

Министерство образования и науки Российской Федерации
ФГБОУ ВПО Уральский государственный лесотехнический университет

Мельник Любовь Юрьевна

ПРОЕКТИРОВАНИЕ ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

Методические указания по выполнению лабораторно-практического цикла для студентов направления «Бизнес - информатика» (очной, очно-заочной и заочной формы обучения)

ЕКАТЕРИНБУРГ 2015

Лабораторная работа 1 «Первое приложение Магазина Windows»

В этой лабораторной работе вы создадите свое первое приложение Магазина Windows, добавите простую функциональность и изучите работу симулятора, после чего создадите пакет этого приложения для дальнейшего развертывания.

- 1) Откройте Visual Studio Express 2012 for Windows 8.
- 2) После того как Visual Studio Express загрузится, вам будет предложено бесплатно получить лицензию разработчика
- 3) Для получения лицензии разработчика нажмите на кнопку **I Agree**.
- 4) В открывшемся диалоговом окне введите данные своей учетной записи и нажмите кнопку **Sign in**. Получите лицензию разработчика. После того, как истечет срок лицензии, вы можете обновить ее, используя VisualStudio 2012 – в VisualStudioExpress нажмите **STORE | Acquire Developer Licence...** в старших версиях VisualStudio нажмите **PROJECT |Store | Acquire Developer Licence....**
- 5) Выберите пункт меню **Файл/Создать/Новый проект**.
- 6) В левой части открывшегося диалогового окна в раздел выберите **Visual C#Пустое приложение (XAML)**.
- 7) Введите в поле **Имя** название приложения Lab1.
- 8) Укажите в поле **Расположение** директорию, в которой будет размещаться проект приложения.
- 9) Нажмите **ОК**. В результате будет создан новый проект Windows Store-приложения шаблона **Пустое приложение(XAML)** и будет открыт файл App.xaml.
- 10) Изучите код App.xaml. Что в нем происходит?

11) Нажмите F5, чтобы запустить приложение на отладку, либо кнопку с зеленой стрелкой на панели инструментов, либо выберите пункт меню **Debug | Start Debugging**.

12) Перейдите на стартовый экран – ваше приложение было установлено в систему.

13) Перейдите обратно в VisualStudio 2012.

14) Изучите структуру проекта. Какие файлы находятся в проекте? Зачем они? Изучите их содержимое.

15) Нажмите два раза левой кнопкой мыши на файле Package.appxmanifest, в **Обозревателе решений**.

16) Изучите редактор конфигурации манифеста.

17) Нажмите правой кнопкой мыши на файле Package.appxmanifest.

18) Нажмите **View Code...**

19) Изучите содержимое манифеста приложения.

Внесите изменения в редакторе конфигурации манифеста, сохраните и изучите, что изменилось в самом коде манифеста.

20) Откройте файл MainPage.xaml.

21) Добавьте вместо тегов <Grid> и </Grid> приведенный ниже код.

```
<Grid Background="{StaticResource
ApplicationPageBackgroundThemeBrush}">
  <Button Content="Hello world!"
    HorizontalAlignment="Center" VerticalAlignment="Center"/>
</Grid>
```

22) Добавьте для кнопки обработчик события **Click**. Это можно сделать, два раза нажав левой кнопкой мыши на кнопке в графическом редакторе. Итоговый вариант XAML-кода MainPage.xaml приведен ниже.

```
<Page
  x:Class="HelloWorldApp.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:HelloWorldApp"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d">

  <Grid Background="{StaticResource
ApplicationPageBackgroundThemeBrush}">
    <Button Content="Hello world!"
      HorizontalAlignment="Center" VerticalAlignment="Center"
Click="Button_Click_1"/>

  </Grid>
</Page>
```

23) Нажмите два раза левой кнопкой мыши на кнопке в графическом редакторе.

24) Добавьте в обработчик события приведенный ниже код.

```
var messagedialog = new MessageDialog("Hello World!",  
"Hello World App Dialog Title");  
messagedialog.ShowAsync();
```

25) Нажмите **F5**, чтобы запустить приложение.

26) Завершите работу приложения.

