

# Лекция 6

## Создание каркаса приложения

Разработка программных модулей

Тимашева Эльза Ринадовна

1. Создать на гитхабе пустой проект под названием FamiliyaTours.
2. Клонировать его через visual studio (см.лек.3).
3. Создать в нем новое решение (проект) под тем же названием FamiliyaTours.
4. Проработать видеоурок по ссылке (**либо просто пройти по слайдам презентации**):

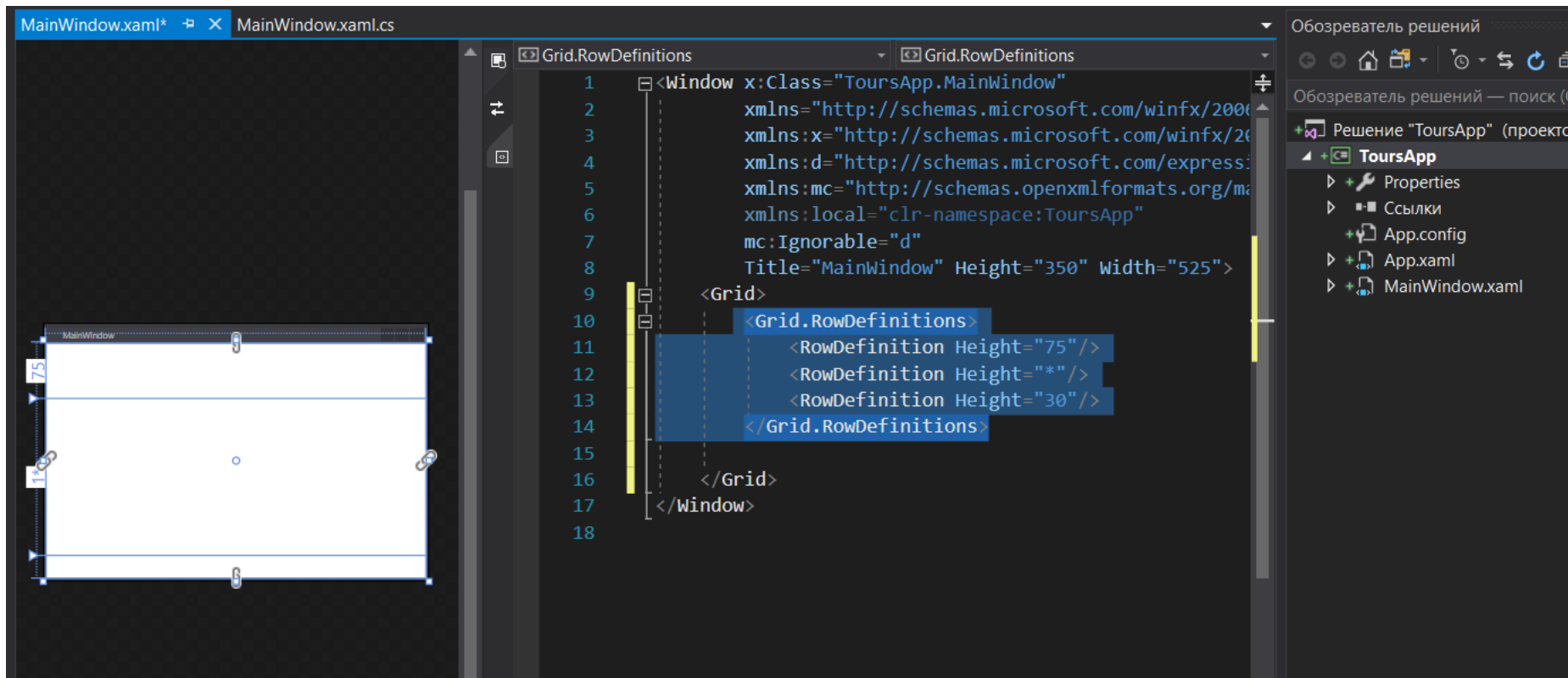
<https://nationalteam.worldskills.ru/skills/sozdanie-karkasa-prilozheniya-sozdanie-i-ispolzovanie-stiley/>

4. Результаты работы загрузить на гитхаб.
5. Ссылку для проверки скинуть в ватсап в личку.



1. Логотип. Сохранить эту картинку в файл logo.png где-нибудь на компе.  
*(Потом добавим этот рисунок в папку проекта)*

