

WPF Data Binding

TECHNICAL PRESENTATION MILESTONE

CHRIS DEVISSER (20521027)

2015-07-08

Presentation Information

General Questions:

- Note the slide number
- Ask at the end

Syntax Questions:

- Some C# 6 features are used
- If syntax not covered, ask right away

Slides and Code:

- Available on Github: <https://github.com/chrisdevisser/TPM>

Outline

- Background
- Data Binding
- Demo: Basic Math
- Summary
- Questions

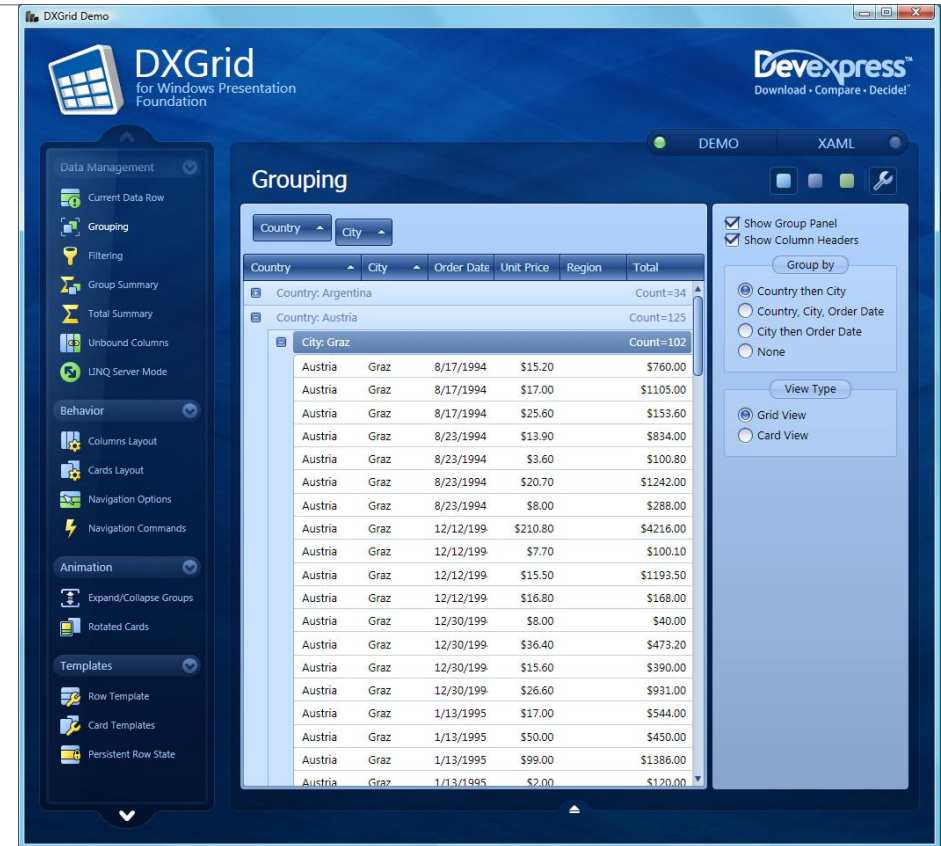
Background

- WPF
- MVVM
- Observer Pattern

WPF

Windows Presentation Foundation

- User interface framework
- XAML
- Windows Forms

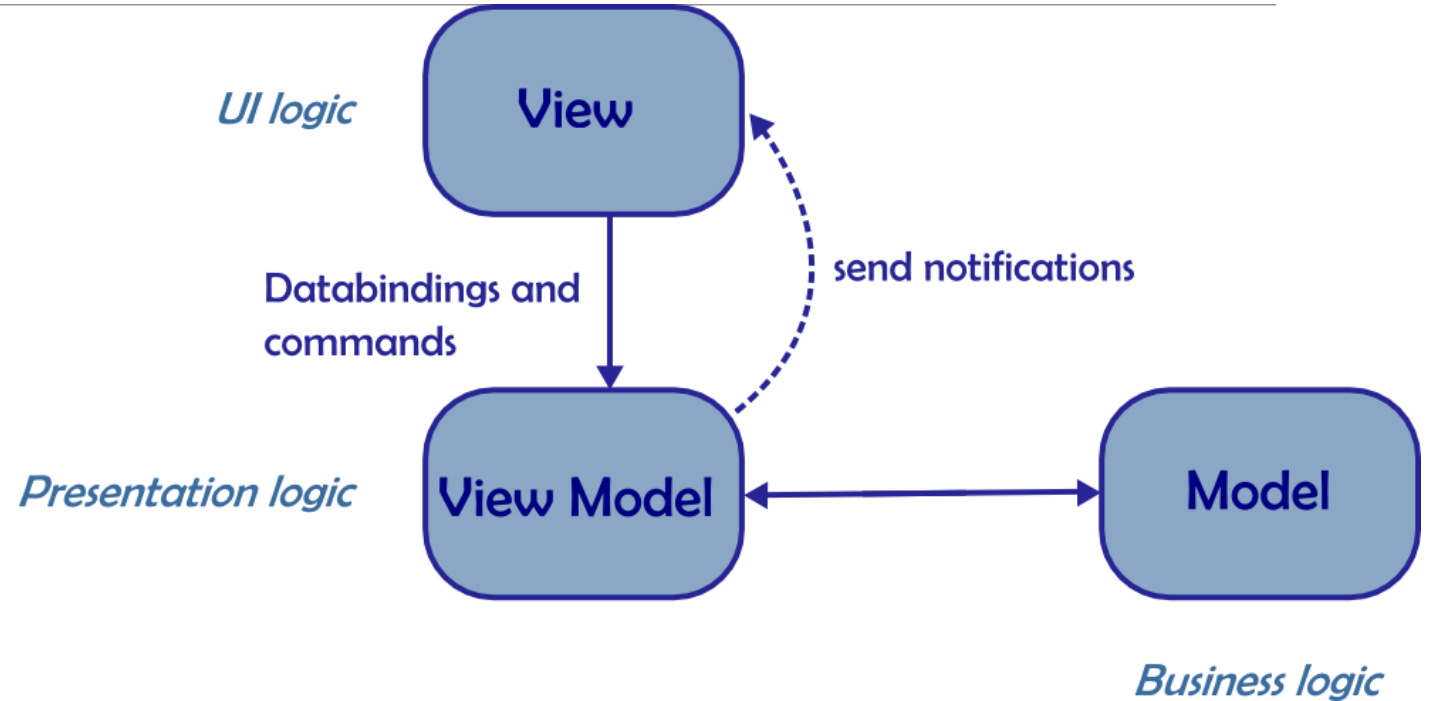


Source: <https://community.devexpress.com/blogs/thinking/DXGrid%20for%20WPF.png>

MVVM

Model – View – View Model

- Decoupled logic
- Little to no UI code
- MVP on steroids
- Testable
- Data binding and commands



Source: <https://benoitpatra.files.wordpress.com/2014/03/mvvm1.png>

Observer Pattern

- Decouples observer and subject
- INotifyPropertyChanged
- Core of MVVM



Source: <http://www.hockeyinsideout.com/wp-content/uploads/2014/10/TV.jpg>

INotifyPropertyChanged Example

- Provide a PropertyChanged event and OnPropertyChanged handler

```
public event PropertyChangedEventHandler PropertyChanged;
```

