Промышленное программирование

Лекция 11. AvaloniaUI (часть 2: Bindings)

1. Свойства зависимостей
2. Привязки
3. Ресурсы (XAML)
4. Ресурсы (Assets)

1. **Свойства зависимостей**
2. Привязки
3. Ресурсы (XAML)
4. Ресурсы (Assets)

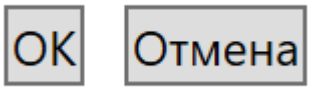
AvaloniaProperty

```
<Border Background="Magenta" />
```

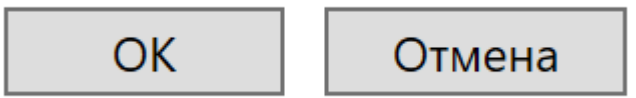
```
class Border : ...
{
  static readonly StyledProperty<IBrush?> BackgroundProperty =
    AvaloniaProperty.Register<Border, IBrush?>(nameof(Background));

  static Border()
  {
    AffectsRender<Border>(BackgroundProperty, ...);
    ...
  }

  IBrush? Background
  {
    get => GetValue(BackgroundProperty);
    set => SetValue(BackgroundProperty, value);
  }
}
```

Зачем это нужно: механизм стилей

```
<Window>
    <StackPanel Orientation="Horizontal" Margin="10">
        <Button Content="OK" />
        <Button Content="Отмена" Margin="10,0,0,0" />
    </StackPanel>
</Window>
```




```
<Window>
    <Window.Styles>
        <Style Selector="Button">
            <Setter Property="MinWidth" Value="70" />
            <Setter Property="Padding" Value="16,2,16,2" />
        </Style>
    </Window.Styles>
    <StackPanel Orientation="Horizontal" Margin="10">
        <Button Content="OK" />
        <Button Content="Отмена" Margin="10,0,0,0" />
    </StackPanel>
</Window>
```



Зачем это нужно: подписка на изменения

```
using Avalonia;
using Avalonia.Controls;

partial class MainWindow : Window
{
    …
    {
        var background = MyTextBox.GetObservable(BackgroundProperty);
        background.Subscribe(value => { ... });
    }
}
```

Прикрепляемые свойства

```
<DragCanvas>
    <Image DragCanvas.IsDraggable="True" ... />
    <Image DragCanvas.IsDraggable="False" ... />
</DragCanvas>
```

Прикрепляемые свойства

```
using Avalonia;
using Avalonia.Controls;

public class DragCanvas : Canvas
{
    public static readonly AttachedProperty<bool?> IsDraggableProperty =
        AvaloniaProperty.RegisterAttached<DragCanvas, AvaloniaObject, bool?>("IsDraggable");

    public static bool? GetIsDraggable(AvaloniaObject target) =>
        target.GetValue(IsDraggableProperty);

    public static void SetIsDraggable(AvaloniaObject target, bool? value) =>
        target.SetValue(IsDraggableProperty, value);

    ...
}
```

1. Свойства зависимостей
2. **Привязки**
3. Ресурсы (XAML)
4. Ресурсы (Assets)

Импорт пространств имён в XAML

```
namespace MyNamespace
{
    public static class MyClass
    {
        public static string MyProperty => string.Empty;
    }
}

<Window xmlns="https://github.com/avaloniaui"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:local="using:MyNamespace"
        x:Class="...">
    <Label Content="{x:Static local:MyClass.MyProperty}" .../>
</Window>
```

Привязка к данным

```
<CheckBox Name="IsRingWorld" .../>

<!-- Вариант 1 -->
<Button IsEnabled="{Binding IsChecked, ElementName=IsRingWorld}" />

<!-- Вариант 2 -->
<Button IsEnabled="{Binding #IsRingWorld.IsChecked}" />
```

Конвертеры значений

```
using System;
using System.Globalization;

using Avalonia.Data.Converters;

namespace MyApp.Converters;

public class IntToEvenConverter : IValueConverter
{
    public object? Convert(object? value, Type targetType, object? parameter, CultureInfo culture) =>
        value is int intValue && intValue % 2 == 0;

    public object? ConvertBack(object? value, Type targetType, object? parameter, CultureInfo culture) =>
        throw new NotImplementedException();
}
```

Конвертеры значений

```
<Window x:Class="MyApp.MainWindow"
        xmlns="..."
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:converters="using:MyApp.Converters">
    <Window.Resources>
        <converters:IntToEvenConverter x:Key="IntToEvenConverter" />
    </Window.Resources>
    ...
        <ListBox Name="IntegerList"
                 Items="{Binding}"
                 Margin="0,10,0,0" />
    ...
        <Button IsEnabled="{Binding #IntegerList.SelectedItem,
                            Converter={StaticResource IntToEvenConverter},
                            Mode=OneWay}"/>
    ...
</Window>
```

Подробнее о привязке

https://docs.avaloniaui.net/docs/data-binding

1. Свойства зависимостей
2. Привязки
3. **Ресурсы (XAML)**
4. Ресурсы (Assets)

Определение ресурсов

```
<Window x:Class="..."
        xmlns="..."
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Window.Resources>
        <SolidColorBrush x:Key="MyDefaultBackground" Color="Magenta" />
    </Window.Resources>
    ...
        <Panel Background="{StaticResource MyDefaultBackground}" />
    ...
</Window>
```

Словари ресурсов

Объявление словаря ресурсов в отдельном XAML-файле:

```
<!-- MyResources.xaml -->
<ResourceDictionary xmlns="..." />
    ...
</ResourceDictionary>
```

Подключение словаря:

```
<Window ...>
    <Window.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="MyResources.xaml" />
                ...
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Window.Resources>
</Window>
```

1. Свойства зависимостей
2. Привязки
3. Ресурсы (XAML)
4. **Ресурсы (Assets)**

Добавление в проект

1. Добавить файлы в проект

Как правило, в отдельную директорию, например, Assets

2. Указать системе сборки, что данные файлы являются файлами ресурсов

В IDE в настройках файла для значения свойства Build Action выставить AvaloniaResource.

Выставление Build Action эквивалентно добавлению следующих элементов в **csproj**-файл:

```
<ItemGroup>
    <AvaloniaResource Include="Assets/*" />
</ItemGroup>
```

Пример: Image

```
<Image Source="Assets/List.png" />
```