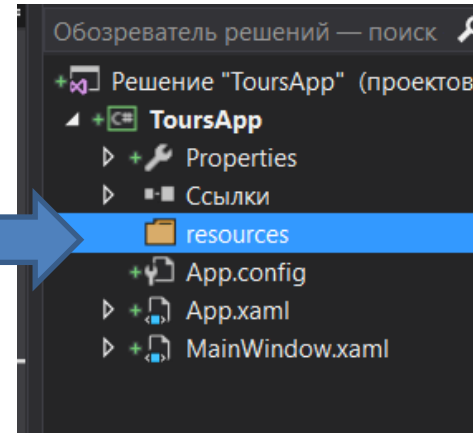
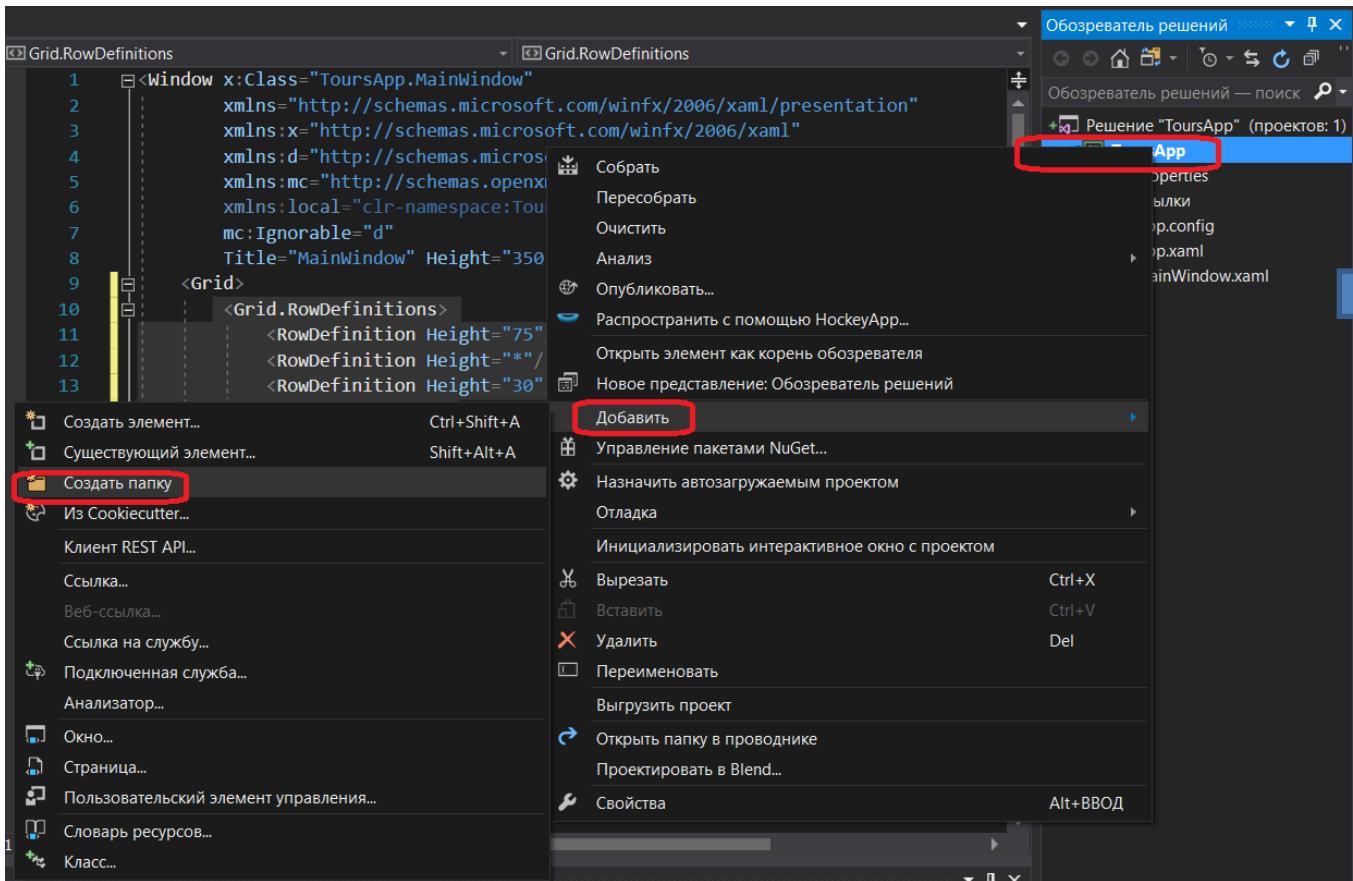
## 2. Выполнить разметку грида:



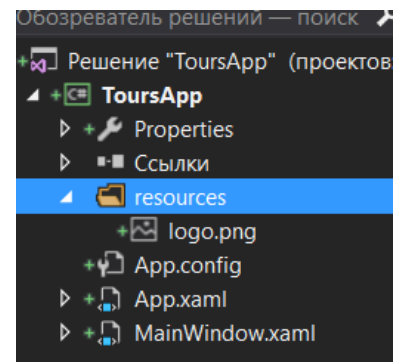
The image shows the Visual Studio IDE with the XAML file `MainWindow.xaml` open. The code defines a `Grid` with three rows. The first row has a height of 75, the second is set to `*`, and the third is 30. The `Grid` is contained within a `Window` titled "MainWindow" with dimensions 350x525. The Solution Explorer on the right shows the project "ToursApp" with files: Properties, Ссылки, App.config, App.xaml, and MainWindow.xaml.

```
1 <Window x:Class="ToursApp.MainWindow"
2       xmlns="http://schemas.microsoft.com/winfx/2006/05/xaml"
3       xmlns:x="http://schemas.microsoft.com/winfx/2006/05/xaml"
4       xmlns:d="http://schemas.microsoft.com/expression/2010/declarative"
5       xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6       xmlns:local="clr-namespace:ToursApp"
7       mc:Ignorable="d"
8       Title="MainWindow" Height="350" Width="525">
9     <Grid>
10      <Grid.RowDefinitions>
11        <RowDefinition Height="75"/>
12        <RowDefinition Height="*/>
13        <RowDefinition Height="30"/>
14      </Grid.RowDefinitions>
15    </Grid>
16  </Window>
```

