

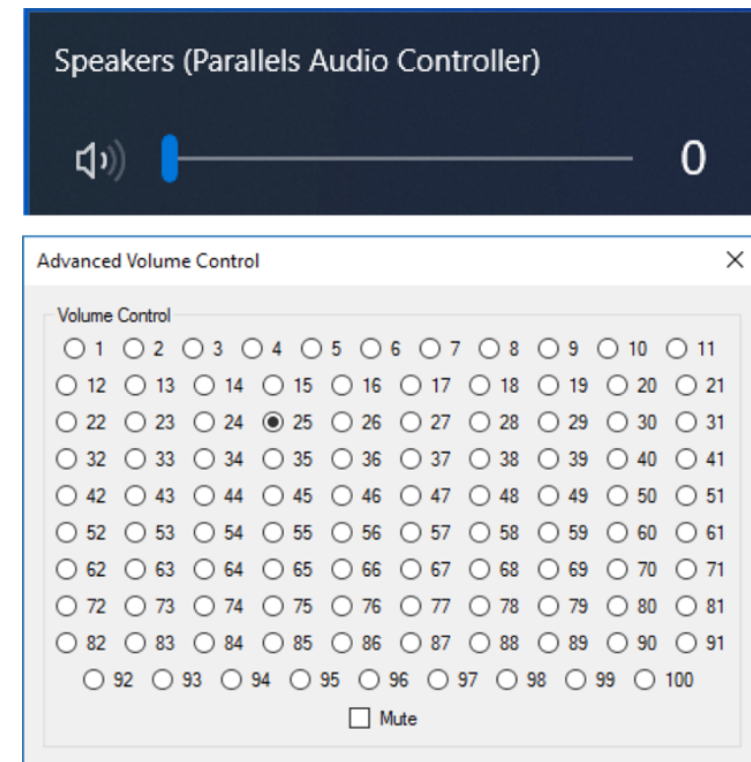
# Görsel Programlama

GUI, WinForms, WPF, WinUI

Emir ÖZTÜRK

# UI-UX

- UI
  - User Interface
- UX
  - User Experience

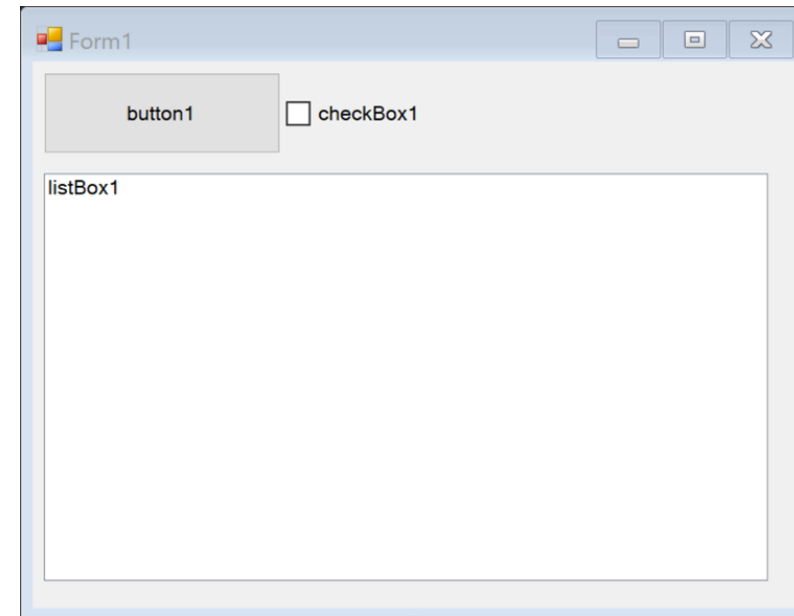


# C# ile Görsel Programlama

- Winforms
- Wpf
