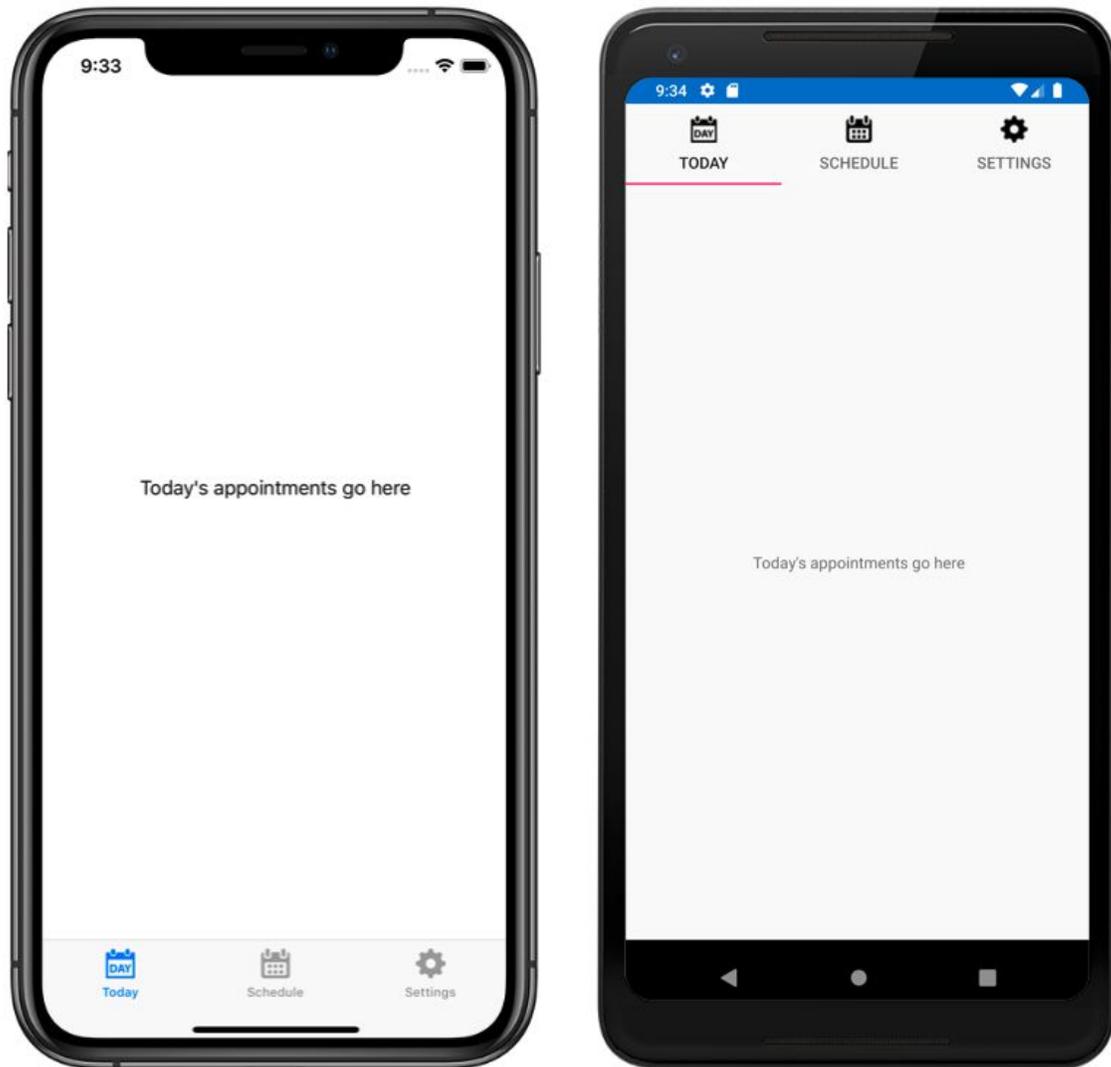


Xamarin Forms. Навигация. TabbedPage.

