

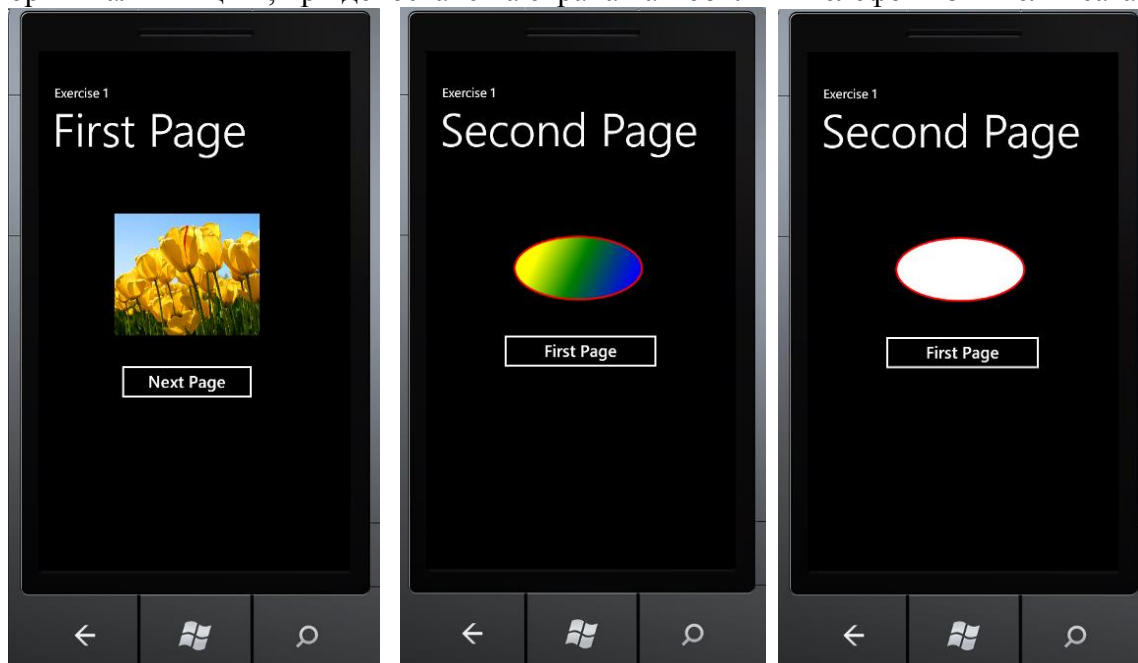
Упражнение 1

Създаване на **Windows Phone** приложение, използващо езика за програмиране **C#**, **Silverlight** и средата **Visual Studio 2010 Express for Windows Phone**.

Целта на приложението е да илюстрира:

- общата структура на **Windows Phone** приложение (един базов елемент **frame**; един или повече елемента **page**; контроли контейнери – **Grid**, **StackPanel**, **Canvas**; „прости” контроли – **TextBlock**, **Ellipse**, **Image**, **Button** и др.);
- възможността за навигация между страниците (pages) на приложението (клас **NavigationService**);
- възможността за добавяне към **.xap** файла на приложението на изображения (**.jpg** , **.png** файлове);
- възможността за обработка на събитие **Click** (**routed** събития, свързани с мишката), както и на събитие **ManipulationStarted** (**touch screen** събития от високо ниво).

Задача. Да се създаде **Windows Phone** приложение, което включва две страници. В първата страница (главната страница - **MainPage.xaml**) да се добави контрол **Image** с показано в него изображение и контрол **Button** за преход към втората страница на приложението. Втората страница на приложението да включва контрол **Ellipse** и контрол **Button** за връщане обратно към първата страница на приложението. Допълнително, във втората страница на приложението да се включи обработка на събитието **ManipulationStarted** за да се променя цветът на запълване на елипсата (атрибут **Fill**), когато се докосне вътрешността на елипсата и да се възстановява оригиналният цвят, при докосване на екрана на мобилния телефон извън елипсата.



За създаване на приложението трябва да се изпълнят следните стъпки.

