

Лабораторная работа № 3.

Тема: «События клавиатуры и мыши. Работа с ресурсами. Создание и применение стилей»

В приложении ЛР 2 были добавлены 3 вкладки: событие клавиатуры, задание 1 и событие мыши (Рис.1)



Рисунок 1 – Добавленные вкладки

Во вкладке событие клавиатуры были добавлены свойства такие, как:

- KeyDown возникает при нажатии клавиши. На это вкладке были добавлены два текстовых поля, в соответствии с рисунком 2-3.

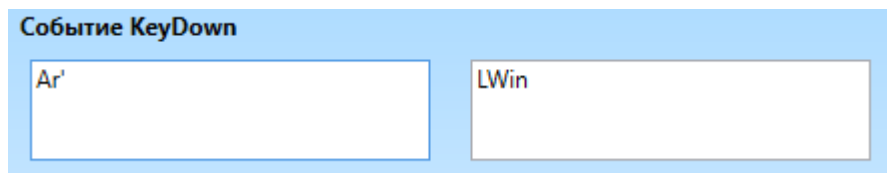


Рисунок 2 – Событие KeyDown

```
private void textBox1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.IsDown)
    {
        textBox2.Text = e.Key.ToString();
    }
}
```

Рисунок 3 – Код KeyDown

Проверил работоспособность данного метода. При нажатии на клавишу, программа распознает на какую именно клавишу было совершено нажатие.

- Свойство IsToggled (Рис.4)

					ККЭП 09.02.07 0053 От		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.	Бреус А.Д.				События клавиатуры и мыши. Работа с ресурсами. Создание и применение стилей.		
Провер.	Шостак А.И.						
					Лит.	Лист	Листов
					у	1	7
					Гр. 22-Д9-ЗИНС		

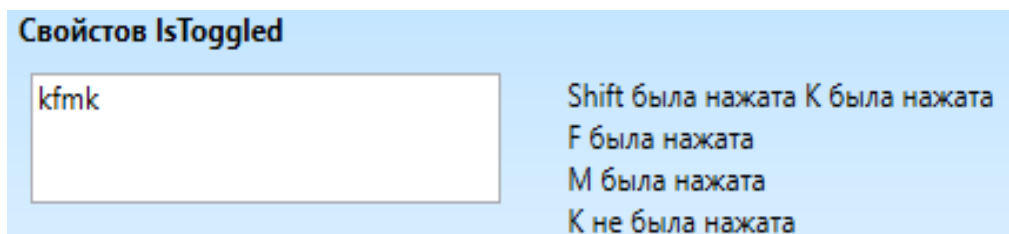


Рисунок 4 – Свойство IsToggled

Для данного метода было создано текстовое поле и label. Для этого текстового поля был использовано событие KeyDown, в соответствии с рисунком 5.

```

Ссылка: 1
private void textBox3_KeyDown(object sender, KeyEventArgs e)
{
    if (e.IsToggled)
    {
        labelToggled.Content += " " + e.Key.ToString() + " была нажата\n";
    }
    else
    {
        labelToggled.Content += " " + e.Key.ToString() + " не была нажата\n";
    }
    if (e.KeyboardDevice.Modifiers == System.Windows.Input.ModifierKeys.Shift)
    {
        labelToggled.Content = " Shift была нажата";
    }
    if (e.KeyboardDevice.Modifiers == System.Windows.Input.ModifierKeys.Alt)
    {
        labelToggled.Content = " Alt была нажата";
    }
    if (e.KeyboardDevice.Modifiers == System.Windows.Input.ModifierKeys.Control)
    {
        labelToggled.Content = " Ctrl была нажата";
    }
}

```

Рисунок 5 – Код KeyDown, свойство IsToggled

- Валидация текста (рис. 6). Создал еще один текстовый блок и label, для которого создал событие PreviewTextInput, которое будет производить валидацию текста, также этот метод был доработан, чтобы можно было вводить свою фамилию, в соответствии с рисунком 7.

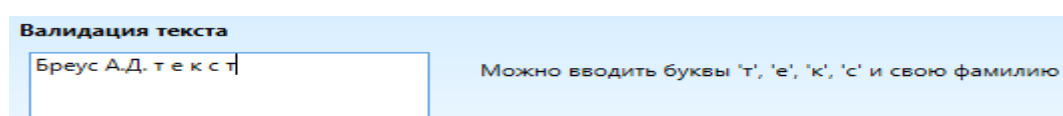


Рисунок 6 – Валидация текста

```
Ссылка: 1
private void textBox4_PreviewTextInput(object sender, TextCompositionEventArgs e)
{
    if (e.Text != "т" && e.Text != "е" && e.Text != "к" && e.Text != "с" && e.Text != "б" && e.Text != "п" && e.Text != "у" && e.Text != "с"
        && e.Text != "А" && e.Text != "Д" && e.Text != "." && e.Text != "Т" && e.Text != "Е" && e.Text != "К" && e.Text != "С")
    {
        e.Handled = true;
        labelValidate.Content = "Символ недопустим";
    }
    else
    {
        labelValidate.Content = "Можно вводить буквы 'т', 'е', 'к', 'с' и свою фамилию";
    }
}

Ссылка: 1
private void textBox4_PreviewKeyDown(object sender, KeyEventArgs e)
{
    if(e.Key == Key.Space)
    {
        labelValidate.Content = "Пробел тоже можно";
    }
}
```

