

# Настольное приложение «Салон красоты» (часть 2)

# Создание страницы MainPage

# Верстка главной страницы

```
<Grid>
  <ListView x:Name="CategoriesListView" VerticalAlignment="Center" >
    <ListView.ItemTemplate>
      <DataTemplate >
        <Grid >
          <Grid.RowDefinitions>
            <RowDefinition/>
            <RowDefinition Height="auto"/>
          </Grid.RowDefinitions>
          <Image Source="{Binding CategoryImage}" Height="200"/>
          <TextBlock Text="{Binding CategoryTitle}"
                    Style="{StaticResource CategoryTextBlock}" />
        </Grid>
      </DataTemplate>
    </ListView.ItemTemplate>
    <!--Вывод блоков в ряд-->
    <ListView.ItemsPanel>
      <ItemsPanelTemplate>
        <WrapPanel />
      </ItemsPanelTemplate>
    </ListView.ItemsPanel>
  </ListView>
</Grid>
```

# Стилевое оформление

```
<Style TargetType="ListView" >
    <Setter Property="Background" Value="{StaticResource MainColor}"/>
</Style>

<Style TargetType="TextBlock" x:Key="CategoryTextBlock">
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="{StaticResource CaptionFont}"/>
    <Setter Property="Border.CornerRadius" Value="10"/>
    <Setter Property="HorizontalAlignment" Value="Center"/>
    <Setter Property="Foreground" Value="{StaticResource ActiveColor}"/>
    <Setter Property="TextWrapping" Value="Wrap"/>
    <Setter Property="Width" Value="200"/>
    <Setter Property="VerticalAlignment" Value="Bottom"/>
</Style>
```

# Создаем класс ServiceCategories (в папке Models)

```
public class ServiceCategories
{
    public int CategoryId { get; set; }
    public string CategoryTitle { get; set; }
    public byte[] CategoryImage { get; set; }
}
```

# Вывод списка категорий (метод GetServiceCategories)

```
public static List<ServiceCategories> GetServiceCategories()
{
    try
    {
        using (HttpClient client = new HttpClient())
        {
            string url = $"{Manager.RootUrl}ServiceCategories";
            Console.WriteLine(url);
            HttpResponseMessage response = client.GetAsync(url).Result;
            var content = response.Content.ReadAsStringAsync();
            var answer =
                JsonConvert.DeserializeObject<List<ServiceCategories>>(content.Result);
            return answer;
        }
    }
    catch
    {
        throw new Exception("Ошибка подключения к серверу");
    }
}
```

