

# Xamarin Forms

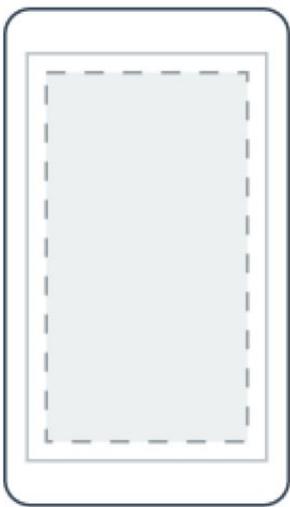
## Навигация

### Иерархическая навигация

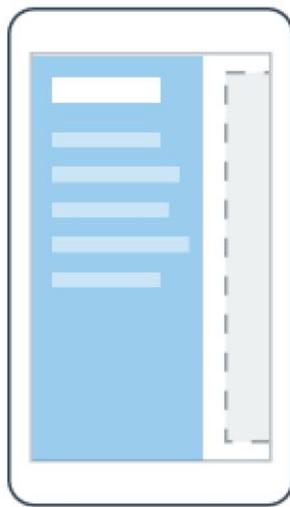
МДК 01.03 «Разработка мобильных приложений»

# Навигация в Xamarin.Forms

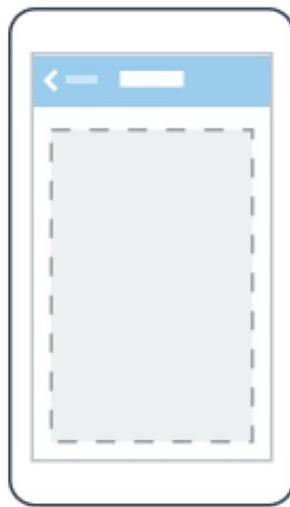
- ▶ Иерархическая навигация
- ▶ TabbedPage
- ▶ CarouselPage
- ▶ FlyoutPage
- ▶ Модальные страницы