27) Замените обработчик события **Button_Click_1** на код, приведенный ниже.

```
private async void Button_Click_1(object sender,  
RoutedEventArgs e)  
{  
    var messagedialog = new MessageDialog("Hello World!", "Hello  
World App Dialog Title ");  
  
    messagedialog.Commands.Add(new UICommand("Button  
1"));  
    messagedialog.Commands.Add(new UICommand("Button 2"));  
    messagedialog.DefaultCommandIndex = 1;  
    messagedialog.CancelCommandIndex = 2;  
  
    var command = await messagedialog.ShowAsync();  
}
```

28) Нажмите **F5**, чтобы запустить приложение.

29) Завершите работу приложения.

30) Измените в выпадающем списке кнопки запуска

пункт **Local Machine** на **Simulator**

31) Нажмите F5, чтобы запустить приложение в симуляторе.

32) Изучите работу всех функций симулятора на примере вашего приложения.

33) Завершите работу симулятора.

34) Соберите пакет приложения:

нажмите **ПРОЕКТ/Магазин/Создать пакеты приложения**.

35) Ответьте "**Нет**" на вопрос о том, необходимо ли создать пакет построения для загрузки в Магазин Windows.

36) Нажмите кнопку **Далее**.

37) Укажите место размещения пакета.

38) Нажмите кнопку **Создать**.

39) Смените расширение созданного файла Lab1_1.0.0.0_AnyCPU_Debug.appx на **zip** и изучите его содержимое.

40) Закройте Visual Studio 2012.

Итоги

В этой лабораторной работе вы создали свое первое приложение Магазина Windows, добавили простую функциональность и изучили работу симулятора. Также вы

создали пакет этого приложения для дальнейшего развертывания.

Лабораторная работа 2 «Страницы и навигация в приложении»

В этой лабораторной работе вы создадите приложение Магазина Windows, добавите страницу шаблона BasicPage, организуете простой обмен сообщениями между двумя страницами и познакомитесь с новым элементом управления HyperLinkButton. Вы узнаете, как определять особые параметры HyperLinkButton, позволяющие переопределить поведение при открытии данных гиперссылок.

41) Откройте Visual Studio 2012.

42) После того как VisualStudioExpress загрузится, вам будет предложено бесплатно получить лицензию разработчика

43) Для получения лицензии разработчика нажмите на кнопку **I Agree**.

44) В открывшемся диалоговом окне введите данные своей учетной записи и нажмите кнопку **Sign in**. Получите лицензию разработчика. После того, как истечет срок лицензии, вы можете обновить ее, используя VisualStudio 2012 – в VisualStudioExpress нажмите

STORE | Acquire Developer Licence... в старших версиях

VisualStudio

нажмите **PROJECT |Store | Acquire Developer Licence....**

45) Выберите пункт меню **Файл/Создать/Новый проект.**

46) В левой части открывшегося диалогового окна выберите **Visual C#Пустое приложение (XAML).**

47) Введите в поле **Имя** название приложения Lab21.

48) Укажите в поле **Location** директорию, в которой будет размещаться проект приложения.

49) Нажмите **ОК.** В результате будет создан новый проект Windows Store-приложения шаблона **Пустое приложение(XAML)** и будет открыт файл App.xaml. Нажмите правой кнопкой мыши на имени проекта в Solution Explorer (**Обозреватель решений**) и нажмите **Add |NewItem (Добавить | Создать элемент)...**

50) В открывшемся диалоговом окне выберите шаблон файла Основная страница.

51) Введите имя новой страницы **Second.xaml.**

52) Нажмите Добавить.

53) На вопрос о добавлении необходимых зависимостей ответьте «Да» (Yes). Выполните построение проекта (Нажмите F5).

54) Изучите содержимое новой страницы. В данном шаблоне используется дополнительно заголовок страницы и кнопка «Назад».

55) Откройте файл MainPage.xaml.

56) Замените теги <Grid> и </Grid> на приведенный ниже код.

```
<Grid Background="{StaticResourceApplicationPageBackgroundThemeBrush}">  
  <Button Content="Hello second page!"  
    HorizontalAlignment="Center" VerticalAlignment="Center"/>  
</Grid>
```

57) Добавьте для кнопки обработчик события **Click**. Это можно сделать, два раза нажав левой кнопкой мыши на кнопке в графическом редакторе.

58) Добавьте в обработчик события приведенный ниже код.

```
this.Frame.Navigate(typeof(Second));
```

59) Нажмите F5, чтобы запустить приложение. Теперь нажатие на кнопку приведет пользователя на страницу Second.