### 3. Добавим папку resources в проект

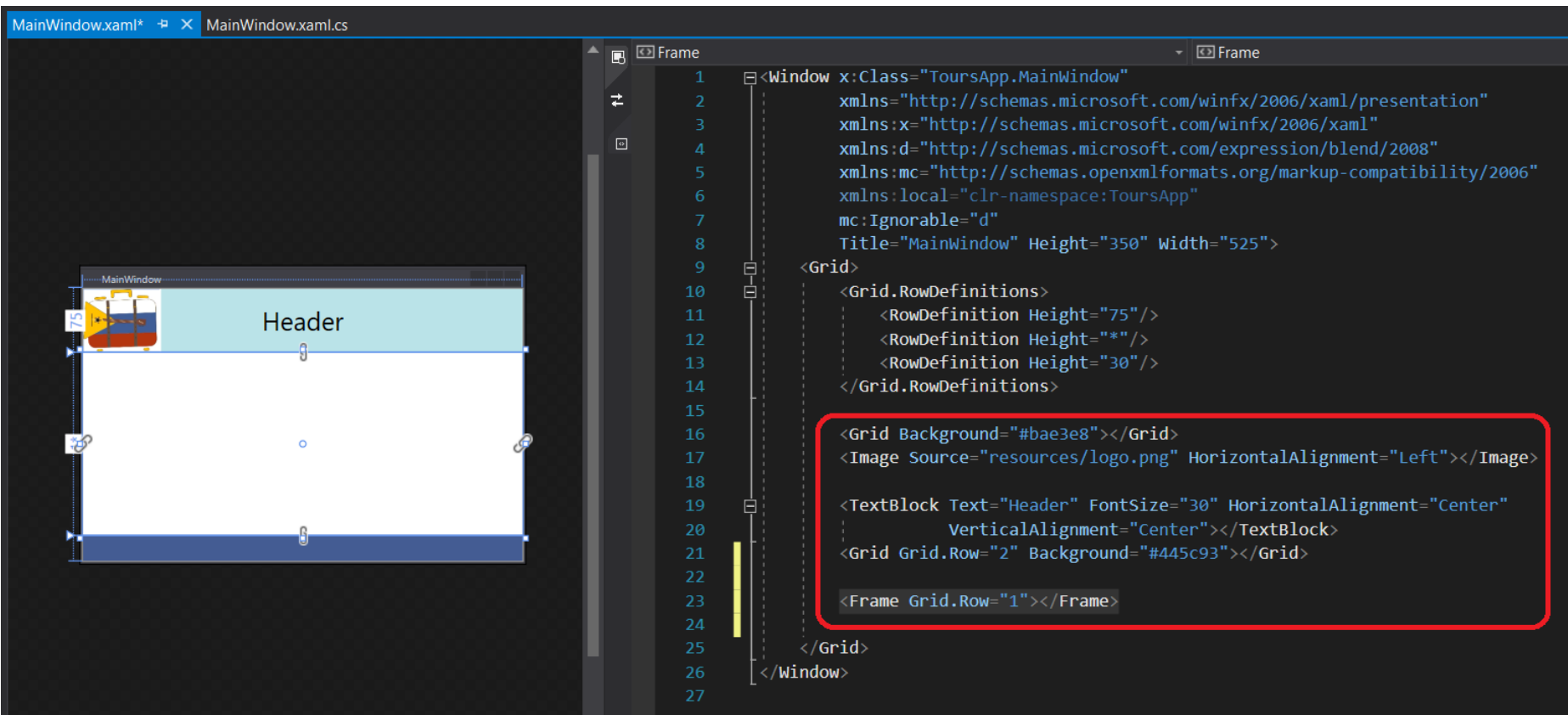


### 4. Перетащить в нее мышкой файл logo.png

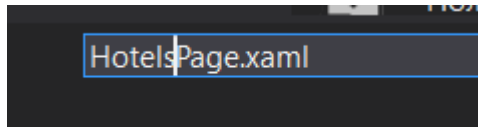
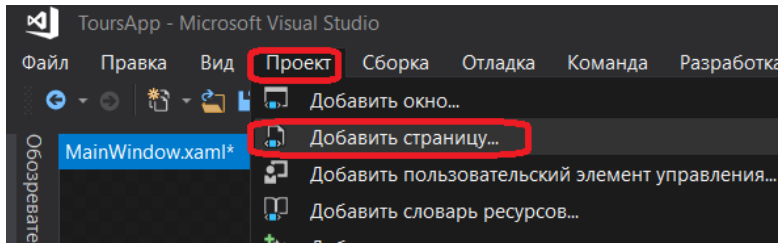


Доделаем разметку главной страницы как показано у меня.

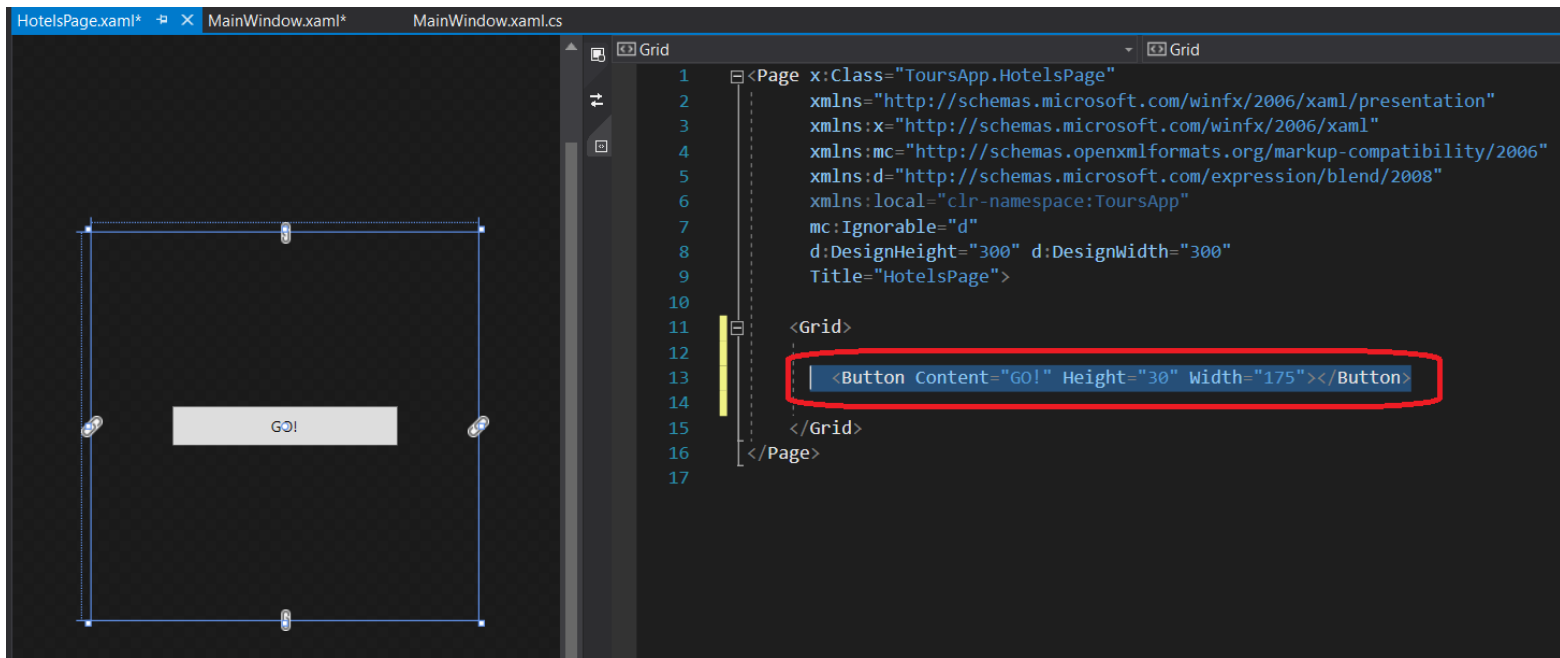
Основной элемент для работы – это элемент Frame, где будут собраны страницы приложения. Для него параметр Grid.Row = “1” установим.