- Uwp
- WinUI

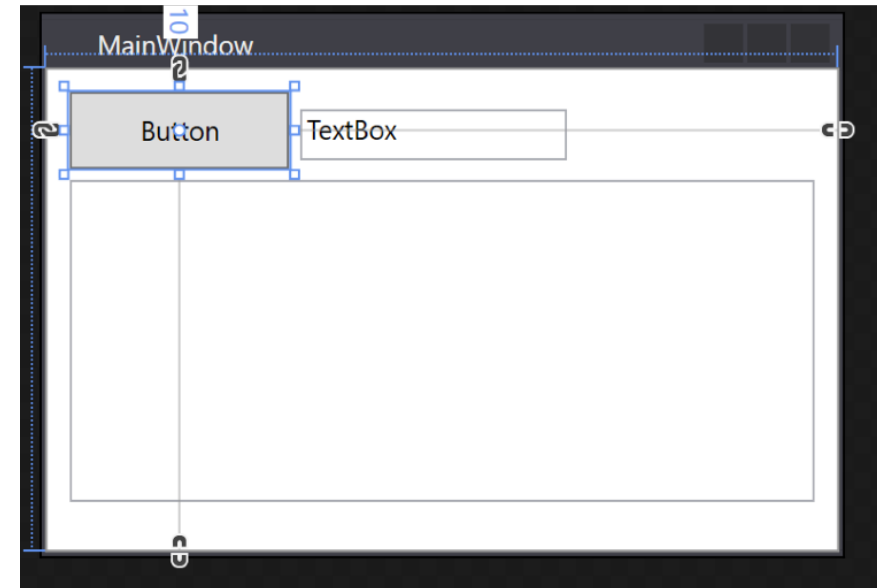
# Winforms

- .Net Framework
  - .Net
- Sürükle bırak



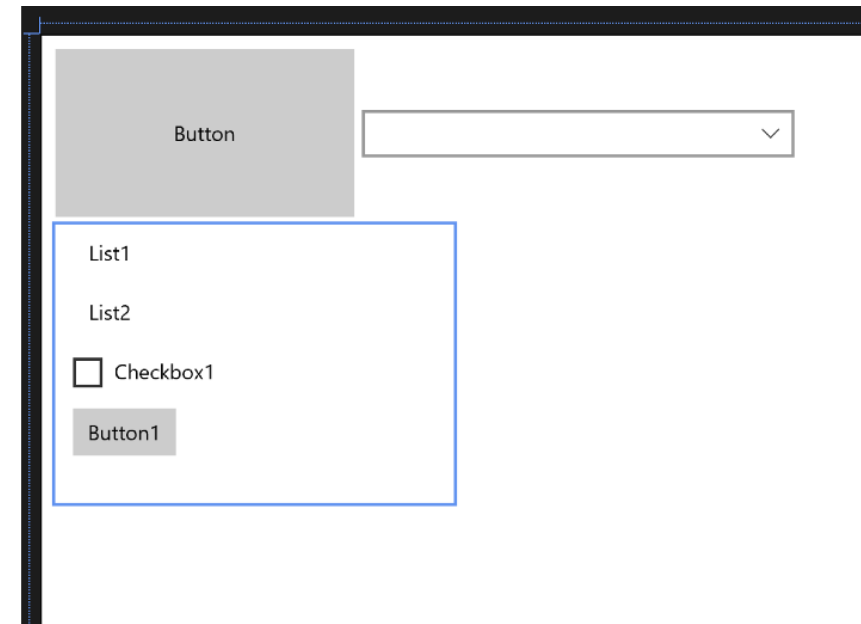
# WPF

- .Net Framework
  - .Net
- Xaml
- Özelleştirme
- Grafik ağırlıklı işlem performansı