60) Завершите работу приложения.

61) Замените код обработчика события на приведенный ниже код.

```
this.Frame.Navigate(typeof(Second), "Hello Second Page!");
```

62) Нажмите правой кнопкой мыши на странице Second.xaml.

63) Нажмите **Перейти к коду...**

64) Добавьте в файл приведенный ниже код.

```
protectedoverridevoidOnNavigatedTo(NavigationEventArgs  
e)  
{
```

```
string parameter = e.ParameterasString;  
var messagedialog = new MessageDialog("Hello Second Page!",  
"" );  
messagedialog.ShowAsync();  
}
```

65) Нажмите F5, чтобы запустить приложение.

66) Нажмите на кнопку, чтобы перейти на вторую страницу и проверить работоспособность приложения.

67) Завершите работу приложения.

68) Откройте файл Second.xaml.

69) Создайте гиперссылку на bing.com, добавьте в элемент Grid после строки <TextBlock... приведенный ниже код.

```
<HyperlinkButtonNavigateUri="http://bing.com"  
Content="Выоткрываете bing.com!" Grid.Column="1"  
Margin="269,209,0,-108" Foreground="#FF17E6DC" />
```

70) Нажмите F5, чтобы запустить приложение.

71) Изучите работу приложения.

72) Завершите работу приложения.

73) Замените код обработчика события Click для гиперссылки на приведенный ниже код.

```
private async void HyperlinkButton_Click_1(object sender,  
RoutedEventArgs e)  
{  
var opts = new LauncherOptions();  
opts.TreatAsUntrusted = true;  
opts.DisplayApplicationPicker = true;
```

```
var success = await Launcher.LaunchUriAsync(  
    new Uri("http://yandex.ru"), opts);  
if (success)  
    {  
    }  
else  
    {  
    }
```

```
var messagedialog = new MessageDialog("Выоткрываете  
bing.com!");  
messagedialog.ShowAsync();  
}
```

Параметр `TreatAsUntrusted` указывает на то, что ссылка не является доверенной и при ее открытии будет показано предупреждение.

74) Нажмите F5, чтобы запустить приложение.

75) Изучите работу приложения.

76) Завершите работу приложения.

77) Закройте Visual Studio 2013.

Итоги

В этой лабораторной работе вы создали приложение Магазина Windows, добавили страницу шаблона `BasicPage`, организовали простой обмен сообщениями между двумя страницами и познакомились с новым элементом управления `HyperLinkButton`. Вы узнали, что у `HyperLinkButton` есть особые параметры, позволяющие переопределить поведение при открытии данных гиперссылок.

Лабораторная работа 3 «Создание приложения магазина Windows. Элементы управления»

В этой лабораторной работе вы создадите приложение Магазина Windows, добавьте элементы управления GridView, ListView и FlipView и изучите базовые принципы их использования.

- 1) Откройте Visual Studio Express 2012 for Windows 8.
- 2) После того как VisualStudioExpress загрузится, вам будет предложено бесплатно получить лицензию разработчика
- 3) Для получения лицензии разработчика нажмите на кнопку **I Agree**.
- 4) В открывшемся диалоговом окне введите данные своей учетной записи и нажмите кнопку **Sign in**. Получите лицензию разработчика. После того, как истечет срок лицензии, вы можете обновить ее, используя VisualStudio 2012 – в VisualStudioExpress нажмите **STORE | Acquire Developer Licence...** в старших версиях VisualStudio нажмите **PROJECT |Store | Acquire Developer Licence**
....
- 5) Выберите пункт меню **File | New Project**.
- 6) В левой части открывшегося диалогового окна **New Project** в разделе **Installed | Templates** выберите **Visual C# | Windows Store | Blank App (XAML)**.
- 7) Введите в поле **Name** название приложения ControlsApp

8) Укажите в поле **Location** директорию, в которой будет размещаться проект приложения.

9) Нажмите **ОК**. В результате будет создан новый проект Windows Store-приложения шаблона **BlankApp (XAML)** и будет открыт файл App.xaml.

10) Откройте файл MainPage.xaml.

