UIViewController

Devteam

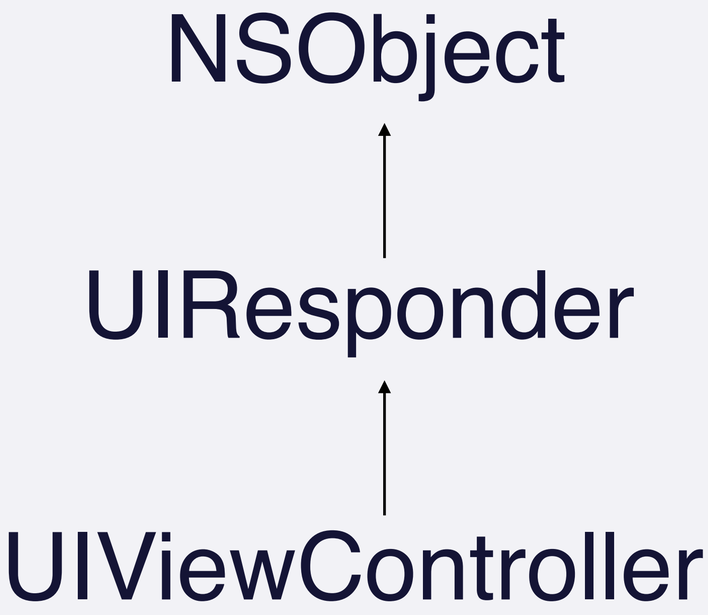
Макаровская Вероника Михайловна @MVeronika

Devteam

Что мы узнаем сегодня?

- UIViewController
- Виды Controller-ов
- Жизненный цикл
- Custom rootViewController
- UITabBarController
- UINavigationController

Devteam



UIViewController

- Обновление содержимого views(UIView), обычно в ответ на изменения базовых данных.
- Реагирование на взаимодействие пользователя с views(UIView).
- Изменение размеров views(UIView) и управление layout-ом всего интерфейса
- Координация с другими объектами, включая другие контроллеры, в вашем приложении
- Обеспечивает взаимосвязь модели и отображения

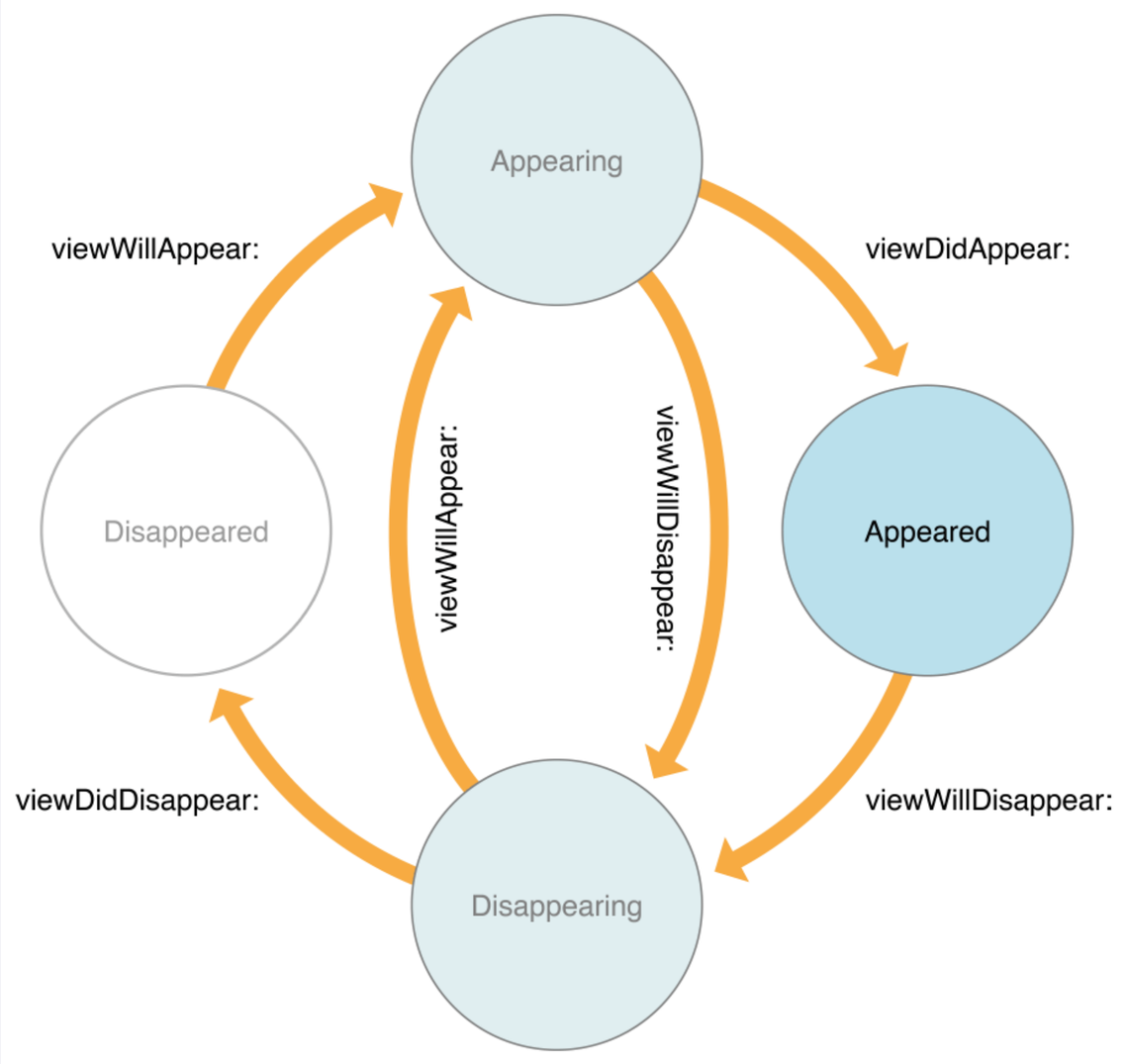
Devteam

Виды UIViewController

- UITableViewController
- UINavigationController
- UISplitViewController
- UIPageViewController
- UICollectionViewController
- …

