

# IT 认证电子书



质 量 更 高 服 务 更 好

半年免费升级服务

<http://www.itrenzheng.com>

**Exam** : **70-502(VB)Chinese**

**Title** : TS: Microsoft .NET  
Framework 3.5 – Windows  
Presentation Foundation

**Version** : DEMO

1. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。您向 Window 元素中添加一个 CommandBinding 元素。该命令具有一个键盘笔势 Ctrl+H。Window 元素中包含下面的 MenuItem 控件。

```
<MenuItem Header="Highlight Content"  
    Command="local:CustomCommands.Highlight" />
```

您需要确保当焦点切换到不包含任何文本的 TextBox 控件时，MenuItem 控件处于禁用状态而且该命令不可执行。

您应该怎么办？

- A. 在 TextBox 控件的 GotFocus 事件处理程序中，设置 MenuItem 控件的 IsEnabled 属性。
- B. 将该命令的 CanExecute 属性设置为 Highlight\_CanExecute。

将以下方法添加到该窗口的代码隐藏文件中。

```
Private Sub Highlight_CanExecute(ByVal sender As Object, _  
    ByVal e As CanExecuteRoutedEventArgs)  
    Dim txtBox As TextBox = CType(sender, TextBox)  
    e.CanExecute = (txtBox.Text.Length > 0)  
End Sub
```

- C. 将该命令的 CanExecute 属性设置为 Highlight\_CanExecute。

将以下方法添加到该窗口的代码隐藏文件中。

```
Private Sub Highlight_CanExecute(ByVal sender As Object, _  
    ByVal e As CanExecuteRoutedEventArgs)  
    Dim txtBox As TextBox  
    txtBox = CType(e.Source, TextBox)  
    e.CanExecute = (txtBox.Text.Length > 0)  
End Sub
```

- D. 将该命令的 CanExecute 属性设置为 Highlight\_CanExecute。

将以下方法添加到该窗口的代码隐藏文件中。

```
Private Sub Highlight_CanExecute(ByVal sender As Object, _  
    ByVal e As CanExecuteRoutedEventArgs)  
    Dim Menu As MenuItem = CType(e.Source, MenuItem)  
    Dim txtBox As TextBox = CType(Menu.CommandTarget, TextBox)
```

```
Menu.IsEnabled = (textBox.Text.Length > 0)
```

```
End Sub
```

**Answer: C**

2. 您使用 Microsoft .NET Framework 3.5 创建了一个 Windows Presentation Foundation 应用程序。该应用程序名为 EnterpriseApplication.exe，通过网络运行。

您使用设计器在用户作用域级别向 Settings.settings 文件中添加 WindowSize 参数和 WindowPosition 参数。窗口的大小和位置是从用户配置文件读取的。

该应用程序必须针对执行它的用户保持最初的窗口大小和位置。

您需要确保满足以下要求：

每个用户的窗口大小都保存在用户配置文件中。

在用户退出应用程序后保留用户设置。

您应该使用哪个配置设置？

A. Private Sub OnClosing(ByVal sender As Object, ByVal e As

System.ComponentModel.CancelEventArgs)

```
My.Settings.Default.WindowPosition = New Point(Me.Left, Me.Top)
```

```
My.Settings.Default.WindowSize = New Size(Me.Width, Me.Height)
```

```
My.Settings.Default.Save()
```

```
End Sub
```

B. Private Sub OnClosing(ByVal sender As Object, ByVal e As

System.ComponentModel.CancelEventArgs)

```
Dim appKey As RegistryKey =
```

```
Registry.CurrentUser.CreateSubKey("Software\EnterpriseApplication")
```

```
Dim settingsKey As RegistryKey =
```

```
appKey.CreateSubKey("WindowSettings")
```

```
Dim windowPositionKey As RegistryKey =
```

```
settingsKey.CreateSubKey("WindowPosition")
```

```
Dim windowSizeKey As RegistryKey =
```

```
settingsKey.CreateSubKey("WindowSize")
```

```
windowPositionKey.SetValue("X", Me.Left)
```

```
windowPositionKey.SetValue("Y", Me.Top)
```

```
windowSizeKey.SetValue("Width", Me.Width)
```

```
windowSizeKey.SetValue("Height", Me.Height)
```

```
End Sub
```

```
C. Private Sub OnClosing(ByVal sender As Object, ByVal e As _
```

```
System.ComponentModel.CancelEventArgs)
```

```
Dim doc As New System.Xml.XmlDocument()
```

```
doc.Load("EnterpriseApplication.exe.config")
```

```
Dim nodePosition As System.Xml.XmlNode = _
```

```
doc.SelectSingleNode("//setting[@name='WindowPosition']")
```

```
nodePosition.ChildNodes(0).InnerText = String.Format("{0},{1}", _
```

```
Me.Left, Me.Top)
```

```
Dim nodeSize As System.Xml.XmlNode = _
```

```
doc.SelectSingleNode("//setting[@name='WindowSize']")
```

```
nodeSize.ChildNodes(0).InnerText = String.Format("{0},{1}", _
```

```
Me.Width, Me.Height)
```

```
doc.Save("UserConfigDistractor2.exe.config")
```

```
End Sub
```

```
D. Private Sub Window_Closing(ByVal sender As Object, ByVal e As _
```

```
System.ComponentModel.CancelEventArgs)
```

```
Dim sw As New StreamWriter("EnterpriseApplication.exe.config", True)
```

```
sw.WriteLine("<EnterpriseApplication.Properties.Settings>")
```

```
sw.WriteLine("<setting name=""WindowSize"" serializeAs=""String"">")
```

```
sw.WriteLine(String.Format("<value>{0},{1}</value>", Me.Width, _
```

```
Me.Height))
```

```
sw.WriteLine("</setting>")
```

```
sw.WriteLine("<setting name=""WindowPosition"" _
```

```
serializeAs=""String"">")
```

```
sw.WriteLine(String.Format("<value>{0},{1}</value>", Me.Left, _
```

```
Me.Top))
```

```
sw.WriteLine("</setting>")
sw.WriteLine("</UserConfigProblem.Properties.Settings>")
sw.Close()
End Sub
```