11) Замените контейнер Grid на приведенный ниже код.

```
<Grid Background="{StaticResource  
ApplicationPageBackgroundThemeBrush}">
```

```
<Grid.RowDefinitions>
```

```
<RowDefinition Height="140"/>
```

```
<RowDefinition Height="*/>
```

```
</Grid.RowDefinitions>
```

```
<TextBlock Text="Controls App" Margin="120,0,30,40"  
Style="{StaticResourcePageHeaderText}">
```

```
<GridView x:Name="gridview" Grid.Row="1"  
Padding="116,37,40,46">
```

```
</GridView>
```

```
</Grid>
```

12) Создайте ПРОЕКТ/Создать класс с именем Person.

13) Замените код класса Person на приведенный ниже код.

```
public class Person
```

```
{  
public string FirstName { get; set; }  
public string LastName { get; set; }  
}
```

- 14) Замените содержимое класса MainPage в файле MainPage.xaml.cs на приведенный ниже код.

```
privateObservableCollection<Person>persons;  
  
publicMainPage()  
{  
this.InitializeComponent();  
  
Person person1 = newPerson  
{  
LastName = "Ivanov",  
FirstName = "Ivan"  
};  
  
Person person2 = newPerson  
{  
LastName = "Petrov",  
FirstName = "Petr"  
};  
  
Person person3 = newPerson  
{  
LastName = "Alexandrov",  
FirstName = "Alexandr"  
};  
  
persons = newObservableCollection<Person>();  
  
persons.Add(person1);  
persons.Add(person2);  
persons.Add(person3);  
  
gridview.ItemsSource = persons;  
}
```

}

Закомментируйте строки, начиная с **PROTECTED** ...

{

}

- 15) Нажмите F5, чтобы запустить приложение.
- 16) Изучите работу приложения.
- 17) Завершите работу приложения.
- 18) Добавьте самостоятельно 2-3 пользователя.
- 19) Замените вставленный ранее код XAMLв MainPage.xaml с разметкой для GridView на приведенный ниже код.

```
<GridView x:Name="gridview" Grid.Row="1"
```

```
Padding="116,37,40,46">
```

```
<GridView.ItemTemplate>
```

```
<DataTemplate>
```

```
<Grid HorizontalAlignment="Left" Width="160" Height="160">
```

```
<Border Background="{StaticResource  
ListViewItemPlaceholderBackgroundThemeBrush}">
```

```
<TextBlock Text="Пользователь:"></TextBlock>
```

```
</Border>
```

```
<StackPanelVerticalAlignment="Bottom"
```

```
Background="{StaticResource  
ListViewItemOverlayBackgroundThemeBrush}">
```

```
<TextBlock Text="{Binding FirstName}"
```

```
Foreground="{StaticResource  
ListViewItemOverlayForegroundThemeBrush}"
```

```
Style="{StaticResourceTitleTextStyle}"
```

```
Margin="15,0,15,0"/>
```

```
<TextBlock Text="{Binding LastName}"  
    Foreground="{StaticResource  
ListViewItemOverlayForegroundThemeBrush}"  
    Style="{StaticResourceTitleTextStyle}"  
    Margin="15,0,15,0"/>
```

```
</StackPanel>
```

```
</Grid>
```

```
</DataTemplate>
```

```
</GridView.ItemTemplate>
```

```
</GridView>
```