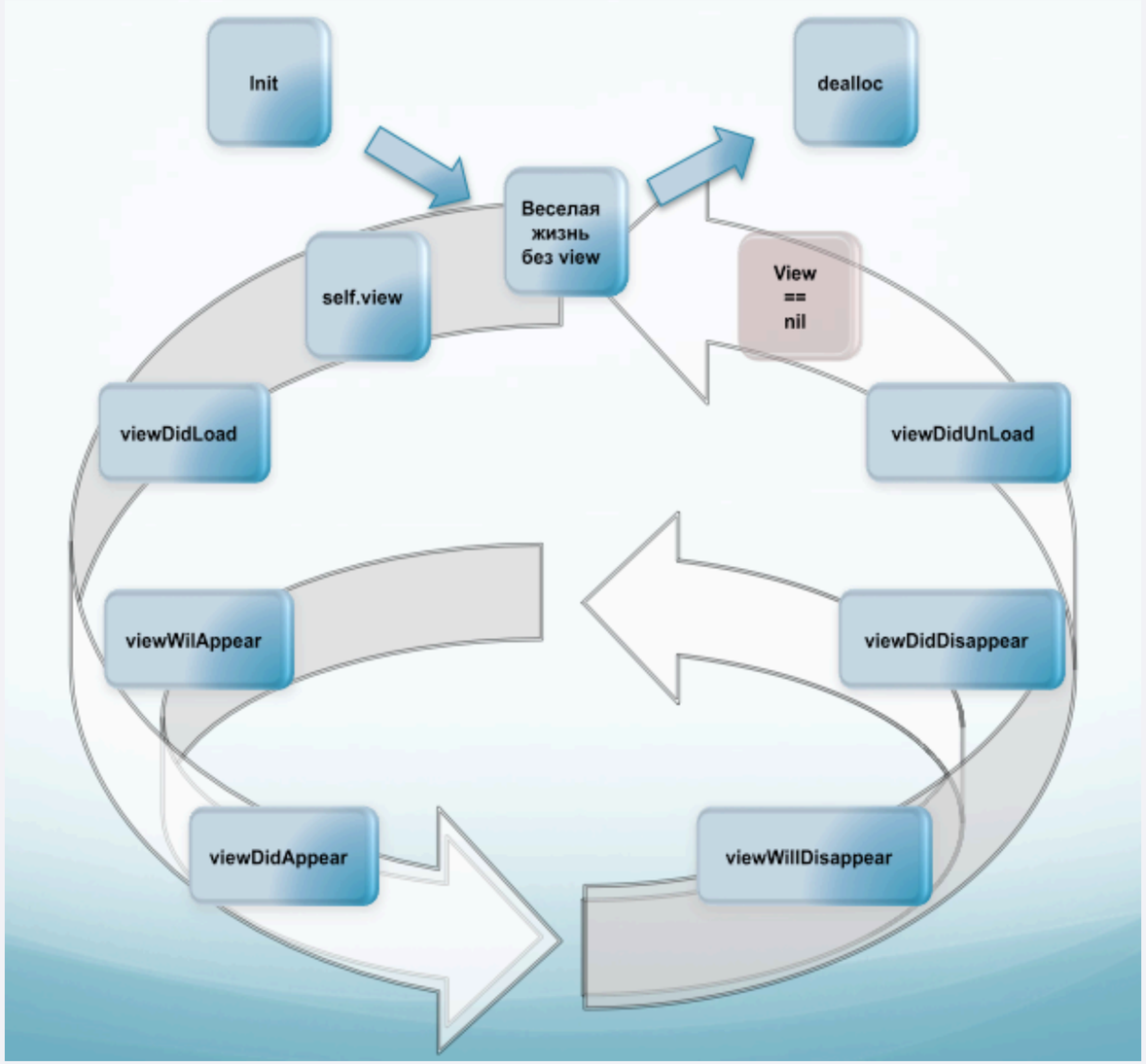
Devteam

Жизненный цикл UIViewController



Devteam

Жизненный цикл UIViewController

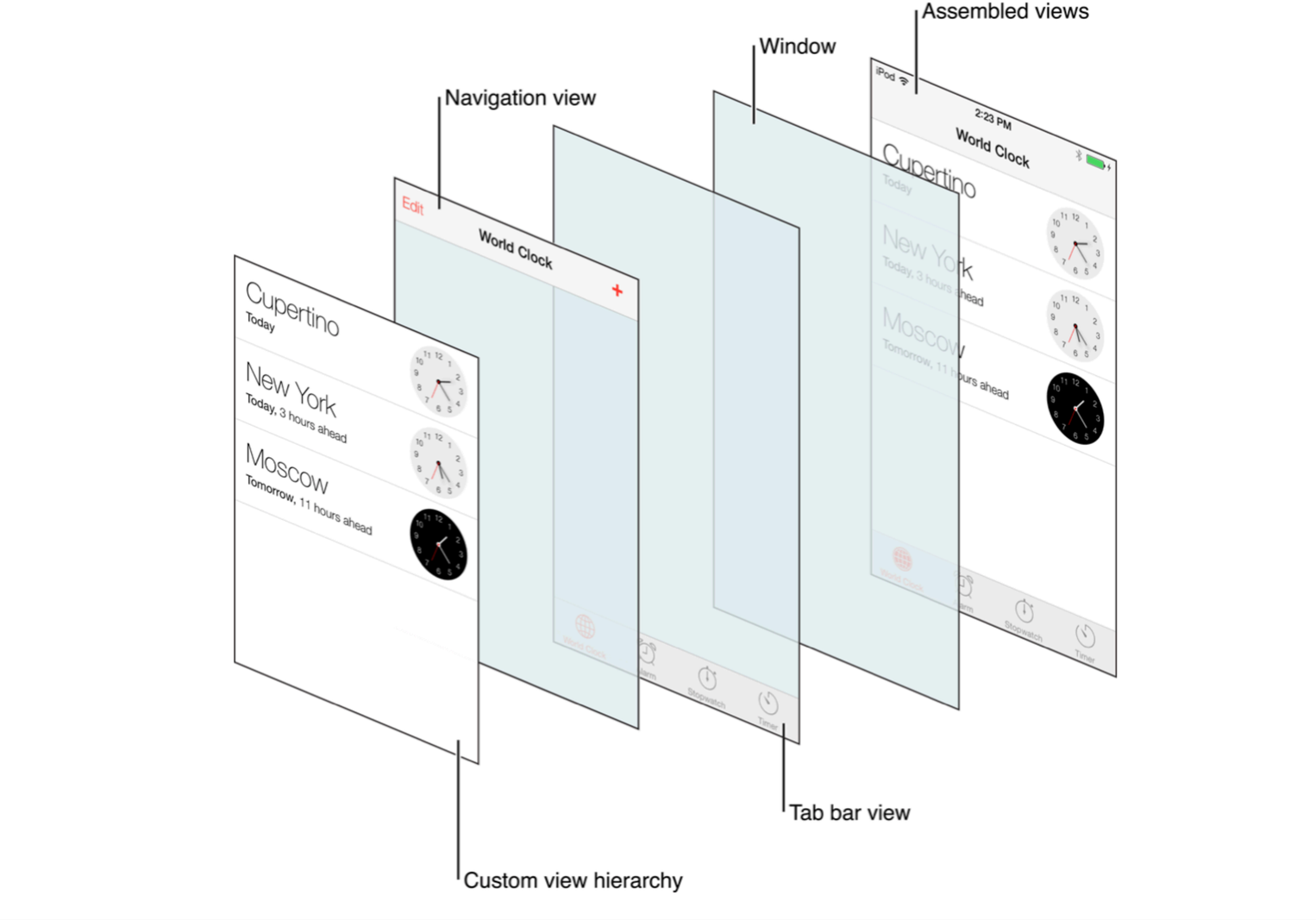


Devteam

UIWindow

- Он содержит видимый контент вашего приложения
- Он играет ключевую роль в доставке сенсорных событий вашим представлениям и другим объектам приложения
- Он работает с контроллерами вашего приложения для обработки изменений ориентации

UIWindow



Devteam

RootViewController

- Корневой контроллер обеспечивает показ содержимого window.
- Назначение контроллера этому свойству (программно или с помощью Interface Builder) устанавливает view контроллера в качестве view содержимого window.
- В приложении может быть несколько реализаций RootViewController

RootViewController

- Значением по умолчанию этого свойства является null.
- Новое view содержимого настроено на отслеживание размера window, изменяющегося при изменении размера window.
- Если window имеет существующую иерархию views, старые views удаляются до установки новых.

