

## Лабораторная работа №3

**Тема:** Создание горячих клавиш в WPF приложениях.

**Цели и задачи:** Научится создавать горячие клавиши в WPF приложениях.

### Краткие сведения

Горячие клавиши - разновидность интерфейса взаимодействия с вычислительным устройством (компьютером, калькулятором), представляющая собой нажатие клавиши (или сочетания клавиш) на клавиатуре, которому назначено (запрограммировано) некое действие — команды (операции), исполняемые данной системой. Как правило, частично дублирует интерфейс меню или кнопок и служит для значительного ускорения работы.

На персональных компьютерах чаще всего в сочетаниях используются клавиши «Ctrl», «Alt», «Shift», а также «Windows» («Super» или «Command»). Их часто называют клавиши-модификаторы. В сочетаниях можно зажимать не только одну из этих клавиш, а несколько одновременно.

То есть при использовании 3-х клавиш-модификаторов на одну обычную клавишу можно «повесить» до восьми ( $8 = 2^3$ ; 2 — поскольку клавиша имеет 2 состояния — «нажата» и «не нажата»). Но следует учесть, что состояние «не нажаты все три клавиши» не может использоваться для выполнения команд, следовательно, доступно для использования только семь состояний, а при использовании клавиши «Windows» — до 16) различных команд.

## Пример создания приложения с горячими клавишами:

1. Создадим проект WPF;
2. В дизайнере форм создадим простейший интерфейс входа в систему (рис. 1);

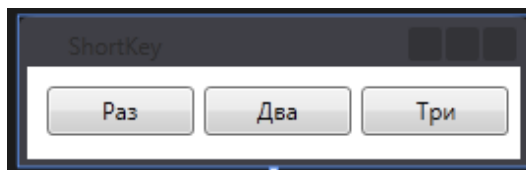


Рисунок 1 – Макет приложения

3. Создадим три метода с сообщениями;

```
private void One(object sender, ExecutedRoutedEventArgs e)
{
    MessageBox.Show("Вы нажали \'Ctrl+Q\'");
}

private void Two(object sender, ExecutedRoutedEventArgs e)
{
    MessageBox.Show("Вы нажали \'Ctrl+Alt+W\'");
}

private void Three(object sender, ExecutedRoutedEventArgs e)
{
    MessageBox.Show("Вы нажали \'Win+S\'");
}
```

4. Создадим команды (x:Key – ключ команды в пространстве имен «x», Text – название команды);

```
<Window.Resources>
    <RoutedUICommand x:Key="First" Text="First" />
    <RoutedUICommand x:Key="Second" Text="Second" />
    <RoutedUICommand x:Key="Third" Text="Third" />
</Window.Resources>
```

5. Объявим связь созданных команд с ранее созданными методами (рис. 2);

```
<Window.CommandBindings>
    <CommandBinding Command="{StaticResource First}"
        Executed="One" />
    <CommandBinding Command="{StaticResource Second}"
        Executed="Two" />
```

```

        <CommandBinding Command="{StaticResource Third}"
                        Executed="Three" />
    </Window.CommandBindings>

```

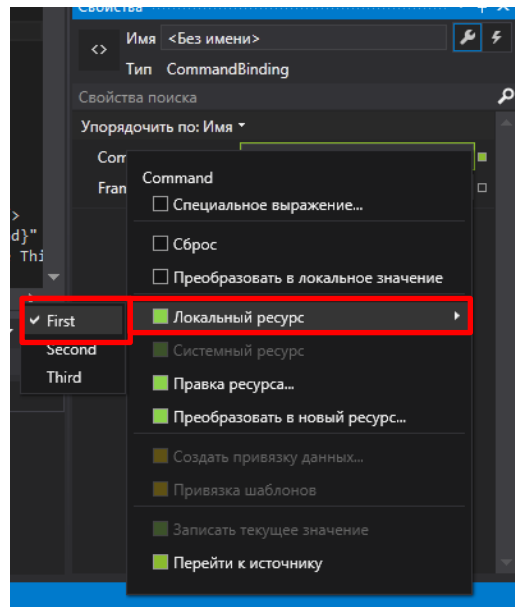


Рисунок 2 – Создание привязки поля «Command» к ранее созданной команде «First»

6. Создадим связь горячих клавиш с созданными командами, где укажем саму клавишу, ее модификаторы и исполняемую команду;

```

<Window.InputBindings>
    <KeyBinding Key="Q" Modifiers="Ctrl" Command="{StaticResource First}" />
    <KeyBinding Key="W" Modifiers="Ctrl+Alt" Command="{StaticResource Second}" />
    <KeyBinding Key="E" Modifiers="Ctrl+Shift+Alt" Command="{StaticResource Third}" />
</Window.InputBindings>

```

7. Добавим к кнопкам на форме эти же команды, создав привязку поля «Command» (рис. 3);

```

<Button x:Name="bOne"
        Margin="10,10,0,0"
        Height="25"
        Width="75"
        HorizontalAlignment="Left"
        VerticalAlignment="Top"
        Content="Паз"
        Command="{DynamicResource First}" />
<Button x:Name="bTwo"
        Margin="90,10,0,0"