- 20) Нажмите F5, чтобы запустить приложение.
- 21) Изучите работу приложения.
- 22) Завершите работу приложения.
- 23) Замените код, содержащий разметку GridView, и все ссылки на gridview в коде MainPage.xaml на приведенный ниже код и gridview соответственно.

```
<ListViewx:Name="listview" Grid.Row="1"  
Margin="0,-10,0,0" Padding="10,0,0,60" Width="320">  
<ListView.ItemTemplate>  
<DataTemplate>  
<Grid Margin="6">  
<Grid.ColumnDefinitions>  
<ColumnDefinition Width="Auto"/>  
<ColumnDefinition Width="*/>  
</Grid.ColumnDefinitions>
```

```
<Border Background="{StaticResource
ListViewItemPlaceholderBackgroundThemeBrush}"
        Width="60" Height="60">
    <TextBlock Text="Пользователь:"></TextBlock>

</Border>
<StackPanelGrid.Column="1" Margin="10,0,0,0">
<TextBlock Text="{Binding FirstName}" MaxHeight="40"/>

<TextBlock Text="{Binding LastName}"
        Style="{StaticResourceCaptionTextStyle}"
TextWrapping="NoWrap"/>

</StackPanel>
</Grid>
</DataTemplate>
</ListView.ItemTemplate>
<ListView.GroupStyle>
<GroupStyle>
<GroupStyle.HeaderTemplate>
<DataTemplate>
<Grid Margin="7,7,0,0">
<Button

Style="{StaticResourceTextPrimaryButtonStyle}">
<StackPanel Orientation="Horizontal">
<TextBlock Text="{Binding GroupName}"
        Margin="3,-7,10,10"
```

```
Style="{StaticResourceGroupHeaderTextStyle}"
/>
<TextBlock Text="{StaticResourceChevronGlyph}"
FontFamily="Segoe UI Symbol" Margin="0,-7,0,10"

Style="{StaticResourceGroupHeaderTextStyle}"/>
</StackPanel>
</Button>
</Grid>
</DataTemplate>
</GroupStyle.HeaderTemplate>
</GroupStyle>
</ListView.GroupStyle>
</ListView>
    24) Нажмите F5, чтобы запустить приложение.
    25) Изучите работу приложения.
    26) Завершите работу приложения.
    27) Замените определение ListView в MainPage.xaml на
        код, приведенный ниже.
<ListView x:Name="listview" Grid.Row="1"
    Margin="0,-10,0,0" Padding="10,0,0,60" Width="320"
SelectionMode="Multiple">
    28) Нажмите F5, чтобы запустить приложение.
    29) Изучите работу приложения. Обратите внимание, что
        с изменением значения SelectionMode у пользователя
        появилась возможность выделять несколько элементов.
        Изучите другие режимы SelectionMode.
    30) Завершите работу приложения.
```

- 31) Замените код, содержащий разметку ListView, и все ссылки на listview в коде MainPage.xaml.cs на приведенный ниже код и flipview соответственно.

```
<FlipView Name="flipview" Grid.Row="1"
Width="250" Height="250">
<FlipView.ItemTemplate>

<DataTemplate>
<Grid Margin="6">
<Grid.ColumnDefinitions>
<ColumnDefinition Width="Auto"/>
<ColumnDefinition Width="*/>
</Grid.ColumnDefinitions>

<StackPanelGrid.Column="1" Margin="10,0,0,0"
Background="Gray">
<TextBlock Text="{Binding FirstName}" MaxHeight="40"/>

<TextBlock Text="{Binding LastName}"
Style="{StaticResourceCaptionTextStyle}"
TextWrapping="NoWrap"/>

</StackPanel>
</Grid>
</DataTemplate>
</FlipView.ItemTemplate>
</FlipView>
```

- 32) Нажмите F5, чтобы запустить приложение.

- 33) Изучите работу приложения.
- 34) Завершите работу приложения.
- 35) Закройте Visual Studio 2012.

Итоги

В этой лабораторной работе вы создали приложение Магазина Windows, добавили элементы управления GridView, ListView и FlipView и изучили базовые принципы их использования.

Лабораторная работа 4: «Модель исполнения приложения», зарегистрировать фоновую задачу»

В этой лабораторной работе вы создадите приложение Магазина Windows, зарегистрируете фоновую задачу, запустите приложение и изучите его работу, после чего добавите функциональность отмены всех фоновых задач.

- 1) Откройте Visual Studio Express 2013 for Windows 8.
- 2) После того как Visual Studio Express загрузится, вам будет предложено бесплатно получить лицензию разработчика
- 3) Для получения лицензии разработчика нажмите на кнопку **I Agree**.
- 4) В открывшемся диалоговом окне введите данные своей учетной записи и нажмите кнопку **Sign in**. Получите лицензию разработчика. После того, как истечет срок лицензии, вы можете обновить ее, используя Visual Studio 2012 – в Visual Studio Express нажмите **STORE|Acquire**