Devteam

RootViewController

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions
    launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    let window = UIWindow(frame: UIScreen.main.bounds)
    window.rootViewController = RootViewController()
    window.makeKeyAndVisible()
    return true
}
```

Devteam

Как показать?

```
let second = SecondViewController()
self.present(second, animated: true, completion: nil)
```

Можно менять анимацию перехода через свойство .modalTransitionStyle:

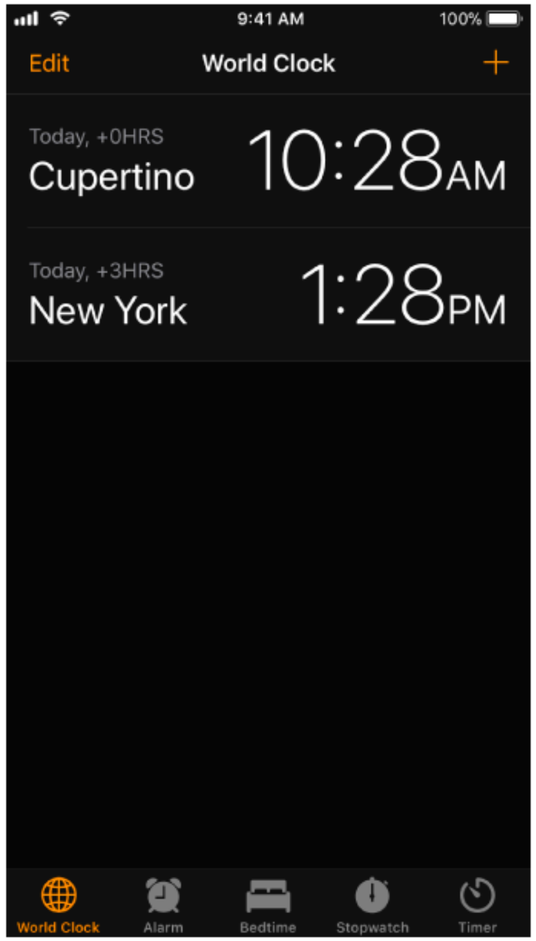
```
    second.modalTransitionStyle = .crossDissolve
