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

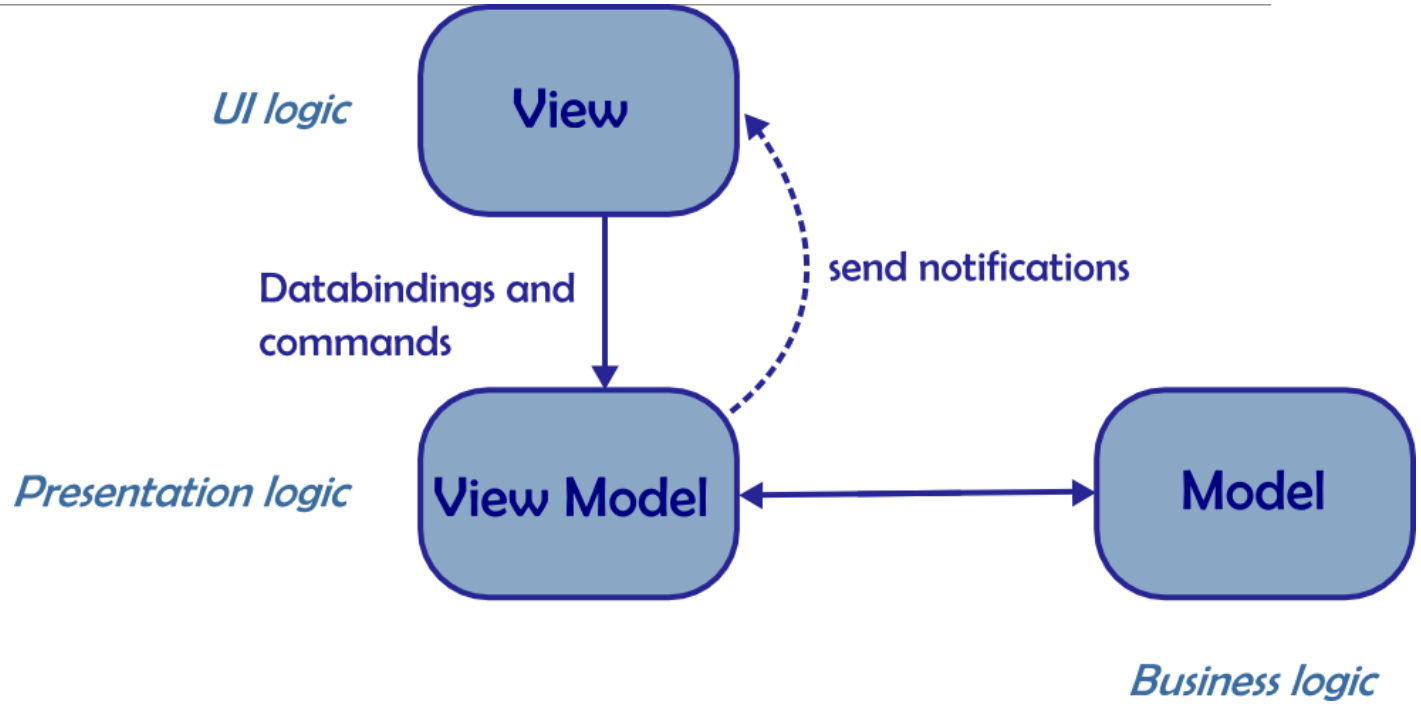
- User interface framework
- XAML
- Windows Forms

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

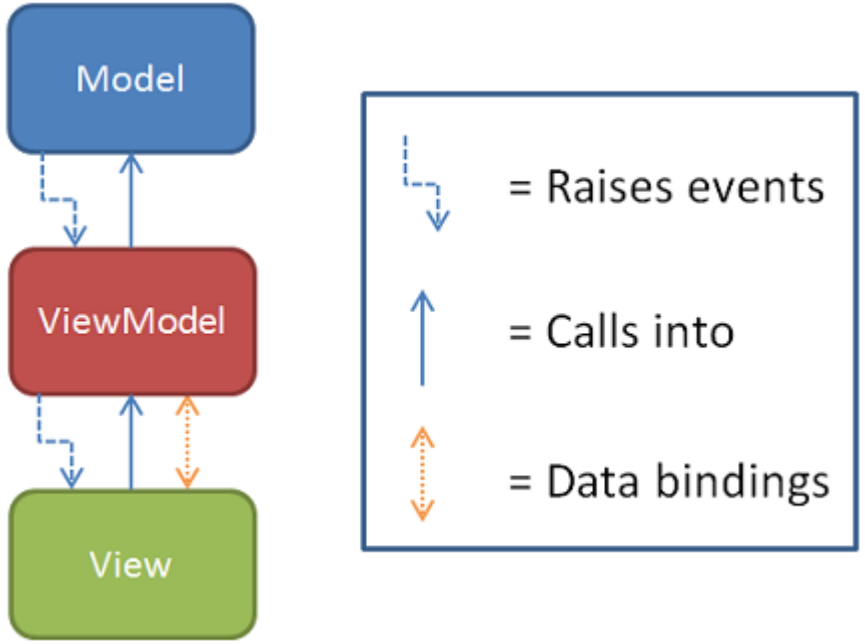
```
public event PropertyChangedEventHandler PropertyChanged;


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();
    }
}
```