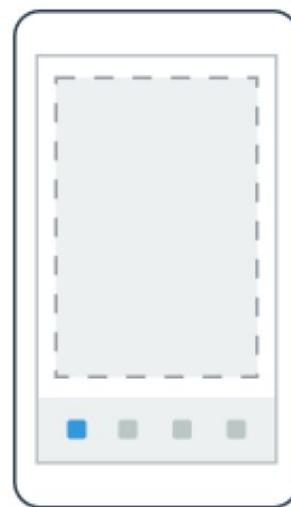
ContentPage



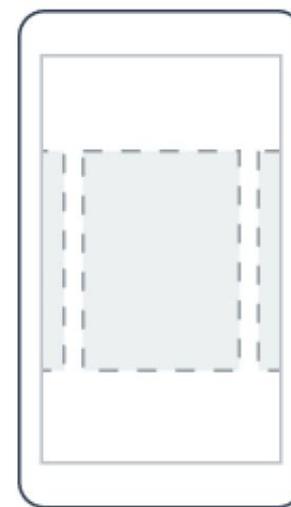
MasterDetailPage



NavigationPage



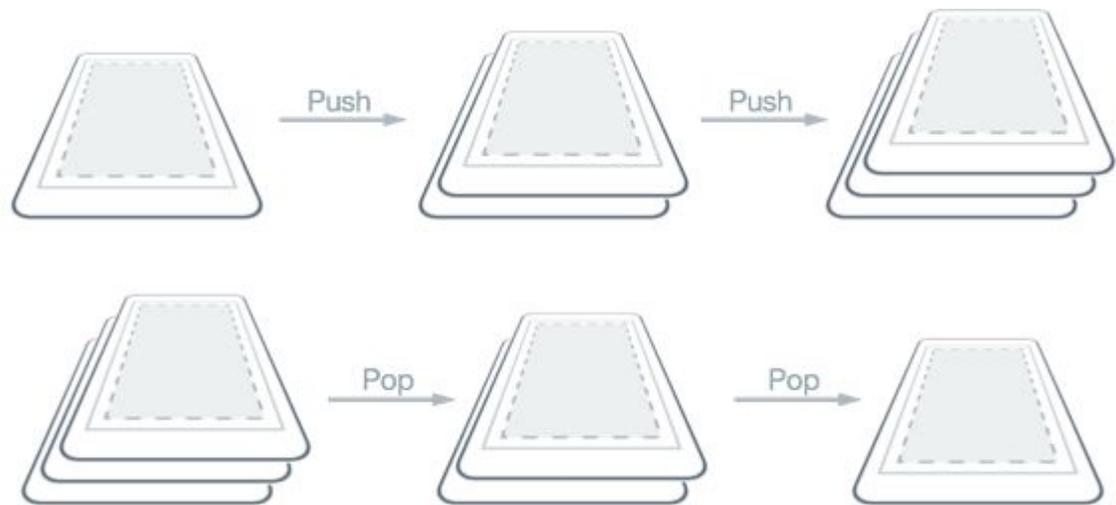
TabbedPage



CarouselPage

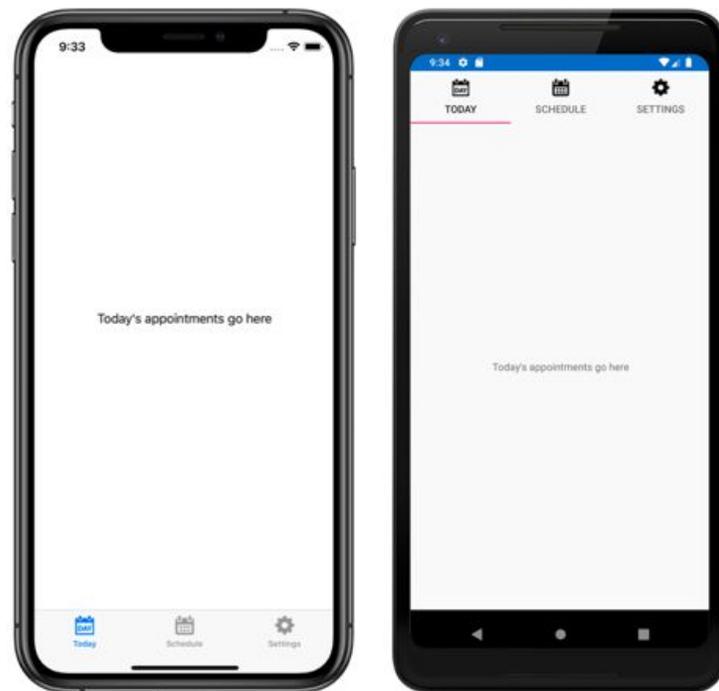
# Иерархическая навигация

Класс `NavigationPage` обеспечивает иерархическую навигацию, при которой пользователь может переходить по страницам вперед и назад по своему желанию. Этот класс реализует навигацию на основе стека объектов `Page` по методу LIFO (последним поступил – первым обслужен).