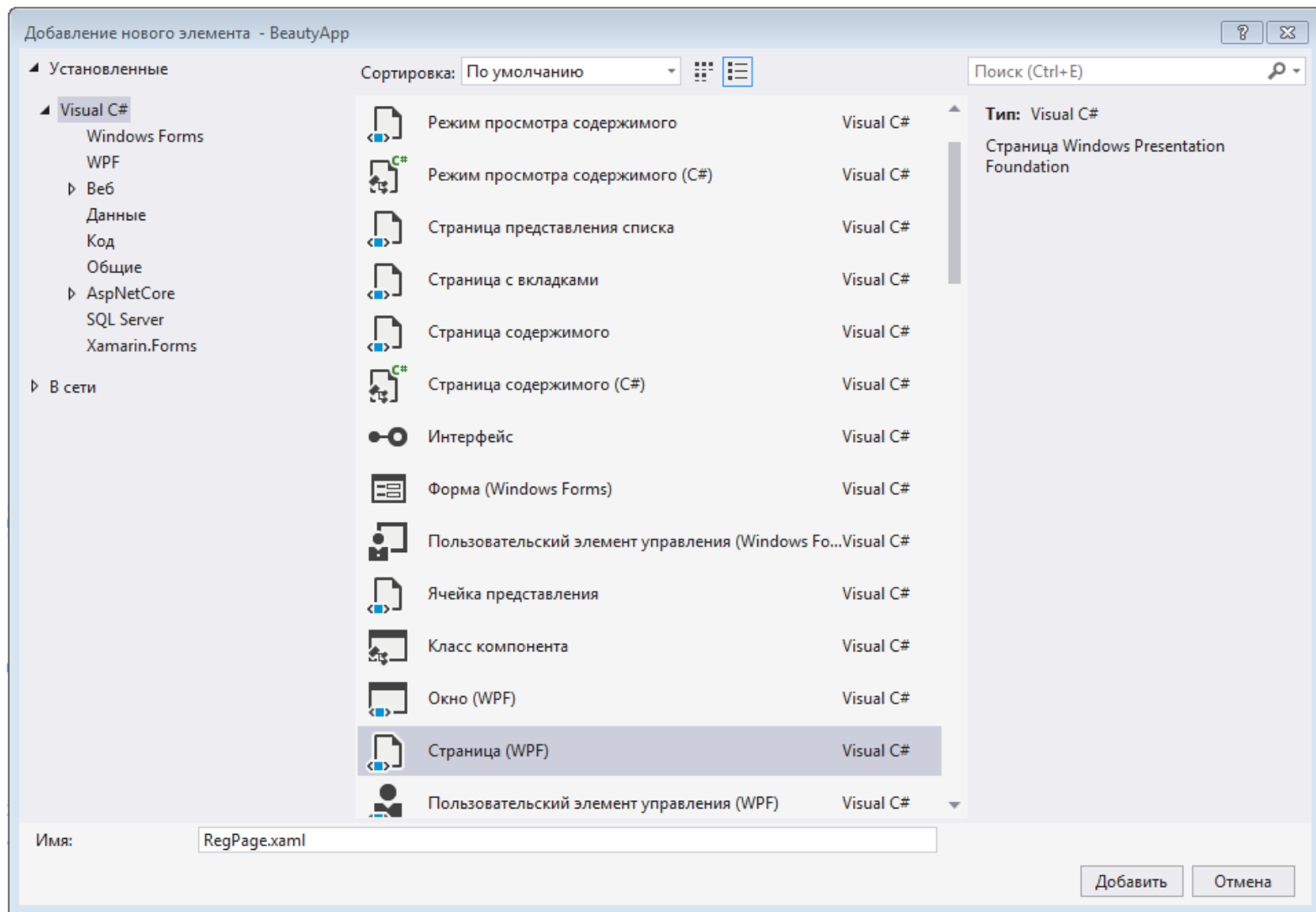
# Передаем в ListView коллекцию

```
public partial class MainPage : Page
{
    List<ServiceCategoryes> arrayCategory ;
    public MainPage()
    {
        InitializeComponent();
        arrayCategory =ServiceCategoryesController.GetServiceCategoryes();
        CategoriesListView.ItemsSource = arrayCategory ;
    }
}
```

Регистрация



# Создаем страницу RegPage



# Верстка страницы RegPage

```
<GroupBox>
  <StackPanel>
    <UniformGrid Columns="2">
      <StackPanel>
        <TextBlock Text="Личные данные" Style="{StaticResource CategoryTextBlock}" />
        <UniformGrid Columns="2">
          <TextBlock Text="Фамилия" HorizontalAlignment="Right"/>
          <TextBox x:Name="SurnameTextBox"/>
        </UniformGrid>
        <UniformGrid Columns="2">
          <TextBlock Text="Имя" HorizontalAlignment="Right"/>
          <TextBox x:Name="NameTextBox"/>
        </UniformGrid>
        <UniformGrid Columns="2">
          <TextBlock Text="Отчество" HorizontalAlignment="Right"/>
          <TextBox x:Name="PatronymicTextBox"/>
        </UniformGrid>
        <UniformGrid Columns="2">
          <TextBlock Text="Пол" HorizontalAlignment="Right"/>
          <ComboBox x:Name="GenderComboBox"/>
        </UniformGrid>
        <UniformGrid Columns="2">
          <TextBlock Text="Дата рождения" HorizontalAlignment="Right"/>
          <DatePicker x:Name="BirthdatePicker"/>
        </UniformGrid>
        <UniformGrid Columns="2">
          <TextBlock Text="Почта" HorizontalAlignment="Right"/>
          <TextBox x:Name="EmailTextBox"/>
        </UniformGrid>
        <UniformGrid Columns="2">
          <TextBlock Text="Телефон" HorizontalAlignment="Right"/>
          <TextBox x:Name="PhoneTextBox"/>
        </UniformGrid>
        <StackPanel HorizontalAlignment="Center">
          <Button Content="Зарегистрировать" />
        </StackPanel>
      </StackPanel>
    </StackPanel>
    <StackPanel>
      <StackPanel>
        <TextBlock Text="Авторизационные данные" Style="{StaticResource CategoryTextBlock}" />
        <UniformGrid Columns="2">
          <TextBlock Text="Логин" HorizontalAlignment="Right"/>
          <TextBox x:Name="LoginTextBox"/>
        </UniformGrid>
        <UniformGrid Columns="2">
          <TextBlock Text="Пароль" HorizontalAlignment="Right"/>
          <TextBox x:Name="PasswordTextBox"/>
        </UniformGrid>
        <UniformGrid Columns="2">
          <TextBlock Text="Повторите пароль" HorizontalAlignment="Right"/>
          <TextBox x:Name="ReturnPasswordTextBox"/>
        </UniformGrid>
      </StackPanel>
      <StackPanel>
        <TextBlock Text="Введите символы с картинки:" Style="{StaticResource CategoryTextBlock}" />
        <UniformGrid Columns="2">
          <StackPanel x:Name="CaptchaStackPanel" Height="50" HorizontalAlignment="Right" Margin="0 0 20 0" Orientation="Horizontal"/>
          <StackPanel>
            <TextBox x:Name="SymbolsTextBox" Height="30"/>
            <Button Content="Обновить Captcha" x:Name="NewCaptcha" Click="NewCaptchaClick"/>
          </StackPanel>
        </UniformGrid>
      </StackPanel>
    </StackPanel>
  </UniformGrid>
</StackPanel>
</GroupBox>
```