```

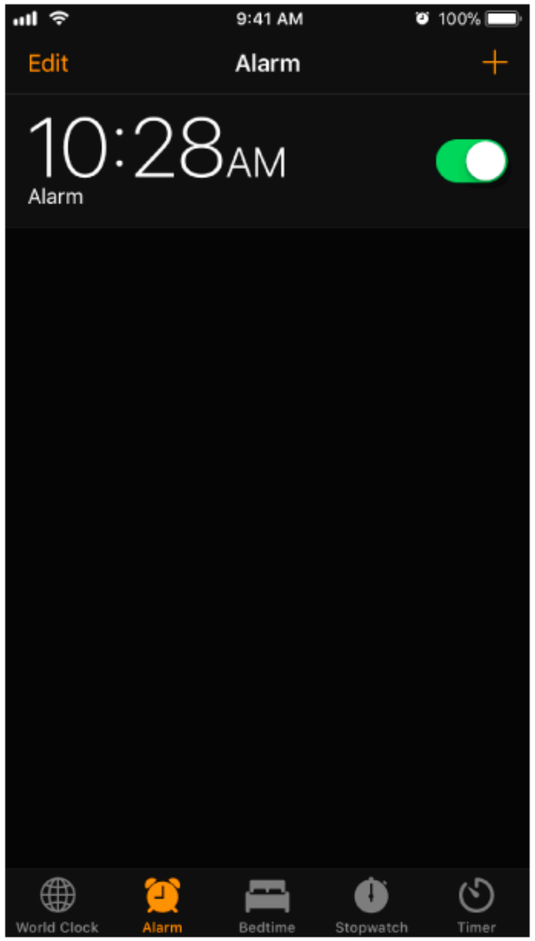
Devteam

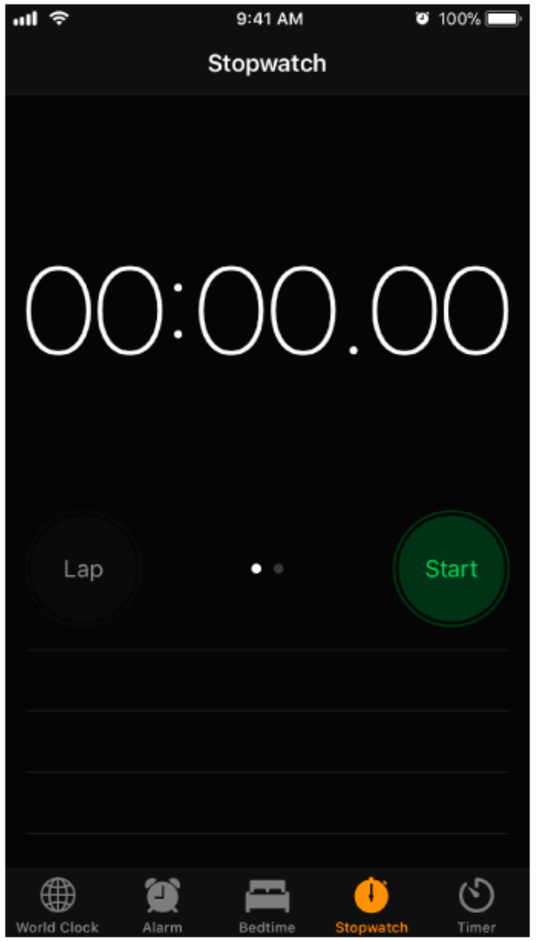
Let’s Code

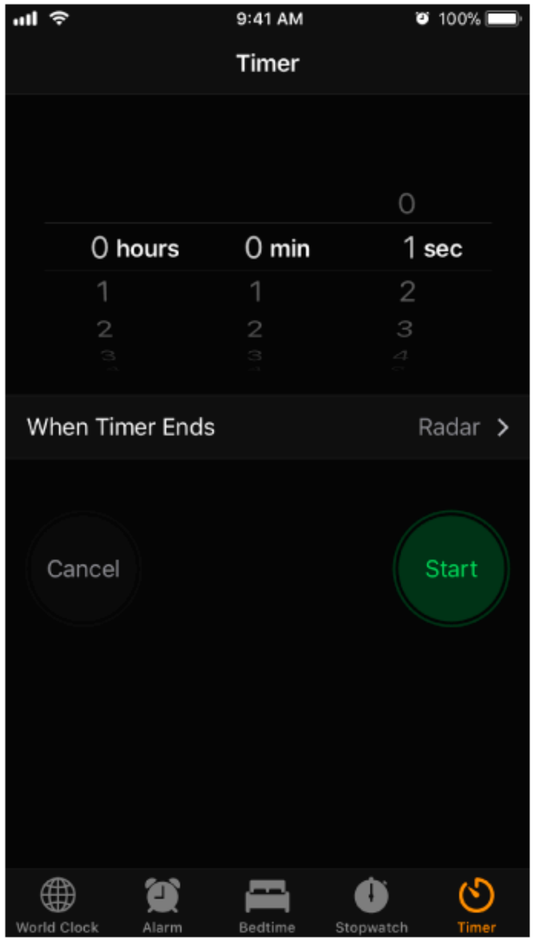
Devteam

UITabBarController



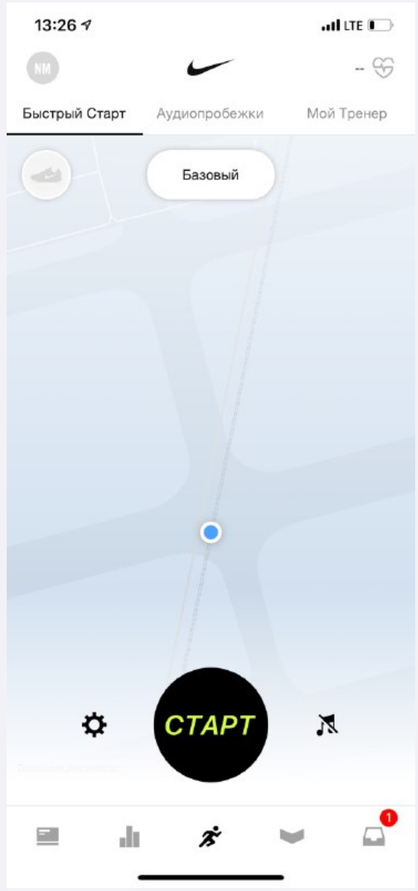


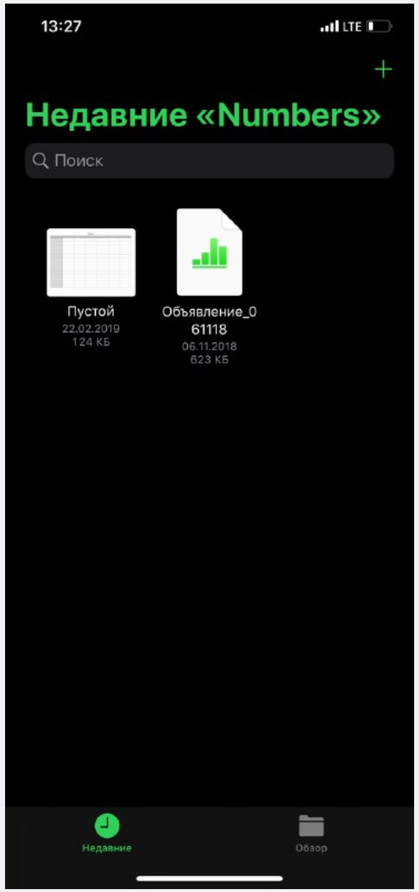


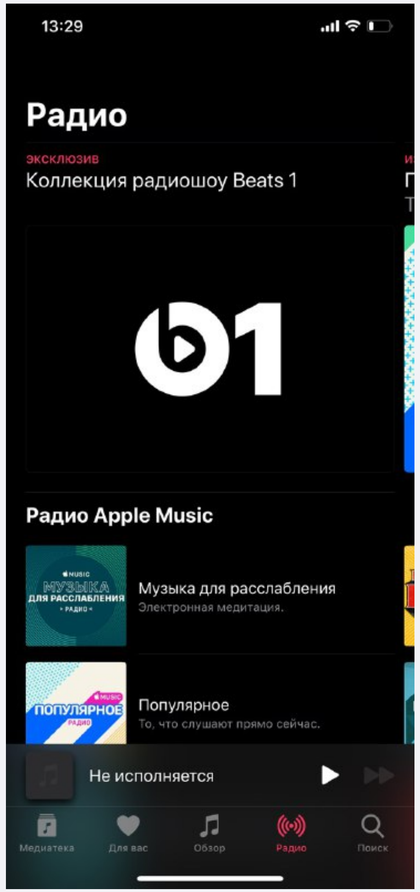


Devteam

UITabBarController

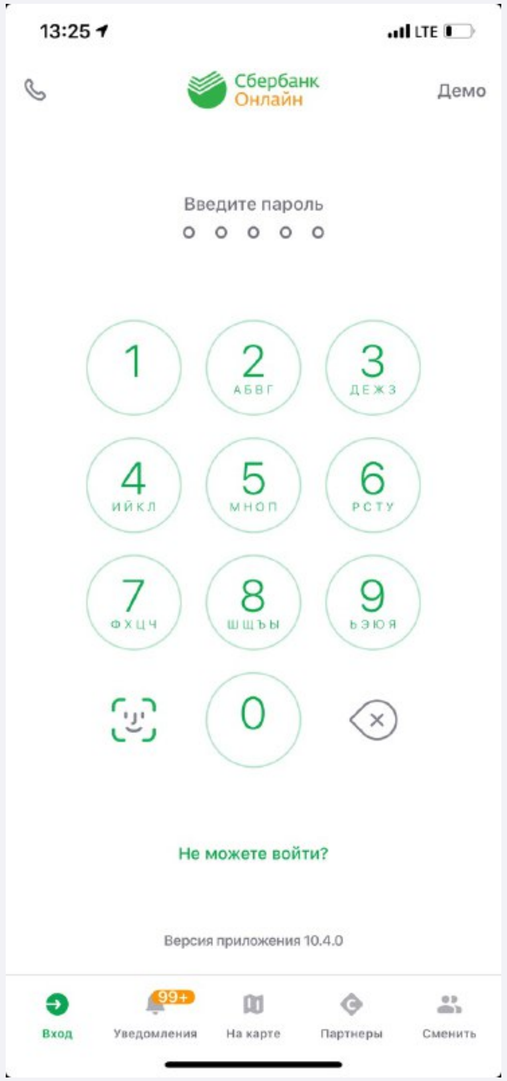


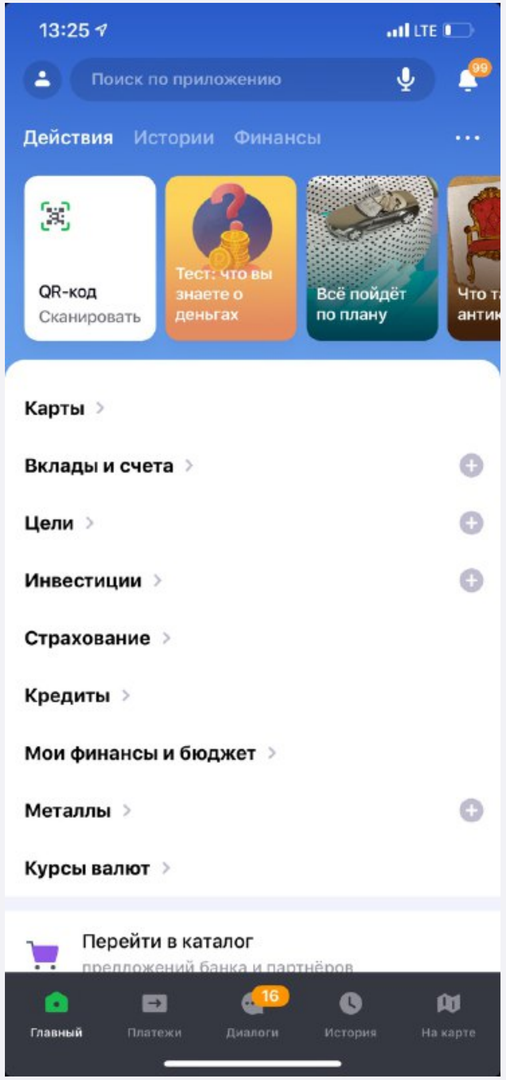




Devteam

UITabBarController и RootViewController





UITabBarController

- Интерфейс панели вкладок отображает вкладки в нижней части окна для выбора между различными режимами и для отображения view для этого режима.

