

Bài 3

CÁC ĐIỀU KHIỂN NÂNG CAO TRONG ỨNG DỤNG WPF

Không dừng lại ở việc cung cấp những điều khiển UI như ComboBox, ListBox, TextBox,..., với những chức năng cơ bản và đặc tính text đơn điệu như trong Windows Form, WPF còn cho phép người lập trình tùy biến thuộc tính của những điều khiển trên để biến chúng thành những điều khiển UI phức hợp, với nhiều đặc tính giao diện phong phú, tinh tế, kết hợp text, hình ảnh,... Để đạt được hiệu quả tương tự, với những công nghệ trước đây như MFC, cần tiêu tốn nhiều công sức lập trình. Qua các ví dụ cụ thể trong bài giảng này, chúng ta sẽ thấy WPF tạo ra chúng đơn giản như thế nào.

1 Hộp lựa chọn phông chữ (Font Chooser)

Mục tiêu của phần này là tạo lập một điều khiển dạng ComboBox, trong đó, liệt kê danh sách các phông chữ hệ thống. Tên của mỗi phông chữ lại được hiển thị dưới dạng chính phông chữ đó. Điều này cho phép người dùng xem trước định dạng phông chữ trước khi chọn chúng. Bạn có thể đã quen thuộc với dạng Combox này khi sử dụng các ứng dụng gần đây của Microsoft Office như Word, Excel, PowerPoint,...

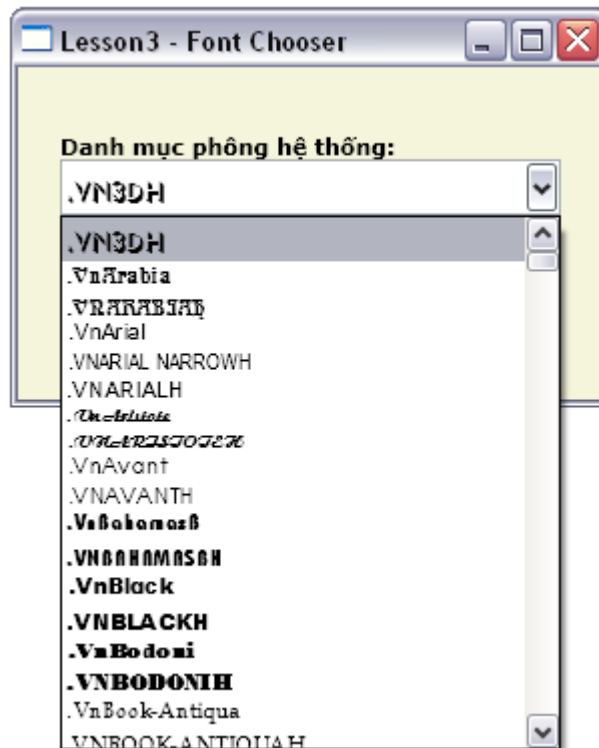
Và sau đây là mã XAML để tạo ra điều khiển này:

```
<!--Sử dụng stack panel theo chiều dọc làm layout chính của form-->
<StackPanel Width="250" Orientation="Vertical" Height="100" >
    <!--Khai báo tạo lập tiêu đề của điều khiển-->
    <TextBlock FontFamily="Verdana" FontWeight="DemiBold">
        Danh mục phông hệ thống:
    </TextBlock>
    <!--Khai báo tạo điều khiển ComboBox-->
    <ComboBox ItemsSource="{x:Static Fonts.SystemFontFamilies}" SelectedIndex="0">
        <!--Khai báo định nghĩa thuộc tính dữ liệu gắn với mỗi mục chọn-->
        <ComboBox.ItemTemplate>
            <DataTemplate>
                <TextBlock Text="{Binding}" FontFamily="{Binding}" />
            </DataTemplate>
        </ComboBox.ItemTemplate>
    </ComboBox>
</StackPanel>
```

Trong phần khai báo tạo điều khiển ComboBox, ta khai báo nguồn dữ liệu được dùng cho các mục trong hộp danh sách thông qua thuộc tính `ItemsSource`. Bằng việc gán `ItemsSource="{x:Static Fonts.SystemFontFamilies}"` ta định nghĩa nguồn dữ liệu này là danh sách các phông chữ hiện có của hệ thống máy tính hiện thời. Thuộc tính `SelectedIndex` cho phép định ra chỉ số của chỉ mục ngầm định được chọn ban đầu trong danh sách phông, cụ thể trong trường hợp này là phông chữ đầu tiên (`SelectedIndex="0"`).