1. Създайте нов проект, като използвате команда **New Project** от меню **File**. От предлаганите платформи за **Visual C#** изберете **Silverlight for Windows Phone**. От шаблоните за различни приложения изберете шаблона **Windows Phone Application** и въведете име за приложението.
2. При извеждане на прозореца за избор на **Windows Phone** платформа изберете **Windows Phone 7.1**.
3. Новото приложение е създадено и се отваря прозорецът, съответстващ на режима на проектиране (**design view**), в който освен панела на мобилния телефон е видим и генерираният по подразбиране код на **XAML** (главната страница на приложението **MainPage.xaml**). Освен **MainPage.xaml** се създава и прозорец с кода на приложението на **C#** (прозорец **MainPage.xaml.cs**)
4. По подразбиране приложението вече съдържа два контрола **Grid**. Първият контрол **Grid** съдържа контрол **StackPanel**. Контролът **StackPanel** от своя страна съдържа (по подразбиране) два контрола **TextBlock**. Първият контрол **TextBlock** се използва за изписване на името на приложението. Променете свойство **Text** на този **TextBlock** контрол, за да зададете име на приложението **Exercise 1**. Вторият контрол **TextBlock** се използва за изписване на името на страницата от приложението (в конкретния случай имаме приложение с две страници). Променете свойство **Text** на този **TextBlock** контрол за да зададете име на страницата на приложението **First Page**.
Забележка. Необходимите свойства за всички участващи в приложението контроли могат да се установят/променят, като се използва Прозорец на свойствата (**Properties window**). Селектира се съответният контрол и се установяват неговите свойства, изведени в Прозореца на свойствата. Свойствата на избран контрол може също така да се установят чрез директни промени в кода на **XAML**.
5. Променете свойство **SupportedOrientation** на страницата на приложението (**PhoneApplicationPage**). За целта е необходимо в **XAML** кода на приложението да промените стойността на свойство **SupportedOrientation**, като го установите: **SupportedOrientation = “PortraitOrLandscape”**. Целта на промяната е приложението да поддържа и двете ориентации.
6. Обичайна практика е нови контроли да се добавят към втория контрол **Grid** в страницата. В режим на проектиране от прозореца **ToolBox** добавете чрез влачене и пускане два контрола: един контрол **Image** и един контрол **Button**.
7. Контролът **Image** ще бъде използван за показване на изображение, което трябва предварително да бъде присъединено към приложението, т.е. да бъде направено част от **.xap** файла на приложението. Процедурата, която се използва за включване на изображение в **.xap** файла на приложение може да се обобщи по следния начин:
 - От меню **Project** се избира команда **Add Existing Item** (или се кликва с десния бутон на мишката върху името на проекта в прозореца **Solution Explorer** и от контекстното меню се избира команда **Add** и след това команда **Existing Item**);

- Намира се исканият файл с изображение и се кликва на бутон **Add** (за целите на настоящето упражнение е добавен файлът **Tulips.jpg**); файлът се появява в прозореца **Solution Explorer** на приложението;
- В прозореца **Solution Explorer** се селектира добавеният вече файл с изображение и за него в прозореца на свойствата се установява свойство **Build Action** = “Content” и свойство **Copy To Output Directory** = “Copy if newer”.

Забележка 1. При построение (**Build**) на приложението, всички „присъединени” към приложението по гореописания начин файлове се вграждат в **.xap** файла на приложението. (**.xap** файла е **.zip** файл и ако бъде преименуван на ***.zip** може да бъде отворен и съдържанието му разгледано).

Забележка 2. **Silverlight** за **Windows Phone** не поддържа всички графични формати.

8. В кода на **XAML** за контрола **Image** добавете атрибут **Source** = “Tulips.jpg”, в резултат на което изображението трябва да се покаже на екрана. (Може да се наложи да добавите атрибут **Stretch** = “Fill” за контрола **Image**).

9. В кода на **XAML** за контрола **Button** променете свойство **Content** = “Next Page”. По подразбиране свойство **Name** за контрола **Button** е установено в „button1“, но тъй като ще участва в кода на **C#** по-добре е да го промените на **Name** = “nextPage”.

10. В режим на проектиране (**design view**) кликнете двукратно върху бутона **Next Page** (другата възможност е да вмъкнете в **XAML** кода за бутона **Next Page** следното свойство, задаващо метод за обработка на събитието **Click**: **Click** = “nextPage_Click”).