МДК 01.03 «Разработка мобильных приложений»

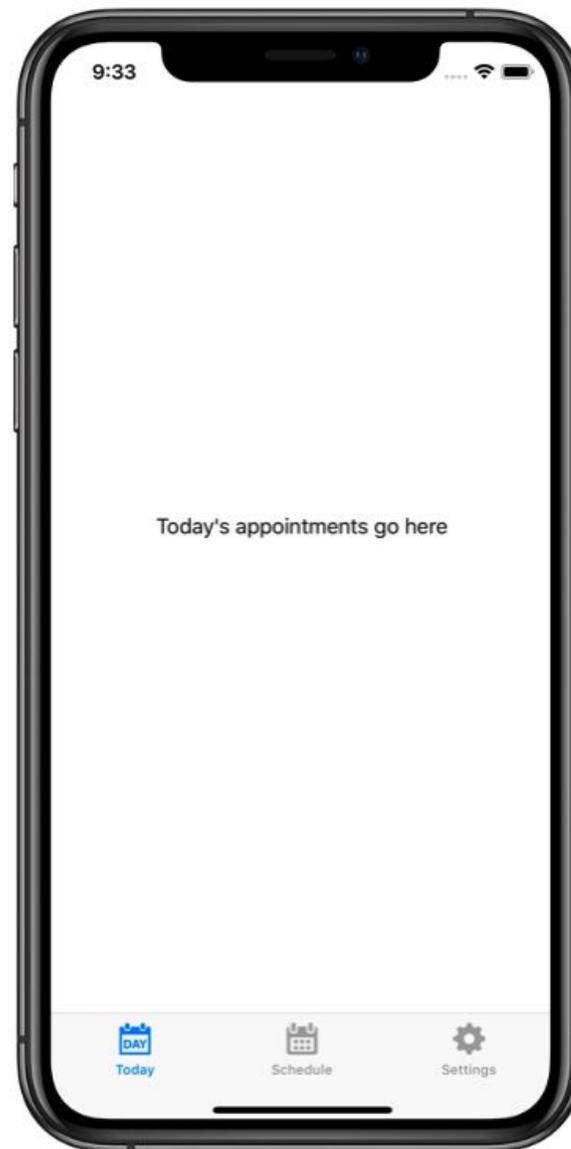
TabPage



Xamarin.Forms **TabPage** состоит из списка вкладок и большой области сведений, где каждая вкладка загружает содержимое в область сведений.