Рисунок 7 – Событие PreviewTextInput

Код самой вкладки представлен на рисунке 8.

```
TabItem Header="События клавиатуры">
<Canvas>
    <Canvas.Background>
        <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
            <GradientStop Color="#FFAFDCFF"/>
            <GradientStop Color="White" Offset="1"/>
        </LinearGradientBrush>
    </Canvas.Background>
    <TextBlock Canvas.Left="15" FontWeight="Bold">Событие KeyDown</TextBlock>
    <TextBox x:Name="textBox1" Canvas.Left="20" Canvas.Top="25" Height="50" Width="200" KeyDown="textBox1_KeyDown"></TextBox>
    <TextBox x:Name="textBox2" Canvas.Left="240" Canvas.Top="25" Height="50" Width="200"></TextBox>

    <TextBlock Canvas.Left="15" Canvas.Top="100" FontWeight="Bold">Свойство IsToggled</TextBlock>
    <TextBox x:Name="textBox3" Canvas.Left="20" Canvas.Top="125" Height="50" Width="200" KeyDown="textBox3_KeyDown"></TextBox>
    <Label x:Name="labelToggled" Canvas.Left="240" Canvas.Top="120">Клавиша</Label>

    <TextBlock Canvas.Left="15" Canvas.Top="200" FontWeight="Bold">Валидация текста</TextBlock>
    <TextBox x:Name="textBox4" Canvas.Left="20" Canvas.Top="225" Height="50" Width="200" PreviewTextInput="textBox4_PreviewTextInput" PreviewKeyDown="textBox4_PreviewKeyDown"></TextBox>
    <Label x:Name="labelValidate" Canvas.Left="240" Canvas.Top="225">Можно вводить буквы 'т', 'е', 'к', 'с' и свою фамилию</Label>
</Canvas>
</TabItem>
```

Рисунок 8 – Код вкладки «События клавиатуры»

На созданной вкладке “Задание 1” были реализованы следующие события клавиатуры (Рис.9)