Trong phần khai báo định nghĩa thuộc tính dữ liệu của mỗi chỉ mục trong ComboBox (phần tử `<ComboBox.ItemTemplate>`), ta lồng vào một điều khiển `TextBlock`, trong đó, nội dung hiển thị là phông chữ tương ứng (`Text="{Binding}"`) và dạng phông hiển thị nội dung cũng chính là phông chữ tương ứng với chỉ mục này (`FontFamily="{Binding}"`). *Những vấn đề liên quan đến kết nối nguồn dữ liệu sẽ được đề cập chi tiết hơn trong các bài giảng tiếp sau.*

Biên dịch và chạy chương trình, ta có kết quả như minh họa ở Hình 3.1. Như vậy, chỉ với không hơn 20 dòng mã XAML, chúng ta đã có thể tạo ra một điều khiển rất hữu dụng.



Hình 3.1 – Hộp lựa chọn phông chữ được xây dựng bằng WPF

2 Hộp danh mục ảnh (Image ListBox)

Trong phần này, ta xây dựng một hộp danh mục (ListBox) các đồ uống có kèm theo ảnh. Rõ ràng tính trực quan của giao diện người dùng sẽ tăng hơn nhiều so với một danh sách dạng text đơn điệu.

2.1 Thêm dữ liệu ảnh vào tài nguyên của project

Trước hết, ta thêm các ảnh đồ uống cần thiết vào tài nguyên của project theo các bước sau:

- Ở cửa sổ *Solution Explorer*, ta nhấp chuột phải vào tên project → Xuất hiện bảng chọn chức năng.
- Chọn mục **Add...>Existing Item** → Xuất hiện cửa sổ cho phép lựa chọn file.
- Trong hộp danh sách **Files of type**, ta chọn **Image Files** → Các file ảnh trong thư mục hiện thời sẽ xuất hiện.
- Tìm đến các file ảnh cần hiển thị trong danh sách và chọn OK.
- Kết quả, trong cửa sổ *Solution Explorer*, ta thấy xuất hiện các file ảnh tương ứng.

2.2 Xây dựng mã XAML

Giả thiết rằng các file ảnh đã được nạp vào project, sau đây là mã XAML tạo lập hộp danh mục đồ uống theo yêu cầu:

```
<!--Khai báo tạo lập một hộp danh mục với các thuộc tính về căn lề, chiều rộng,...-->
<ListBox Margin="10,10,0,13" Width="280" Name="listBox1" HorizontalAlignment="Left"
    VerticalAlignment="Top">
    <!--Khai báo tạo lập một chỉ mục con trong hộp danh mục với thuộc tính màu nền-->
    <ListBoxItem Background="Beige">
        <!--Lồng vào chỉ mục này một StackPanel để có thể chứa nhiều hơn 1 phần tử UI
        con theo chiều ngang-->
        <StackPanel Orientation="Horizontal">
            <!--Khai báo tạo lập một ảnh đồ uống ở đầu mỗi chỉ mục-->
```

```

        <Image Width="50" Height="50" Source="orange_juice.jpg"></Image>
        <!--Khai báo tạo lập một dòng chữ chỉ tên đồ uống-->
        <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Nước cam tươi</TextBlock>
    </StackPanel>
</ListBoxItem>

<!--Khai báo chỉ mục 2 tương tự như trên-->
<ListBoxItem>
    <StackPanel Orientation="Horizontal">
        <Image Width="50" Height="50" Source="kiwi_juice.jpg"></Image>
        <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Nước kiwi ép</TextBlock>
    </StackPanel>
</ListBoxItem>

<!--Khai báo chỉ mục 3 tương tự như trên-->
<ListBoxItem Background="Beige">
    <StackPanel Orientation="Horizontal">
        <Image Width="50" Height="50" Source="mango_juice.jpg"></Image>
        <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Nước soài ép</TextBlock>
    </StackPanel>
</ListBoxItem>

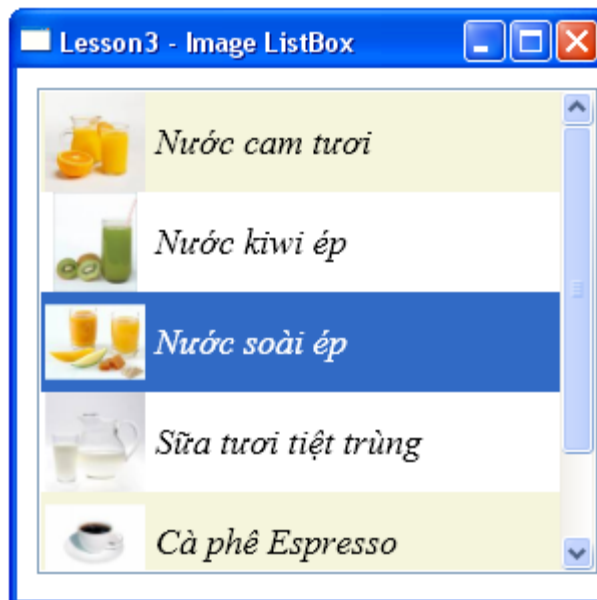
<!--Khai báo chỉ mục 4 tương tự như trên-->
<ListBoxItem>
    <StackPanel Orientation="Horizontal">
        <Image Width="50" Height="50" Source="milk.jpg"></Image>
        <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Sữa tươi tiệt trùng</TextBlock>
    </StackPanel>
</ListBoxItem>

<!--Khai báo chỉ mục 5 tương tự như trên-->
<ListBoxItem Background="Beige">
    <StackPanel Orientation="Horizontal">
        <Image Width="50" Height="50" Source="coffee.jpg"></Image>
        <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Cà phê Espresso</TextBlock>
    </StackPanel>
</ListBoxItem>
</ListBox>