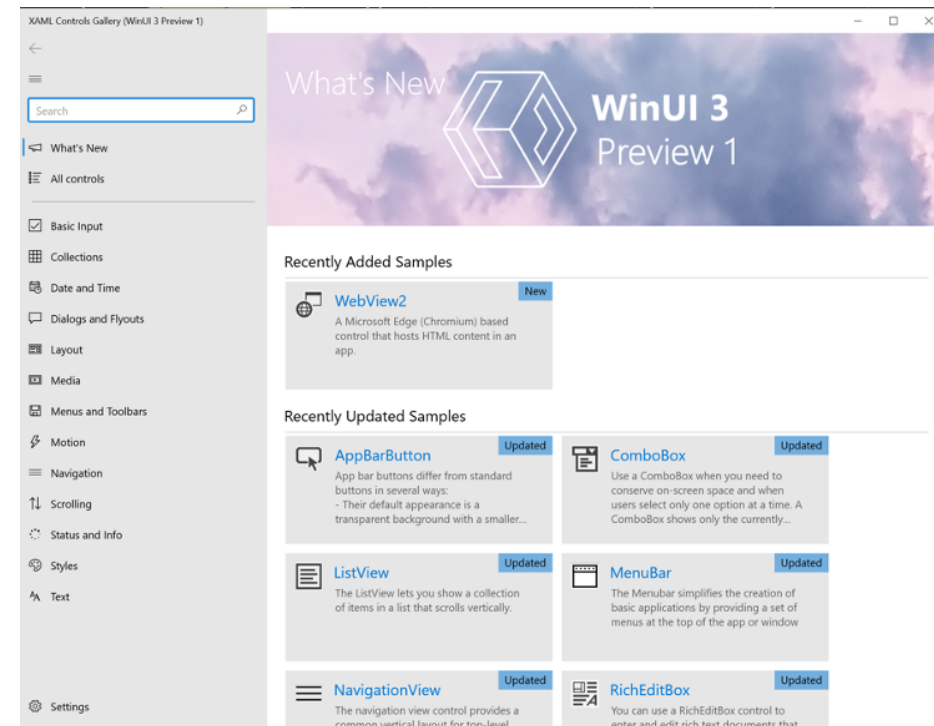
# UWP

- Windows 10
- Xbox, Hololens, Surface Hub
- Xaml
- Fluent Design



# WinUI

- Bileşen ve stiller
- Fluent Design
- WinUI 2
- WinUI 3



# WinUI

	WinUI 3	UWP XAML & WinUI 2	WPF	WinForms	MFC
Windows app types supported	UWP and Win32	UWP	Win32	Win32	Win32
Windows versions supported	Windows 10 (1803+)	Windows 10 (1703+)	Windows XP or higher	Windows XP or higher	Windows XP or higher
Supported on all Windows device families	✓ *	✓			
Native C/C++	✓	✓			✓
.NET 5 Support	✓		✓	✓	
WebView2 (Chromium-based engine)	✓		✓ **	✓ **	
Built-in Fluent Design controls	✓	✓			
Built-in support for modern input (e.g. touch, pen, gamepad*)	✓	✓			
Uses latest DirectX version for graphics performance	✓	✓			
High performance data binding (x:Bind)	✓	✓			
Input Validation	✓		✓	✓	✓



# Olay Tabanlı Programlama

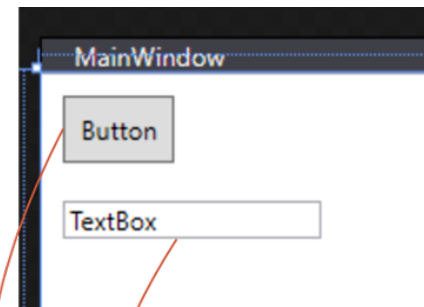
- Event Driven
- Ana döngü
- Kullanıcıdan alınan giriş
- Callback

# Olay Tabanlı Programlama

```
public static string Decode(MWISStream Stream)
{
    string[] D1 = (" " + Stream.D1).Split(" ");
    string[] D2 = Stream.D2.Split(" ");
    string[] S3 = Stream.S3.Split(" ");
    int S1Counter, S2Counter, S3Counter, CodeWord;
    StringBuilder Output = new StringBuilder();

    S1Counter = 0;
    S2Counter = 0;
    S3Counter = 0;
    bool[] BV = Stream.BV.SelectMany(GetBits).ToArray();

    for (int i = 0; i < BV.Length - Stream.RedundantBitLength; i++)
    {
        if (!BV[i])
        {
            CodeWord = Stream.S1[S1Counter++];
            if (CodeWord == 0)
                Output.Append(S3[S3Counter++] + " ");
            else
                Output.Append(D1[CodeWord]);
        }
        else if (BV[i])
        {
            byte[] parca = new byte[2];
            parca[0] = Stream.S2[S2Counter++];
            parca[1] = Stream.S2[S2Counter++];
            CodeWord = BitConverter.ToInt16(parca, 0);
            Output.Append(D2[CodeWord]);
        }
    }
}
```



```
2 references
public partial class MainWindow : Window
{
    References
    public MainWindow()
    {
        InitializeComponent();
    }

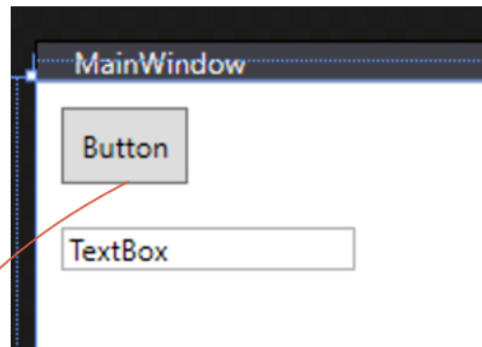
    References
    private void TextBox_TextChanged(object sender, TextChangedEventArgs e)
    {
    }

    References
    private void Button_Click(object sender, RoutedEventArgs e)
    {
    }
}
```