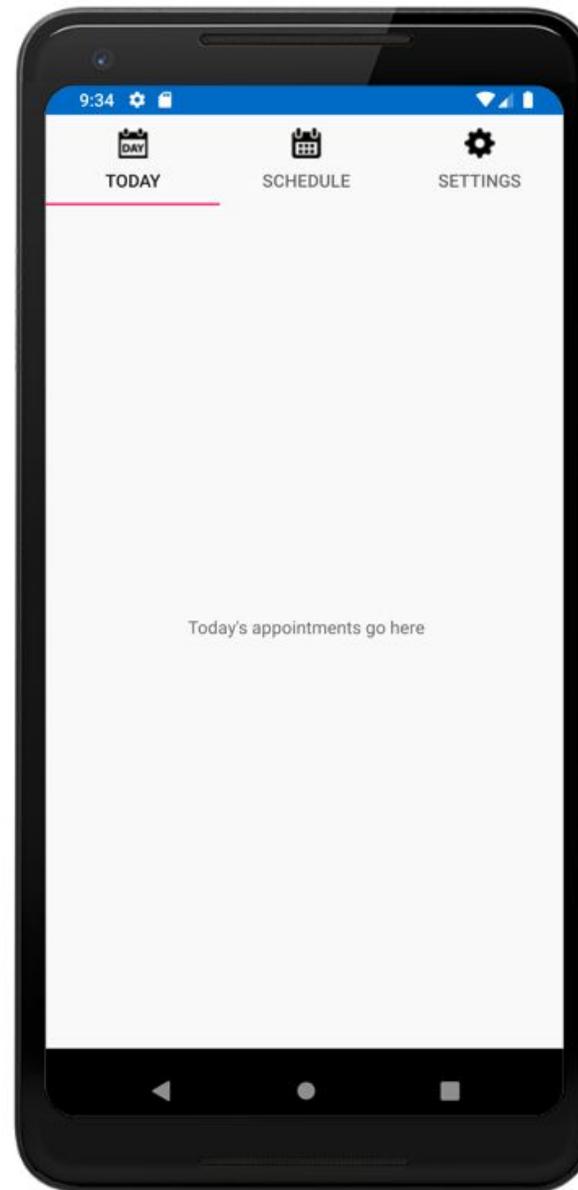
TabbedPage в iOS

В iOS список вкладок отображается в нижней части экрана, а также в области данных сверху. У каждой вкладки есть заголовок и значок, который должен быть PNG-файлом с альфа-каналом. В книжной ориентации значки панели вкладок отображаются над заголовками вкладок. В альбомной ориентации значки и заголовки отображаются рядом. Кроме того, в зависимости от устройства и ориентации может отображаться обычная или компактная панель вкладок. При наличии более пяти вкладок появляется вкладка **Дополнительно**, используемая для доступа к дополнительным вкладкам.



TabPage в Android

В Android список вкладок отображается в верхней части экрана, а также в области данных ниже. У каждой вкладки есть заголовок и значок, который должен быть PNG-файлом с альфа-каналом. Но эти вкладки можно переместить в нижнюю часть экрана в зависимости от конкретной платформы. При наличии более пяти вкладок, список которых отображается внизу экрана, появляется вкладка *Дополнительно*, которую можно использовать для доступа к дополнительным вкладкам.



Создание TabbedPage

Создать **TabbedPage** можно двумя способами:

- ▶ Заполните **TabbedPage** коллекцией дочерних объектов **Page**, например **ContentPage**. Дополнительные сведения см. в разделе **Заполнение TabbedPage** коллекцией **Page**;
- ▶ Назначьте коллекции свойство **ItemsSource**, а затем присвойте **DataTemplate** свойству **ItemTemplate** для возврата страницы объектов в коллекции. Дополнительные сведения см. в разделе **Заполнение TabbedPage** с помощью шаблона.

При обоих подходах **TabbedPage** отображает каждую страницу при выборе пользователем каждой из вкладок.

Важно!

Рекомендуется заполнять **TabPage** только экземплярами **NavigationPage** и **ContentPage**. Это позволит обеспечить согласованность пользовательского интерфейса на всех платформах.

Свойства TabbedPage

- ▶ `BarBackgroundColor` с типом `Color` — цвет фона панели вкладок;
- ▶ `BarTextColor` с типом `Color` — цвет текста на панели вкладок;
- ▶ `SelectedTabColor` с типом `Color` — цвет выбранной вкладки.
- ▶ `UnselectedTabColor` с типом `Color` — цвет вкладки, если она не выбрана.

Все эти свойства поддерживаются объектами `BindableProperty`, то есть к ним можно применить стиль и их можно указывать в *качестве целевых для привязки данных*

Предупреждение

В **TabPage** каждый объект **Page** создается при формировании **TabPage**.

Это может повлечь проблемы с произвольностью, особенно если **TabPage** является корневой страницей приложения.

Тем не менее оболочка **Xamarin.Forms** позволяет по запросу создавать страницы, доступ к которым осуществляется через панель вкладок, в ответ на навигацию.