```

Như vậy, điểm mấu chốt để bổ sung thêm các thuộc tính giao diện như ảnh, text, checkbox,..., vào mỗi chỉ mục của hộp danh sách chính là việc kết hợp các phần tử UI riêng lẻ tương ứng vào cùng một phần tử Panel nằm trong khai báo chỉ mục. Trong trường hợp này, với mỗi khai báo chỉ mục `<ListBoxItem>` ta thêm vào một `<StackPanel Orientation="Horizontal">` theo chiều ngang, trong đó, chứa một phần tử `<Image>` và 1 phần tử `<TextBlock>`. Nguồn dữ liệu ảnh được xác định qua thuộc tính `Source="<tên ảnh đã được bổ sung vào project>"`.

Kết quả của đoạn code được minh hoạ trong hình 3.2.



Hình 3.2 – Danh mục đồ uống có kèm ảnh minh họa xây dựng bằng WPF

3 Hộp mở rộng (Expander)

Hộp mở rộng Expander là một trong những điều khiển UI mới được giới thiệu trong WPF như một điều khiển cơ bản. Expander cho phép thu gọn hoặc mở rộng một nội dung nào đó chứa trong nó, giống như một node trong TreeView, bằng việc click vào biểu tượng mũi tên (hướng lên, nếu điều khiển đang ở trạng thái mở rộng; hướng xuống, nếu đang ở trạng thái thu gọn). Điều khiển này rất tiện lợi: Khi diện tích form chính quá chật hẹp vì nhiều chức năng được trình bày trên cùng giao diện, ta có thể sử dụng Expander để chứa một số chức năng ít dùng có thể tạm thời được ẩn dưới một tên nhóm chung.