# Olay Tabanlı Programlama

- Event handler
- Arayüze bağlama
- Event args

# Olay Tabanlı Programlama



## XAML

```
<Button x:Name="button" Content="Button" HorizontalAlignment="Left" Margin="10,10,0,0"
        VerticalAlignment="Top" Height="32" Width="52" Click="Button_Click"/>
```

## KOD

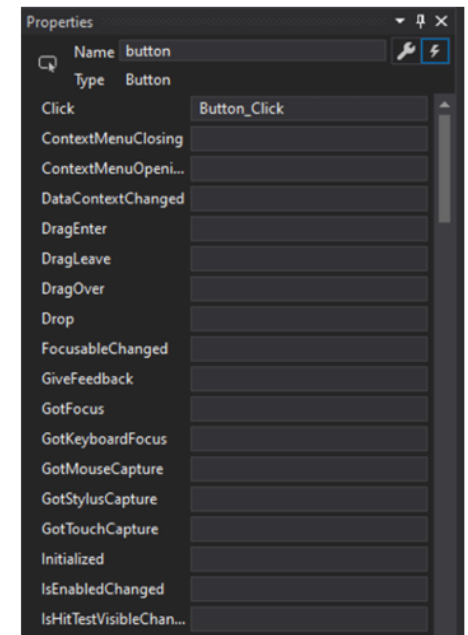
```
0 references
public MainWindow()
{
    InitializeComponent();
    button.Click += Button_Click;
}
```

```
2 references
public partial class MainWindow : Window
{
    0 references
    public MainWindow()
    {
        InitializeComponent();
    }

    0 references
    private void TextBox_TextChanged(object sender, TextChangedEventArgs e)
    {
    }

    0 references
    private void Button_Click(object sender, RoutedEventArgs e)
    {
    }
}
```

## Designer



# Olay Tabanlı Programlama

- Bir bileşen birden fazla olay
- Bir olay birden fazla bileşen

# Olay Tabanlı Programlama

The image illustrates event-based programming in a WPF application. It shows the visual tree, the Properties window, the XAML code, and the C# code.

**Visual Tree:** The visual tree shows a **MainWindow** containing a **Button** and a **TextBox**.

**Properties Window:** The Properties window shows the **textBox** control. The **KeyDown** event is set to **textBox\_KeyDown**, and the **KeyUp** event is set to **textBox\_KeyUp**.

**XAML Code:** The XAML code defines the **textBox** control with the following properties: `HorizontalAlignment="Left", Margin="10,59,0,0", Text="TextBox", TextWrapping="Wrap", VerticalAlignment="Top", Width="120", TextChanged="textBox_TextChanged", KeyDown="textBox_KeyDown", KeyUp="textBox_KeyUp"/>`.