**Answer: A**

3. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。该应用程序定义一个 **BrowserWindow** 类。**BrowserWindow** 类的每个实例允许用户在一个单独的窗口中浏览网站。在打开新的浏览器窗口时，用户将重定向到一个预定义的 URL。

您编写以下代码段。

```
01 Private Sub OpenNewWindow(ByVal sender As Object, _
02   ByVal e As RoutedEventArgs)
03   Dim newWindowThread As New Thread(New _
04     ThreadStart(AddressOf NewThreadProc))
05
06   newWindowThread.Start()
07 End Sub
08 Private Sub NewThreadProc()
09
10 End Sub
```

您需要确保满足以下要求：

在创建其他浏览器窗口时，应用程序的主窗口不被阻止。

在应用程序的主窗口关闭时，应用程序即执行完毕。

您应该怎么办？

A. 在第 05 行处插入以下代码段。

```
newWindowThread.SetApartmentState(ApartmentState.STA)
newWindowThread.IsBackground = True
```

在第 09 行处插入以下代码段。

```
Dim newWindow As New BrowserWindow()
newWindow.Show()
```

```
Dim app As New Application()
```

```
app.Run(newWindow)
```

B. 在第 05 行处插入以下代码段。

```
newWindowThread.IsBackground = True
```

在第 09 行处插入以下代码段。

```
newWindowThread.SetApartmentState(ApartmentState.STA)
```

```
Dim newWindow As New BrowserWindow()
```

```
newWindow.Show()
```

```
Dim app As New Application()
```

```
app.Run(newWindow)
```

C. 在第 05 行处插入以下代码段。

```
newWindowThread.SetApartmentState(ApartmentState.STA)
```

```
newWindowThread.IsBackground = False
```

在第 09 行处插入以下代码段。

```
Dim newWindow As New BrowserWindow()
```

```
System.Windows.Threading.Dispatcher.Run()
```

```
newWindow.Show()
```

D. 在第 05 行处插入以下代码段。

```
newWindowThread.SetApartmentState(ApartmentState.STA)
```

```
newWindowThread.IsBackground = True
```

在第 09 行处插入以下代码段。

```
Dim newWindow As New BrowserWindow()
```

```
newWindow.Show()
```

```
System.Windows.Threading.Dispatcher.Run()
```

**Answer: D**

4. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

该应用程序使用若干个异步操作来计算显示给用户的数据。名为 `tommorowsWeather` 的操作执行将由其他操作使用的计算。

您需要确保 `tommorowsWeather` 以尽可能高的优先级运行。

您应该使用哪个代码段？

- A. tomorrowWeather.Dispatcher.BeginInvoke( \_  
System.Windows.Threading.DispatcherPriority.Normal, \_  
New OneArgDelegate(AddressOf UpdateUserInterface), weather)
- B. tomorrowWeather.Dispatcher.BeginInvoke( \_  
System.Windows.Threading.DispatcherPriority.DataBind, \_  
New OneArgDelegate(AddressOf UpdateUserInterface), weather)
- C. tomorrowWeather.Dispatcher.BeginInvoke( \_  
System.Windows.Threading.DispatcherPriority.Send, \_  
New OneArgDelegate(AddressOf UpdateUserInterface), weather)
- D. tomorrowWeather.Dispatcher.BeginInvoke( \_  
System.Windows.Threading.DispatcherPriority.Render, \_  
New OneArgDelegate(AddressOf UpdateUserInterface), weather)

**Answer: C**

5. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

您为该应用程序创建一个窗口。

您需要确保满足以下要求：

使用 **ListBox** 控件以双列格式显示字符串数组。

**ListBox** 控件中数据的流动方向为从左向右、自上向下。

您应该怎么办？

A. 使用按如下方式定义的 **ListBox** 控件。

```
<ListBox Name="myList">  
  <ListBox.ItemsPanel>  
    <ItemsPanelTemplate>  
      <UniformGrid Columns="2"/>  
    </ItemsPanelTemplate>  
  </ListBox.ItemsPanel>  
</ListBox>
```