Devteam

UITabBarController

- Каждая вкладка интерфейса контроллера со своим настраиваемым контроллером.
- Когда пользователь выбирает конкретную вкладку, UITabBarController корневой вид соответствующего контроллера представления, заменяя любые предыдущие представления.

UITabBarController

- Фактически, интерфейсы панели вкладок обычно используются либо для представления различных типов информации, либо для представления одной и той же информации с использованием совершенно другого стиля интерфейса.

Devteam

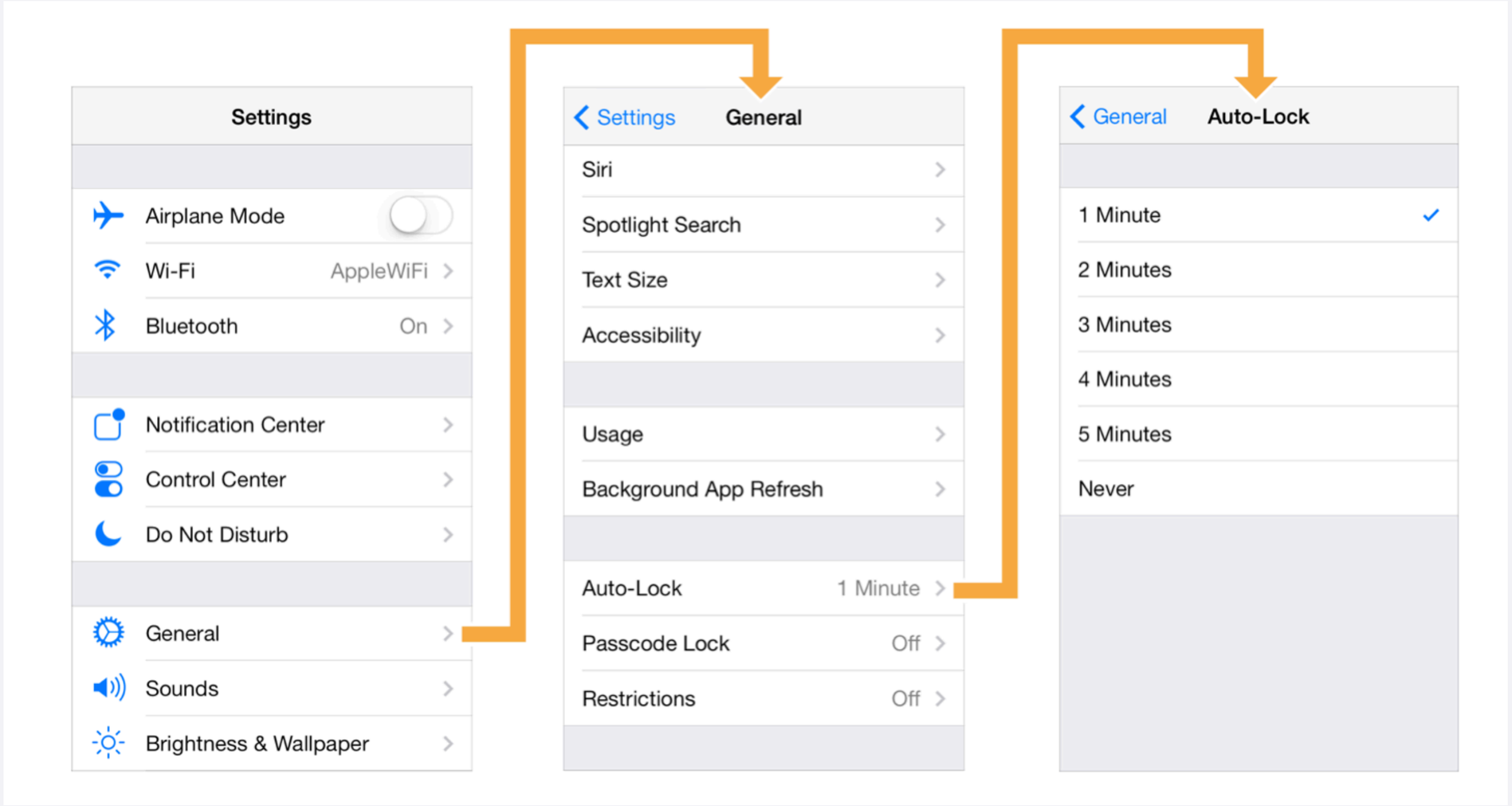
UITabBarItem и UITabBar

- Содержит массив контроллеров и управляет UITabBar
- Заголовок .tabBarItem.title
- Иконка .tabBarItem.image
- .hidden - UITabBar

Let’s Code

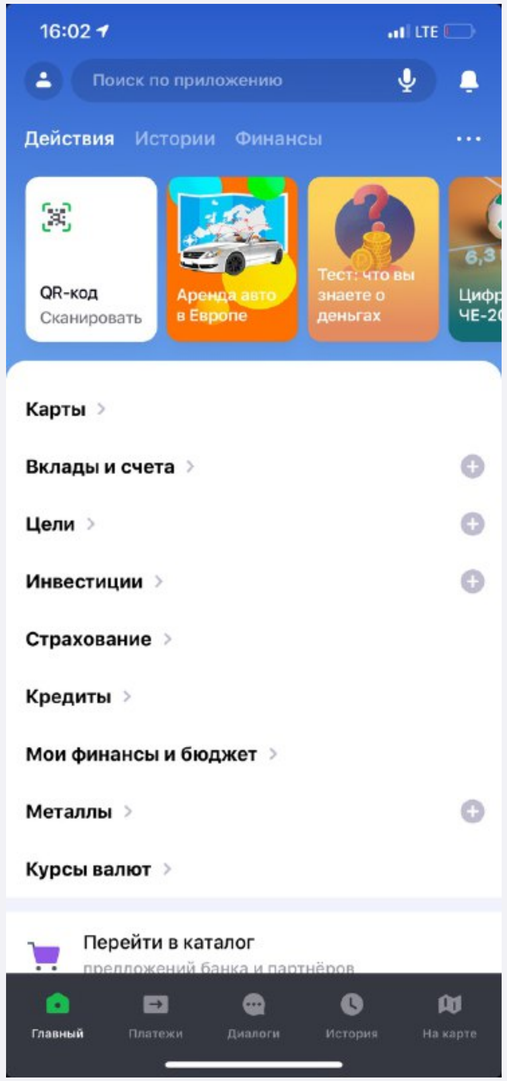
Devteam

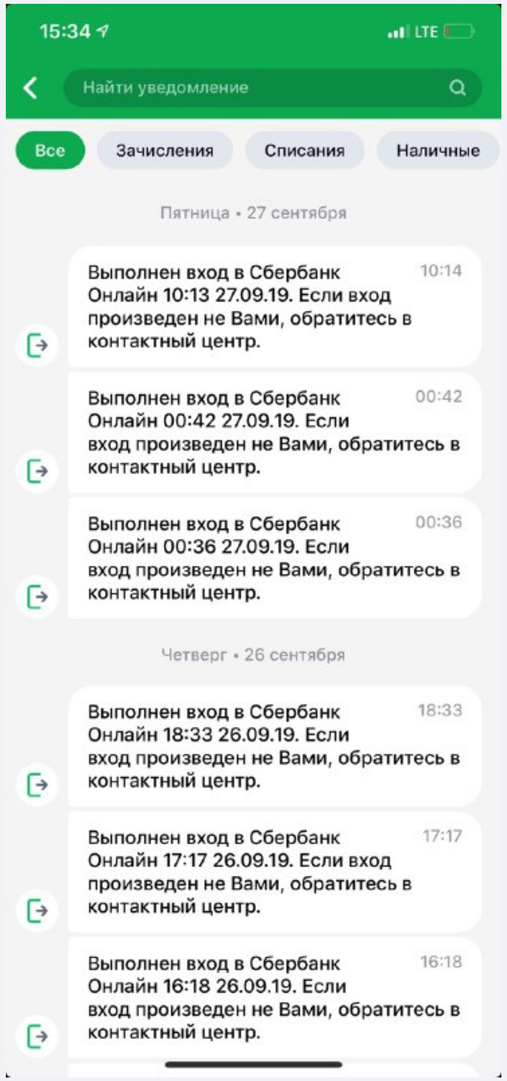
UINavigationController

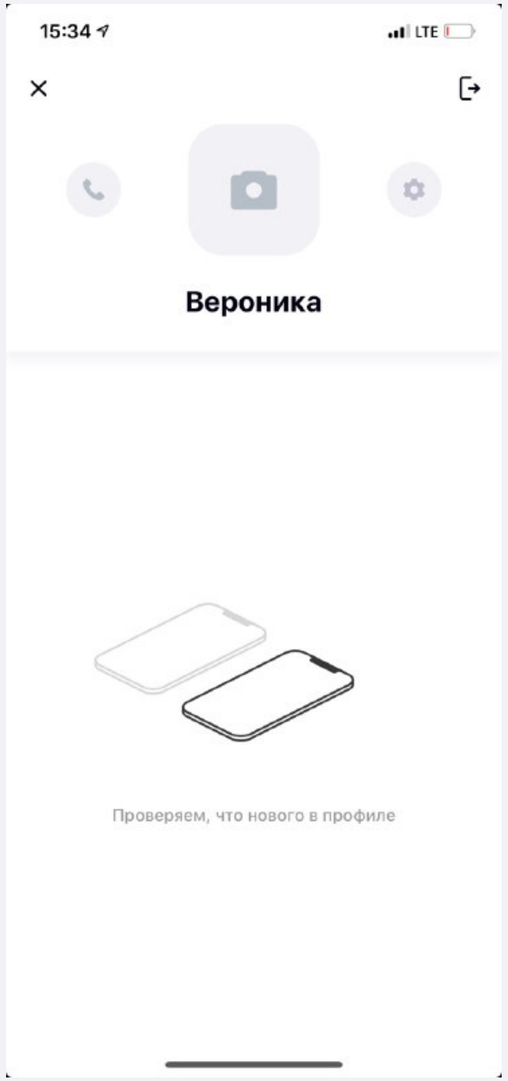


Devteam

UINavigationController

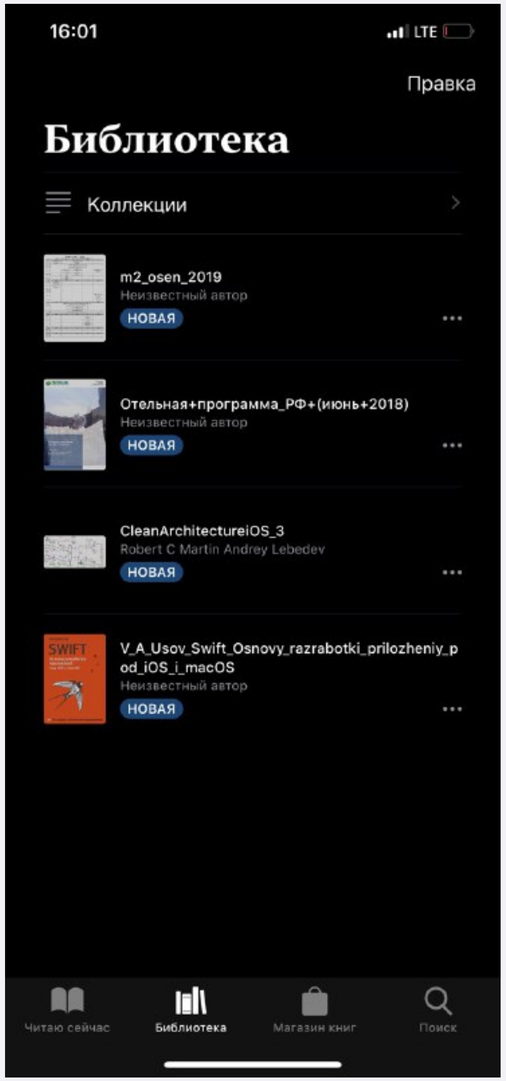


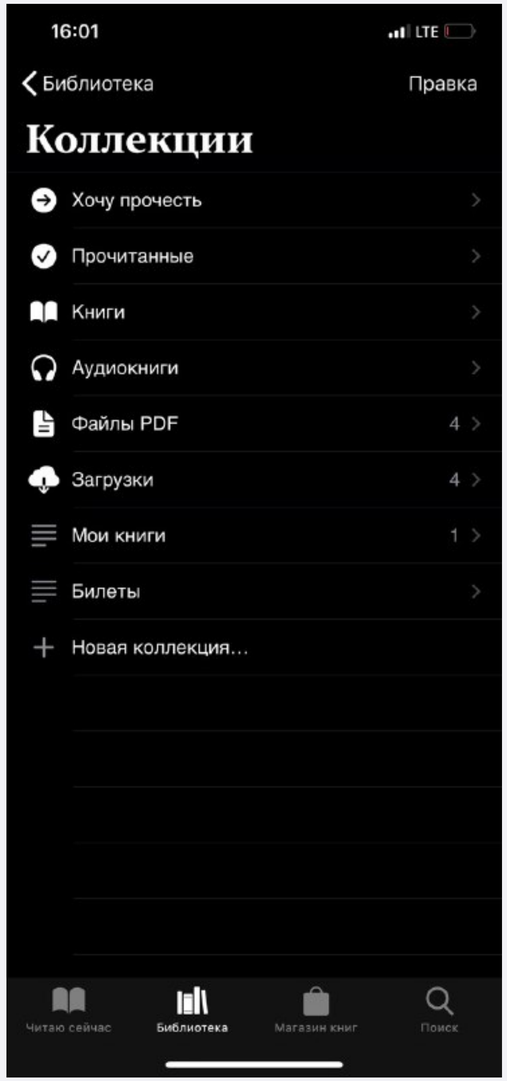




Devteam

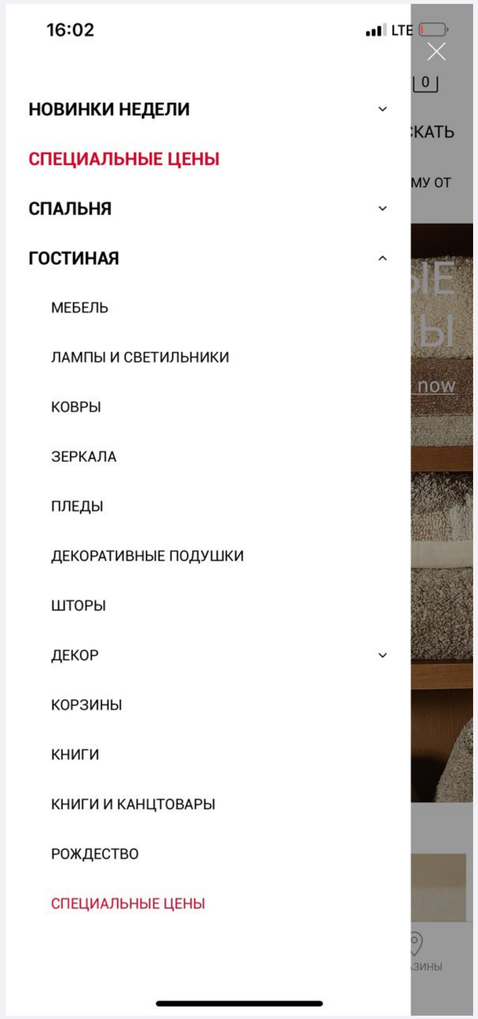
UINavigationController

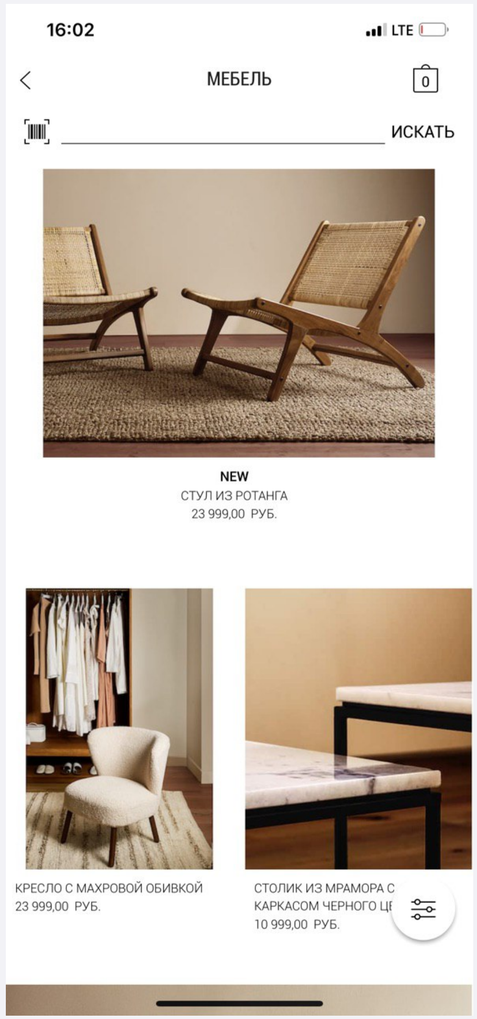