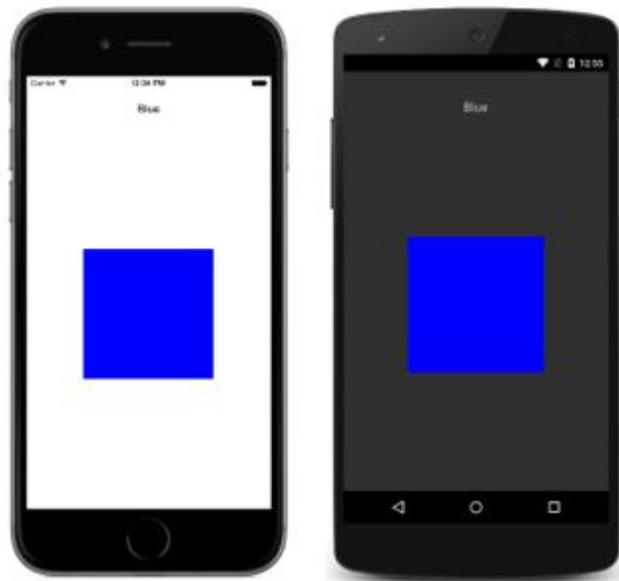
# TabbedPage

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



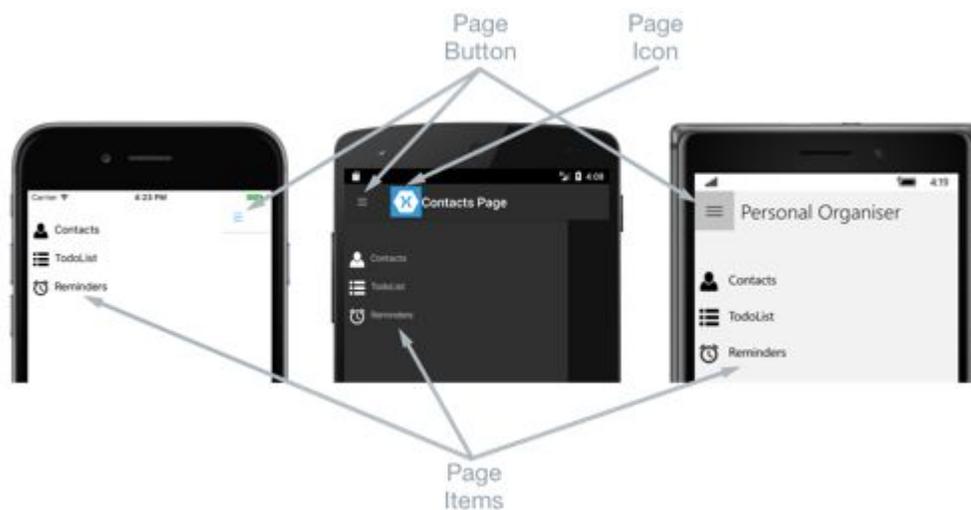
# CarouselPage

Xamarin.Forms CarouselPage – это страница, по которой пользователи могут проводить из стороны в сторону, чтобы переходить по страницам содержимого, например по страницам коллекции.



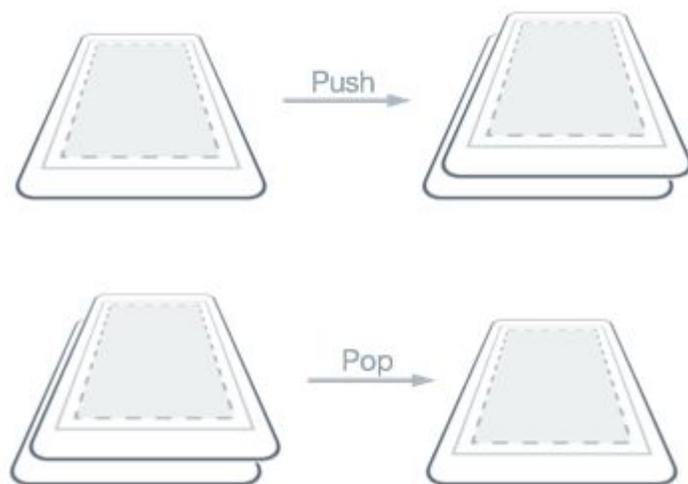
# FlyoutPage

FlyoutPage в Xamarin.Forms представляет собой страницу, которая обеспечивает доступ к двум страницам со связанными данными: всплывающей странице, где представлены элементы, и странице со сведениями об этих элементах.



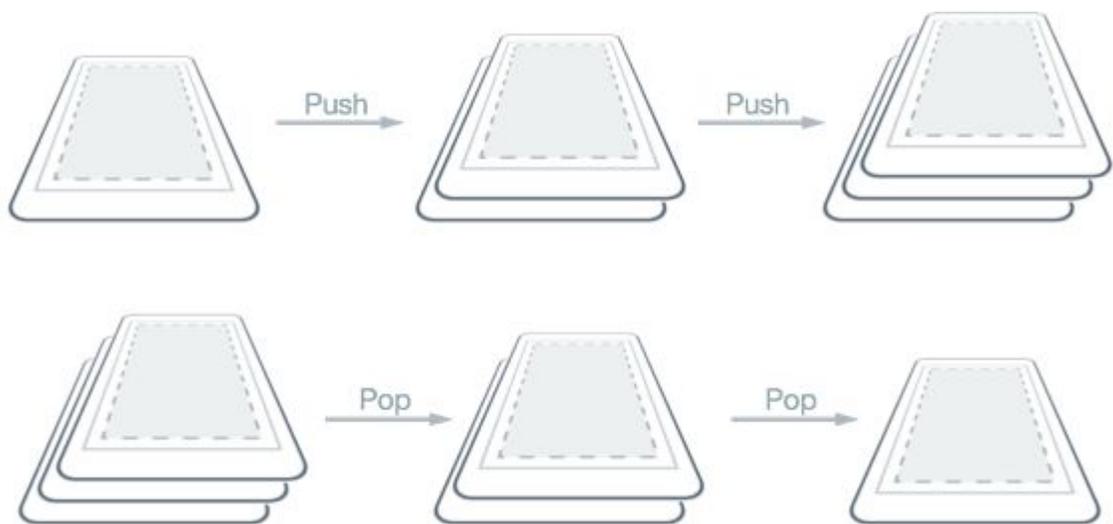
# Модальные страницы

Xamarin.Forms также поддерживает модальные страницы. На модальной странице пользователь должен выполнить отдельную задачу, причем он не может уйти с этой страницы, пока задача не будет выполнена или отменена.