使用下面的 VB.NET 代码将字符串数组与 **ListBox** 控件关联。

```
myList.ItemsSource = arrayOfString
```



B. 使用按如下方式定义的 **ListBox** 控件。

```
<ListBox Name="myList">
  <ListBox.ItemsPanel>
    <ItemsPanelTemplate>
      <StackPanel />
    </ItemsPanelTemplate>
  </ListBox.ItemsPanel>
</ListBox>
```

使用下面的 VB.NET 代码将字符串数组与 **ListBox** 控件关联。

```
myList.ItemsSource = arrayOfString
```

C. 使用按如下方式定义的 **ListBox** 控件。

```
<ListBox Name="myList">
  <ListBox.ItemsPanel>
    <ItemsPanelTemplate>
      <WrapPanel />
    </ItemsPanelTemplate>
  </ListBox.ItemsPanel>
</ListBox>
```

使用下面的 VB.NET 代码将字符串数组与 **ListBox** 控件关联。

```
myListView.ItemsSource = arrayOfString
```

D. 使用按如下方式定义的 **ListBox** 控件。

```
<ListBox Name="myList">
  <ListBox.ItemsPanel>
    <ItemsPanelTemplate>
      <Grid>
        <Grid.ColumnDefinitions>
          <ColumnDefinition />
          <ColumnDefinition />
        </Grid.ColumnDefinitions>
      </Grid>
    </ItemsPanelTemplate>
  </ListBox.ItemsPanel>
</ListBox>
```

```
</ItemsPanelTemplate>
```

```
</ListBox.ItemsPanel>
```

```
</ListBox>
```

使用下面的 VB.NET 代码将字符串数组与 `ListBox` 控件关联。

```
myList.ItemsSource = arrayOfString
```

**Answer: A**

6. 您使用 `Windows Presentation Foundation` 和 `Microsoft .NET Framework 3.5` 创建一个窗体。该窗体中包含一个状态栏。

您打算向该状态栏添加一个 `ProgressBar` 控件。

您需要确保 `ProgressBar` 控件显示您无法预测完成时间的任务的进度。

您应该使用哪个代码段？

A. `progressbar.IsIndeterminate = True`

B. `progressbar.IsIndeterminate = False`

C. `progressbar.HasAnimatedProperties = True`

D. `progressbar.HasAnimatedProperties = False`

**Answer: A**

7. 您正在将 `Windows` 窗体应用程序转换为 `Windows Presentation Foundation (WPF)` 应用程序。您使用 `Microsoft .NET Framework 3.5` 创建该 `WPF` 应用程序。

该 `WPF` 应用程序将重用此 `Windows` 窗体应用程序的 30 个窗体。

该 `WPF` 应用程序中包含下面的类定义。

```
Public Class OwnerWindow
```

```
    Implements System.Windows.Forms.IWin32Window
```

```
    Private handle_Renamed As IntPtr
```

```
    Public Property Handle() As IntPtr _
```

```
        Implements System.Windows.Forms.IWin32Window.Handle
```

```
        Get
```

```
            Return handle_Renamed
```

```
        End Get
```

```
Set(ByVal value As IntPtr)
```

```
handle_Renamed = value
```

```
End Set
```

```
End Property
```

```
End Class
```

您在该 WPF 应用程序中编写以下代码段。(所包括的行号仅供参考。)

```
01 Public Function LaunchWindowsFormsDialog(ByVal dialog As _  
02 System.Windows.Forms.Form, ByVal wpfParent As Window) As _  
03 System.Windows.Forms.DialogResult  
04 Dim helper As New  
05 System.Windows.Interop.WindowInteropHelper(wpfParent)  
07 Dim owner As New OwnerWindow()  
08  
09 End Function
```

您需要确保该应用程序能够以模式对话框形式启动可重用的窗体。

您应该在第 08 行处插入哪个代码段？

A. owner.Handle = helper.Owner

```
Dim db As New System.Windows.Forms.DialogResult()
```

```
Return db
```

B. owner.Handle = helper.Owner

```
Return dialog.ShowDialog(owner)
```

C. owner.Handle = helper.Owner

```
Dim result As Nullable(Of Boolean) = wpfParent.ShowDialog()
```

```
If result.HasValue Then
```

```
Return If(result.Value, System.Windows.Forms.DialogResult.OK, _
```

```
System.Windows.Forms.DialogResult.Cancel)
```

```
Else
```

```
Return System.Windows.Forms.DialogResult.Cancel
```

```
End If
```

D. owner.Handle = helper.Handle

```
Dim result As Nullable(Of Boolean) = wpfParent.ShowDialog()  
If result.HasValue Then  
    Return If(result.Value, System.Windows.Forms.DialogResult.OK, _  
        System.Windows.Forms.DialogResult.Cancel)  
Else  
    Return System.Windows.Forms.DialogResult.Cancel  
End If
```

**Answer: B**

8. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation (WPF) 应用程序。

该 WPF 应用程序有一个名为 rootGrid 的 Grid 控件。

您编写以下 XAML 代码片段。

```
<Window x:Class="MCP.HostingWinFormsControls"  
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/  
presentation"  
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
Title="HostingWinFormsControls"  
Loaded="Window_Loaded">  
    <Grid x:Name="rootGrid">  
    </Grid>  
</Window>
```