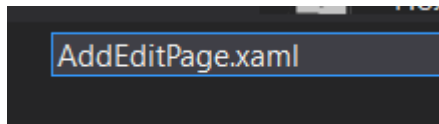
Далее создадим 2 страницы: первая (HotelsPage) будет отображаться при первом запуске приложения:



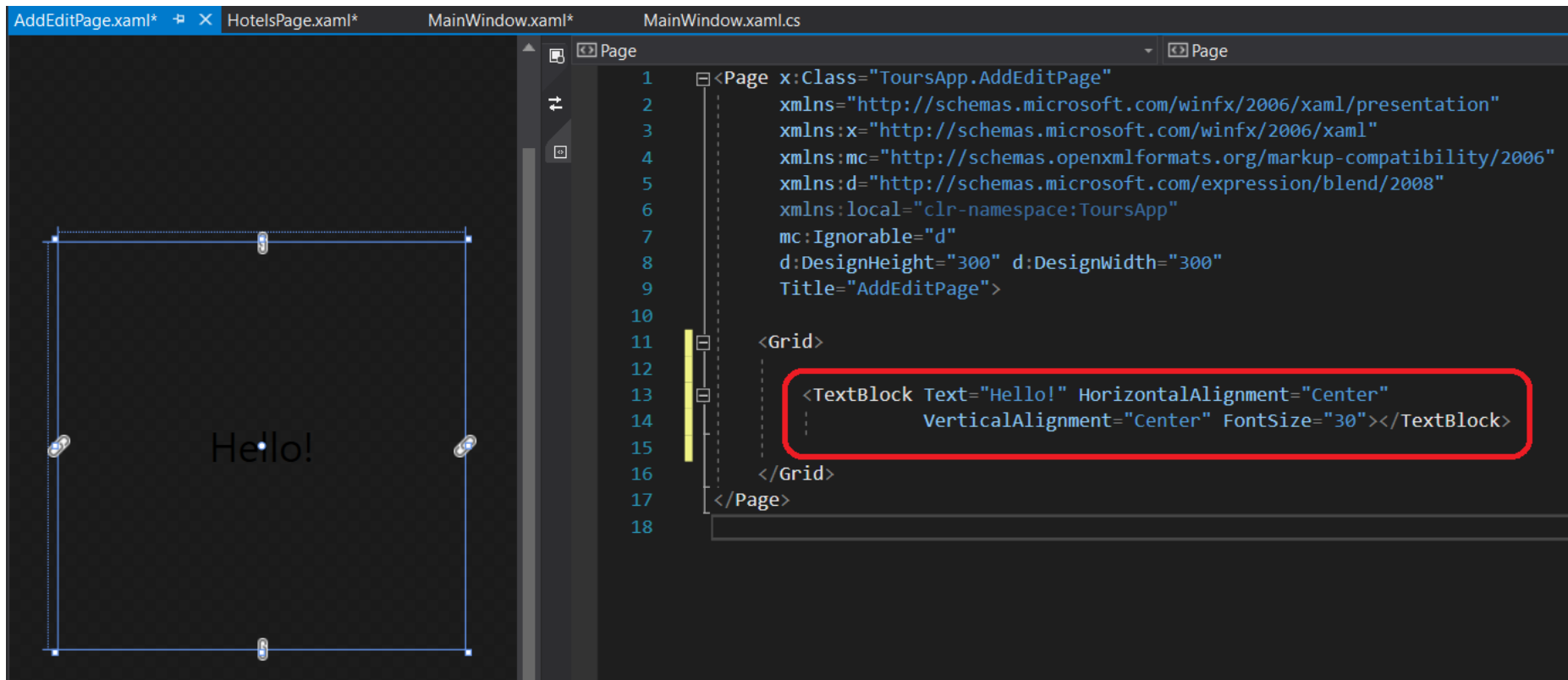
Размещаем на этой странице одну кнопку:



Вторая страница (AddEditPage) для тестирования навигации между ними.



Размещаем в ней TextBlock:





Далее продолжим работу на основной форме (MainWindow.xaml).

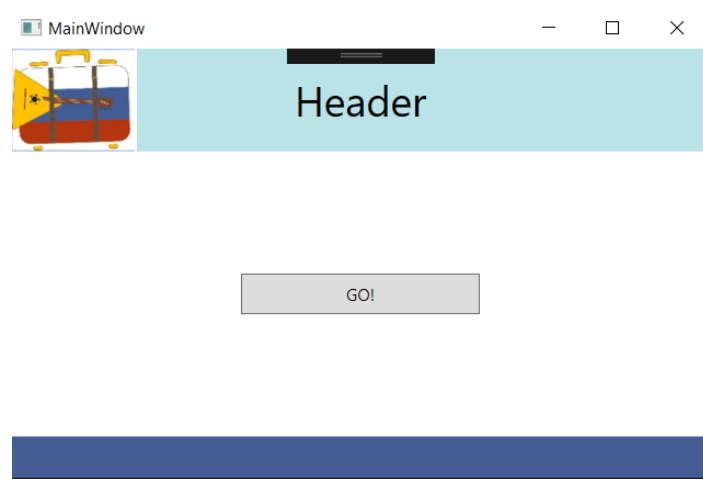
Зададим имя (MainFrame) фрейму для того, чтобы программировать действия для него.

```
<Frame Grid.Row="1" Name="MainFrame"></Frame>
```