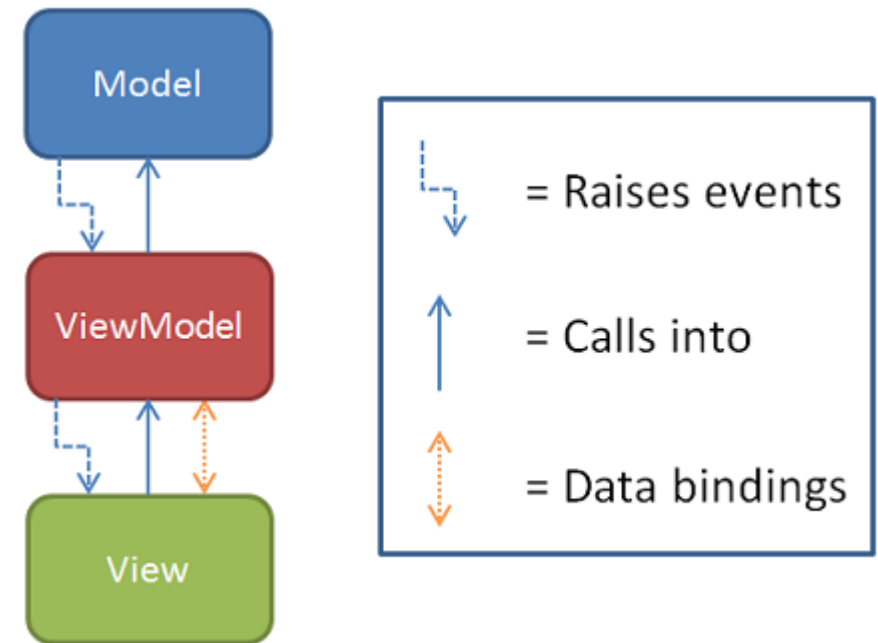
```
protected virtual void OnPropertyChanged([CallerMemberName] string propertyName = null) {  
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));  
}
```

- Notify observers when properties are changed

```
private string _property;  
public string Property {  
    get {return _property;}  
    set {  
        if (_property != value) {  
            _property = value;  
            OnPropertyChanged();  
        }  
    }  
}
```


Data Binding

- View model exposes properties
- View binds directly to those properties
- Value conversions happen if needed
- That's it!



Source: <http://rarcher.azurewebsites.net/Images/binding00b.png>

Preparing to Bind

- We need to set up the view's **data context** (the view model)

```
public partial class MainWindow : Window {  
    public MainWindow() {  
        InitializeComponent();  
        DataContext = new MainWindowViewModel();  
    }  
}
```

Simple Bindings

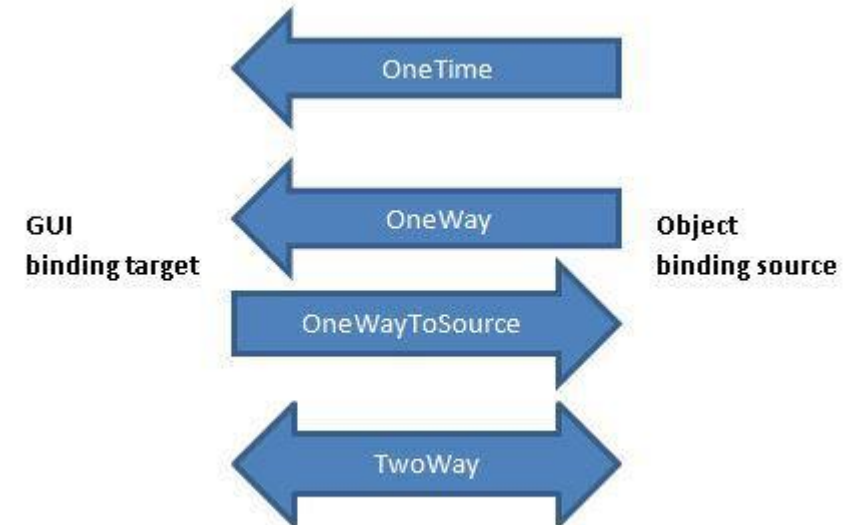
- We just need to specify the name of the property to which to bind

```
<Label x:Name="result" Content="{Binding Result}"/>
```

```
<TextBox x:Name="operation" Text="{Binding Operation, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"/>
```