您需要确保每次打开 WPF 窗口时，都会向 rootGrid 中添加一个名为 MyCustomFormsControl 的 Windows 窗体控件。

您应该使用哪个代码段？

```
A. Private Sub Window_Loaded(ByVal sender As Object, ByVal e As _  
RoutedEventArgs)  
    Dim host As New WindowsFormsHost()  
    Dim formsControl As New MyCustomFormsControl()  
    host.Child = formsControl;
```

```
rootGrid.Children.Add(host);
```

End Sub

B. Private Sub Window\_Loaded(ByVal sender As Object, ByVal e As \_

RoutedEventArgs)

```
Dim host As New ElementHost()
```

```
Dim formsControl As New MyCustomFormsControl()
```

```
host.Child = formsControl;
```

```
rootGrid.Children.Add(host);
```

End Sub

C. Private Sub Window\_Loaded(ByVal sender As Object, ByVal e As \_

RoutedEventArgs)

```
Dim formsControl As New MyCustomFormsControl()
```

```
formsControl.CreateControl()
```

```
Dim target As New HwndTarget(formsControl.Handle)
```

```
Dim formsElement As UIElement = TryCast(target.RootVisual, _  
UIElement)
```

```
rootGrid.Children.Add(formsElement)
```

End Sub

D. Private Sub Window\_Loaded(ByVal sender As Object, ByVal e As \_

RoutedEventArgs)

```
Dim formsControl As New MyCustomFormsControl()
```

```
formsControl.CreateControl()
```

```
Dim source As HwndSource = HwndSource.FromHwnd(formsControl.Handle)
```

```
Dim formsElement As UIElement = TryCast(source.RootVisual, _  
UIElement)
```

```
rootGrid.Children.Add(formsElement)
```

End Sub

**Answer: A**

9. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

您在此程序中包含了功能以解决窗口行为问题。

您需要在窗口中鼠标单击的位置显示 UI 元素列表。您还需要确保此元素列表显示在消息框中。

您应该将下面哪个代码段包括在代码隐藏文件中？

A. Dim controlsToDisplay As String = String.Empty

```
Private Sub Window_MouseDown(ByVal sender As Object, _  
    ByVal e As MouseButtonEventArgs)  
    controlsToDisplay = CType(sender, UIElement).ToString()  
    MessageBox.Show(controlsToDisplay)  
End Sub
```

B. Dim controlsToDisplay As String = String.Empty

```
Private Sub Window_MouseDown(ByVal sender As Object, _  
    ByVal e As MouseButtonEventArgs)  
    For i = 0 To VisualChildrenCount - 1  
        controlsToDisplay += GetVisualChild(i).ToString() + "\r\n"  
    Next  
    MessageBox.Show(controlsToDisplay)  
End Sub
```

C. Dim controlsToDisplay As String = String.Empty

```
Private Sub Window_MouseDown(ByVal sender As Object, _  
    ByVal e As MouseButtonEventArgs)  
    Dim myVisual As Visual()  
    For i = 0 To VisualTreeHelper.GetChildrenCount(CType(sender, _  
        Visual)) - 1  
        myVisual(i) = CType(VisualTreeHelper.GetChild(CType(sender, _  
            Visual), i), Visual)  
        controlsToDisplay += myVisual.GetType().ToString() + "\r\n"  
    Next  
    MessageBox.Show(controlsToDisplay)  
End Sub
```

D. Dim controlsToDisplay As String = String.Empty

```
Private Sub Window_MouseDown(ByVal sender As Object, _
    ByVal e As MouseButtonEventArgs)
    Dim pt As Point = e.GetPosition(Me)
    VisualTreeHelper.HitTest(Me, Nothing, _
        New HitTestResultCallback(AddressOf HitTestCallback), _
        New PointHitTestParameters(pt))
    MessageBox.Show(controlsToDisplay)
End Sub

Private Function HitTestCallback(ByVal result As HitTestResult) As _
    HitTestResultBehavior
    controlsToDisplay += result.VisualHit.GetType().ToString() + "\r\n"
    Return HitTestResultBehavior.Continue
End Function
```

**Answer: D**

10. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。您编写以下代码段（所包括的行号仅供参考）。

```
01 Dim content As Object
02 Dim fileName As String = "theFile"
03 Using xamlFile As New FileStream(fileName & ".xaml", _
04     FileMode.Open, FileAccess.Read)
06     content = TryCast(XamlReader.Load(xamlFile), Object)
07 End Using
08 Using container As Package = Package.Open(fileName & ".xps", _
09     FileMode.Create)
10
11 End Using
```

您需要确保满足以下要求：

此应用程序将现有流文档转换为 XPS 文档。

XPS 文档使用流文档格式生成。

XPS 文档的大小为可能的最小值。

您应该在第 10 行插入下面哪个代码段？

A. Using xpsDoc As New XpsDocument(container, \_

CompressionOption.SuperFast)