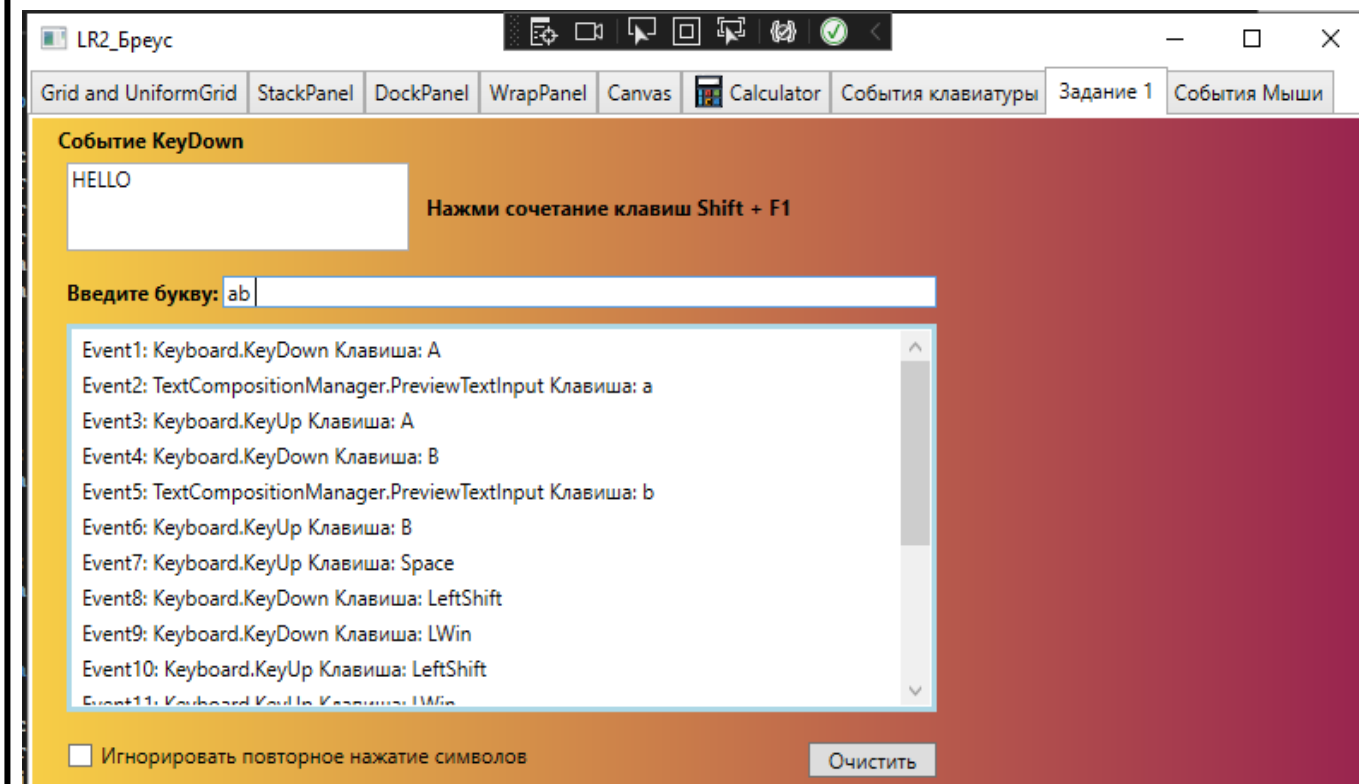


Рисунок 9 – Вкладка «Задание1»

```

TabItem Header="Задание 1">
<Canvas>
<Canvas.Background>
<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0" Opacity="1">
<LinearGradientBrush.RelativeTransform>
<TransformGroup>
<ScaleTransform CenterY="0.5" CenterX="0.5"/>
<SkewTransform CenterX="0.5" CenterY="0.5"/>
<RotateTransform Angle="90" CenterX="0.5" CenterY="0.5"/>
<TranslateTransform/>
</TransformGroup>
</LinearGradientBrush.RelativeTransform>
<GradientStop Color="#FF9924" Offset="0"/>
<GradientStop Color="#FF7CE4" Offset="1"/>
</LinearGradientBrush>
</Canvas.Background>
<TextBlock Canvas.Left="15" HorizontalAlignment="Center" Canvas.Top="4" FontWeight="Bold">Событие KeyDown</TextBlock>
<TextBox x:Name="TextBox6" Canvas.Left="20" Canvas.Top="25" Height="50" Width="200" KeyDown="TextBox6_KeyDown"></TextBox>
<Label x:Name="LabelValidate2" Canvas.Left="225" Canvas.Top="37" FontWeight="Bold">Нажми сочетание клавиш Shift + F1</Label>
<Label x:Name="Label3" Canvas.Left="15" Canvas.Top="85" FontWeight="Bold">Введите букву:</Label>
<TextBox MinWidth="100" Canvas.Left="111" Canvas.Top="89" KeyDown="KeyEvents" KeyUp="KeyEvents" PreviewTextInput="TextInputEvent" Name="txtContent" HorizontalAlignment="Left" VerticalAlignment="Center" Width="417"/>
<ListBox BorderBrush="LightBlue" BorderThickness="3" Name="LbxEvents" Canvas.Left="20" Canvas.Top="116" HorizontalAlignment="Center" VerticalAlignment="Top" Height="219" Width="500"/>
<CheckBox Name="chkIgnoreRepeat" Canvas.Left="20" Canvas.Top="352">Игнорировать повторное нажатие символов</CheckBox>
<Button HorizontalAlignment="Left" Name="Clear" Click="Clear_Click" Canvas.Left="453" Canvas.Top="352" VerticalAlignment="Center" Width="75">Очистить</Button>
</Canvas>
</TabItem>

```

Рисунок 10 – Код вкладки «Задание1»

```

Ссылка: 2
private void TextBox6_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyboardDevice.Modifiers == ModifierKeys.Shift && e.Key == Key.F1)
    {
        TextBox6.Text += "HELLO ";
    }
}

Ссылка: 1
private void Clear_Click(object sender, RoutedEventArgs e)
{
    lbxEvents.Items.Clear();
    txtContent.Clear();
    i = 0;
}

protected int i = 0;

Ссылка: 2
private void KeyEvents(object sender, KeyEventArgs e)
{
    if ((bool)chkIgnoreRepeat.IsChecked && e.IsRepeat) return;
    i++;
    string s = "Event" + i + ": " + e.RoutedEvent + " Клавиша: " + e.Key;
    lbxEvents.Items.Add(s);
}

Ссылка: 1
private void TextInputEvent(object sender, TextCompositionEventArgs e)
{
    i++;
    string s = "Event" + i + ": " + e.RoutedEvent + " Клавиша: " + e.Text;
    lbxEvents.Items.Add(s);
}

```

Рисунок 11 – Код вкладки «Задание1»

На следующей вкладке “Событие мыши”, на нее добавил кнопку и два label, в соответствии с рисунком 12.

