

Công nghệ .NET

Bài 11 – Style, Template

Nguyễn Thành Trung – Khoa CNTT

Email: trung.nguyenthanh1@phenikaa-uni.edu.vn

WPF

- Style (kiểu)
- Template (mẫu)

Style

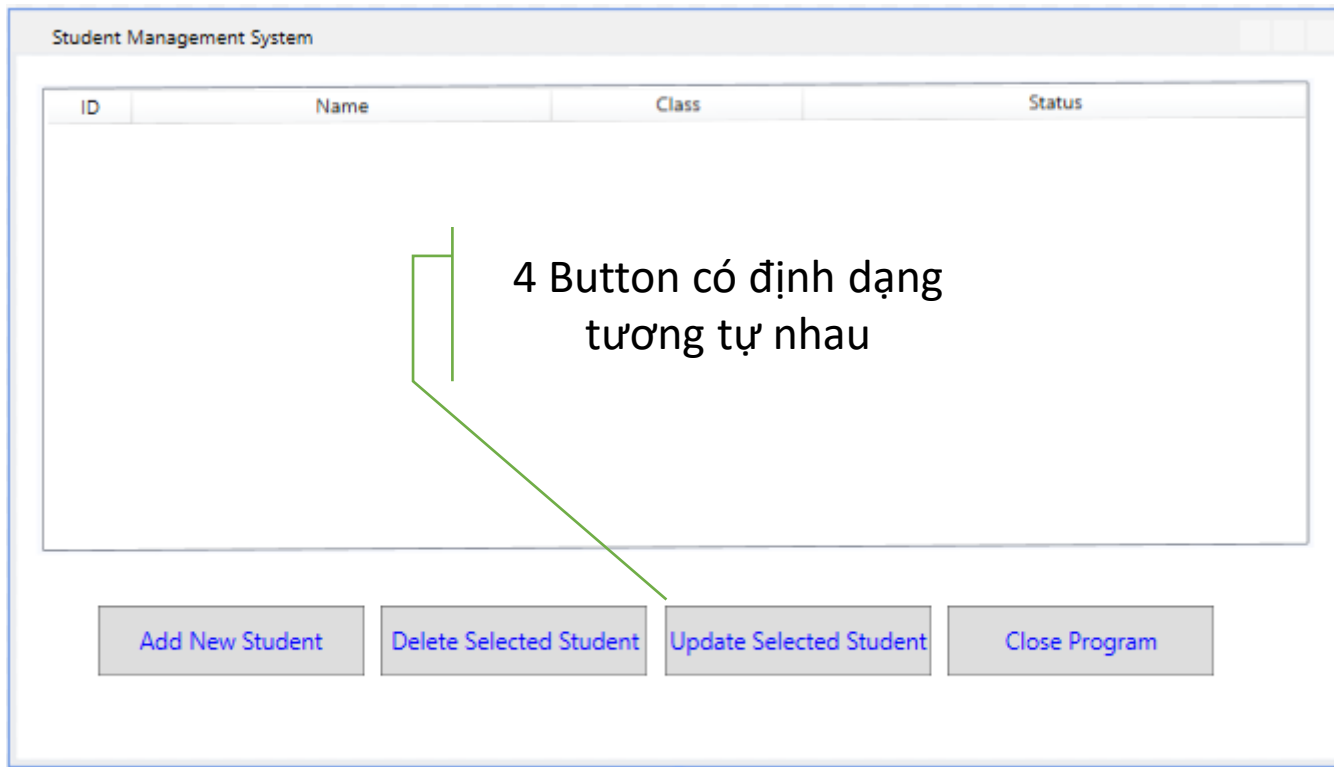
- Thực tế: khi bạn muốn sử dụng cùng một cỡ và màu chữ, định dạng khoảng cách,... cho các điều khiển TextBlock khác nhau thì làm như nào ?