Dim rsm As XpsSerializationManager = New \_

System.Windows.Xps.XpsSerializationManager(New \_

XpsPackagingPolicy(xpsDoc), False)

rsm.SaveAsXaml(paginator)

End Using

B. Using xpsDoc As New XpsDocument(container, \_

CompressionOption.SuperFast)

Dim rsm As New XpsSerializationManager(New \_

XpsPackagingPolicy(xpsDoc), False)

rsm.Commit()

End Using

C. Using xpsDoc As New XpsDocument(container, \_

CompressionOption.Maximum)

Dim rsm As New XpsSerializationManager(New \_

XpsPackagingPolicy(xpsDoc), False)

Dim paginator As DocumentPaginator = (CType(content, \_

IDocumentPaginatorSource)).DocumentPaginator

rsm.SaveAsXaml(paginator)

End Using

D. Using xpsDoc As New XpsDocument(container, \_

CompressionOption.SuperFast)

Dim rsm As New XpsSerializationManager(New \_

XpsPackagingPolicy(xpsDoc), False)

Dim paginator As DocumentPaginator = (CType(content, \_

IDocumentPaginatorSource)).DocumentPaginator

rsm.SaveAsXaml(paginator)



End Using

**Answer: C**

11. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。此应用程序将使用 FlowDocumentPageViewer 类的实例来显示文档。该实例的名称为 fdpv。用户可以突出显示文档内容并为其添加注释。

您需要确保为文档作的注释得以保存，且这些注释在文档再次显示时呈现出来。

您应该使用下面哪个代码段？

A. Protected Sub OnTextInput(ByVal sender As Object, \_

ByVal e As RoutedEventArgs)

Dim service As AnnotationService = \_

AnnotationService.GetService(fdpv)

If service Is Nothing Then

AnnotationStream = New FileStream("annotations.xml", \_

FileMode.Open, FileAccess.ReadWrite)

service = New AnnotationService(fdpv)

Dim store As AnnotationStore = \_

New XmlStreamStore(AnnotationStream)

service.Enable(store)

End If

End Sub

Private Sub OnClosing(ByVal sender As Object, \_

ByVal e As System.ComponentModel.CancelEventArgs)

Dim service As AnnotationService = \_

AnnotationService.GetService(fdpv)

If Not service Is Nothing AndAlso service.IsEnabled Then

service.Store.Flush()

service.Disable()

AnnotationStream.Close()

End If

End Sub

B. Protected Sub OnLoaded(ByVal sender As Object, \_

ByVal e As RoutedEventArgs)

Dim service As AnnotationService = \_

AnnotationService.GetService(fdpv)

If service Is Nothing Then

AnnotationStream = New FileStream("annotations.xml", \_

FileMode.Open, FileAccess.ReadWrite)

service = New AnnotationService(fdpv)

End If

End Sub

Private Sub OnClosing(ByVal sender As Object, \_

ByVal e As System.ComponentModel.CancelEventArgs)

Dim service As AnnotationService = \_

AnnotationService.GetService(fdpv)

If Not service Is Nothing AndAlso service.IsEnabled Then

service.Store.Flush()

service.Disable()

AnnotationStream.Close()

End If

End Sub

C. Protected Sub OnLoaded(ByVal sender As Object, \_

ByVal e As RoutedEventArgs)

Dim service As AnnotationService = \_

AnnotationService.GetService(fdpv)

If service Is Nothing Then

AnnotationStream = New FileStream("annotations.xml", \_

FileMode.Open, FileAccess.ReadWrite)

service = New AnnotationService(fdpv)

Dim store As AnnotationStore = New \_

```
XmlStreamStore(AnnotationStream)
service.Enable(store)
End If
End Sub
Private Sub OnClosing(ByVal sender As Object, _
ByVal e As System.ComponentModel.CancelEventArgs)
Dim service As AnnotationService = _
AnnotationService.GetService(fdpv)
If Not service Is Nothing AndAlso service.IsEnabled Then
service.Store.Flush()
service.Disable()
AnnotationStream.Close()
End If
End Sub
D. Protected Sub OnLoaded(ByVal sender As Object, _
ByVal e As RoutedEventArgs)
Dim service As AnnotationService = _
AnnotationService.GetService(fdpv)
If service Is Nothing Then
AnnotationStream = New FileStream("annotations.xml", _
FileMode.Open, FileAccess.ReadWrite)
service = New AnnotationService(fdpv)
Dim store As AnnotationStore = New _
XmlStreamStore(AnnotationStream)
service.Enable(store)
End If
End Sub
Private Sub OnClosing(ByVal sender As Object, _
ByVal e As System.ComponentModel.CancelEventArgs)
Dim service As AnnotationService = _
```

```
AnnotationService.GetService(fdpv)
If Not service Is Nothing AndAlso service.IsEnabled Then
service.Disable()
AnnotationStream.Close()
End If
End Sub
```