- Developer Licence...** в старших версиях Visual Studio нажмите **PROJECT | Store | Acquire Developer Licence....**
- 5) Выберите пункт меню **Файл | Создать Проект.**
 - 6) В левой части открывшегося диалогового окна **New Project** в разделе **Installed | Templates** выберите **Visual C# | Магазин Windows | Пустое приложение (XAML).**
 - 7) Введите в поле **Name** название приложения **Фоновая задача**
 - 8) Укажите в поле **Location** директорию, в которой будет размещаться проект приложения.
 - 9) Нажмите **ОК.** В результате будет создан новый проект Windows Store-приложения шаблона **Пустое приложение (XAML)** и будет открыт файл **Фоновая задача.xaml.**
 - 10) Выберите **Файл | Добавить | Новый проект...**
 - 11) В открывшемся диалоговом окне укажите тип проекта — **Windows Runtime Component** и имя проекта — **BackgroundTaskComponent (ФоноваяЗадача).**
 - 12) Нажмите **ОК.**
 - 13) Нажмите правой кнопкой мыши на директории **References (Рекомендации)** проекта **BackgroundTaskApp.**
 - 14) Нажмите **Add Reference (Добавить ссылку)...**
 - 15) Выберите проект **BackgroundTaskComponent (Фоновая Задача),** который будет находиться в ветке **Solution.**
 - 16) Нажмите правой кнопкой мыши на проекте **BackgroundTaskComponent.**
 - 17) Выберите **Добавить | Новый элемент...**

- 18) Выберите шаблон Class.
- 19) Введите имя файла ЗадачаОбразец.cs.
- 20) Нажмите ОК.
- 21) Замените содержимое созданного файла на приведенный ниже код.

```
public sealed class ЗадачаОбразец : IBackgroundTask
{
    public void Run(IBackgroundTaskInstance taskInstance)
    {
        var deferral = taskInstance.GetDeferral();
        try
        {

        }
        finally
        {
            deferral.Complete();
        }
    }
}
```

- 22) Откройте файл Package.appxmanifest.
- 23) Перейдите на вкладку Declarations (Объявление).
- 24) Добавьте объявление типа Background Tasks (Фоновые задачи).
- 25) Выберите в списке Supported task types (Поддерживаемые типы задач) тип задачи System event.

- 26) Введите значение входной точки (Entry point) `BackgroundTaskComponent.SampleTask` (Фоновая Задача Образец).
- 27) Откройте файл `MainPage.xaml`.
- 28) Добавьте между тегами `<Grid>` и `</Grid>` приведенный ниже код.

```
<Grid Background="{StaticResource
ApplicationPageBackgroundThemeBrush}">
    <Button Content="Test Фоновая задача."
        HorizontalAlignment="Center" VerticalAlignment="Center"
    />
</Grid>
```

- 29) Добавьте для кнопки обработчик события **Click**. Это можно сделать, два раза нажав левой кнопкой мыши на кнопке в графическом редакторе либо задав событие **Click** элемента `Button` непосредственно в XAML-коде с помощью кнопки **New Event handler (Новый обработчик события)**.
- 30) Нажмите правой кнопкой мыши на событии **Click**.
- 31) Нажмите **Navigate to event handler (Перейдите в обработчик события)**, чтобы перейти к коду обработчика события.
- 32) Замените содержимое метода обработчика события на приведенный ниже код.

```
var timerTrigger = new
```

```
SystemTrigger(SystemTriggerType.TimeZoneChange, false);
```

```
var builder = new BackgroundTaskBuilder();
```

```
builder.Name = "SampleTask";
```

```
builder.TaskEntryPoint =
```

```
"BackgroundTaskComponent.SampleTask";
```

```
builder.SetTrigger(timerTrigger);
```

```
IBackgroundCondition condition = new
```

```
SystemCondition(SystemConditionType.InternetAvailable);
```

```
builder.AddCondition(condition);
```

```
BackgroundTaskRegistration task = builder.Register();
```

- 33) Добавьте между тегами <Grid> и </Grid> еще одну кнопку.

```
<Button Content="Cancel background task."  
HorizontalAlignment="Center" VerticalAlignment="Center"  
>
```

- 34) Добавьте для кнопки обработчик события **Click**. Это можно сделать, два раза нажав левой кнопкой мыши на кнопке в графическом редакторе либо задав событие **Click** элемента Button непосредственно в XAML-коде с помощью кнопки **New Event handler**.

- 35) Нажмите правой кнопкой мыши на событии **Click**.