Style

- WPF cung cấp Style để nhóm một tập các thuộc tính và gán cho các điều khiển.
 - ⇒ Hay gọi cách khác là XAML CSS
- Cấu trúc
 - <Style ... >
 - ...
 - </Style>

Style

- Ví dụ



Style

- Ví dụ
 - Thay vì đặt định dạng cho từng Button, ta đặt chung cho chúng một style

```
<StackPanel Orientation="Horizontal" Margin="43,322,59,0">
    <StackPanel.Resources>
        <Style TargetType="Button">
            <Setter Property="Foreground" Value="■"Blue" />
            <Setter Property="FontSize" Value="14" />
            <Setter Property="Margin" Value="5" />
        </Style>
    </StackPanel.Resources>
    <Button x:Name="btnCreate" Content="Add New Student" HorizontalAlignment="Left" VerticalAlig
    <Button x:Name="btnDelete" Content="Delete Selected Student" HorizontalAlignment="Left" \
    <Button x:Name="btnUpdate" Content="Update Selected Student" HorizontalAlignment="Left" \
    <Button x:Name="btnClose" Content="Close Program" HorizontalAlignment="Left" VerticalAlig
</StackPanel>
```

Style

- Các thành phần thuộc tính trong Style
 - BasedOn
 - Mang tính chất kế thừa
 - Một Style kế thừa thuộc tính chung của một Style khác

Style

- Các thành phần thuộc tính trong Style
 - TargetType
 - Được sử dụng để giới hạn loại điều khiển (Control) sử dụng Style đó.
 - Ví dụ: một Style với thuộc tính TargetType thiết lập cho Button, thì Style này sẽ không thể áp dụng cho các kiểu điều khiển khác (TextBox, Label,...)

```
<Style TargetType="{x:Type Button}">
```

```
....
```

```
</Style>
```


Style

- Các thành phần thuộc tính trong Style

- Triggers

- đối tượng cho phép bạn áp dụng những thay đổi về thuộc tính giao diện khi những điều kiện nhất định

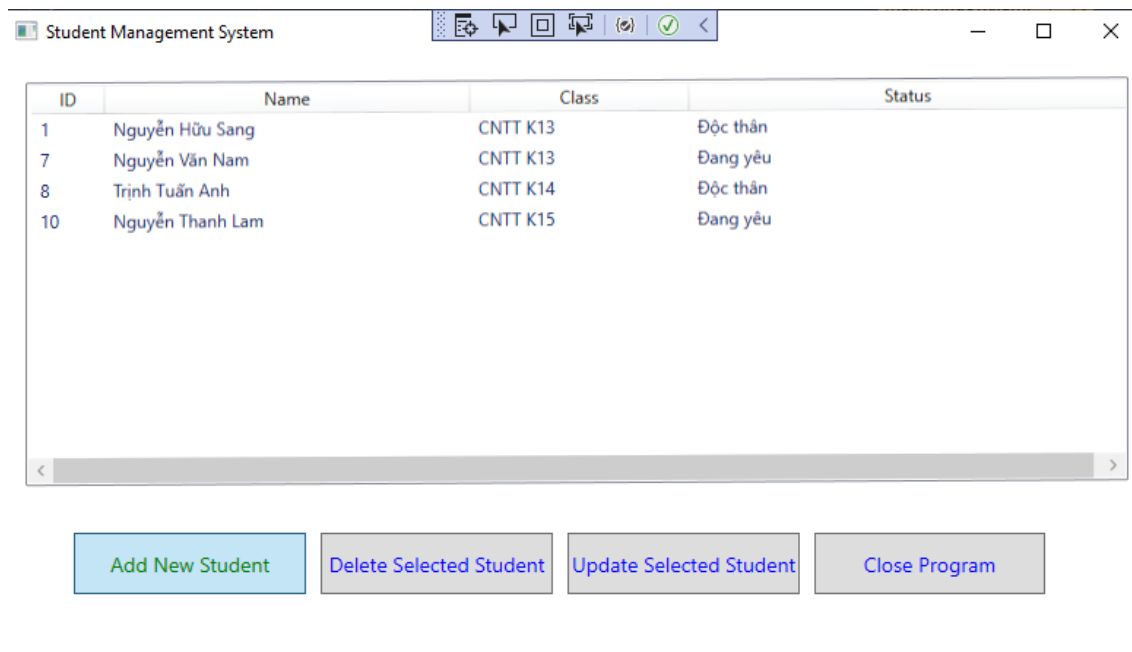
- Ví dụ

```
<Style.Triggers>
  <Trigger Property="IsPressed" Value="true">
    <Setter Property="Foreground" Value="■"Green"/>
  </Trigger>
</Style.Triggers>
```

=> Tác dụng: thay đổi thuộc tính màu chữ của Button khi thuộc tính IsPressed (khi bấm nút) có giá trị True.