Trong ví dụ sau đây, ta sẽ làm một menu chứa 2 mục là đồ uống và đồ ăn, mỗi mục sẽ chứa danh sách các sản phẩm tương ứng mà nhà hàng cung cấp. Ta sử dụng Expander để có thể mở rộng/thu gọn từng mục nêu trên. Sau đây là mã XAML của ứng dụng:

```
<Window x:Class="Lesson3.Window2"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Lesson 3 - Expander WPF Sample" Height="250" Width="200">

    <!--Sử dụng stackpanel làm layout chính-->
    <StackPanel>
        <!--Khai báo tạo lập Expander chứa danh mục đồ uống-->
        <Expander FontFamily="Times New Roman" FontSize="14" FontStyle="Oblique"
Header="Đồ uống" Background="#acbf43" >
            <!--Khai báo ListBox chứa danh mục đồ uống-->
            <ListBox >
                <!--Khai báo các chỉ mục con trong danh mục đồ uống-->
                <!--Mỗi mục con được trang trí bằng một biểu tượng đi kèm text-->
                <ListBoxItem>
                    <StackPanel Orientation="Horizontal">
                        <Image Source="glass.jpg" Height="20"></Image>
                        <TextBlock VerticalAlignment="Center">Nước chanh leo</TextBlock>
                    </StackPanel>
                </ListBoxItem>

                <ListBoxItem>
                    <StackPanel Orientation="Horizontal">
                        <Image Source="glass.jpg" Height="20"></Image>
```

```

        <TextBlock VerticalAlignment="Center">Nước cam vắt</TextBlock>
    </StackPanel>
</ListBoxItem>

<ListBoxItem>
    <StackPanel Orientation="Horizontal">
        <Image Source="glass.jpg" Height="20"></Image>
        <TextBlock VerticalAlignment="Center">Nước mơ muối</TextBlock>
    </StackPanel>
</ListBoxItem>

<ListBoxItem>
    <StackPanel Orientation="Horizontal">
        <Image Source="glass.jpg" Height="20"></Image>
        <TextBlock VerticalAlignment="Center">Sữa chua đánh đá</TextBlock>
    </StackPanel>
</ListBoxItem>
</ListBox>
</Expander>

<!--Khai báo Expander chứa danh mục đồ ăn-->
<Expander FontFamily="Times New Roman" FontSize="14" FontStyle="Oblique" Header="Đồ
ăn" Background="DarkOrange">
    <ListBox>
        <!--Khai báo các chỉ mục con trong danh mục đồ ăn-->
        <ListBoxItem>
            <StackPanel Orientation="Horizontal">
                <Image Source="bowl.gif" Height="20"></Image>
                <TextBlock VerticalAlignment="Center">Phở tái gà</TextBlock>
            </StackPanel>
        </ListBoxItem>

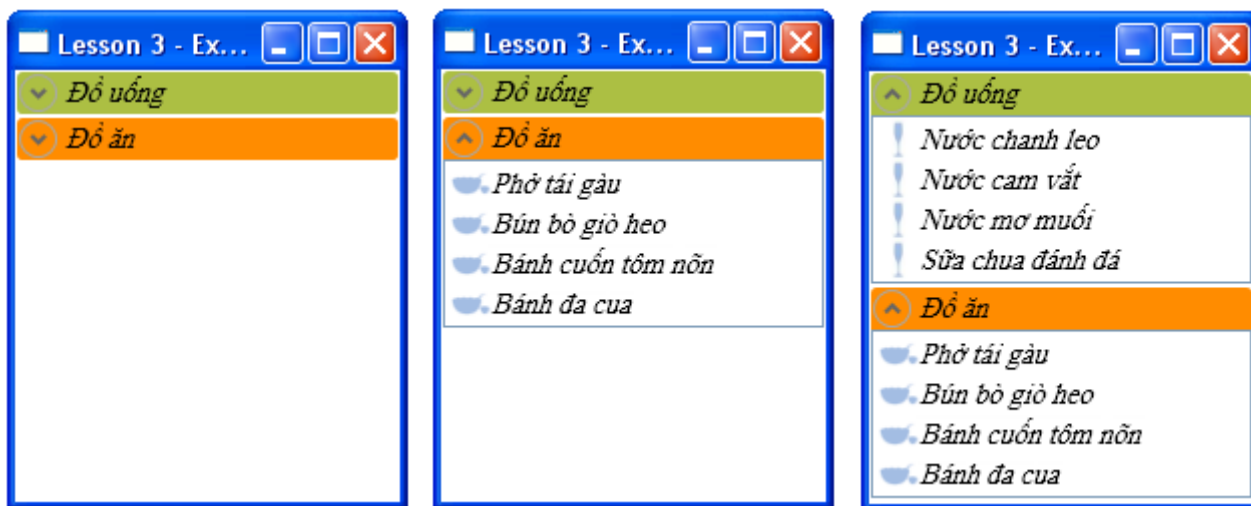
        <ListBoxItem>
            <StackPanel Orientation="Horizontal">
                <Image Source="bowl.gif" Height="20"></Image>
                <TextBlock VerticalAlignment="Center">Bún bò giò heo</TextBlock>
            </StackPanel>
        </ListBoxItem>

        <ListBoxItem>
            <StackPanel Orientation="Horizontal">
                <Image Source="bowl.gif" Height="20"></Image>
                <TextBlock VerticalAlignment="Center">Bánh cuốn tôm nơn</TextBlock>
            </StackPanel>
        </ListBoxItem>

        <ListBoxItem>
            <StackPanel Orientation="Horizontal">
                <Image Source="bowl.gif" Height="20"></Image>
                <TextBlock VerticalAlignment="Center">Bánh đa cua</TextBlock>
            </StackPanel>
        </ListBoxItem>
    </ListBox>
</Expander>
</StackPanel>
</Window>

```

Như ta thấy việc sử dụng Expander trong WPF rất đơn giản, với cùng nguyên tắc như các điều khiển UI cơ bản khác. Ở đây ta sử dụng 2 hộp mở rộng Expander: một chứa danh mục đồ uống được đặt trong một ListBox; một chứa danh mục đồ ăn trong một ListBox. Kết quả của đoạn code được minh họa trong Hình 3.3.



a)

b)

c)

Hình 3.3 – Tạo lập và sử dụng hộp mở rộng bằng WPF: a) Hai danh mục cùng thu gọn; b) Danh mục Đồ ăn được mở rộng; c) Cả hai danh mục được mở rộng (Danh mục trên đẩy danh mục dưới xuống)

4 Hộp soạn văn bản đa năng (RichTextBox)

Hộp soạn văn bản đa năng RichTextBox là một trong những điều khiển có chức năng phong phú. Không chỉ cho phép soạn sửa và hiển thị các nội dung text đơn thuần, RichTextBox còn cho phép thay đổi phông chữ (Verdana, Times New Roman,...), kiểu chữ (ngiêng, đậm, gạch chân),... Đặc biệt, điều khiển RichTextBox trong WPF/.NET 3.0 còn cho phép kiểm tra/gợi ý sửa đổi lỗi chính tả tiếng Anh của nội dung văn bản chứa trong đó. RichTextBox trong WPF/.NET 3.0 là phần tử được cải tiến về cơ bản so với phiên bản trước của điều khiển RichTextBox trong .NET 2.0. Tuy nhiên, cùng với sự mở rộng về chức năng là việc bổ sung các API mới cũng như những cách thức sử dụng khác.