При запуске приложения во фрейме будет отображаться первая страница (HotelsPage), для этого пропишем в коде строку:

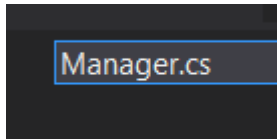
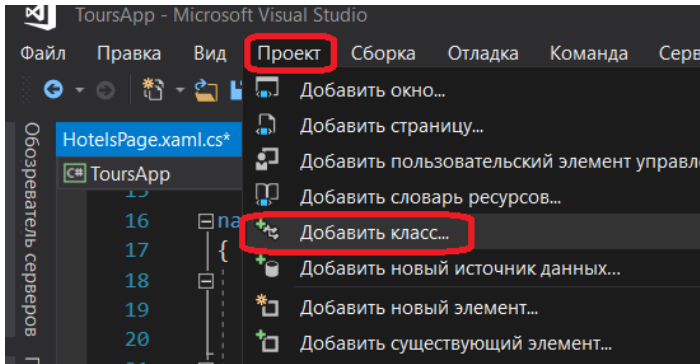
```
16 namespace ToursApp
17 {
18     /// <summary>
19     /// Логика взаимодействия для MainWindow.xaml
20     /// </summary>
21     public partial class MainWindow : Window
22     {
23         public MainWindow()
24         {
25             InitializeComponent();
26             MainFrame.Navigate(new HotelsPage());
27         }
28     }
29 }
30 }
```

Запустить, проверить (появится кнопка).

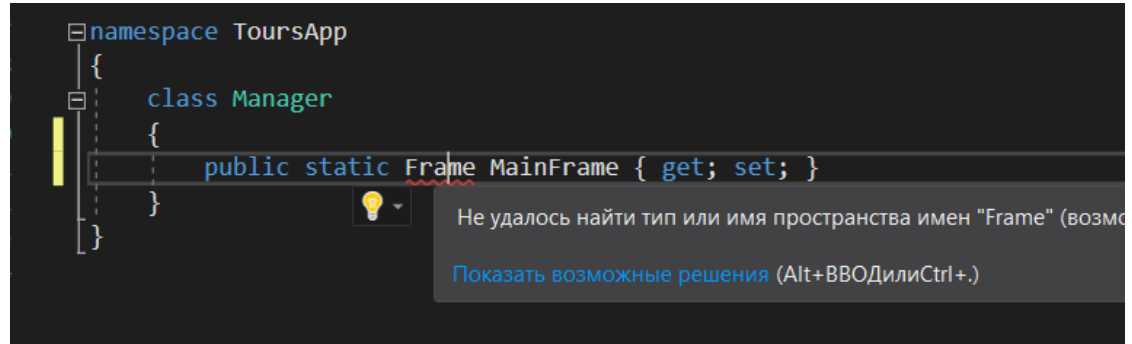


Для реализации перехода между страницами, нужно взаимодействовать с фреймом на главной форме, передавая ему для отображения одну из двух страниц, а также со страницы HotelsPage с кнопкой реализовать переход на вторую страницу и отображение ее во фрейме.

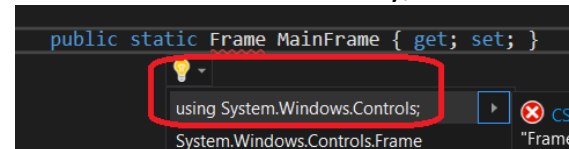
НО, эти две страницы «не знают» про фрейм, и для того, чтобы им можно было взаимодействовать с ним, опишем класс (Manager), в котором будет глобальный параметр фрейм (название такое же, как на главной форме - MainFrame).



Прописать в классе строку:



При этом происходит выделение слова Frame, т.к. нужно подключить соотв.библиотеку, нажав на лампочку и выбрав первый предложенный вариант решения проблемы:

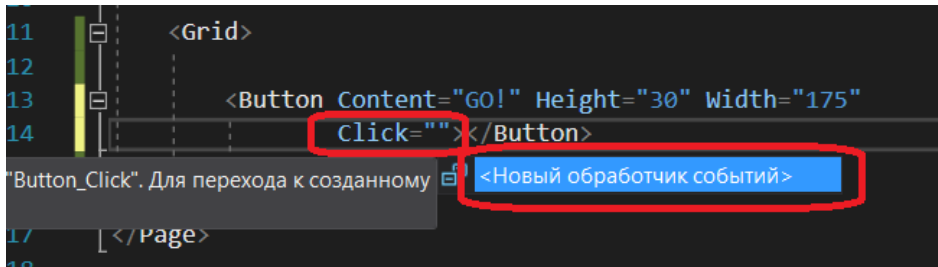


Далее в коде главной формы (MainWindow.xaml.cs) прописать строку присваивания значения фрейму из класса фрейма с главной формы.

```
16 namespace ToursApp
17 {
18     /// <summary>
19     /// Логика взаимодействия для MainWindow.xaml
20     /// </summary>
21     public partial class MainWindow : Window
22     {
23         public MainWindow()
24         {
25             InitializeComponent();
26
27             MainFrame.Navigate(new HotelsPage());
28
29             Manager.MainFrame = MainFrame;
30         }
31     }
32 }
33
34
```

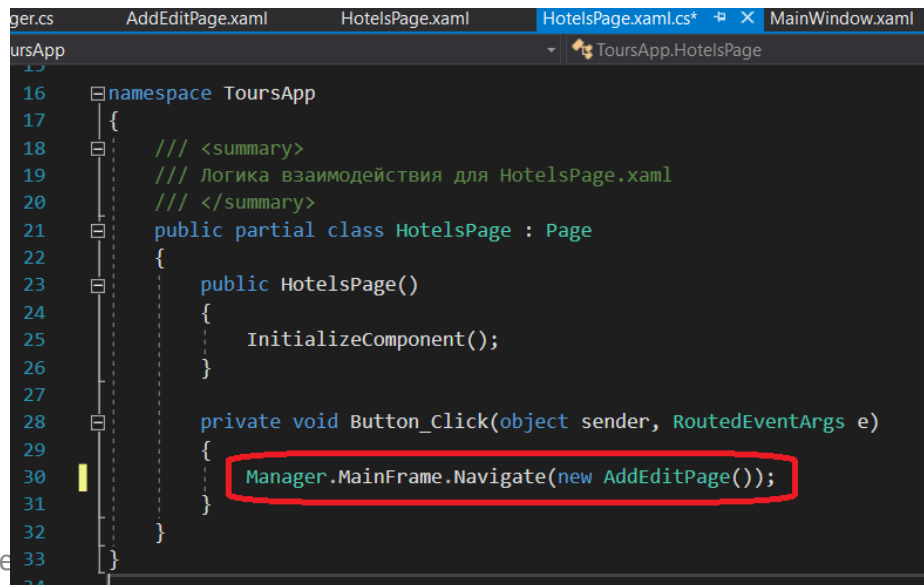
Далее перейдем на страницу HotelsPage.