**Answer: C**

12. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。您打算使用此应用程序预览视频文件。

您编写以下 XAML 代码片段。

```
01 <Window
01 x:Class="myClass" xmlns=
01 "http://schemas.microsoft.com/winfx/2006/xaml/presentation"
01 xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
01 Title="myWindow" Height="300" Width="300">
02   <StackPanel Background="Black">
03
04     <StackPanel HorizontalAlignment="Center"
04     Orientation="Horizontal">
05       <Button Name="btnPlay" Margin="10" Content="Play" />
06     </StackPanel>
07
08   </StackPanel>
09 </Window>
```

您需要确保此应用程序仅播放您要预览的视频的前 10 秒。

您应该执行哪两个操作？（每个正确的答案提供部分解决方案。请选择两个答案。）

A. 在第 03 行插入以下 XAML 片段。

```
<MediaElement Name="myMediaElement" Stretch="Fill" />
```

B. 在第 03 行插入以下 XAML 片段。

```
<MediaElement Name="myMediaElement"  
Source="MediaFileSelected.wmv" Stretch="Fill" />
```

C. 在代码隐藏文件中创建以下方法。

```
Public Sub PlayMedia(ByVal sender As Object, _  
ByVal args As RoutedEventArgs)  
    myMediaElement.Play()  
End Sub
```

D. 在第 07 行插入以下 XAML 片段。

```
<StackPanel.Triggers>  
    <EventTrigger RoutedEvent="Button.Click" SourceName="btnPlay">  
        <EventTrigger.Actions>  
            <BeginStoryboard Name="myBegin">  
                <Storyboard SlipBehavior="Slip">  
                    <MediaTimeline Source="MediaFileSelected.wmv"  
Storyboard.TargetName="myMediaElement"  
BeginTime="0:0:0" Duration="0:0:10" />  
                </Storyboard>  
            </BeginStoryboard>  
        </EventTrigger.Actions>  
    </EventTrigger>  
</StackPanel.Triggers>
```

E. 在第 07 行插入以下 XAML 片段。

```
<StackPanel.Triggers>  
    <EventTrigger RoutedEvent="Button.Click" SourceName="btnPlay">  
        <EventTrigger.Actions>  
            <BeginStoryboard Name="myBegin">  
                <Storyboard SlipBehavior="Slip">  
                    <MediaTimeline  
Storyboard.TargetName="myMediaElement"  
BeginTime="0:0:0" Duration="0:0:10" />  
            </BeginStoryboard>  
        </EventTrigger.Actions>  
    </EventTrigger>  
</StackPanel.Triggers>
```

```
</Storyboard>
</BeginStoryboard>
</EventTrigger.Actions>
</EventTrigger>
</StackPanel.Triggers>
```

**Answer: AD**

13. 您正在创建一个 Windows Presentation Foundation 应用程序。

您为此应用程序创建一个窗口。此应用程序包含一个名为 `AudioFileToPlay.wav` 的音频文件。

您需要确保满足以下要求：

每次单击该窗口的工作区时都会播放该音频文件。

该窗口在音频文件播放时提供最佳性能。

您应该怎么办？

A. 将以下 XAML 代码行添加到该窗口中。

```
<MediaElement Source="AudioFileToPlay.wav" />
```

B. 将以下代码段添加到代码隐藏文件中的 `window constructor` 方法中。

```
Dim player As New SoundPlayer()
player.SoundLocation = "AudioFileToPlay.wav"
player.Play()
```

C. 将以下代码段添加到代码隐藏文件中的 `window MouseDown` 方法中。

```
Dim player As New MediaElement()
player.Source = New Uri("AudioFileToPlay.wav", UriKind.Relative)
player.LoadedBehavior = MediaState.Manual
player.Play()
```

D. 将以下 XAML 代码片段添加到窗口中。

```
<Window.Triggers>
  <EventTrigger RoutedEvent="Window.MouseDown">
    <EventTrigger.Actions>
      <SoundPlayerAction Source="AudioFileToPlay.wav"/>
    </EventTrigger.Actions>
  </EventTrigger>
</Window.Triggers>
```

```
</EventTrigger>
```

```
</Window.Triggers>
```

**Answer: D**

14. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。您的项目包含一个名为 Data 的文件夹。

您在此 Data 文件夹中添加一个名为 song.mp3 的 .MP3 文件。您将此应用程序的 Build Action 属性设置为 Resource。

您需要通过其中一个 Application 类访问此 .MP3 文件。

您应该使用哪个代码段？

A. Dim uri As New Uri("/Data/song.mp3", UriKind.Relative)

```
Dim sri As StreamResourceInfo = Application.GetContentStream(uri)
```

```
Dim stream As Stream = sri.Stream
```

B. Dim uri As New Uri("/Data/song.mp3", UriKind.Relative)

```
Dim sri As StreamResourceInfo = Application.LoadComponent(uri)
```

```
Dim stream As Stream = sri.Stream
```

C. Dim uri As New Uri("/Data/song.mp3", UriKind.Relative)

```
Dim sri As StreamResourceInfo = Application.GetRemoteStream(uri)
```

```
Dim stream As Stream = sri.Stream
```

D. Dim uri As New Uri("/Data/song.mp3", UriKind.Relative)

```
Dim sri As StreamResourceInfo = Application.GetResourceStream(uri)
```

```
Dim stream As Stream = sri.Stream
```