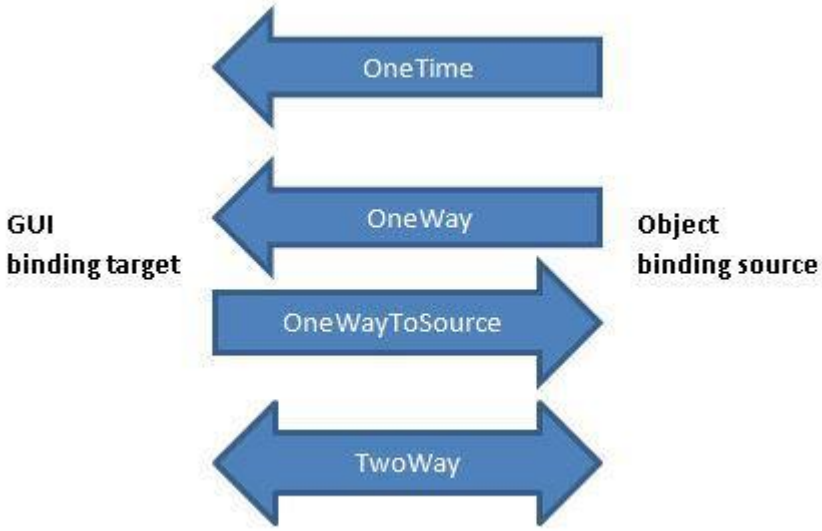
Simple Bindings

- 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://csharphardcoreprogramming.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;
    }
}
```

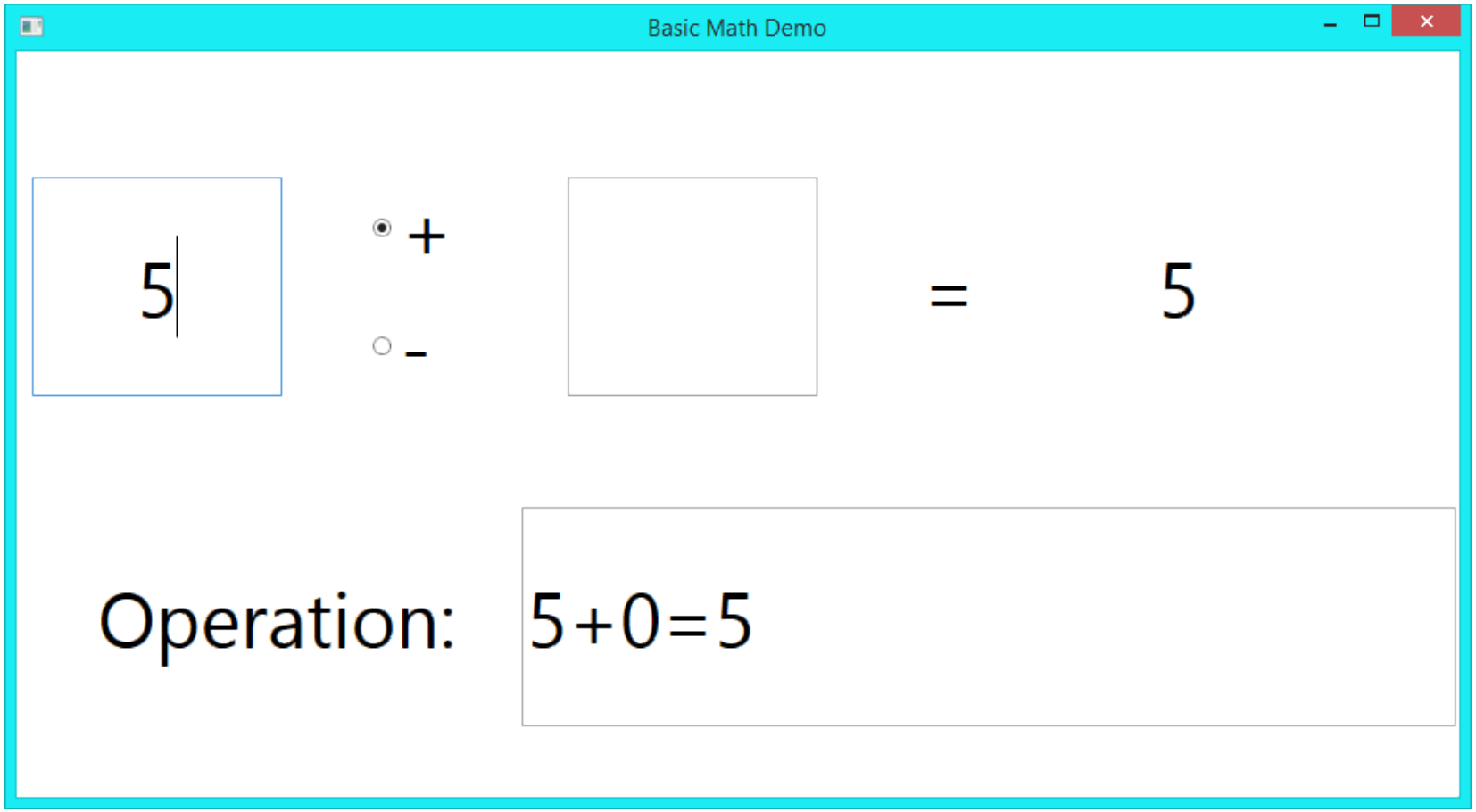
```
<RadioButton x:Name="plus" GroupName="OperationKind" Content="+"
             IsChecked="{Binding OperationKind, Mode=TwoWay, Converter={StaticResource EnumToBoolConverter}, ConverterParameter={x:Static local:OperationKinds.Plus}}"
<RadioButton x:Name="minus" GroupName="OperationKind" Content="-"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