- 36) Нажмите **Navigate to event handler (Перейдите в обработчик события)**, чтобы перейти к коду обработчика события.
- 37) Замените содержимое метода обработчика события на приведенный ниже код.

```
foreach (var task in BackgroundTaskRegistration.AllTasks)
{
    task.Value.Unregister(true);
}
```

- 38) Нажмите F5, чтобы запустить приложение.
- 39) Изучите работу приложения. Во время работы приложения инициируется смена зоны вручную либо вызвав фоновую задачу из Visual Studio с помощью кнопки на панели Debug Location, на которой при запуске приложения должна появиться кнопка запуска фоновой задачи SampleTask.
- 40) Завершите работу приложения.

Итоги

В этой лабораторной работе вы создали приложение Магазина Windows, зарегистрировали фоновую задачу, запустили приложение и изучили его работу, после чего добавили функциональность отмены всех фоновых задач.

Лабораторная работа 5: «Живые плитки и уведомления»

Прочтите и выполните указания, приведенные в <http://www.spugachev.com/windows8book>, Глава 11. Уведомления, Глава 12. «Живые» плитки.

Лабораторная работа 6: «Интеграция с системой и облаком»

В этой лабораторной работе вы создадите приложение Магазина Windows, зарегистрируете и изучите контракты Поиск и Настройка, запустите приложение и изучите его работу.

- 1) Откройте Visual Studio Express 2013 for Windows 8.
- 2) После того как Visual Studio Express загрузится, вам будет предложено бесплатно получить лицензию разработчика
- 3) Для получения лицензии разработчика нажмите на кнопку **I Agree**.
- 4) В открывшемся диалоговом окне введите данные своей учетной записи и нажмите кнопку **Sign in**. Получите лицензию разработчика. После того, как истечет срок лицензии, вы можете обновить ее, используя Visual Studio 2012 – в Visual Studio Express нажмите **STORE|Acquire Developer Licence...** в старших версиях Visual Studio нажмите **PROJECT |Store | Acquire Developer Licence....**
- 5) Выберите пункт меню **File | New Project**.
- 6) В левой части открывшегося диалогового окна **New Project** в разделе **Installed | Templates** выберите **Visual C# | Windows Store | Blank App (Пустое приложение) (XAML)**.
- 7) Введите в поле **Name** название приложения ContractsApp (Lab7)

- 8) Укажите в поле **Location** директорию, в которой будет размещаться проект приложения.
- 9) Нажмите **OK**. В результате будет создан новый проект Windows Store-приложения шаблона **Blank App (Пустое приложение)(XAML)** и будет открыт файл App.xaml.
- 10) Нажмите правой кнопкой мыши на файле Package.appxmanifest, чтобы открыть редактор конфигурации манифеста.
- 11) Перейдите на вкладку Declarations (Объявления).
- 12) В выпадающем списке Available Declarations выберите пункт Search (Поиск)
- 13) Нажмите кнопку Add.
- 14) Добавьте в проект страницу SearchPage.xaml, используя шаблон Basic Page (Основная страница).
- 15) Добавьте в файле App.xaml.cs приведенный ниже код.

(Добавлять нужно после строк

```
private void OnSuspending(object sender, SuspendingEventArgs e)
{
    var deferral = e.SuspendingOperation.GetDeferral();
    //TODO: Save application state and stop any background activity
    deferral.Complete();
}
```

```
)
protected override void OnSearchActivated(
    SearchActivatedEventArgs args)
{
    Frame rootFrame = Window.Current.Content as
Frame;

    if (rootFrame == null)
    {
```

```
rootFrame = new Frame();  
Window.Current.Content = rootFrame;  
}
```

```
rootFrame.Navigate(typeof(SearchPage),  
args.QueryString);
```

```
Window.Current.Activate();  
}
```

- 16) Замените в файле SearchPage.xaml.cs содержимое метода OnNavigatedTo приведенный ниже код.

```
protected override void  
OnNavigatedTo(NavigationEventArgs e)  
{  
    var queryText = (string)e.Parameter;  
    pageTitle.Text = String.Format("Вы искали \"{0}\"",  
        queryText);  
  
    navigationHelper.OnNavigatedTo(e);  
}
```

- 17) Добавьте на страницу SearchPage.xaml приведенный ниже код разметки для кнопки и панели приложения.

```
<Page.BottomAppBar>  
    <AppBar>  
        <AppBarButton Icon="Home"/>  
    </AppBar>
```