4.1 Chức năng cơ bản

Để thêm mới một hộp soạn thảo đa năng vào form, ta dùng mã XAML như sau:

```
<RichTextBox x:Name="XAMLRichBox" SpellCheck.IsEnabled="True"
    MinHeight="100"></RichTextBox>
```

Thuộc tính `x:Name` là từ khoá xác định danh tính của RichTextBox được tạo. Thuộc tính này đóng vai trò là tham chiếu cho phép ta sau này buộc mã lệnh C# vào điều khiển. Thuộc tính `MinHeight` xác định số dòng có thể thấy được của hộp soạn thảo, giá trị này ngầm định bằng 1.

Thuộc tính `SpellCheck.IsEnabled="True"` kích hoạt tính năng kiểm tra lỗi chính tả tiếng Anh trong nội dung văn bản và gợi ý những từ đúng có thể để thay thế, giống như Microsoft Word.

Tuy nhiên, nếu chỉ với một RichTextBox, ta không có cách nào để sửa đổi định dạng của văn bản trong RichTextBox như đánh chữ nghiêng, chữ đậm, đổi phông chữ, v.v. Muốn đạt được điều này, ta phải buộc mã lệnh vào giao diện Command của RichTextBox.

4.2 Giao diện Command

Microsoft chủ trương để người phát triển làm việc với RichTextBox thông qua giao diện Command. Mặc dù khái niệm này không mới đối với phần lớn người phát triển giao diện đồ họa người dùng, việc cài đặt và cú pháp trong XAML có chút khác biệt.

Ta cần thêm một `ToolBar` và một số nút bấm hai trạng thái (`ToggleButton`) để gắn lệnh điều khiển `RichTextBox` đã tạo. Thuộc tính `Command` trên mỗi điều khiển kể trên sẽ xác định chức năng mà ta muốn kích hoạt trên `RichTextBox`. Trong khi đó, thuộc tính `CommandTarget` xác định `RichTextBox` nào ta muốn chức năng kích hoạt của các nút bấm nhằm vào. Sau đây là đoạn mã XAML bổ sung thêm một `ToolBar` và 3 nút bấm hai trạng thái:

```
<!--Khai báo sử dụng Toolbar-->
<ToolBar>
    <!--Khai báo nút bấm kích hoạt lệnh tô đậm đoạn chữ được chọn trong
RichTextBox-->
    <ToggleButton MinWidth="40"
        Command="EditingCommands.ToggleBold"
        CommandTarget="{Binding ElementName=XAMLRichBox}"
        TextBlock.FontWeight="Bold">B</ToggleButton>

    <!--Khai báo nút bấm kích hoạt lệnh in nghiêng đoạn chữ được chọn trong
RichTextBox-->
    <ToggleButton MinWidth="40"
        Command="EditingCommands.ToggleItalic"
        CommandTarget="{Binding ElementName=XAMLRichBox}"
        TextBlock.FontStyle="Italic">I</ToggleButton>

    <!--Khai báo nút bấm kích hoạt lệnh gạch chân đoạn chữ được chọn trong
RichTextBox-->
    <ToggleButton MinWidth="40"
        Command="EditingCommands.ToggleUnderline"
        CommandTarget="{Binding ElementName=XAMLRichBox}"
        <TextBlock TextDecorations="Underline">U</TextBlock>
    </ToggleButton>
</ToolBar>
```

Mặc dù đoạn mã ví dụ chỉ bao gồm một số ít các nút lệnh (`Command="EditingCommands.ToggleBold"`, `Command="EditingCommands.ToggleBold"`, `Command="EditingCommands.ToggleItalic"`), có tổng cộng 47 lệnh khác nhau mà ta có thể lựa chọn (có thể xem chúng bằng cách khảo sát lớp `EditingCommands`).