Demo: Basic Math

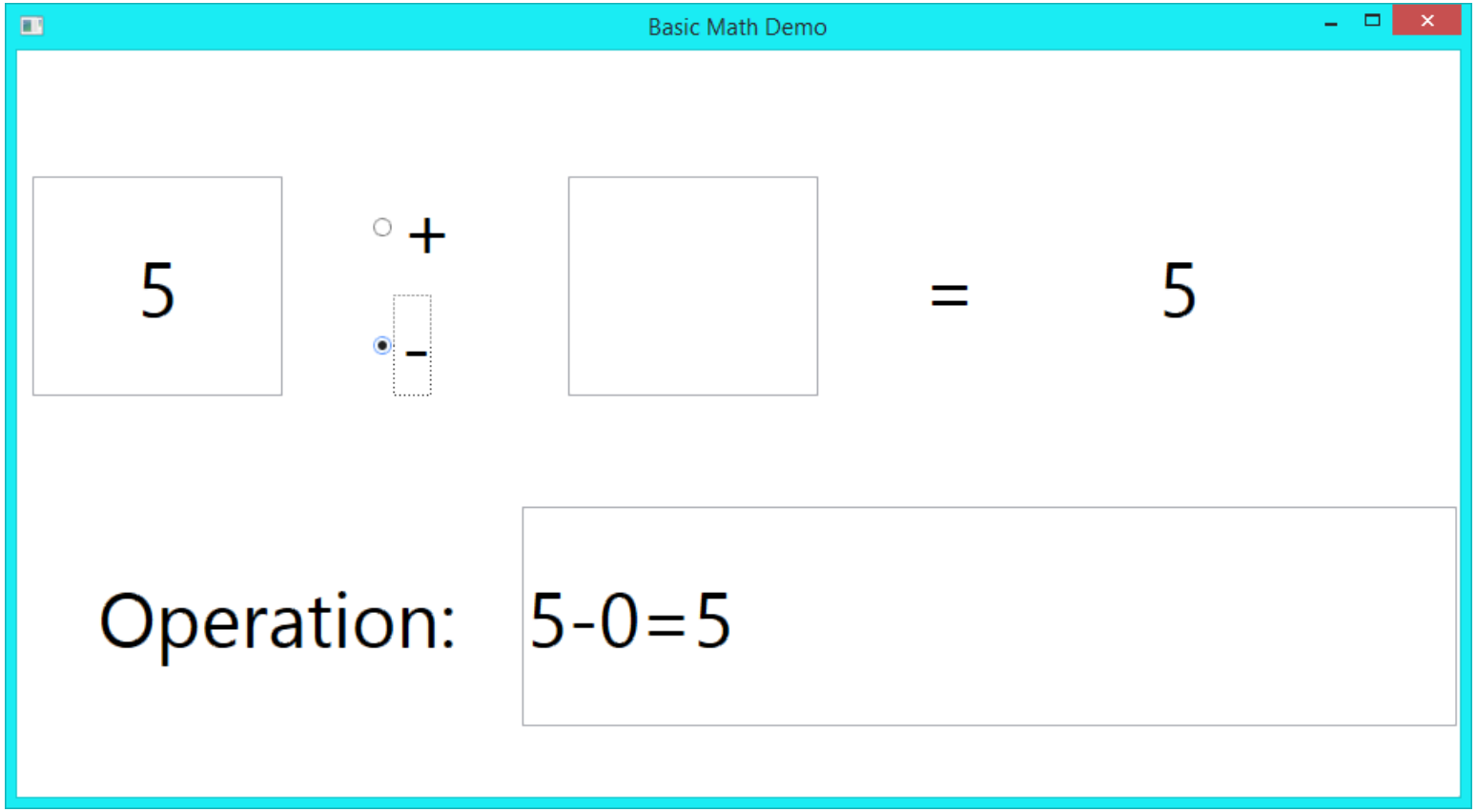
- See provided code and program