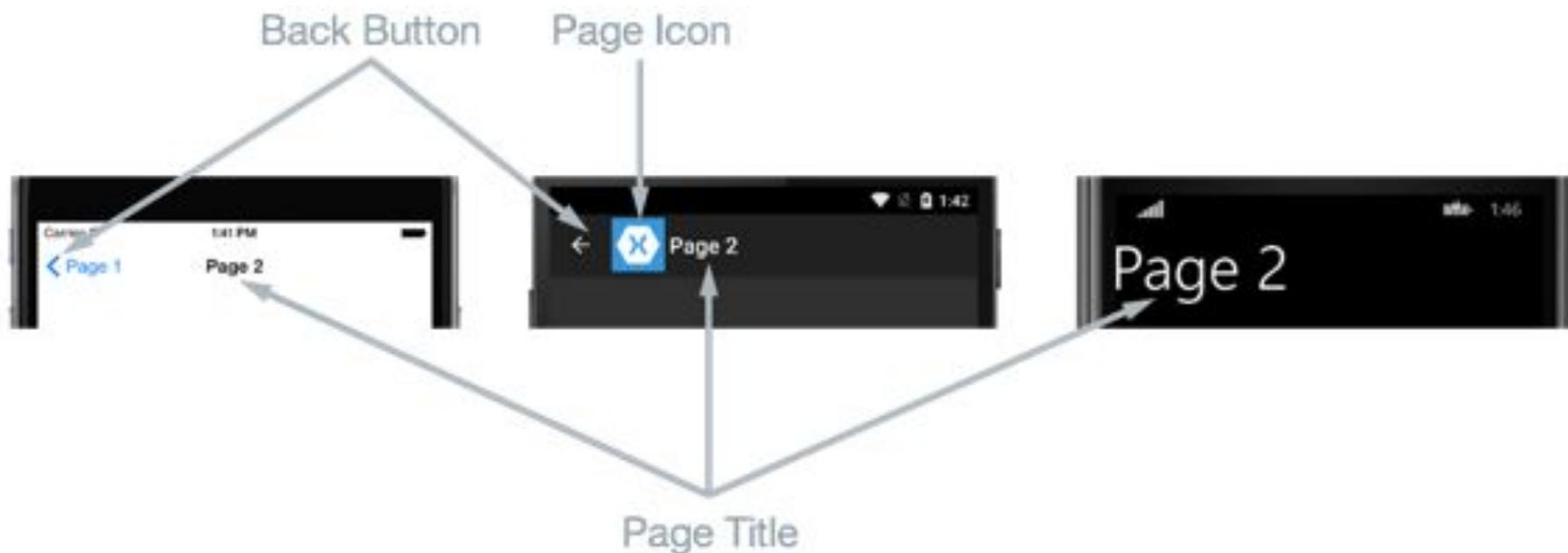
# Иерархическая навигация

Класс `NavigationPage` обеспечивает иерархическую навигацию, при которой пользователь может переходить по страницам вперед и назад по своему желанию. Этот класс реализует навигацию на основе стека объектов `Page` по методу ЛИФО (последним поступил – первым обслужен).



# Выполнение навигации

- ▶ При иерархической навигации класс `NavigationPage` используется для перехода по стеку объектов `ContentPage`.



# Создание корневой страницы

```
public App ()  
{  
    MainPage = new NavigationPage (new Page1Xaml ());  
}
```

# Помещение страниц в стек навигации

```
async void OnNextPageButtonClicked (object sender, EventArgs e)
{
    await Navigation.PushAsync (new Page2Xaml ());
}
```

# Извлечение страниц из стека навигации

```
async void OnPreviousPageButtonClicked (object sender, EventArgs e)
{
    await Navigation.PopAsync ();
}
```

# Извлечение страниц из стека навигации

```
async void OnRootPageButtonClicked (object sender, EventArgs e)
{
    await Navigation.PopToRootAsync ();
}
```

# Анимация переходов по страницам

```
async void OnNextPageButtonClicked (object sender, EventArgs e)
{
    // Page appearance not animated
    await Navigation.PushAsync (new Page2Xaml (), false);
}

async void OnPreviousPageButtonClicked (object sender, EventArgs e)
{
    // Page appearance not animated
    await Navigation.PopAsync (false);
}

async void OnRootPageButtonClicked (object sender, EventArgs e)
{
    // Page appearance not animated
    await Navigation.PopToRootAsync (false);
}
```

# Передача данных при переходе

Иногда странице необходимо передать данные другой странице во время навигации. Существует два способа:

- ▶ передача данных с помощью конструктора страниц
- ▶ указание данных для объекта `BindingContext` новой страницы.