11. Независимо от начина на създаване, в метода за обработка на събитието **Click** на бутона **Next Page** трябва да се вложи някаква функционалност. За целта в метода за обработка на събитието **Click** за бутона **Next Page** извикайте метод **Navigate** на класа **NavigationService** за да осъществите преход към втората страница на приложението:

```
private void nextPage_Click(object sender, RoutedEventArgs e)
{
    this.NavigationService.Navigate(new Uri("/Second.xaml", UriKind.Relative));
}
```

12. Кодът на **XAML** за главната страница на приложението до момента изглежда по следния начин (подчертан е само допълнително въведеният или коригиран код):

```
<phone:PhoneApplicationPage
    x:Class="Ex1.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="PortraitOrLandscape" Orientation="Portrait"
    shell:SystemTray.IsVisible="True">
```

```

<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="Transparent">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>

    <!--TitlePanel contains the name of the application and page title-->
    <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
        <TextBlock x:Name="ApplicationTitle" Text="Exercise 1"
Style="{StaticResource PhoneTextNormalStyle}"/>
        <TextBlock x:Name="PageTitle" Text="First Page" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
    </StackPanel>

    <!--ContentPanel - place additional content here-->
    <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
        <Image Source="Tulips.jpg" Height="187" HorizontalAlignment="Left"
Margin="104,57,0,0" Name="image1" Stretch="Fill" VerticalAlignment="Top" Width="223"
/>
        <Button Content="Next Page" Height="72" HorizontalAlignment="Left"
Margin="104,280,0,0" Name="nextPage" VerticalAlignment="Top" Width="223"
Click="nextPage Click" />
    </Grid>
</Grid>

<!--Sample code showing usage of ApplicationBar-->
<!--phone:PhoneApplicationPage.ApplicationBar>
    <shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
        <shell:ApplicationBarIconButton IconUri="/Images/appbar_button1.png"
Text="Button 1"/>
        <shell:ApplicationBarIconButton IconUri="/Images/appbar_button2.png"
Text="Button 2"/>
        <shell:ApplicationBar.MenuItems>
            <shell:ApplicationBarMenuItem Text="MenuItem 1"/>
            <shell:ApplicationBarMenuItem Text="MenuItem 2"/>
        </shell:ApplicationBar.MenuItems>
    </shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationBar-->
</phone:PhoneApplicationPage>

```

13. Добавете към приложението нова (втора) страница, като за целта кликнете с десния бутон на мишката върху името на проекта в прозорец **Solution Explorer** и от контекстното меню изберете команда **Add** и след това команда **New Item** - елемент **Windows Phone Portrait Page**. Сменете името на новия елемент на **Second.xaml**.

14. По подразбиране и новата страница на приложението съдържа два контрола **Grid**. В първия контрол **Grid** се съдържа отново контрол **StackPanel**, а в него два контрола **TextBlock**. Променете свойство **Text** на първия контрол **TextBlock**, за да зададете името на приложението - **Exercise 1**. Променете свойство **Text** на втория контрол **TextBlock** за да зададете име на втората страница на приложението - **Second Page**.

15. Във втория контрол **Grid** добавете (като привлечите от **ToolBox**) един контрол **Ellipse** и един контрол **Button**.

16. За контрола **Ellipse** направо в текста на **XAML** установете атрибути: **Stroke = “Red”, StrokeThickness=“3”** за да стане видим контурът на елипсата.

17. За да се запълни с цвят елипсата, установете атрибут **Fill**. Като използвате правилата на **XAML** за атрибут **Fill** създайте четка **LinearGradientBrush** с преливащи се три цвята (в примера – жълт, зелен и син). За четката е зададено име **myBrush** за да може да се обърнем към нея от кода на **C#**:

```
:  
<Ellipse . . . >  
    <Ellipse.Fill>  
        <LinearGradientBrush x:Name="myBrush">  
            <GradientStop Color="Yellow" Offset="0.2" />  
            <GradientStop Color="Green" Offset="0.5" />  
            <GradientStop Color="Blue" Offset="0.9" />  
        </LinearGradientBrush>  
    </Ellipse.Fill>  
</Ellipse>
```

18. За контрола **Ellipse** добавете събитие **ManipulationStarted**, заедно със съответния метод за обработка.