Skip demo slides

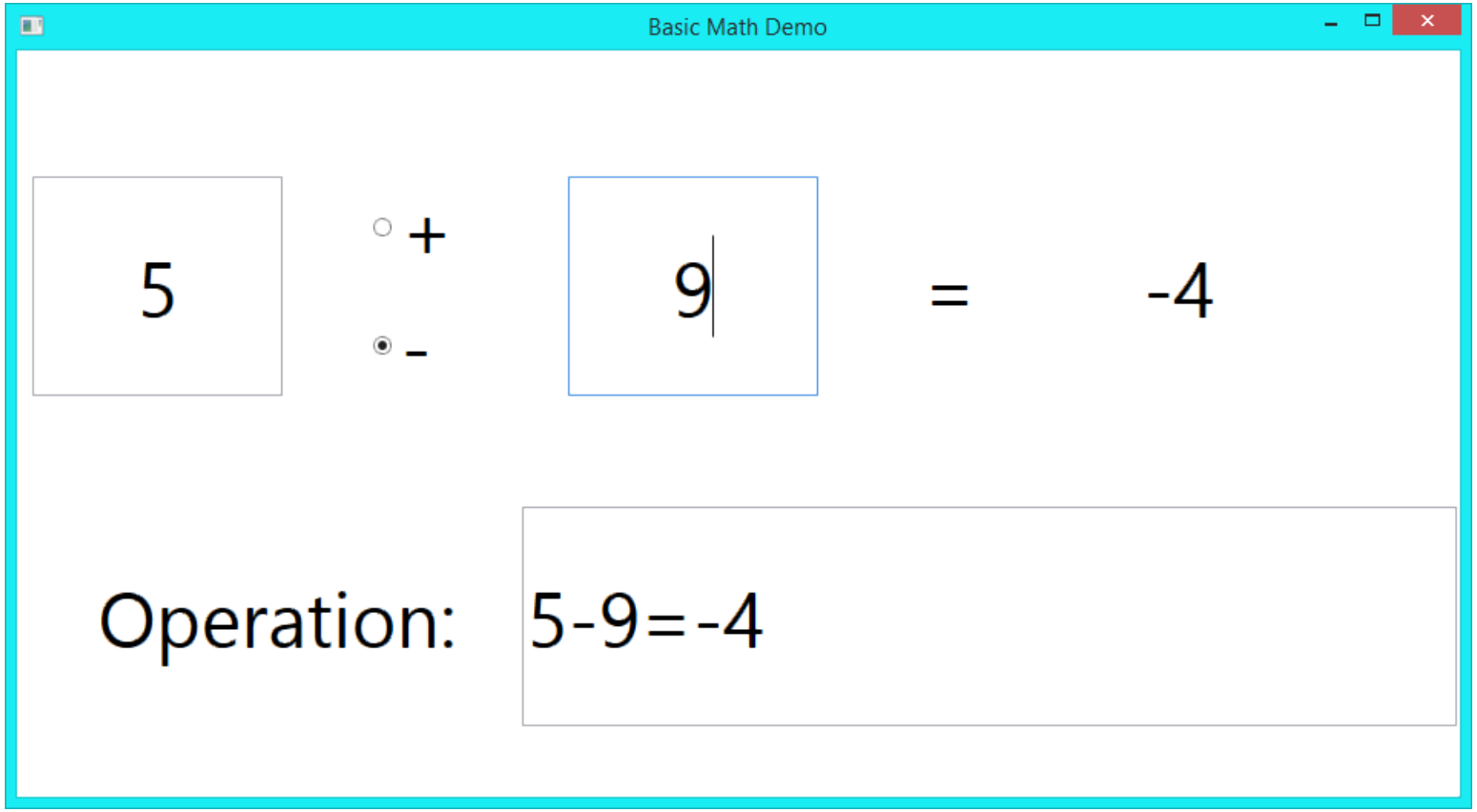
Demo: Basic Math



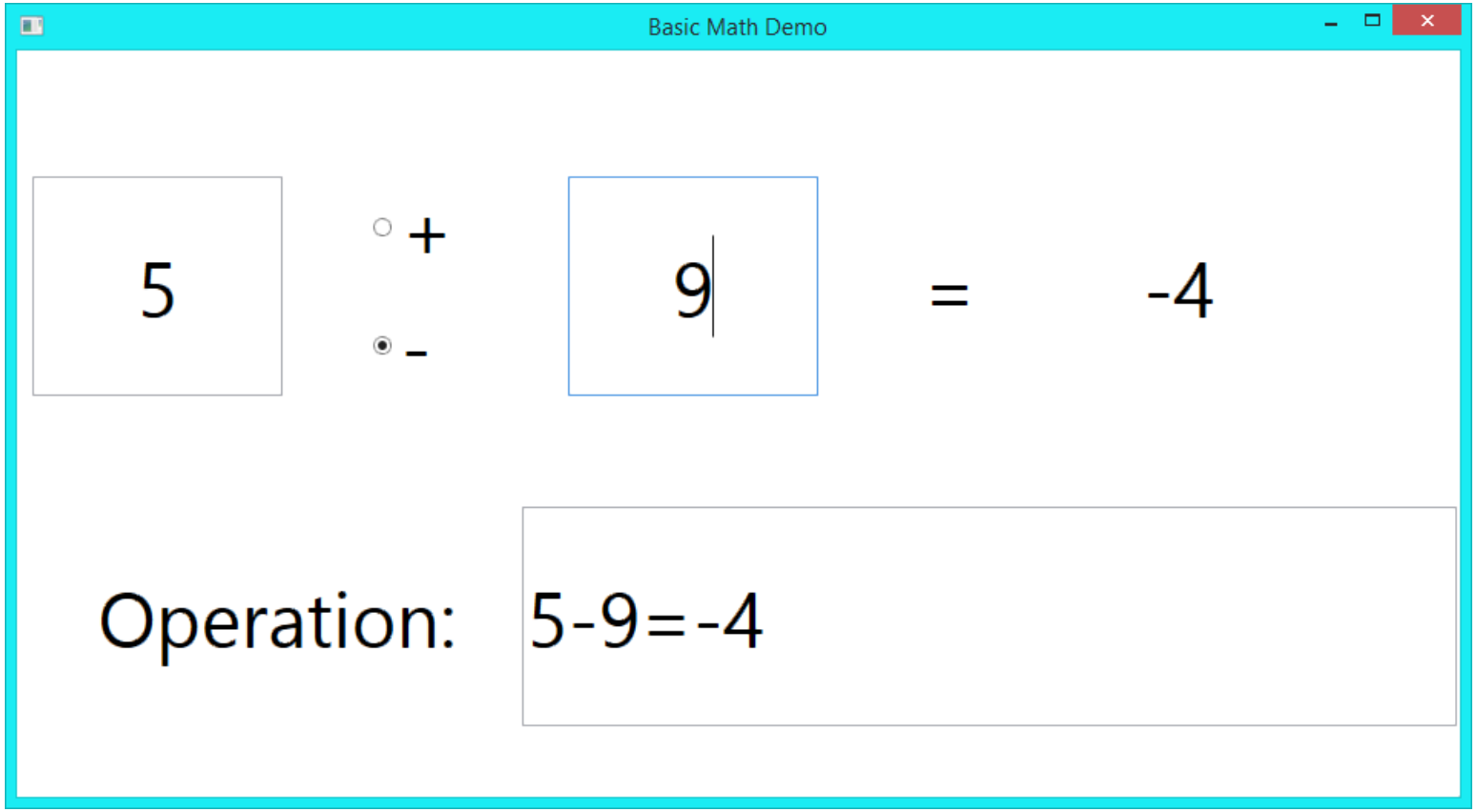