Binding Modes

- One way (default)
- Two way
- One way to source



Source: <https://csharpcoreprogramming.files.wordpress.com/2014/02/mode.jpg>

Value Conversions

- The binding might not be a perfect match (e.g., different types)

```
[ValueConversion(typeof(int), typeof(string))]  
internal class IntToStringConverter : IValueConverter {  
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture) {  
        return value.ToString();  
    }  
  
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture) {  
        return int.Parse((string)value);  
    }  
}
```

```
<TextBox x:Name="num1" Text="{Binding FirstNumber, UpdateSourceTrigger=PropertyChanged, Mode=TwoWay, Converter={StaticResource IntToStringConverter}}"/>
```

Value Conversions

- The binding might not be a perfect match (e.g., different types)

```
[ValueConversion(typeof(Enum), typeof(bool))]  
internal class EnumToBoolConverter : IValueConverter {  
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture) {  
        return value.Equals(parameter);  
    }  
  
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture) {  
        return value.Equals(true) ? parameter : Binding.DoNothing;  
    }  
}
```

```
<RadioButton x:Name="plus" GroupName="OperationKind" Content="+"  
    IsChecked="{Binding OperationKind, Mode=TwoWay, Converter={StaticResource EnumToBoolConverter}, ConverterParameter={x:Static local:OperationKinds.Plus}}"  
<RadioButton x:Name="minus" GroupName="OperationKind" Content="-"  
    IsChecked="{Binding OperationKind, Mode=TwoWay, Converter={StaticResource EnumToBoolConverter}, ConverterParameter={x:Static local:OperationKinds.Minus}}"
```

Demo: Basic Math

- See provided code and program

[Skip demo slides](#)

Demo: Basic Math

Basic Math Demo

☒ +
☐ -

5 | = 5

Operation:

Demo: Basic Math

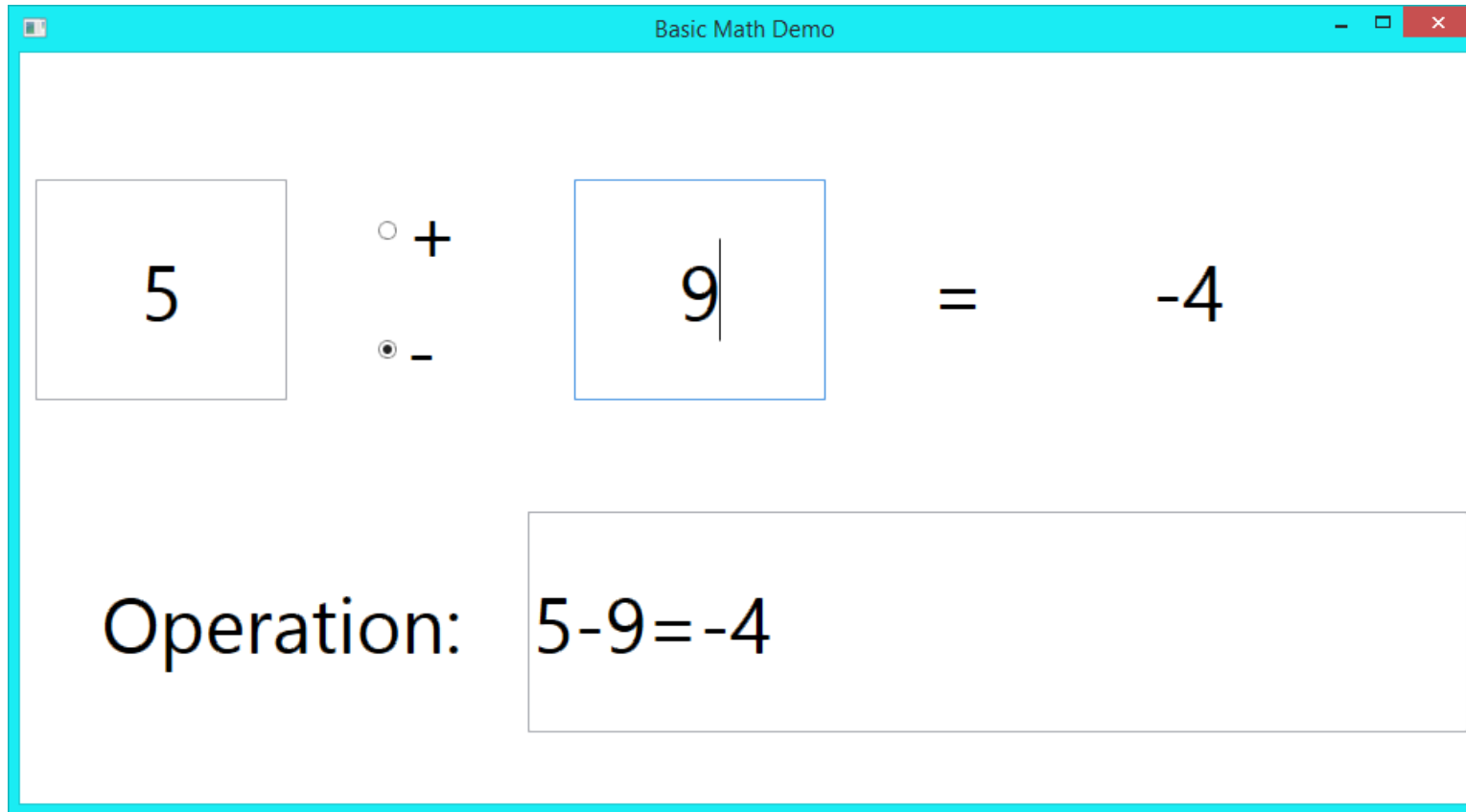
Basic Math Demo

5 + = 5

☐ +
☒ -

Operation: 5-0=5

Demo: Basic Math



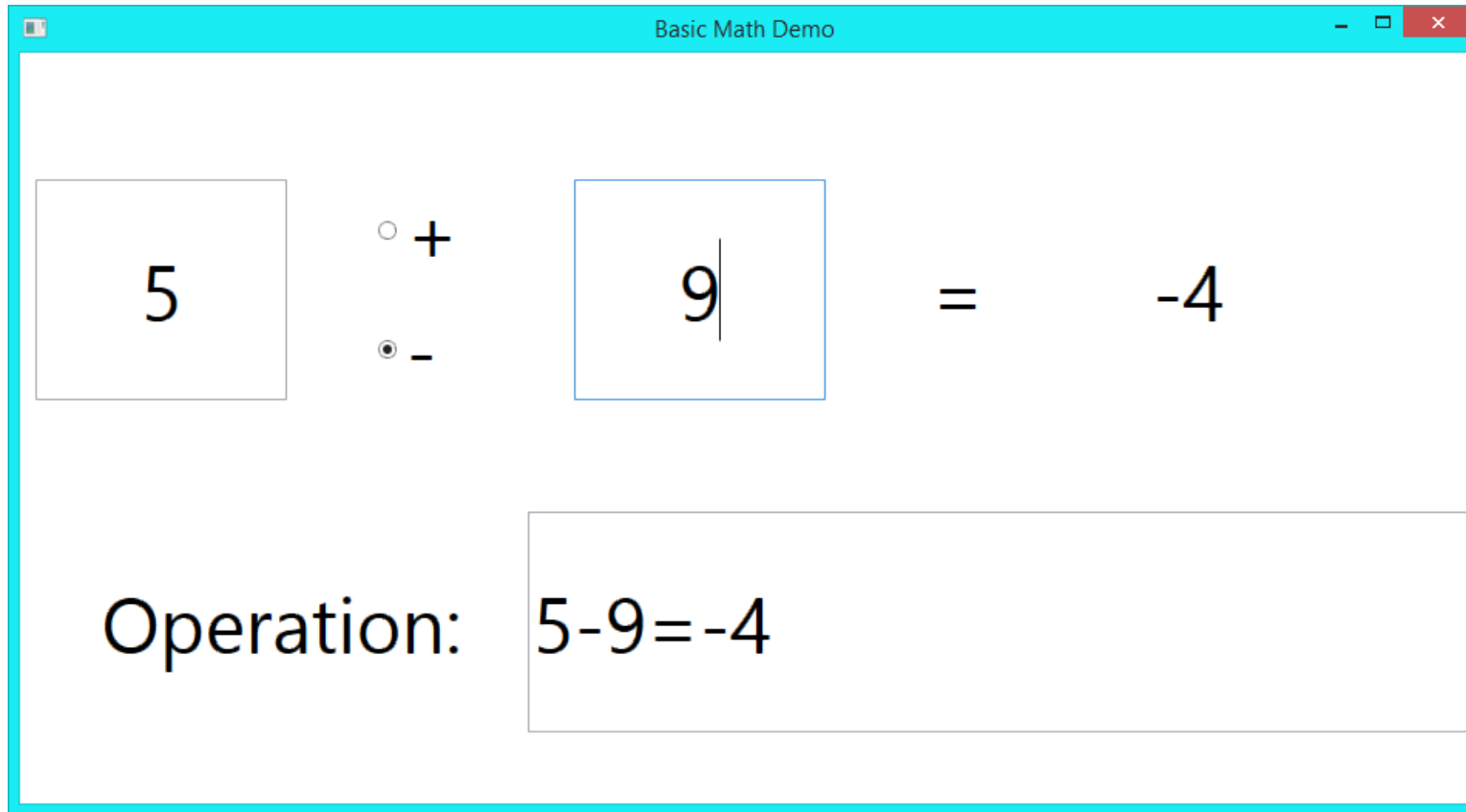
A screenshot of a software window titled "Basic Math Demo". The window has a cyan border and standard Windows-style window controls (minimize, maximize, close) in the top right corner. Inside the window, there is a mathematical expression: a box containing the number "5", followed by a radio button next to a "+" sign and a checked radio button next to a "-" sign, then a box containing the number "9" with a vertical line to its right, followed by an equals sign and the number "-4". Below this expression, the text "Operation:" is followed by a box containing the equation "5-9=-4".