Для взаимодействия с кнопкой будем использовать событие Click(). С помощью него реализуем переход с этой страницы на вторую (AddEditPage). Для создания события пропишем соответствующее значение в верстке:



```
11 <Grid>
12
13 <Button Content="GO!" Height="30" Width="175"
14 Click="\"\"></Button>
```

Далее нажмем F12 и попадем в окно обработки нажатия на кнопку. Пропишем код навигации для фрейма:



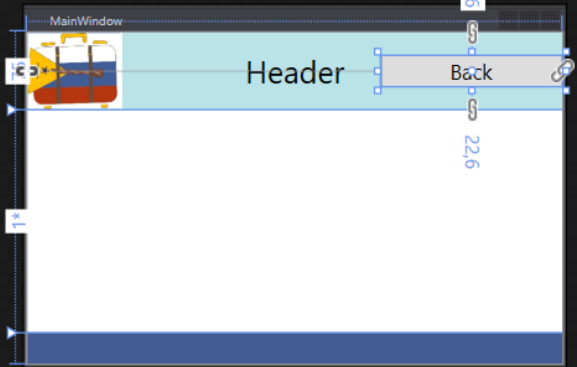
```
ger.cs AddEditPage.xaml HotelsPage.xaml HotelsPage.xaml.cs* x MainWindow.xaml
ToursApp
16 namespace ToursApp
17 {
18     /// <summary>
19     /// Логика взаимодействия для HotelsPage.xaml
20     /// </summary>
21     public partial class HotelsPage : Page
22     {
23         public HotelsPage()
24         {
25             InitializeComponent();
26         }
27
28         private void Button_Click(object sender, RoutedEventArgs e)
29         {
30             Manager.MainFrame.Navigate(new AddEditPage());
31         }
32     }
33 }
34
```

Запустить, проверить.

Добавим кнопку «Назад» на главной форме и реализуем переходы по ее нажатию (свойство Click).

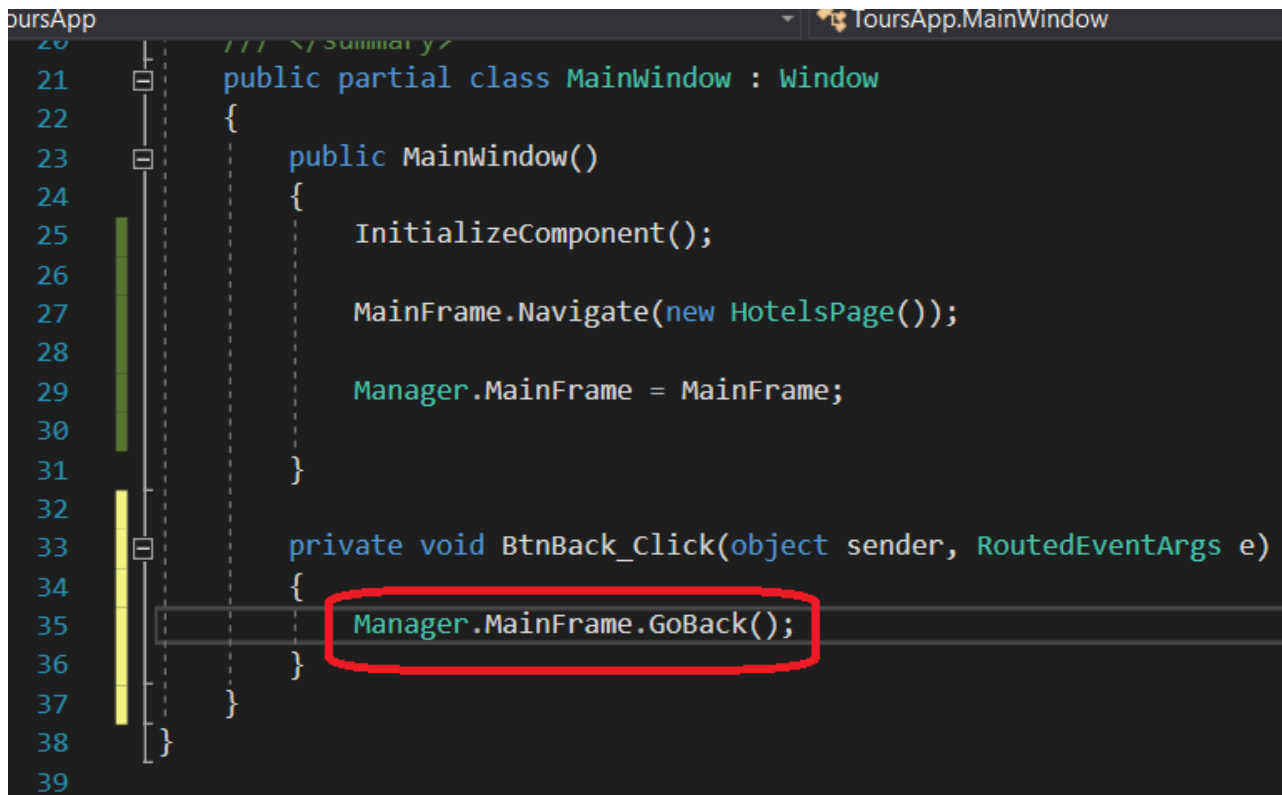
Manager.cs AddEditPage.xaml HotelsPage.xaml HotelsPage.xaml.cs MainWindow.xaml\* X MainWindow.xaml.cs\*