Dưới đây là đoạn mã XAML đầy đủ cho phép ta xây dựng một hộp soạn thảo văn bản có thể thay đổi được kiểu chữ (đậm, nghiêng, gạch chân):

```
<Window x:Class="Lesson3.Window3"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Lesson3 - Rich Text Box" Height="300" Width="300"
>
<!--Sử dụng StackPanel làm layout chính-->
<StackPanel Orientation="Vertical">
    <!--Khai báo toolbar-->
    <ToolBar>
        <!--Nút tô đậm-->
        <ToggleButton MinWidth="40"
            Command="EditingCommands.ToggleBold"
            CommandTarget="{Binding ElementName=XAMLRichBox}"
            TextBlock.FontWeight="Bold">B
        </ToggleButton>

        <!--Nút in nghiêng-->
        <ToggleButton MinWidth="40"
            Command="EditingCommands.ToggleItalic"
            CommandTarget="{Binding ElementName=XAMLRichBox}"
```



```

        TextBlock.FontStyle="Italic">I
    </ToggleButton>

    <!--Nút gạch chân-->
    <ToggleButton MinWidth="40"
        Command="EditingCommands.ToggleUnderline"
        CommandTarget="{Binding ElementName=XAMLRichTextBox}">
        <TextBlock TextDecorations="Underline">U</TextBlock>
    </ToggleButton>

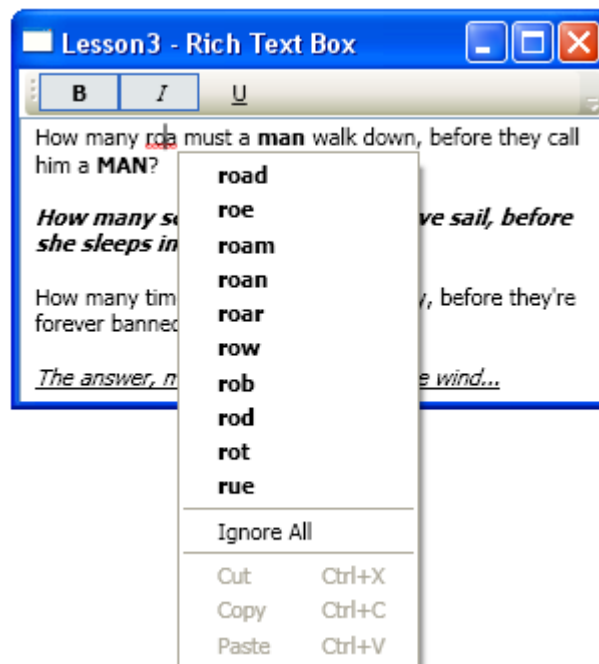
</ToolBar>

<!--Khai báo tạo lập RichTextBox-->
<RichTextBox x:Name="XAMLRichTextBox" SpellCheck.IsEnabled="True" MinHeight="100">
</RichTextBox>

</StackPanel>
</Window>

```

Kết quả được minh họa trong Hình 3.4.



Hình 3.4 – Xây dựng hộp soạn thảo đa năng đơn giản với các chức năng thay đổi kiểu chữ bằng WPF

Câu hỏi ôn tập

1. Thuộc tính nào của một ComboBox cho phép khai báo nguồn dữ liệu?

- A. SelectedIndex
- B. ItemSource
- C. Text
- D. FontFamily

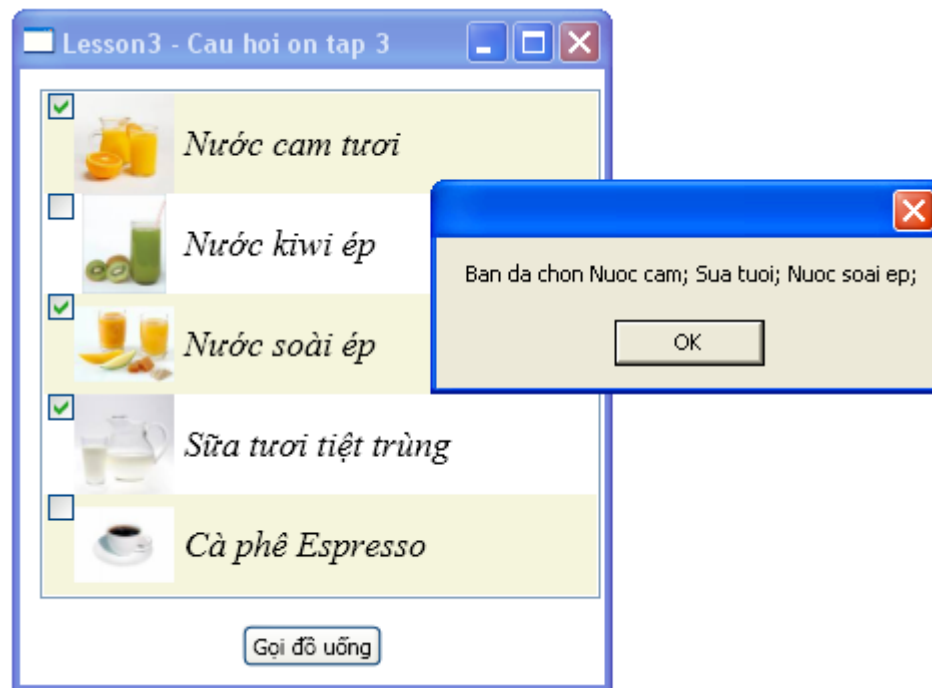
Trả lời: B

2. Trong hộp chọn phông chữ, danh sách phông chữ hệ thống được lấy từ lớp nào?

- A. Fonts.SystemFontFamilies
- B. Fonts.SystemTypefaces
- C. FontFamilyMapCollection