Basic Math Demo

5 ☐ + ☒ - 9 = -4

Operation: 5-9=-4

Demo: Basic Math



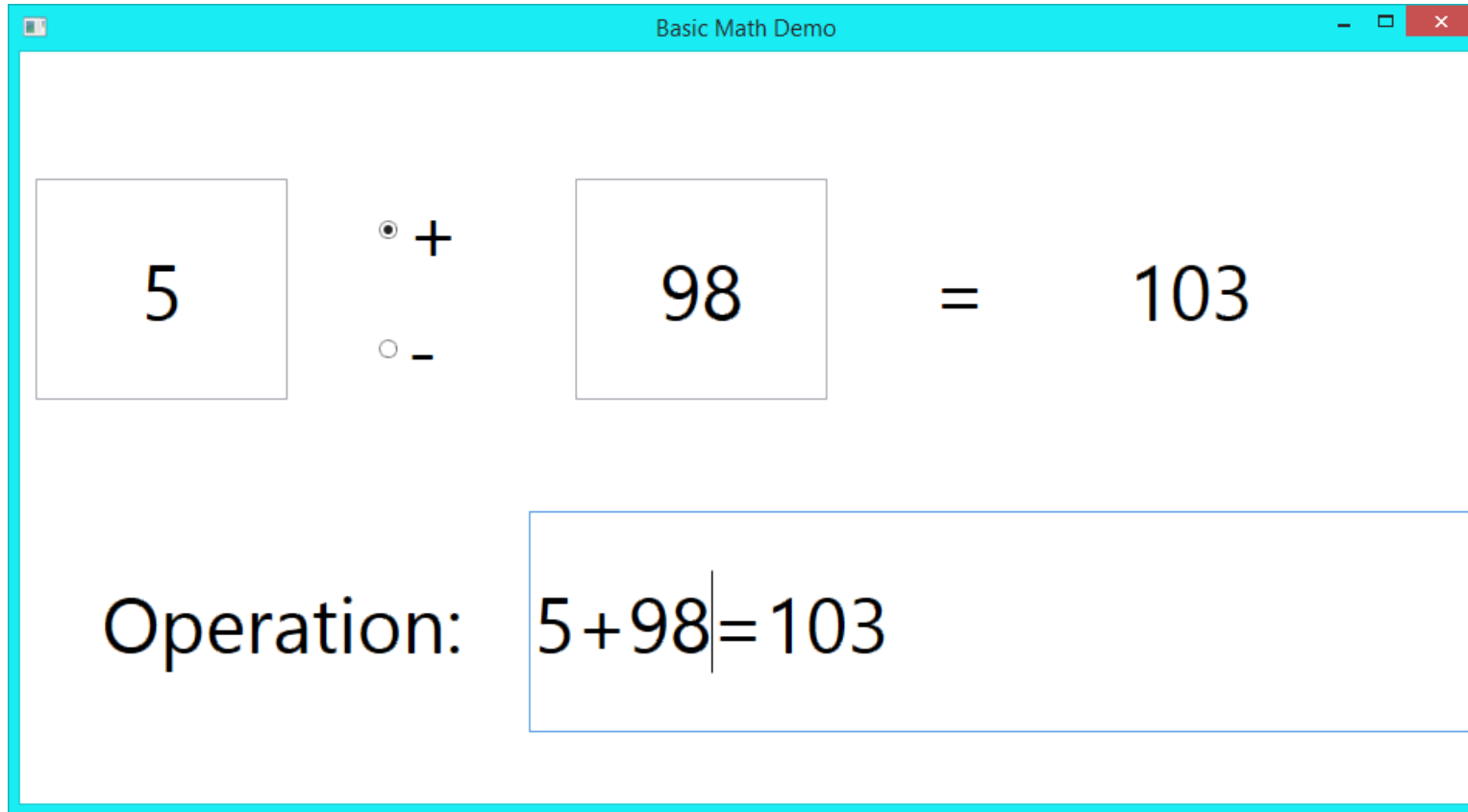
A screenshot of a software window titled "Basic Math Demo". The window has a cyan border and standard Windows window controls (minimize, maximize, close) in the top right corner. Inside the window, there is a mathematical expression: a box containing the number "5", followed by a radio button next to a "+" sign and a checked radio button next to a "-" sign, then a box containing the number "9" with a vertical line to its right, followed by an equals sign and the number "-4". Below this expression, the text "Operation:" is followed by a box containing the equation "5-9=-4".

Basic Math Demo

5 ☐ + ☒ - 9 | = -4

Operation: 5-9=-4

Demo: Basic Math



A screenshot of a software window titled "Basic Math Demo". The window has a cyan border and standard Windows window controls (minimize, maximize, close) in the top right corner. Inside the window, there is a visual representation of a math problem. On the left, a light gray square box contains the number "5". To its right are two radio buttons; the top one is selected and is next to a "+" sign, while the bottom one is unselected and is next to a "-" sign. Further right is another light gray square box containing the number "98". To the right of this box is an equals sign "=", followed by the number "103". Below this visual representation, the text "Operation:" is displayed. To the right of this text is a light gray rectangular box containing the text "5+98|=103", where the vertical bar is positioned between "98" and "=". The entire window is set against a white background.

Basic Math Demo

5 + 98 = 103

Operation: 5+98|=103

Summary

- Data bindings:
 - Are powerful
 - Do not require code to set up
 - UI designers do not need to write C#
 - Contribute to easy creation of reactive UIs

Questions



Source: <http://images.clipartpanda.com/cool-question-marks-QuestionMarks.jpg>