**Answer: D**

15. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。该应用程序有一个以编程方式显示图像的窗口。该窗口包含一个名为 theGrid 的网格。

该窗口按图像的实际尺寸(宽度为 1024 像素)或更大尺寸显示图像。您希望这些图像的宽度为 200 像素。

您编写以下代码段。(所包括的行号仅供参考。)

```
01 Image theImage=new Image();
```

```
02 theImage.Width=200;
```

03 `BitmapImage theBitmapImage=new BitmapImage();`

04

05 `theImage.Source=theBitmapImage;`

06 `theGrid.Children.Add(theImage);`

您需要确保应用程序满足以下要求：

窗口使用最少的内存显示图像。

图像不扭曲。

您应该在第 04 行插入下面哪个代码段？

A. `theBitmapImage.UriSource=new Uri(@"imageToDisplay.jpg");`

`theBitmapImage.DecodePixelWidth=200;`

B. `theBitmapImage.BeginInit();`

`theBitmapImage.UriSource=new Uri(@"imageToDisplay.jpg");`

`theBitmapImage.EndInit();`

C. `theBitmapImage.BeginInit();`

`theBitmapImage.UriSource=new Uri(@"imageToDisplay.jpg");`

`theBitmapImage.DecodePixelWidth=200;`

`theBitmapImage.EndInit();`

D. `theBitmapImage.BeginInit();`

`theBitmapImage.UriSource=new Uri(@"imageToDisplay.jpg");`

`theBitmapImage.DecodePixelWidth=200;`

`theBitmapImage.DecodePixelHeight=200;`

`theBitmapImage.EndInit();`

**Answer: C**

16. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

该应用程序有一个以编程方式显示图像的窗口。该窗口包含一个名为 `theGrid` 的网格。

该窗口按图像的实际尺寸显示图像。您希望这些图像的宽度为 200 像素。您编写以下代码段。

01 `Dim theImage As New Image()`

02 `theImage.Width = 200`

03 `Dim theBitmapImage As New BitmapImage()`



04

05 theImage.Source = theBitmapImage

06 theGrid.Children.Add(theImage)

您需要确保应用程序满足以下要求：

窗口使用最少的内存显示图像。

图像不扭曲。

您应该在第 04 行插入下面哪个代码段？

A. theBitmapImage.UriSource = New Uri("imageToDisplay.jpg")

theBitmapImage.DecodePixelWidth = 200

B. theBitmapImage.BeginInit()

theBitmapImage.UriSource = New Uri("imageToDisplay.jpg")

theBitmapImage.EndInit()

C. theBitmapImage.BeginInit()

theBitmapImage.UriSource = New Uri("imageToDisplay.jpg")

theBitmapImage.DecodePixelWidth = 200

theBitmapImage.EndInit()

D. theBitmapImage.BeginInit()

theBitmapImage.UriSource = New Uri("imageToDisplay.jpg")

theBitmapImage.EndInit()

theBitmapImage.DecodePixelWidth = 200

theBitmapImage.DecodePixelHeight = 200

**Answer: C**

17. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation (WPF) 应用程序。

该应用程序包含一个以编程方式显示图像的窗口。该窗口包含一个名为 **theGrid** 的网格控件。您编写以下代码段。（所包括的行号仅供参考。）

09 theGrid.Children.Add(theImage);

01 Dim theImage As New Image()

02 theImage.Width = 200

03 Dim theBitmapImage As New BitmapImage()

04 theBitmapImage.BeginInit()

05 theBitmapImage.UriSource = New Uri("imageToDisplay.jpg")

06 theBitmapImage.DecodePixelWidth = 200

07 theBitmapImage.EndInit()

08

09 theGrid.Children.Add(theImage)

您需要裁剪图像的椭圆形区域，以使椭圆在位置 X=75 和 Y=50 处居中。

您应该在第 08 行插入下面哪个代码段？

A. Dim croppedBitmap As New CroppedBitmap(theBitmapImage, \_

New Int32Rect(75, 50, 105, 50))

theImage.Source = theBitmapImage

B. Dim croppedBitmap As New CroppedBitmap()

croppedBitmap.Source = theBitmapImage

croppedBitmap.SourceRect = New Int32Rect(75, 50, 105, 50)

theImage.Source = theBitmapImage

C. theImage.Source = theBitmapImage

Dim clipGeometry As New EllipseGeometry(New Point(75, 50), 50, 25)

theImage.Clip = clipGeometry

D. theImage.Source = theBitmapImage

Dim clipGeometry As New EllipseGeometry()

clipGeometry.Center = New Point(75, 50)

theImage.Clip = clipGeometry

**Answer: C**

18. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

该应用程序有一个针对给定目录生成目录树的 `TreeView` 类。

您编写以下代表目录的类。

`Class Folder`

`Public ReadOnly Property Name() As String`

Get

‘ ...

End Get

End Property

Public ReadOnly Property Subfolders() As List(Of Folder)

Get

‘ ...