Demo: Basic Math



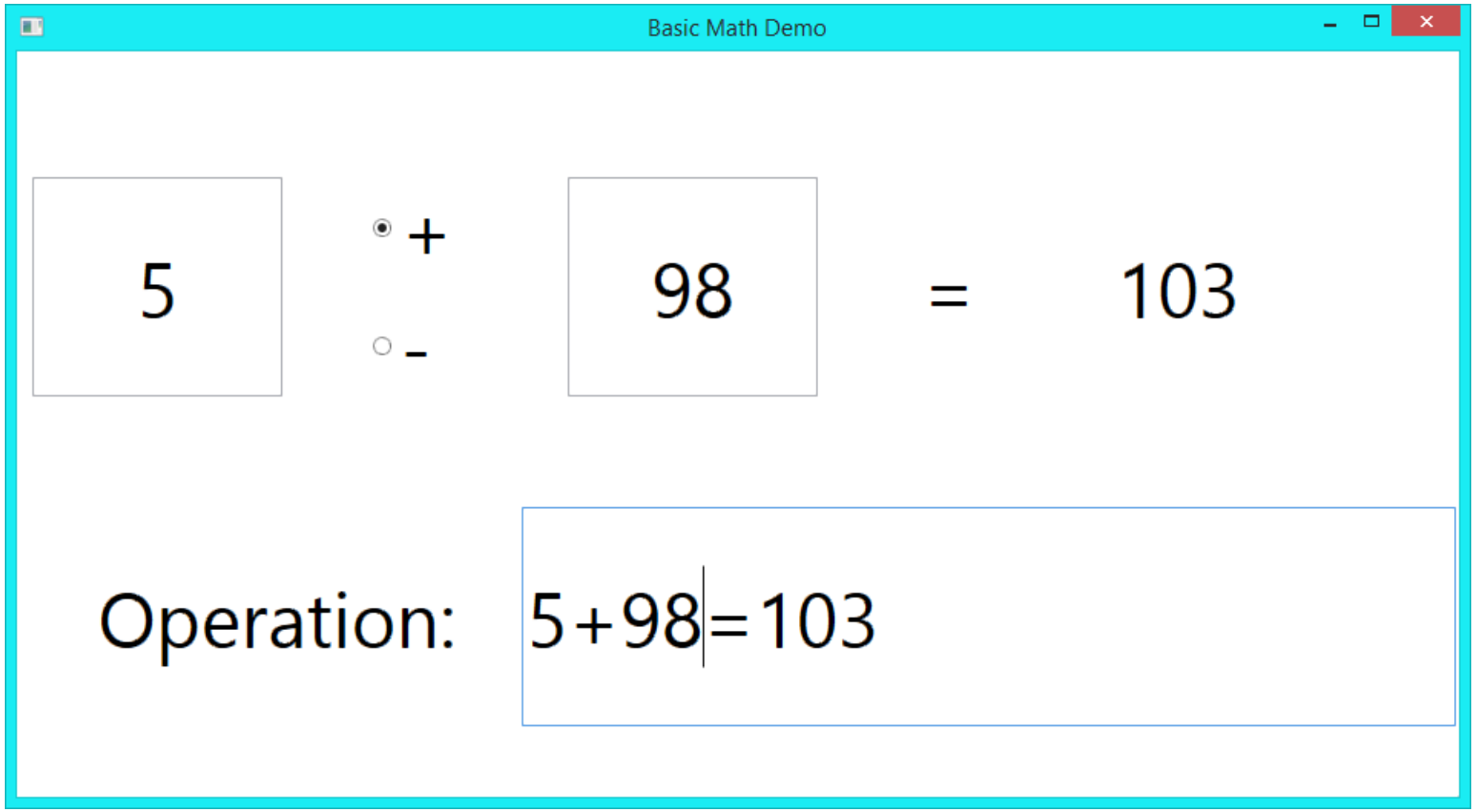
Demo: Basic Math



Demo: Basic Math



Demo: Basic Math



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