## Статические и динамические ресурсы в WPF

Pесурсы могут быть статическими и динамическими. Статические ресурсы устанавливается только один раз. А динамические ресурсы могут меняться в течение работы программы. Например, у нас есть ресурс кисти:

|  |  |
| --- | --- |
| 1 | <SolidColorBrush Color="LightGray" x:Key="buttonBrush" /> |

Для установки ресурса в качестве статического используется выражение StaticResource:

|  |  |
| --- | --- |
| 1 | <Button MaxWidth="80" MaxHeight="40" Content="OK" Background="{StaticResource buttonBrush}" /> |

А для установки ресурса как динамического применяется выражение DynamicResource:

|  |  |
| --- | --- |
| 1 | <Button MaxWidth="80" MaxHeight="40" Content="OK" Background="{DynamicResource buttonBrush}" /> |

Причем один и тот же ресурс может быть и статическим и динамическим. Чтобы посмотреть различие между ними, добавим к кнопке обработчик нажатия:

|  |  |
| --- | --- |
| 1  2 | <Button x:Name="button1" MaxWidth="80" MaxHeight="40" Content="OK"          Background="{DynamicResource buttonBrush}"  Click="Button\_Click" /> |

А в файле кода определим в этом обработчике изменение ресурса:

|  |  |
| --- | --- |
| 1  2  3  4 | private void Button\_Click(object sender, RoutedEventArgs e)  {      this.Resources["buttonBrush"] = new SolidColorBrush(Colors.LimeGreen);  } |

И если после запуска мы нажмем на кнопку, то ресурс изменит свой цвет, что приведет к изменению цвета кнопки. Если бы ресурс был бы определен как статический, то изменение цвета кисти никак бы не повлияло на цвет фона кнопки.

В то же время надо отметить, что мы все равно может изменить статический ресурс - для этого нужно менять не сам объект по ключу, а его отдельные свойства:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | private void Button\_Click(object sender, RoutedEventArgs e)  {      // данное изменение будет работать и со статическими ресурсами      SolidColorBrush buttonBrush = (SolidColorBrush)this.TryFindResource("buttonBrush");      buttonBrush.Color = Colors.LimeGreen;  } |

### **Иерархия ресурсов**

Еще одно различие между статическими и динамическими ресурсами касается поиска системой нужного ресурса. Так, при определении статических ресурсов ресурсы элемента применяются только к вложенным элементам, но не к внешним контейнерам. Например, ресурс кнопки мы не можем использовать для грида, а только для тех элементов, которые будут внутри этой кнопки. Поэтому, как правило, большинство ресурсов определяются в коллекции Window.Resources в качестве ресурсов всего окна, чтобы они были доступны для любого элемента данного окна.

В случае с динамическими ресурсами такого ограничения нет.

### **Установка динамических ресурсов в коде C#**

Ранее мы рассмотрели, как устанавливать в коде C# статические ресурсы:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | LinearGradientBrush gradientBrush = new LinearGradientBrush();  gradientBrush.GradientStops.Add(new GradientStop(Colors.LightGray, 0));  gradientBrush.GradientStops.Add(new GradientStop(Colors.White, 1));  this.Resources.Add("buttonGradientBrush", gradientBrush);    button1.Background = (Brush)this.TryFindResource("buttonGradientBrush"); |

Установка динамического ресурса призводится немного иначе:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | LinearGradientBrush gradientBrush = new LinearGradientBrush();  gradientBrush.GradientStops.Add(new GradientStop(Colors.LightGray, 0));  gradientBrush.GradientStops.Add(new GradientStop(Colors.White, 1));  this.Resources.Add("buttonGradientBrush", gradientBrush);    button1.SetResourceReference(Button.BackgroundProperty, "buttonGradientBrush"); |

Для установки применяется метод **SetResourceReference()**, который есть у большинства элементов WPF. Первым параметром в него передается свойство зависимости объекта, для которого предназначен ресурс, а вторым - ключ ресурса. Общая форма установки:

|  |  |
| --- | --- |
| 1 | объект.SetResourceReference(Класс\_объекта.Свойство\_КлассаProperty, ключ\_ресурса); |

### **Элементы StaticResource и DynamicResource**

В ряде случае в разметке XAML бывает удобнее использовать не расширения разметки тип "{StaticResource}", а полноценные элементы DynamicResource и StaticResource. Например:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <Window x:Class="ResourcesApp.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:ResourcesApp"          mc:Ignorable="d"          Title="Ресурсы" Height="250" Width="300">      <Window.Resources>          <SolidColorBrush Color="LimeGreen" x:Key="buttonBrush" />      </Window.Resources>      <Grid>          <Button x:Name="button1" MaxWidth="80" MaxHeight="40" Content="OK">              <Button.Background>                  <DynamicResource ResourceKey="buttonBrush" />              </Button.Background>          </Button>      </Grid>  </Window> |

Элементы StaticResource и DynamicResource имеют свойство **ResourceKey**, которое позволяет установить ключ применяемого ресурса.

Особенно это эффективно может быть с контейнерами:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <Window x:Class="ResourcesApp.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:ResourcesApp"          mc:Ignorable="d"          Title="Ресурсы" Height="250" Width="300">      <Window.Resources>          <Button x:Key="buttonRes" x:Shared="False" Content="OK" MaxHeight="40" MaxWidth="80" Background="Azure" />      </Window.Resources>      <StackPanel>          <StaticResource ResourceKey="buttonRes" />          <StaticResource ResourceKey="buttonRes" />          <StaticResource ResourceKey="buttonRes" />          <StaticResource ResourceKey="buttonRes" />      </StackPanel>  </Window> |