Window



```
1 <Window x:Class="ToursApp.MainWindow"
2       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4       xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5       xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6       xmlns:local="clr-namespace:ToursApp"
7       mc:Ignorable="d"
8       Title="MainWindow" Height="350" Width="525">
9     <Grid>
10      <Grid.RowDefinitions>
11        <RowDefinition Height="75"/>
12        <RowDefinition Height="*/>
13        <RowDefinition Height="30"/>
14      </Grid.RowDefinitions>
15
16      <Grid Background="#bae3e8"></Grid>
17      <Image Source="resources/logo.png" HorizontalAlignment="Left"></Image>
18
19      <TextBlock Text="Header" FontSize="30" HorizontalAlignment="Center"
20                VerticalAlignment="Center"></TextBlock>
21      <Grid Grid.Row="2" Background="#445c93"></Grid>
22
23      <Frame Grid.Row="1" Name="MainFrame"></Frame>
24
25      <Button Content="Back" Name="BtnBack" Width="175" Height="30" FontSize="20"
26            HorizontalAlignment="Right" Click="BtnBack_Click"></Button>
27    </Grid>
28  </Window>
```

Далее нажимаем F12 и попадаем в окно обработки нажатия на кнопку. Используем следующую логику: обращаемся к классу Manager, в нем к фрейму MainFrame, для которого и вызываем метод GoBack().



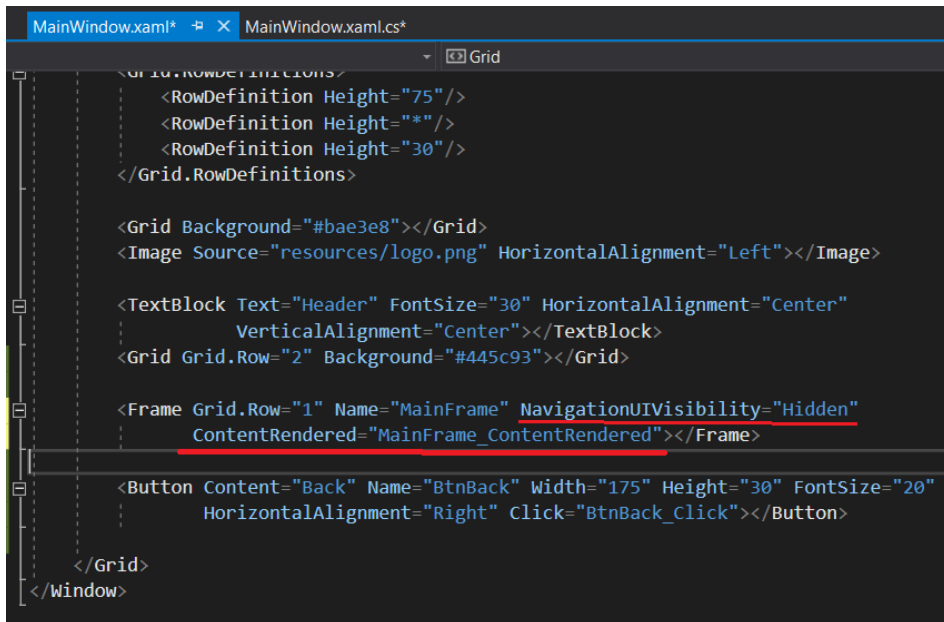
```
20 // Summary
21 public partial class MainWindow : Window
22 {
23     public MainWindow()
24     {
25         InitializeComponent();
26
27         MainFrame.Navigate(new HotelsPage());
28
29         Manager.MainFrame = MainFrame;
30     }
31
32     private void BtnBack_Click(object sender, RoutedEventArgs e)
33     {
34         Manager.MainFrame.GoBack();
35     }
36 }
37
38
39
```

Запустить, проверить.

Но она не нужна когда нельзя идти назад.

Чтобы скрыть ее, воспользуемся событием фрейма ContentRendered (Новый обработчик событий) и пропишем код для него.