# Передача данных через конструктор страниц

- ▶ Самый простой способ передачи данных на другую страницу во время навигации — в качестве параметра конструктора страниц.

```
public App ()  
{  
    MainPage = new NavigationPage (new MainPage (DateTime.Now.ToString ("u")));  
}
```

```
public MainPage (string date)  
{  
    InitializeComponent ();  
    dateLabel.Text = date;  
}
```

# Передача данных через объект BindingContext

- ▶ Альтернативный способ передачи данных на другую страницу во время навигации – указание данных для объекта BindingContext новой страницы

```
async void OnNavigateButtonClicked (object sender, EventArgs e)
{
    var contact = new Contact {
        Name = "Jane Doe",
        Age = 30,
        Occupation = "Developer",
        Country = "USA"
    };

    var secondPage = new SecondPage ();
    secondPage.BindingContext = contact;
    await Navigation.PushAsync (secondPage);
}
```

# Управление стеком навигации

- ▶ Свойство `Navigation` предоставляет свойство `NavigationStack`, из которого могут быть получены страницы в стеке навигации. Хотя `Xamarin.Forms` поддерживает доступ к стеку навигации, свойство `Navigation` предоставляет методы `InsertPageBefore` и `RemovePage` для управления стеком путем вставки страниц или их удаления.

# Управление стеком навигации

- ▶ Метод `InsertPageBefore` вставляет указанную страницу в стек навигации перед указанной существующей страницей



- ▶ Метод `RemovePage` удаляет указанную страницу из стека навигации



# Отображение представлений на панели навигации

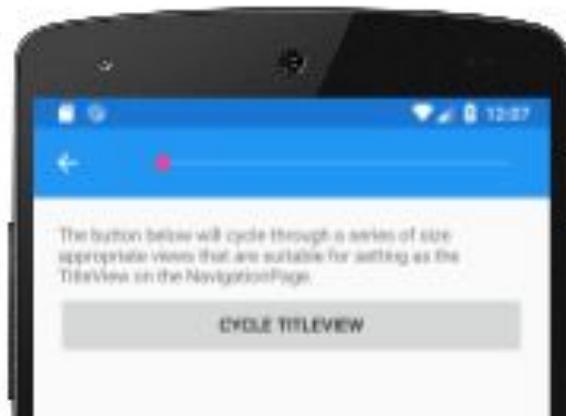
- ▶ Любое представление Xamarin.Forms View может отображаться на панели навигации `NavigationPage`. Для этого нужно установить присоединенное свойство `NavigationPage.TitleView` в `View`. Это присоединенное свойство может быть задано для любого объекта `Page`, и, когда `Page` помещается в `NavigationPage`, `NavigationPage` будет учитывать значение этого свойства.

# Пример

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigationPageTitleView.TitleViewPage">
  <NavigationPage.TitleView>
    <Slider HeightRequest="44" WidthRequest="300" />
  </NavigationPage.TitleView>
  ...
</ContentPage>
```

```
public class TitleViewPage : ContentPage
{
  public TitleViewPage()
  {
    var titleView = new Slider { HeightRequest = 44, WidthRequest = 300 };
    NavigationPage.SetTitleView(this, titleView);
    ...
  }
}
```

# Пример



# Ограничения

Существует ряд ограничений, которые следует учитывать при отображении объекта `View` на панели навигации `NavigationPage`.

- ▶ В iOS представления, размещенные на панели навигации `NavigationPage`, отображаются по-разному в зависимости от того, включены ли крупные заголовки.
- ▶ В Android можно поместить представления на панели навигации `NavigationPage` только в приложениях, использующих совместимость приложений.
- ▶ Не рекомендуется помещать большие и сложные представления, например `ListView` и `TableView`, в строке навигации `NavigationPage`.