Trả lời: A

3. Trong ví dụ về danh mục ảnh ở mục 3, phát triển thêm tính năng mới bằng việc bổ sung các check box vào đầu mỗi mục đồ uống. Một nút bấm có trách nhiệm hiển thị những đồ uống mà người dùng đã chọn bằng việc đánh dấu checkbox (chú ý có thể nhiều hơn 1 lựa chọn). Kết quả như trong hình minh họa dưới đây:



Gợi ý:

- Thêm điều khiển CheckBox vào mỗi chỉ mục con của ListBox đang dùng. Với mỗi điều khiển CheckBox, thêm hàm xử lý sự kiện Uncheck và Check. Viết mã C# các hàm tương ứng trong file code-behind của form chứa.
- Thêm điều khiển Button vào sau ListBox (ví dụ, dùng StackPanel). Thêm hàm xử lý sự kiện Click của Button.

Sau đây là mã ví dụ:

Phần mã XAML:

```
<Window x:Class="Lesson3.Window4"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Lesson3 - Cau hoi on tap 3" Height="400" Width="300"
>
<StackPanel>
<!--Khai báo tạo lập một hộp danh mục với các thuộc tính về căn lề, chiều rộng, tên gọi,...-->
<ListBox Margin="10,10,0,13" Width="280" Name="listBox1" HorizontalAlignment="Left"
    VerticalAlignment="Top">
<!--Khai báo tạo lập một chỉ mục con trong hộp danh mục với thuộc tính màu nền-->
<ListBoxItem Background="Beige">
<!--Lồng vào chỉ mục này một StackPanel để có thể chứa nhiều hơn 1 phần tử UI con theo chiều ngang-->
<StackPanel Orientation="Horizontal">
<CheckBox x:Name="chkOrangeJuice" Checked="HandleChecked"
Unchecked="HandleUnchecked"></CheckBox>
<!--Khai báo tạo lập một ảnh đồ uống ở đầu mỗi chỉ mục-->
<Image Width="50" Height="50" Source="orange_juice.jpg"></Image>
```

```

        <!--Khái báo tạo lập một dòng chữ chỉ tên đồ uống-->
        <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Nước cam tươi</TextBlock>
    </StackPanel>
</ListBoxItem>

    <!--Khái báo chỉ mục 2 tương tự như trên-->
    <ListBoxItem>
        <StackPanel Orientation="Horizontal">
            <CheckBox x:Name="chkKiwiJuice" Checked="HandleChecked"
Unchecked="HandleUnchecked"></CheckBox>
            <Image Width="50" Height="50" Source="kiwi_juice.jpg"></Image>
            <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Nước kiwi ép</TextBlock>
        </StackPanel>
    </ListBoxItem>

    <!--Khái báo chỉ mục 3 tương tự như trên-->
    <ListBoxItem Background="Beige">
        <StackPanel Orientation="Horizontal">
            <CheckBox x:Name="chkMangoJuice" Checked="HandleChecked"
Unchecked="HandleUnchecked"></CheckBox>
            <Image Width="50" Height="50" Source="mango_juice.jpg"></Image>
            <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Nước soài ép</TextBlock>
        </StackPanel>
    </ListBoxItem>

    <!--Khái báo chỉ mục 4 tương tự như trên-->
    <ListBoxItem>
        <StackPanel Orientation="Horizontal">
            <CheckBox x:Name="chkMilk" Checked="HandleChecked"
Unchecked="HandleUnchecked"></CheckBox>
            <Image Width="50" Height="50" Source="milk.jpg"></Image>
            <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Sữa tươi tiệt trùng</TextBlock>
        </StackPanel>
    </ListBoxItem>

    <!--Khái báo chỉ mục 5 tương tự như trên-->
    <ListBoxItem Background="Beige">
        <StackPanel Orientation="Horizontal">
            <CheckBox x:Name="chkCafe" Checked="HandleChecked"
Unchecked="HandleUnchecked"></CheckBox>
            <Image Width="50" Height="50" Source="coffee.jpg"></Image>
            <TextBlock Margin="5" VerticalAlignment="Center" FontFamily="Times New Roman"
FontStyle="Italic" FontSize="18">Cà phê Espresso</TextBlock>
        </StackPanel>
    </ListBoxItem>
</ListBox>

    <!--Khái báo nút bấm kích hoạt hiển thị đồ uống được chọn -->
    <Button Click="DislayCustomerChoices" Width="70">Gọi đồ uống</Button>
</StackPanel>
</Window>