Заполнение TabbedPage коллекцией Page

- ▶ TabbedPage можно заполнить коллекцией дочерних объектов Page, например ContentPage. Для этого необходимо добавить объекты Page в коллекцию TabbedPage.Children.

```
1 <TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
2           xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
3           xmlns:local="clr-namespace:TabbedPageWithNavigationPage;
4           assembly=TabbedPageWithNavigationPage"
5           x:Class="TabbedPageWithNavigationPage.MainPage">
6     <local:TodayPage />
7     <NavigationPage Title="Schedule" IconImageSource="schedule.png">
8       <x:Arguments>
9         <local:SchedulePage />
10      </x:Arguments>
11    </NavigationPage>
12 </TabbedPage>
```

Заполнение TabbedPage коллекцией Page

Эквивалентный код на C#

```
1 public class MainPageCS : TabbedPage
2 {
3     public MainPageCS ()
4     {
5         NavigationPage navigationPage = new NavigationPage (new SchedulePageCS ());
6         navigationPage.IconImageSource = "schedule.png";
7         navigationPage.Title = "Schedule";
8
9         Children.Add (new TodayPageCS ());
10        Children.Add (navigationPage);
11    }
12 }
```

Перемещение в пределах вкладки

- ▶ Навигацию можно осуществлять в пределах одной вкладки при условии, что объект **ContentPage** упакован в объект **NavigationPage**. Для этого необходимо вызвать метод **PushAsync** для свойства **Navigation** объекта **ContentPage**

```
C# Копировать  
  
await Navigation.PushAsync (new UpcomingAppointmentsPage ());
```

- ▶ Страница, к которой осуществляется переход, указывается в качестве аргумента метода **PushAsync**. В этом примере страница **UpcomingAppointmentsPage** принудительно отправляется в стек навигации, где становится активной

Заполнение TabbedPage с помощью шаблона

TabbedPage можно заполнить страницами, назначив коллекцию данных свойству ItemsSource, а также назначив DataTemplate свойству ItemTemplate, которое записывает данные в шаблон в виде объектов Page.

XAML

```
1 <TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
2           xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
3           xmlns:local="clr-namespace:TabbedPageDemo;assembly=TabbedPageDemo"
4           x:Class="TabbedPageDemo.TabbedPageDemoPage"
5           ItemsSource="{x:Static local:MonkeyDataModel.All}">
6   <TabbedPage.Resources>
7     <ResourceDictionary>
8       <local:NonNullToBooleanConverter x:Key="booleanConverter" />
9     </ResourceDictionary>
10  </TabbedPage.Resources>
11  <TabbedPage.ItemTemplate>
12    <DataTemplate>
13      <ContentPage Title="{Binding Name}" IconImageSource="monkeyicon.png">
14        <StackLayout Padding="5, 25">
15          <Label Text="{Binding Name}" Font="Bold, Large" HorizontalOptions="Center" />
16          <Image Source="{Binding PhotoUrl}" WidthRequest="200" HeightRequest="200" />
17          <StackLayout Padding="50, 10">
18            <StackLayout Orientation="Horizontal">
19              <Label Text="Family:" HorizontalOptions="FillAndExpand" />
20              <Label Text="{Binding Family}" Font="Bold, Medium" />
21            </StackLayout>
22            ...
23          </StackLayout>
24        </StackLayout>
25      </ContentPage>
26    </DataTemplate>
27  </TabbedPage.ItemTemplate>
28 </TabbedPage>
```

```
1 public class TabbedPageDemoPageCS : TabbedPage
```