</Page.BottomAppBar>

- 18) Добавьте обработчик нажатия кнопки панели (при помощи двойного клика мышью).
- 19) Добавьте в обработчик нажатия кнопки на панели приведенный ниже код.

```
Frame.Navigate(typeof(MainPage));
```

- 20) Нажмите F5, чтобы запустить приложение.
- 21) Вызовите панель чудо-кнопок и нажмите на чудо-кнопку Поиск.
- 22) Найдите свое приложение и введите поисковой запрос.
- 23) Изучите работу приложения.
- 24) Завершите работу приложения.
- 25) Создайте новое приложение шаблона Blank App.
- 26) Введите имя приложения SettingsContractApp.
- 27) Нажмите Ok.
- 28) Замените содержимое класса App в файле App.xaml.cs на приведенный ниже код.

```
public App()  
{  
    this.InitializeComponent();  
    this.Suspending += OnSuspending;  
}
```

```
protected override void
OnLaunched(LaunchActivatedEventArgs args)
{
    Frame rootFrame = Window.Current.Content as Frame;

    // Do not repeat app initialization when the Window
already has content,
    // just ensure that the window is active
    if (rootFrame == null)
    {
        // Create a Frame to act as the navigation context and
navigate to the first page
        rootFrame = new Frame();

        if (args.PreviousExecutionState ==
ApplicationExecutionState.Terminated)
        {
            //TODO: Load state from previously suspended
application
        }

        // Place the frame in the current Window
        Window.Current.Content = rootFrame;
    }

    if (rootFrame.Content == null)
    {
```

Электронный архив УГЛТУ

```
// When the navigation stack isn't restored navigate to
the first page,
// configuring the new page by passing required
information as a navigation
// parameter
if (!rootFrame.Navigate(typeof(MainPage),
args.Arguments))
{
    throw new Exception("Failed to create initial
page");
}
}
// Ensure the current window is active
Window.Current.Activate();
Subscribe();

Window.Current.Activate();

}

private void OnSuspending(object sender,
SuspendingEventArgs e)
{
    var deferral = e.SuspendingOperation.GetDeferral();
    Unsubscribe();
    deferral.Complete();
}
```

```
public void Subscribe()  
{  
    var currentPane = SettingsPane.GetForCurrentView();  
    currentPane.CommandsRequested +=  
currentPane_CommandsRequested;  
}
```

```
public void Unsubscribe()  
{  
    var currentPane = SettingsPane.GetForCurrentView();  
    currentPane.CommandsRequested -=  
currentPane_CommandsRequested;  
}
```

```
void currentPane_CommandsRequested(SettingsPane  
sender,  
SettingsPaneCommandsRequestedEventArgs args)  
{  
  
}
```

- 29) Нажмите F5, чтобы запустить приложение.
- 30) Изучите работу приложения. Во время работы приложения вызовите панель чудо-кнопок и нажмите на чудо-кнопку Параметры.
- 31) Завершите работу приложения.

- 32) Замените содержимое метода `currentPane_CommandsRequested` в файле `App.xaml.cs` на приведенный ниже код.

```
void currentPane_CommandsRequested(SettingsPane sender, SettingsPaneCommandsRequestedEventArgs args)
{
    var applicationCommands = args.Request.ApplicationCommands;

    var newCommand = new SettingsCommand("Команда", "Общая информация", cmd =>
    {

    });
    applicationCommands.Add(newCommand);

    var aboutCommand = new SettingsCommand("Команда 2", "О приложении", cmd =>
    {

    });
    applicationCommands.Add(aboutCommand);

    var siteCommand = new SettingsCommand("Bing.com", "Bing.com", cmd =>
    {
```

```
Launcher.LaunchUriAsync(new  
Uri("http://bing.com"));  
});  
applicationCommands.Add(siteCommand);  
}
```

- 33) Нажмите F5, чтобы запустить приложение.
- 34) Изучите работу приложения. Во время работы приложения вызовите панель чудо-кнопок и нажмите на чудо-кнопку Параметры.
- 35) Завершите работу приложения.
- 36*)** <http://habrahabr.ru/company/microsoft/blog/206702/> - SearchBox и многое другое
- 36) Закройте Visual Studio 2013.

Итоги

В этой лабораторной работе вы создали приложение Магазина Windows, зарегистрировали и изучили контракты Поиск и Настройка, запустили приложение и изучили его работу.