Style

- Các thành phần thuộc tính trong Style
 - Triggers



Style

- Lời khuyên:
 - Không đặt khai báo Style trong App.xaml hay các file chức năng xaml. Hãy lưu chúng trong một file xaml tài nguyên riêng.
 - Cách tạo một file xaml tài nguyên riêng:
 - Add -> ResourceDictionary (WPF)
 - Thêm thẻ tag trong file XAML chức năng:

```
<Window.Resources>  
    <ResourceDictionary Source="<tên file xaml tài  
nguyên>"></ResourceDictionary>  
</Window.Resources>
```

Style

- Lời khuyên:

- Vậy, khi đã chia thành các file tài nguyên riêng thì vấn đề là làm sao để tìm tham chiếu tới tài nguyên đó ?

⇒ dùng một khoá duy nhất: Khi định nghĩa một tài nguyên trong XAML, định nghĩa một giá trị khoá duy nhất cho tài nguyên đó thông qua thuộc tính x:Key.

⇒ sau đó, có thể tham chiếu tới tài nguyên này bằng việc sử dụng giá trị này thông qua thuộc tính:

```
Style="{StaticResource ResourceKey=<tên_khoá>}"
```

Style

- Bài tập 11.1: Từ chương trình quản lý sinh viên đã có, tạo một file XAML tài nguyên sử dụng Style để nhóm các điều khiển có những thuộc tính giống nhau

Template

- Muốn đổi Button từ hình chữ nhật thành hình tròn / uốn viền,... ?
⇒ Khi đó, cần sử dụng đến Template (khuôn mẫu)

Template

- Muốn đổi Button từ hình chữ nhật thành hình tròn / uốn viền,... ?

⇒ Khi đó, cần sử dụng đến Template (khuôn mẫu)

- Trong WPF, có hai dạng khuôn mẫu được sử dụng:
 - ControlTemplate: thay đổi cấu trúc hiển thị cho điều khiển UI
 - DataTemplate: thay đổi cách thức hiển thị dữ liệu

Template

- ControlTemplate
 - ControlTemplate của một điều khiển định nghĩa diện mạo cho điều khiển đó.

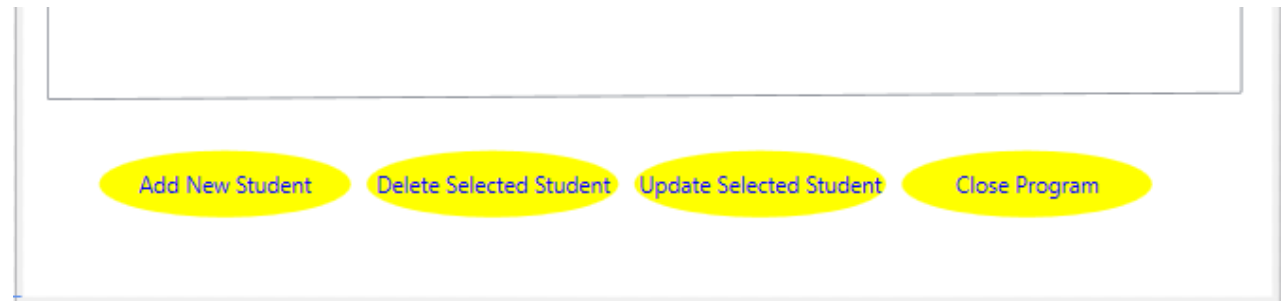
Template

- ControlTemplate

- Ví dụ

```
<Setter Property="Template">
  <Setter.Value>
    <ControlTemplate TargetType="Button">
      <Grid>
        <Ellipse Fill="Yellow"/>
        <ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>
      </Grid>
    </ControlTemplate>
  </Setter.Value>
</Setter>
```

- Kết quả



Template

- Bài tập 11.2: Từ chương trình quản lý sinh viên đã có và kết quả bài 11.1, tiếp tục sử dụng các Template để thiết kế giao diện sinh động hơn.