```

```

Height="25"
Width="75"
HorizontalAlignment="Left"
VerticalAlignment="Top"
Content="Два"
Command="{DynamicResource Second}" />
<Button x:Name="bThree"
Margin="170,10,0,0"
Height="25"
Width="75"
HorizontalAlignment="Left"
VerticalAlignment="Top"
Content="Три"
Command="{DynamicResource Third}" />

```

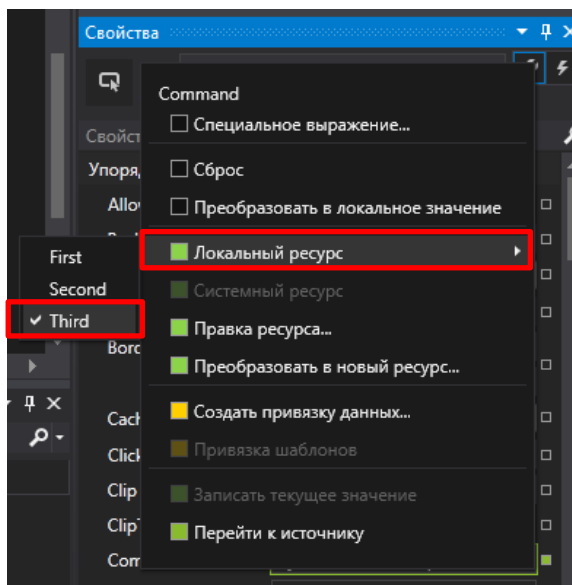


Рисунок 3 - Создание привязки поля «Command» к ранее созданной команде «Third»

8. Запустим программу и проверим работоспособность горячих клавиш и выполнение команд по нажатию кнопок на форме (рис. 4).

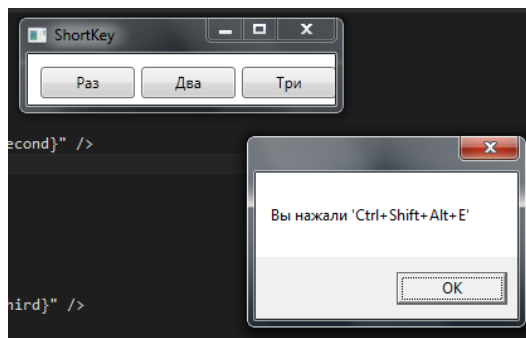


Рисунок 4 – Нажатие сочетания клавиш «Ctrl+Shift+Alt+E»

## Итоговая XAML разметка для тестового приложения

```
<Window x:Class="ShortKey.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="ShortKey" Height="78"
        Width="260">
    <Window.Resources>
        <RoutedUICommand x:Key="First" Text="First" />
        <RoutedUICommand x:Key="Second" Text="Second" />
        <RoutedUICommand x:Key="Third" Text="Third" />
    </Window.Resources>

    <Window.CommandBindings>
        <CommandBinding Command="{StaticResource First Executed="One" />
        <CommandBinding Command="{StaticResource Second}" Executed="Two" />
        <CommandBinding Command="{StaticResource Third}" Executed="Three" />
    </Window.CommandBindings>

    <Window.InputBindings>
        <KeyBinding Key="Q" Modifiers="Ctrl" Command="{StaticResource
First}" />
        <KeyBinding Key="W" Modifiers="Ctrl+Alt" Command="{StaticResource
Second}" />
        <KeyBinding Key="E" Modifiers="Ctrl+Shift+Alt" Command="{StaticRe-
source Third}" />
    </Window.InputBindings>

    <Grid>
        <Button x:Name="bOne"
            Margin="10,10,0,0"
            Height="25"
            Width="75"
            HorizontalAlignment="Left"
            VerticalAlignment="Top"
            Content="Один"
            Command="{DynamicResource First}" />
        <Button x:Name="bTwo"
            Margin="90,10,0,0"
            Height="25"
            Width="75"
            HorizontalAlignment="Left"
            VerticalAlignment="Top"
            Content="Два"
            Command="{DynamicResource Second}" />
        <Button x:Name="bThree"
            Margin="170,10,0,0"
            Height="25"
            Width="75"
            HorizontalAlignment="Left"
            VerticalAlignment="Top"
            Content="Три"
            Command="{DynamicResource Third}" />
    </Grid>
</Window>
```

## **Задания на лабораторную работу**

1. Изучить материал, представленный в данных методических указаниях к лабораторной работе;
2. Разработать WPF приложение, согласно заданию преподавателя;
3. Составить отчет. В отчете отобразить:
  - цели и задачи лабораторной работы;
  - XAML разметку;
  - исходный код приложения (отображающий только логику программы и описание файлов ресурсов);
  - скриншоты работы приложения;
  - выводы по проделанной лабораторной работе.