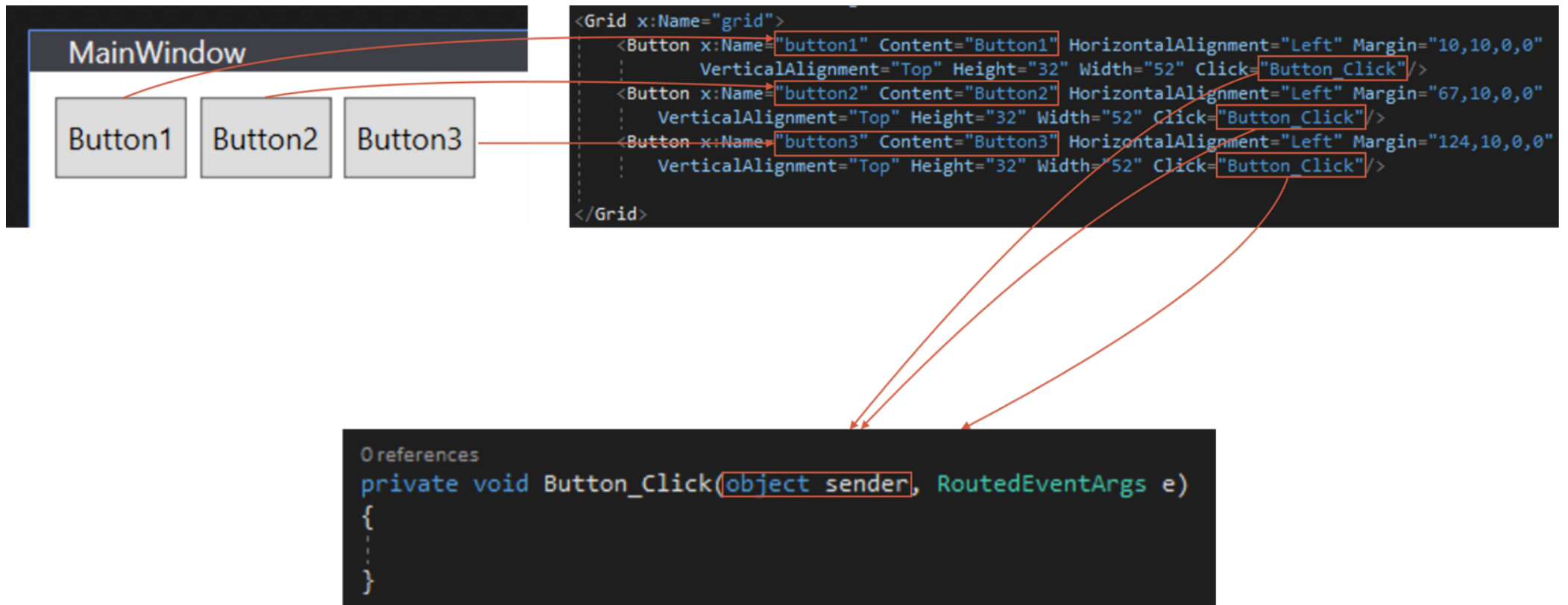
**C# Code:** The C# code shows the event handler methods for the **textBox** control:

```
private void textBox_KeyDown(object sender, KeyEventArgs e)
{
}

private void textBox_KeyUp(object sender, KeyEventArgs e)
{
}

private void textBox_TextChanged(object sender, TextChangedEventArgs e)
{
}
```

