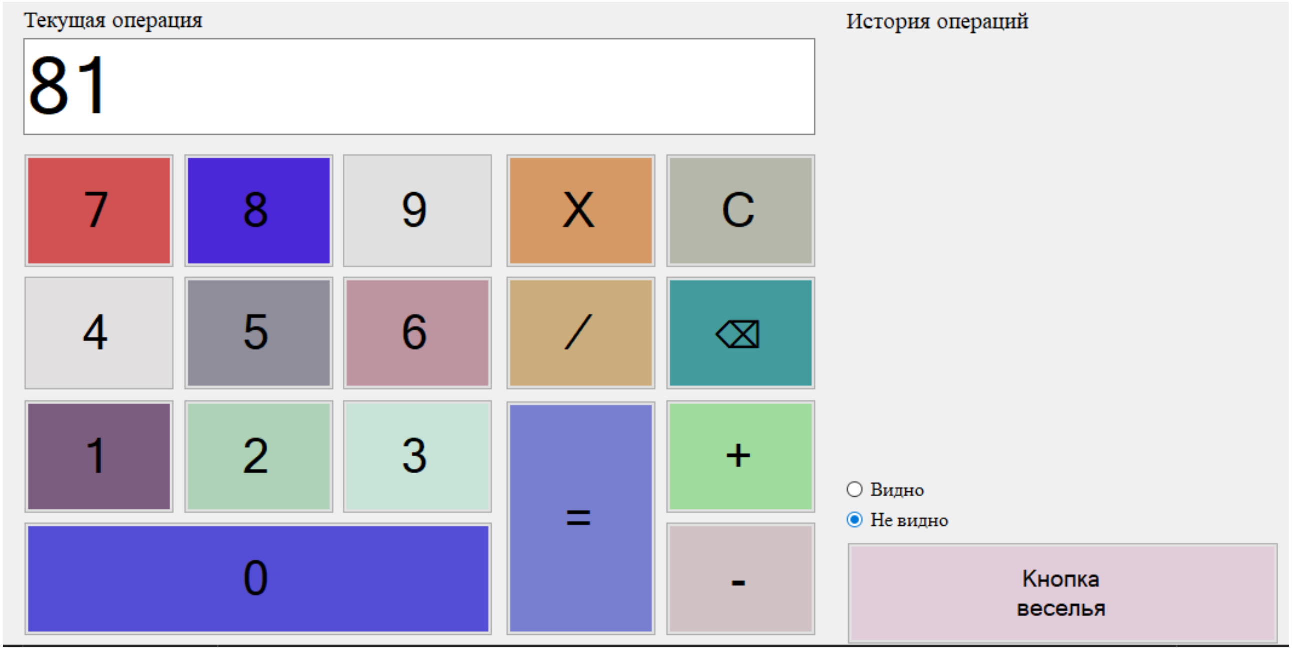
Текущая операция

81

История операций

9*9 = 81
81*6 = 486

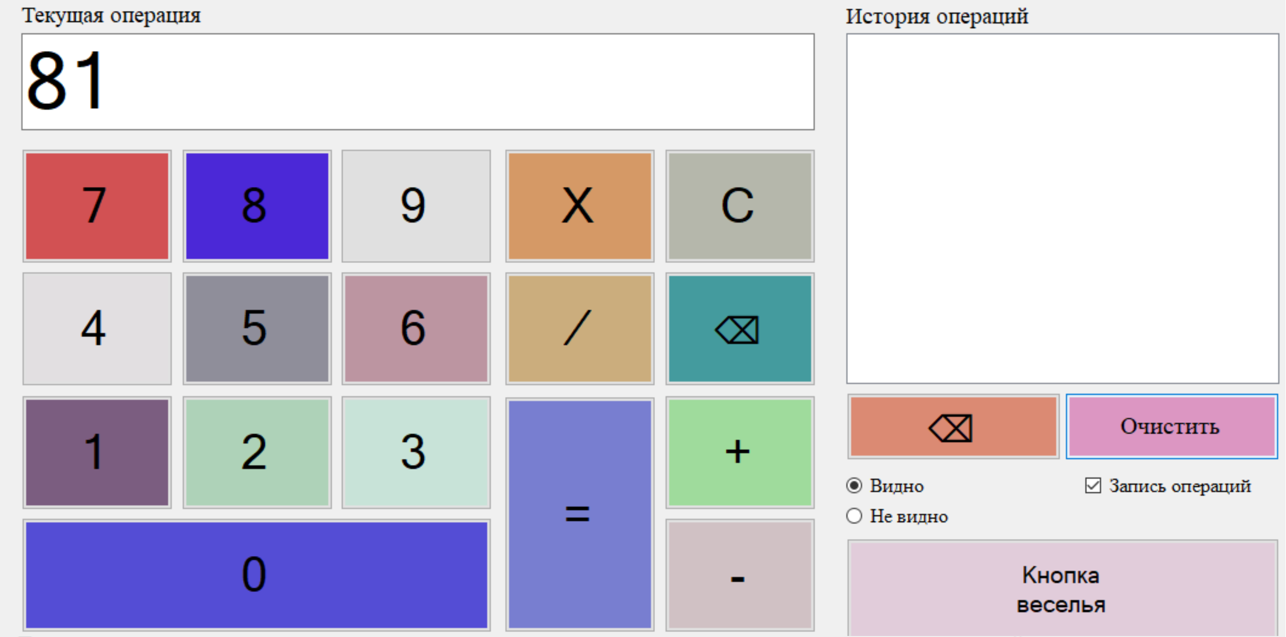
● Видно
○ Не видно
☑ Запись операций



Текущая операция: 81

История операций