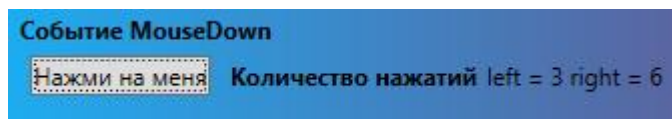


Рисунок 12 – Событие MouseDown

Затем прописал на эту кнопку событие MouseDown. Данный метод подсчитывает нажатия правой и левой кнопки мыши (рис. 13).

```

Ссылка: 2
private void previewDown(object sender, MouseButtonEventArgs e)
{
    if (e.ChangedButton == MouseButton.Left) l++;
    if (e.ChangedButton == MouseButton.Right) r++;
    textClick.Text = $"left = {l} right = {r}";
}

```

Рисунок 13 – Код MouseDown

На вкладке реализовал ProgressBar (максимум 100, минимум 0) и лейбл, в соответствии с рисунком 14.



Рисунок 14 – Свойство MiddleButton

Для того, чтобы он работал прописал событие MouseWheel, в соответствии с рисунком 15.

```
Ссылка: 1
private void progressBar_2_MouseWheel(object sender, MouseEventArgs e)
{
    progressBar_2.Value += e.Delta / 100;
    labelMiddle.Content = progressBar_2.Value + "%";
}
```

Рисунок 15 – Код MouseWheel

Delta – обозначение конечной разности при изменении какого-то параметра. Без нее метод выдает значения только 0% и 100%.

Создал текстовое поле и реализовал событие перетаскивания текста (рис. 16-17)



Рисунок 16 – Событие перетаскивания

```
private void textBox1_MouseDown(object sender, System.Windows.Input.MouseButtonEventArgs e)
{
    DragDrop.DoDragDrop(textBox1, textBox1.Text, DragDropEffects.Copy);
}

private void button1_Drop(object sender, DragEventArgs e)
{
    button1.Content = e.Data.GetData(DataFormats.Text);
}
```

Рисунок 17 – Код для события перетаскивания

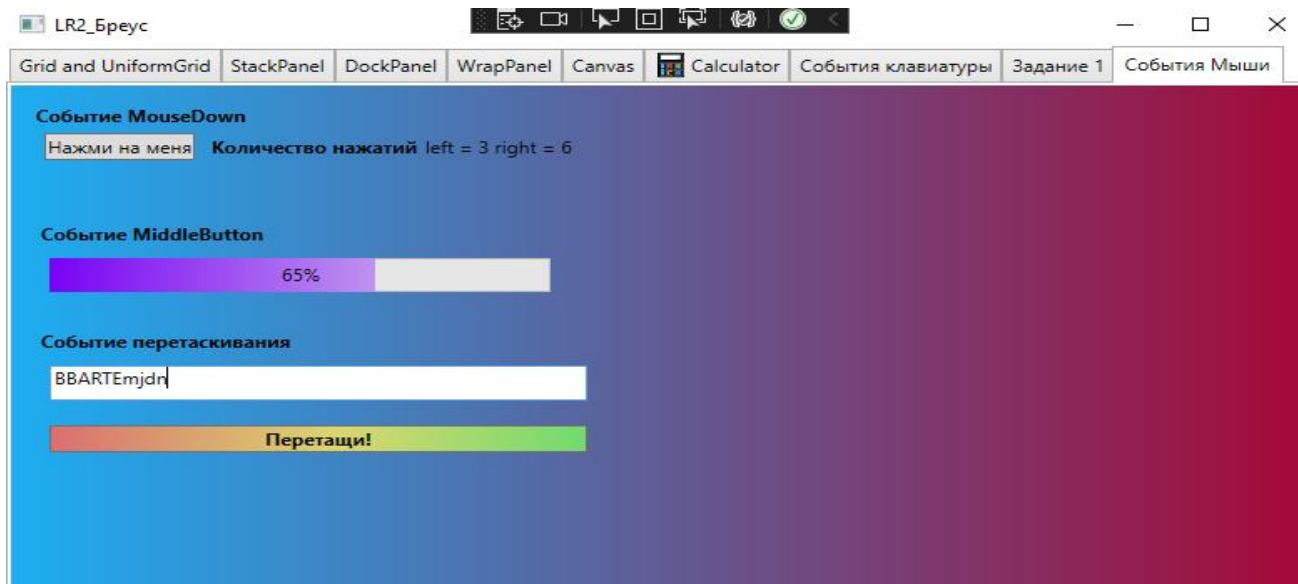


Рисунок 18 – Вкладка «Событие мыши»