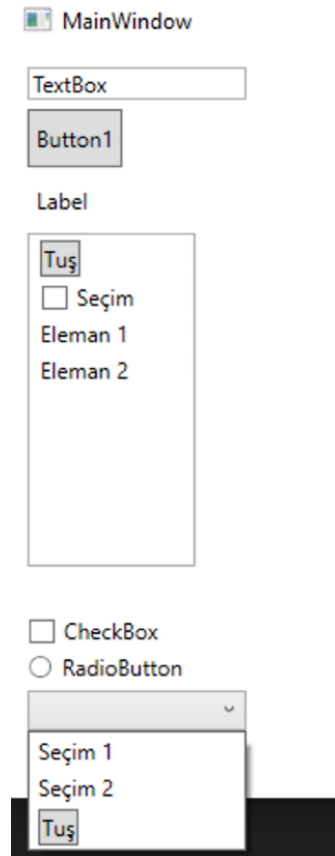
# Olay Tabanlı Programlama



# Temel Bileşenler

## Textbox

- Özellik
  - Text
- Olay
  - KeyDown
  - KeyPress
  - TextChanged

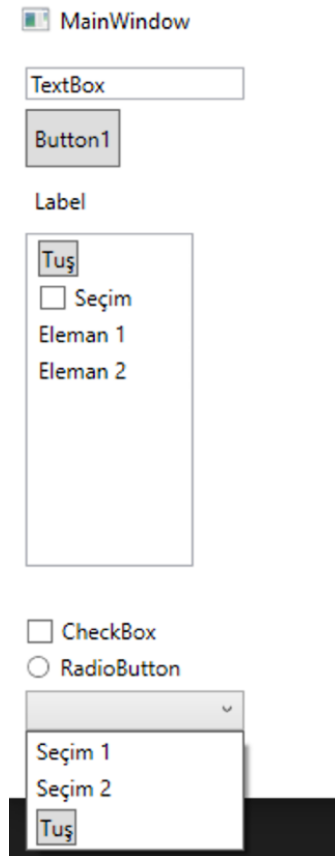




# Temel Bileşenler

## Button

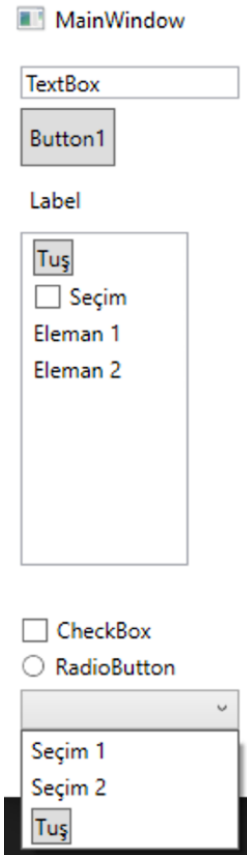
- Özellik
  - Content
- Olay
  - Click
  - MouseDown
  - MouseUp



# Temel Bileşenler

## Label

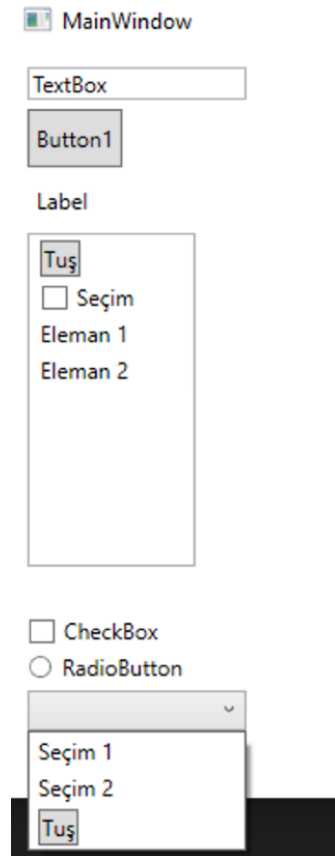
- Özellik
  - Content
- Olay
  - -



# Temel Bileşenler

## Listbox

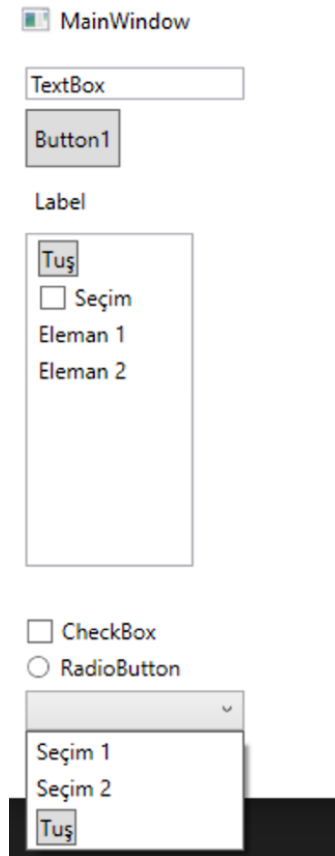
- Özellik
  - Items (ItemSource)
  - SelectedIndex
  - SelectedItem
- Olay
  - SelectionChanged



# Temel Bileşenler

## Checkbox

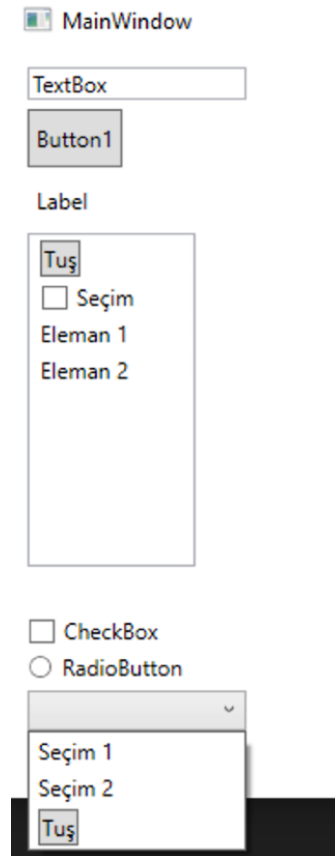
- Özellik
  - Content
  - IsChecked
- Olay
  - Checked



# Temel Bileşenler

## RadioButton

- Özellik
  - Content
  - IsChecked
- Olay
  - Checked



# Temel Bileşenler

## Combobox

- Özellik
  - Items (ItemSource)
  - SelectedItem
  - SelectedIndex
- Olay
  - SelectionChanged

