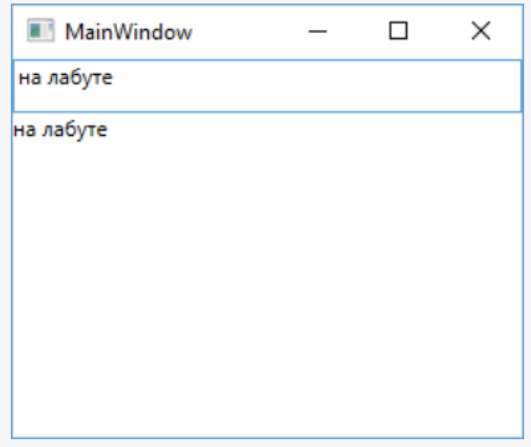
Привязка данных в WPF

В WPF привязка (binding) является мощным инструментом программирования, без которого не обходится ни одно серьезное приложение.

Привязка подразумевает взаимодействие двух объектов: источника и приемника. Объект-приемник создает привязку к определенному свойству объекта-источника. В случае модификации объекта-источника, объект-приемник также будет модифицирован. Например, простейшая форма с использованием привязки:

```
<Window x:Class="BindingApp.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:BindingApp"
        mc:Ignorable="d"
        Title="MainWindow" Height="250" Width="300">
    <StackPanel>
        <TextBox x:Name="myTextBox" Height="30" />
        <TextBlock x:Name="myTextBlock" Text="{Binding ElementName=myTextBox,Path=
Height="30" />
    </StackPanel>
</Window>
```



Для определения привязки используется выражение типа:

```
{Binding ElementName=Имя_объекта-источника, Path=Свойство_объекта-источника}
```

То есть в данном случае у нас элемент TextBox является источником, а TextBlock - приемником привязки. Свойство Text элемента TextBlock привязывается к свойству Text элемента TextBox. В итоге при осуществлении ввода в текстовое поле синхронно будут происходить изменения в текстовом блоке.

Привязка с EntityFramework

При работе с EntityFramework можно просто указывать {Binding имя_свойства}, но перед этим необходимо произвести установку this.DataContext – в случае если на окне будет производиться работа с одной записью, и DataGrid.ItemSource – для отображения данных в DataGrid.

Например, задана следующая модель.

```
public class Phone
{
   public int Id { get; set; }
   public string Name { get; set; }
   public int Price { get; set; }
}
```

В форме которая будет отвечать за работу с элементами Phone, необходимо указать this.DataContext;

```
public MainWindow()
{
   InitializeComponent();
   This.DataContext = new Phone();
}
```

Тогда на форме необходимо дописать следующее.

```
<TextBox x:Name="myTextBox" Text="{Binding Name}" Height="30" />
```

Теперь все изменения которые будут вноситься в TextBox, будут автоматически применяться к полю Name объекта Phone, который находится в this.DataContext.

TargetNullValue

На случай, если свойство в источнике привязки вдруг имеет значение null, то есть оно не установлено, мы можем задать некоторое значение по умолчанию. Например:

```
<Window.Resources>
   <local:Phone x:Key="nexusPhone" Company="Google" Price="25000" />
</Window.Resources>
<StackPanel>
   <TextBlock x:Name="titleTextBlock"
      Text="{Binding Source={StaticResource nexusPhone}, Path=Title,
TargetNullValue=Текст по умолчанию}" />
</StackPanel>
```

В данном случае у ресурса nexusPhone не установлено свойство Title, поэтому текстовый блок будет выводить значение по умолчанию, указанное в параметре TargetNullValue.