End Get

End Property

End Class

您编写以下代码段。（所包括的行号仅供参考。）

01 Dim tree As New TreeView()

02 Dim folder As New Folder("C:\")

03

04 Dim labelFactory As New

05 FrameworkElementFactory(GetType(TextBlock))

06

07 template.VisualTree = labelFactory

08 tree.ItemTemplate = template

09 tree.ItemsSource = folder.Subfolders

您需要确保 `TreeView` 类显示与 `C:\` 驱动器的子文件夹对应的节点。

您应该在第 03 行和第 06 行分别插入下面哪个代码段？

A. 1. 在第 03 行插入以下代码段。

```
Dim template As New HierarchicalDataTemplate(folder)
```

```
template.ItemsSource = New Binding("Subfolders")
```

2. 在第 06 行插入以下代码段。

```
labelFactory.SetBinding(TextBlock.TextProperty, New Binding("Name"))
```

B. 1. 在第 03 行插入以下代码段。

```
Dim template As New HierarchicalDataTemplate(GetType(Folder))
```

```
template.ItemsSource = New Binding("Subfolders")
```

2. 在第 06 行插入以下代码段。

```
labelFactory.SetBinding(TextBlock.TextProperty, New Binding("Name"))
```

C. 1. 在第 03 行插入以下代码段。

```
Dim template As New HierarchicalDataTemplate("Folder")
```

```
template.ItemsSource = New Binding("Name")
```

2. 在第 06 行插入以下代码段。

```
labelFactory.SetBinding(TextBlock.TextProperty, _
```

```
New Binding("Subfolders"))
```

D. 1. 在第 03 行插入以下代码段。

```
Dim template As New HierarchicalDataTemplate("Folder")
```

```
template.ItemsSource = New Binding("Folder.Subfolders")
```

2. 在第 06 行插入以下代码段。

```
labelFactory.SetBinding(TextBlock.TextProperty, _
```

```
New Binding("Folder.Name"))
```

**Answer: B**

19. 您正在为汽车经销商创建一个 Windows Presentation Foundation 应用程序。您使用 Microsoft .NET Framework 3.5 创建该应用程序。

您正在创建一个将显示可用汽车列表的窗口。该列表将根据 **Make** 属性进行排序。

您尝试针对该窗口运行 XAML 代码片段。以下代码片段部分没有编译。（所包括的行号仅供参考。）

```
01 <Window.Resources>
```

```
02 ...
```

```
03 <CollectionViewSource x:Key="vw"
```

```
04 Source="{StaticResource cars}">
```

```
05 <CollectionViewSource.SortDescriptions>
```

```
06 <SortDescription PropertyName="Make" />
```

```
07 </CollectionViewSource.SortDescriptions>
```

```
08 </Window.Resources>
```

您收到以下错误消息：“找不到类型 ‘SortDescriptions’”。

您需要确保 XAML 代码段对可用汽车的列表正确地进行排序。

您应该怎么办？

A. 将 XAML 代码段的第 06 行替换为以下代码段。

```
<componentModel:SortDescription PropertyName="Make" xmlns:componentModel=
"clr-namespace:System.ComponentModel;assembly=System"/>
```

B. 将 XAML 代码段的第 06 行替换为以下代码段。

```
<componentModel:SortDescription PropertyName="Make" xmlns:componentModel=
"clr-namespace:System.ComponentModel;assembly=WindowsBase"/>
```

C. 在构造函数的 InitializeComponent 方法调用之后立即添加以下代码段。

```
Dim view As BindingListCollectionView = _
TryCast(CollectionViewSource.GetDefaultView(lst.ItemsSource), _
BindingListCollectionView)
If view IsNot Nothing Then
view.SortDescriptions.Add(New SortDescription("Make", _
ListSortDirection.Ascending))
End If
```

D. 从 XAML 代码片段中删除第 03 行到第 07 行。在构造函数的 InitializeComponent 方法调用之后立即添加以下代码段。

```
Dim view As BindingListCollectionView = _
TryCast(CollectionViewSource.GetDefaultView(lst.ItemsSource), _
BindingListCollectionView)
If view IsNot Nothing Then
view.SortDescriptions.Add(New SortDescription("Make", _
ListSortDirection.Ascending))
End if
```

**Answer: B**

20. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。该应用程序包含一个数据源和一个名为 txtUnitPrice 的 TextBox 控件。txtUnitPrice 控件的 Text 属性使用双向绑定绑定到数据源。

您编写以下代码段。

```
01 Private Sub OverridePrice(ByVal newPrice As Decimal)
```

```
02
```

```
03     txtUnitPrice.Text = newPrice
```

```
04 End Sub
```

您需要确保在调用 `OverridePrice` 方法之后，对数据源进行的更改不传播到 `txtUnitPrice`。

您应该在第 02 行插入下面哪个代码段？

A. `txtUnitPrice.Clear()`

B. `txtUnitPrice.DataContext = Nothing`

C. `BindingOperations.ClearBinding(txtUnitPrice, TextBox.TextProperty)`

D. `Dim binding As Binding = BindingOperations.GetBinding(txtUnitPrice, _  
 TextBox.TextProperty)`

```
binding.NotifyOnSourceUpdated = False
```

**Answer: C**