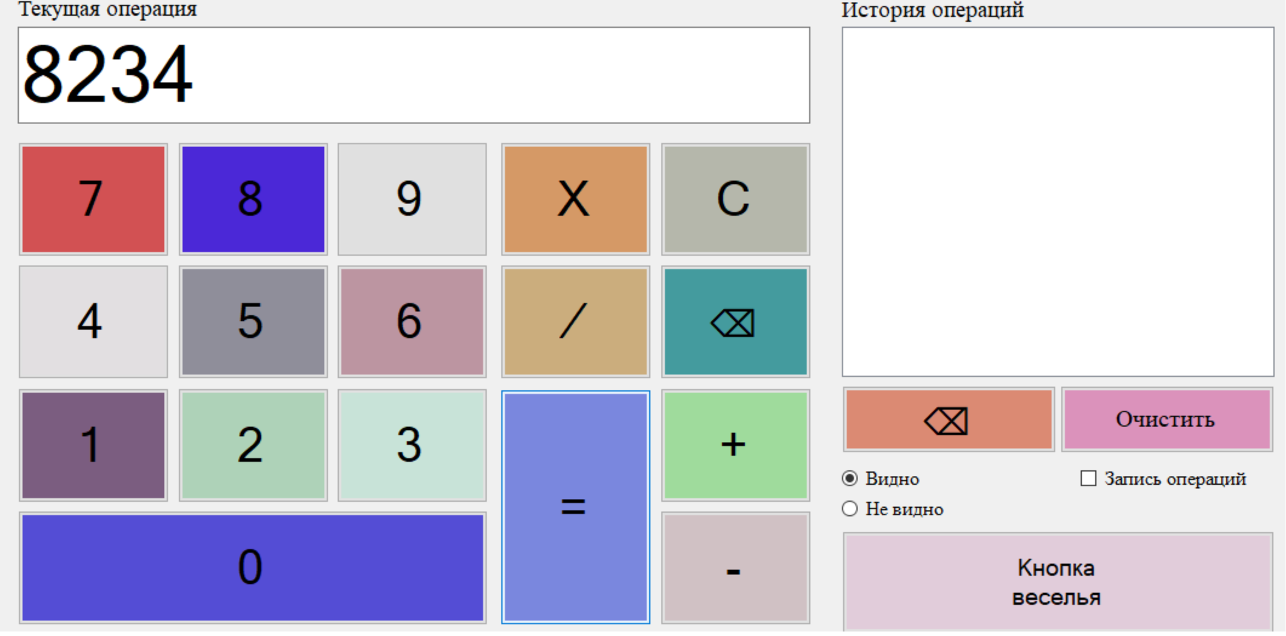
○ Видно
● Не видно



Текущая операция: 81

История операций

● Видно
○ Не видно
☑ Запись операций



Текущая операция: 8234

История операций

● Видно
○ Не видно
☐ Запись операций