# Создание класса Clients

```
public class Clients
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Patronymic { get; set; }
    public DateTime? Birthday { get; set; }
    public DateTime? RegistrationDate { get; set; } = DateTime.Now;
    public string Email { get; set; }
    public string Phone { get; set; }
    public string GenderCode { get; set; }
    public string PhotoPath { get; set; }
    public string Login { get; set; }
}
```

# Метод RegUser (добавление в Users)

```
public static bool RegUser(Users user)
{
    try
    {
        string jsonStr = JsonConvert.SerializeObject(user);
        var buffer = System.Text.Encoding.UTF8.GetBytes(jsonStr);
        var byteContent = new ByteArrayContent(buffer);
        byteContent.Headers.ContentType = new MediaTypeHeaderValue("application/json");
        using (HttpClient client = new HttpClient())
        {
            string query = $"{Manager.RootUrl}Users";
            HttpResponseMessage response = client.PostAsync(query, byteContent).Result;
            return response.IsSuccessStatusCode;
        }
    }
    catch
    {
        throw new Exception("Ошибка подключения к серверу");
    }
}
```

# Метод RegUser (добавление в Users)

```
public static bool RegUser(Users user)
{
    try
    {
        string jsonStr = JsonConvert.SerializeObject(user);
        var buffer = System.Text.Encoding.UTF8.GetBytes(jsonStr);
        var byteContent = new ByteArrayContent(buffer);
        byteContent.Headers.ContentType = new MediaTypeHeaderValue("application/json");
        using (HttpClient client = new HttpClient())
        {
            string query = $"{Manager.RootUrl}Users";
            HttpResponseMessage response = client.PostAsync(query, byteContent).Result;
            return response.IsSuccessStatusCode;
        }
    }
    catch
    {
        throw new Exception("Ошибка подключения к серверу");
    }
}
```

# Метод RegClient (добавление в Clients)

```
public static bool RegClient(Clients client)
{
    try
    {
        string jsonStr = JsonConvert.SerializeObject(client);
        var buffer = System.Text.Encoding.UTF8.GetBytes(jsonStr);
        var byteContent = new ByteArrayContent(buffer);
        byteContent.Headers.ContentType = new MediaTypeHeaderValue("application/json");
        using (HttpClient obj = new HttpClient())
        {
            string query = $"{Manager.RootUrl}Clients";
            HttpResponseMessage response = obj.PostAsync(query, byteContent).Result;
            return response.IsSuccessStatusCode;
        }
    }
    catch
    {
        throw new Exception("Ошибка подключения к серверу");
    }
}
```

# Регистрация

```
private void RegButtonClick(object sender, RoutedEventArgs e)
{
    if (PasswordTextBox.Text != String.Empty && LoginTextBox.Text != String.Empty)
    {
        if (ReturnPasswordTextBox.Text == PasswordTextBox.Text)
        {
            Users user = new Users()
            {
                UserLogin = LoginTextBox.Text,
                UserPassword = PasswordTextBox.Text
            };
            Clients cl = new Clients()
            {
                FirstName = SurnameTextBox.Text,
                LastName = NameTextBox.Text,
                Patronymic = PatronymicTextBox.Text,
                GenderCode = "1",
                Birthday = BirthdatePicker.SelectedDate.Value,
                Email = EmailTextBox.Text,
                Phone = PhoneTextBox.Text,
                Login = LoginTextBox.Text
            };
            if (UsersController.RegUser(user) && ClientsController.RegClient(cl))
            {
                MessageBox.Show("Вы успешно зарегистрировались!");
                this.NavigationService.Navigate(new AuthPage());
            }
        }
        else
        {
            MessageBox.Show("Пароли не совпадают!");
        }
    }
    else
    {
        MessageBox.Show("не заполнены поля!");
    }
}
```

Занятие окончено