19. В метода за обработка на събитието **ManipulationStarted** вмъкнете следния код, с който промените цвета на запълване на елипсата от преливащ цвят в чисто бял цвят (едноцветна четка **SolidColorBrush**), когато се докосне екрана вътре в очертанията на елипсата:

```
private void ellipse1_ManipulationStarted(object sender, ManipulationStartedEventArgs e)  
{  
    ellipse1.Fill = new SolidColorBrush(Colors.White);  
    e.Complete();  
    e.Handled = true;  
}
```

20. В кода на **C#** за втората страница на приложението **override** метод **OnManipulationStarted**, като добавите нова функционалност, позволяваща да се възстанови оригиналният преливащ се цвят на елипсата, когато се докосне екрана извън очертанията ѝ:

```
protected override void OnManipulationStarted(ManipulationStartedEventArgs e)  
{  
    ellipse1.Fill = myBrush;  
    e.Complete();  
    e.Handled = true;  
    base.OnManipulationStarted(e);  
}
```

21. За контрола **Button** променете свойство **Content = “First Page”**. По подразбиране свойство **Name** за контрола **Button** е установено в „**button1**“, но тъй като ще участва в кода на **C#** по-добре е да го промените на **Name = “firstPage”**.

22. За контрола **Button** добавете събитие **Click**, заедно със съответния метод за обработка.

23.В метода за обработка на събитието **Click** за бутона **First Page** извикайте метод **GoBack()** на класа **NavigationService**, за да се върнете обратно към първата страница на приложението:

```
private void firstPage_Click(object sender, RoutedEventArgs e)
{
    this.NavigationService.GoBack();
}
```

24. Кодът на **XAML** за втората страница на приложението до момента изглежда по следния начин (подчертан е само допълнително въведеният или коригиран код):

```
<phone:PhoneApplicationPage
    x:Class="Ex1.Second"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    mc:Ignorable="d" d:DesignHeight="768" d:DesignWidth="480"
    shell:SystemTray.IsVisible="True">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions>

        <!--TitlePanel contains the name of the application and page title-->
        <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
            <TextBlock x:Name="ApplicationTitle" Text="Exercise 1"
                Style="{StaticResource PhoneTextNormalStyle}"/>
            <TextBlock x:Name="PageTitle" Text="Second Page" Margin="9,-7,0,0"
                Style="{StaticResource PhoneTextTitle1Style}"/>
        </StackPanel>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
            <Ellipse Stroke="Red" Height="100" HorizontalAlignment="Left"
                Margin="123,92,0,0" Name="ellipse1" StrokeThickness="3" VerticalAlignment="Top"
                Width="200" ManipulationStarted="ellipse1_ManipulationStarted">
                <Ellipse.Fill>
                    <LinearGradientBrush x:Name="myBrush">
                        <GradientStop Color="Yellow" Offset="0.2" />
                        <GradientStop Color="Green" Offset="0.5" />
                        <GradientStop Color="Blue" Offset="0.9" />
                    </LinearGradientBrush>
                </Ellipse.Fill>
            </Ellipse>

            <Button Content="First Page" Height="72" HorizontalAlignment="Left"
                Margin="97,234,0,0" Name="firstPage" VerticalAlignment="Top" Width="259"
                Click="firstPage_Click" />
        </Grid>
    </Grid>
</phone:PhoneApplicationPage>
```

```
        </Grid>
</Grid>

<!--Sample code showing usage of ApplicationBar-->
<!--<phone:PhoneApplicationPage.ApplicationBar>
    <shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
        <shell:ApplicationBarIconButton IconUri="/Images/appbar_button1.png"
Text="Button 1"/>
        <shell:ApplicationBarIconButton IconUri="/Images/appbar_button2.png"
Text="Button 2"/>
        <shell:ApplicationBar.MenuItems>
            <shell:ApplicationBarMenuItem Text="MenuItem 1"/>
            <shell:ApplicationBarMenuItem Text="MenuItem 2"/>
        </shell:ApplicationBar.MenuItems>
    </shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationBar-->

</phone:PhoneApplicationPage>
```

25. Компилирайте и построете приложението като изберете команда **Build Solution** от меню **Debug**. Ако има грешки, коригирайте ги и компилирайте отново.

26. От стандартната лента с икони (standard tool bar) изберете как ще бъде изпълнено приложението, като посочите **Windows Phone Emulator** (или **Windows Phone Device**, ако има свързан към компютъра мобилен телефон).

27. Стартирайте приложението като изберете команда **Start Debugging** от меню **Debug**.