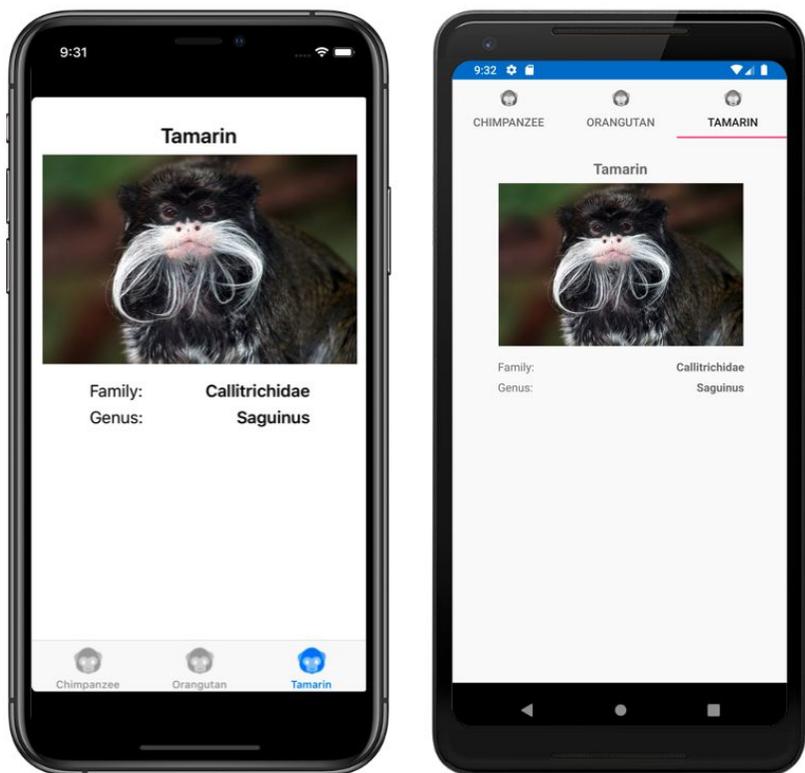
```
2 {  
3     public TabbedPageDemoPageCS ()  
4     {  
5         var booleanConverter = new NonNullToBooleanConverter ();  
6  
7         ItemTemplate = new DataTemplate (() =>  
8         {  
9             var nameLabel = new Label  
10            {  
11                FontSize = Device.GetNamedSize (NamedSize.Large, typeof(Label)),  
12                FontAttributes = FontAttributes.Bold,  
13                HorizontalOptions = LayoutOptions.Center  
14            };  
15            nameLabel.SetBinding (Label.TextProperty, "Name");  
16  
17            var image = new Image { WidthRequest = 200, HeightRequest = 200 };  
18            image.SetBinding (Image.SourceProperty, "PhotoUrl");  
19  
20            var familyLabel = new Label  
21            {  
22                FontSize = Device.GetNamedSize (NamedSize.Medium, typeof(Label)),  
23                FontAttributes = FontAttributes.Bold  
24            };  
25            familyLabel.SetBinding (Label.TextProperty, "Family");  
26            ...  
27  
28            var contentPage = new ContentPage  
29            {  
30                IconImageSource = "monkeyicon.png",  
31                Content = new StackLayout {
```

C#

```
32                Padding = new Thickness (5, 25),  
33                Children =  
34                {  
35                    nameLabel,  
36                    image,  
37                    new StackLayout  
38                    {  
39                        Padding = new Thickness (50, 10),  
40                        Children =  
41                        {  
42                            new StackLayout  
43                            {  
44                                Orientation = StackOrientation.Horizontal,  
45                                Children =  
46                                {  
47                                    new Label { Text = "Family:", HorizontalOptions = LayoutOptions.FillAndExpand },  
48                                    familyLabel  
49                                }  
50                            },  
51                            // ...  
52                        }  
53                    }  
54                }  
55            };  
56            contentPage.SetBinding (TitleProperty, "Name");  
57            return contentPage;  
58        });  
59        ItemsSource = MonkeyDataModel.All;  
60    }  
61 }  
62 }
```

Заполнение TabbedPage с помощью шаблона

- ▶ В этом примере каждая вкладка состоит из объекта `ContentPage`, который использует объекты `Image` и `Label` для вывода данных на вкладку



- ▶ При выборе другой вкладки отображается объект `ContentPage`, представляющий вкладку

Спасибо за внимание!