```

Phần mã C#:

```

public partial class Window4 : System.Windows.Window
{
    //Đặt các cờ xác định lựa chọn tương ứng với các loại đồ uống
    bool selectedOrange;

```

```

bool selectedKiwi;
bool selectedMango;
bool selectedMilk;
bool selectedEspresso;

public Window4()
{
    InitializeComponent();
    //
    //Khởi tạo biến đánh dấu chọn
    selectedOrange = false;
    selectedKiwi = false;
    selectedMango = false;
    selectedMilk = false;
    selectedEspresso = false;
}

//Hàm xử lý sự kiện bỏ chọn (Unchecked) của mỗi checkbox
//Lưu ý: Ở đây ta chỉ sử dụng một hàm duy nhất xử lý sự kiện này cho mọi
checkbox
//Để phân biệt checkbox nào phát động sự kiện, ta dựa vào tham số sender và so
sánh nó với các checkbox
private void HandleUnchecked(object sender, RoutedEventArgs e)
{
    if (sender.Equals(chkCafe)) selectedEspresso = false;
    if (sender.Equals(chkKiwiJuice)) selectedKiwi = false;
    if (sender.Equals(chkMangoJuice)) selectedMango = false;
    if (sender.Equals(chkMilk)) selectedMilk = false;
    if (sender.Equals(chkOrangeJuice)) selectedOrange = false;
}

//Hàm xử lý sự kiện bỏ chọn (checked) của mỗi checkbox (tương tự như trên)
private void HandleChecked(object sender, RoutedEventArgs e)
{
    if (sender.Equals(chkCafe)) selectedEspresso = true;
    if (sender.Equals(chkKiwiJuice)) selectedKiwi = true;
    if (sender.Equals(chkMangoJuice)) selectedMango = true;
    if (sender.Equals(chkMilk)) selectedMilk = true;
    if (sender.Equals(chkOrangeJuice)) selectedOrange = true;
}

//Hàm xử lý sự kiện hiển thị các đồ uống được chọn
private void DisplayCustomerChoices(object sender, RoutedEventArgs e)
{
    String choices = "Ban da chon ";
    bool selected = false;
    //
    if (selectedOrange)
    {
        choices += "Nuoc cam; ";
        selected = true;
    }
    //
    if (selectedMilk)
    {
        choices += "Sua tuoi; ";
        selected = true;
    }
    //
    if (selectedMango)
    {

```

```

        choices += "Nuoc soai ep; ";
        selected = true;
    }
    //
    if (selectedEspresso)
    {
        choices += "Cafe Espresso;";
        selected = true;
    }
    //
    if (!selected) choices = "Ban chua chon do uong nao";
    //
    MessageBox.Show(choices);
}
}

```

4. Trong điều khiển *Expander*, ta thường tạo dòng text mô tả nội dung bên trong của *Expander*, luôn xuất hiện trên *Expander* bên cạnh mũi tên chỉ trạng thái của *Expander*. Muốn thiết lập nội dung của dòng text này, ta dùng thuộc tính gì của điều khiển *Expander*?

- A. Content
- B. Text
- C. Header
- D. Source

Trả lời: C

5. Trong *ToolBar* kết hợp với *RichTextBox*, thuộc tính nào của nút bấm hai trạng thái *ToggleButton* xác định chức năng sửa đổi văn bản cần kích hoạt?

- A. Command
- B. CommandTarget
- C. Cả hai thuộc tính trên

Trả lời: A

6. Trong *ToolBar* kết hợp với *RichTextBox*, thuộc tính nào của nút bấm hai trạng thái *ToggleButton* xác định đối tượng *RichTextBox* có chức năng sửa đổi văn bản cần kích hoạt?

- A. Command
- B. CommandTarget
- C. Cả hai thuộc tính trên

Trả lời: B

Tài liệu tham khảo

1. Creating ImageListBox in WPF, URL: <http://windowsclient.net/blogs/ilves/archive/2008/02/14/creating-imageListBox-in-wpf.aspx>

2. WPF ListBox Tutorial, URL: <http://www.longhorncorner.com/UploadFile/lheditor/WPFLB08262008103013AM/WPFLB.aspx>

3. WPF Sample Series - Expander Control With Popup Content. URL: <http://karlshifflett.wordpress.com/2008/02/05/wpf-sample-series-expander-control-with-popup-content/>

4. WPF RichTextBox, URL: <http://www.c-sharpcorner.com/UploadFile/mahesh/WPFRichTextBox09072008194855PM/WPFRichTextBox.aspx>

5. Mastering the WPF RichTextBox, URL: <http://www.devx.com/dotnet/Article/34644>