Заодно скроем стандартное навигационное меню через свойство фрейма — *NavigationUIVisibility*.



```
<Grid.RowDefinitions>
  <RowDefinition Height="75"/>
  <RowDefinition Height="*/>
  <RowDefinition Height="30"/>
</Grid.RowDefinitions>

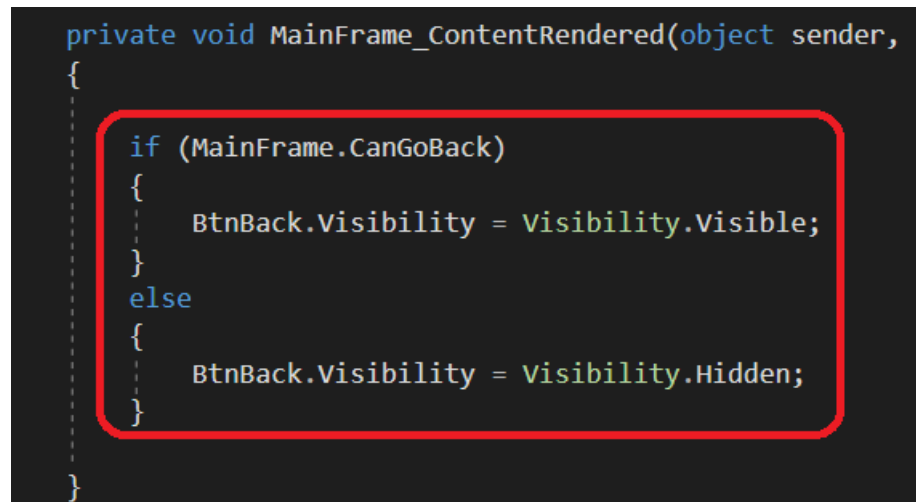
<Grid Background="#bae3e8"></Grid>
<Image Source="resources/logo.png" HorizontalAlignment="Left"></Image>

<TextBlock Text="Header" FontSize="30" HorizontalAlignment="Center"
  VerticalAlignment="Center"></TextBlock>
<Grid Grid.Row="2" Background="#445c93"></Grid>

<Frame Grid.Row="1" Name="MainFrame" NavigationUIVisibility="Hidden"
  ContentRendered="MainFrame_ContentRendered"></Frame>

<Button Content="Back" Name="BtnBack" Width="175" Height="30" FontSize="20"
  HorizontalAlignment="Right" Click="BtnBack_Click"></Button>

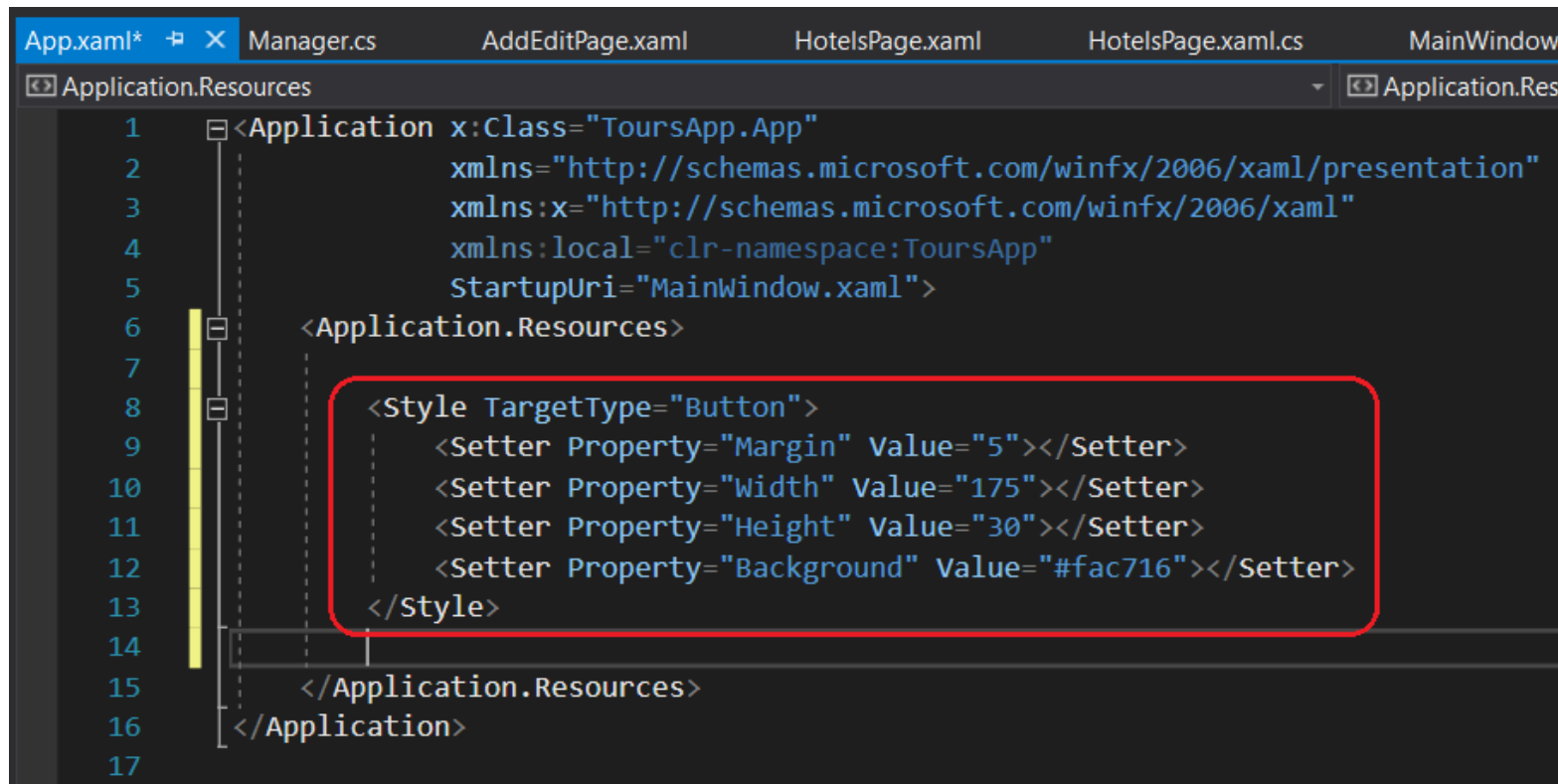
</Grid>
</Window>
```



```
private void MainFrame_ContentRendered(object sender, EventArgs e)
{
    if (MainFrame.CanGoBack)
    {
        BtnBack.Visibility = Visibility.Visible;
    }
    else
    {
        BtnBack.Visibility = Visibility.Hidden;
    }
}
```

Запустить, проверить.

Для большинства созданных элементов использовался похожий набор свойств: ширина, высота, размер шрифта, отступы и др. Чтобы применять определенные наборы свойств для элементов, WPF предлагает использование глобальных стилей в проекте. Чтобы создавать их в проекте, есть файл App.xaml. Используем тег Style и свойство TargetType, чтобы указать, для каких элементов предназначен данный стиль.



```
App.xaml*  Manager.cs  AddEditPage.xaml  HotelsPage.xaml  HotelsPage.xaml.cs  MainWindow.xaml.cs
Application.Resources
1  <Application x:Class="ToursApp.App"
2      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4      xmlns:local="clr-namespace:ToursApp"
5      StartupUri="MainWindow.xaml">
6      <Application.Resources>
7
8          <Style TargetType="Button">
9              <Setter Property="Margin" Value="5"></Setter>
10             <Setter Property="Width" Value="175"></Setter>
11             <Setter Property="Height" Value="30"></Setter>
12             <Setter Property="Background" Value="#fac716"></Setter>
13         </Style>
14
15     </Application.Resources>
16 </Application>
17
```

И теперь  
удалить  
описанные  
ранее  
свойства  
элементов,  
для которых  
определили  
стиль.

Запустить,  
проверить.