Devteam

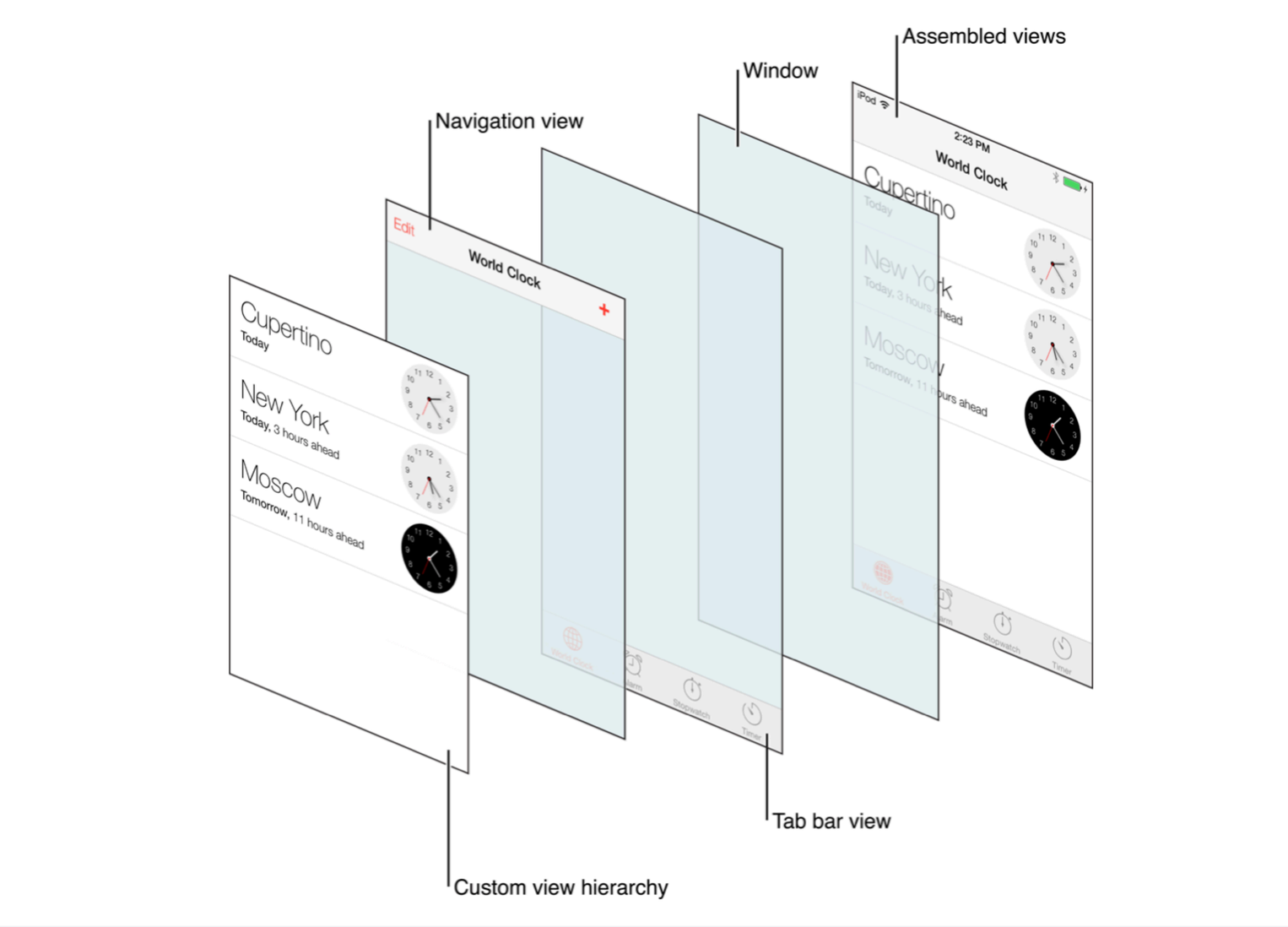
UINavigationController





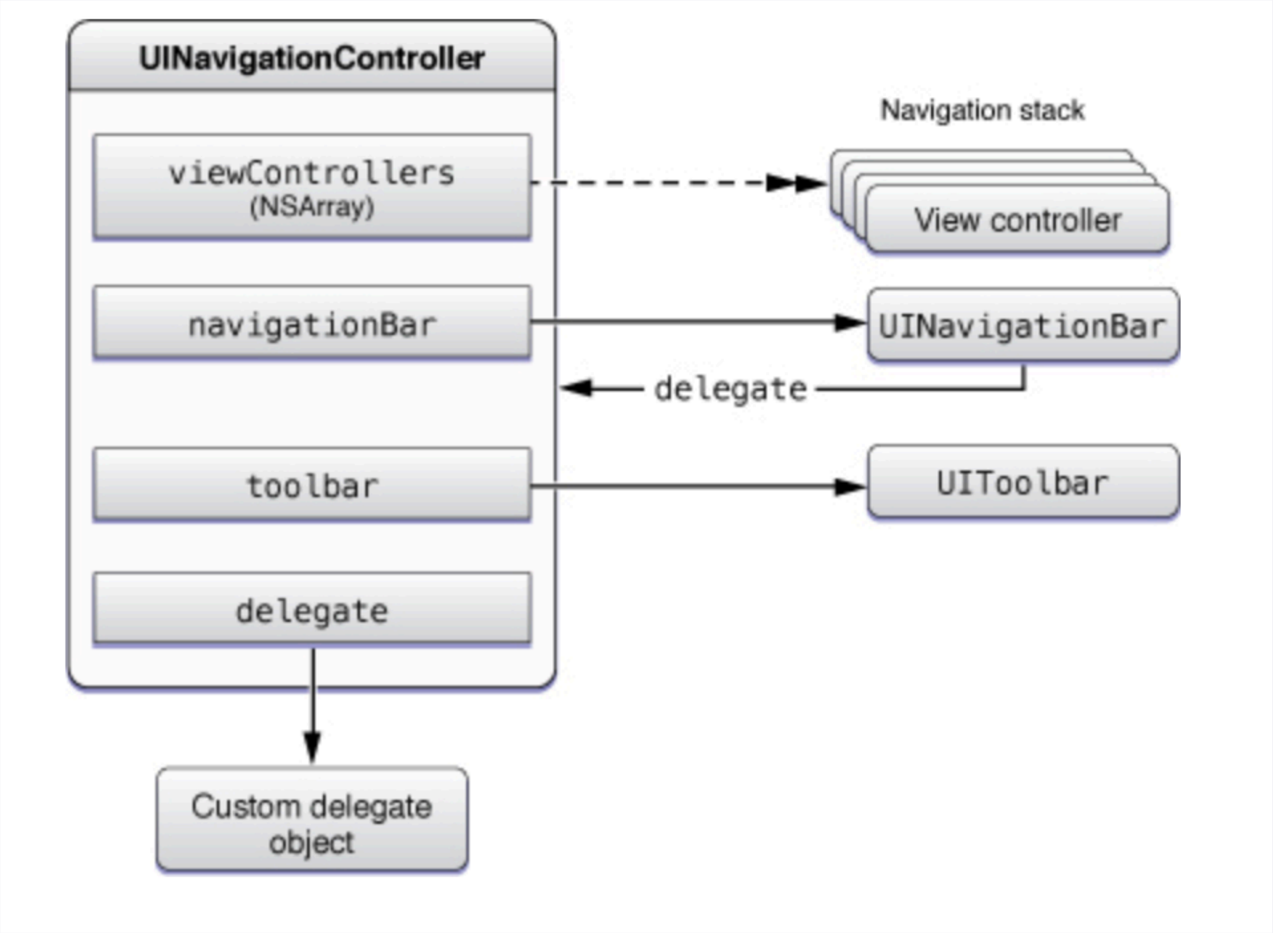
Devteam

UINavigationController



Devteam

UINavigationController




```swift
func pushViewController(_
viewController: UIViewController,
animated: Bool) // добавляет сверху
```

```swift
func popViewController(animated: Bool)
-> UIViewController? // снимает
верхний
```

```swift
func
popToRootViewController(animated:
Bool) -> [UIViewController]? //
сбрасывает до  первого контроллера
```

Devteam

UINavigationController

- Контроллер навигации представляет собой контроллер представления контейнера, который управляет одним или несколькими контроллерами дочернего представления в интерфейсе навигации.
- В этом типе интерфейса одновременно виден только один дочерний контроллер представления.

UINavigationController

- При выборе элемента в контроллере представления новый контроллер представления выводится на экран с помощью анимации, тем самым скрывая предыдущий контроллер представления.
- Нажатие кнопки «Назад» в навигационной панели в верхней части интерфейса удаляет контроллер вида сверху, тем самым раскрывая контроллер вида снизу.

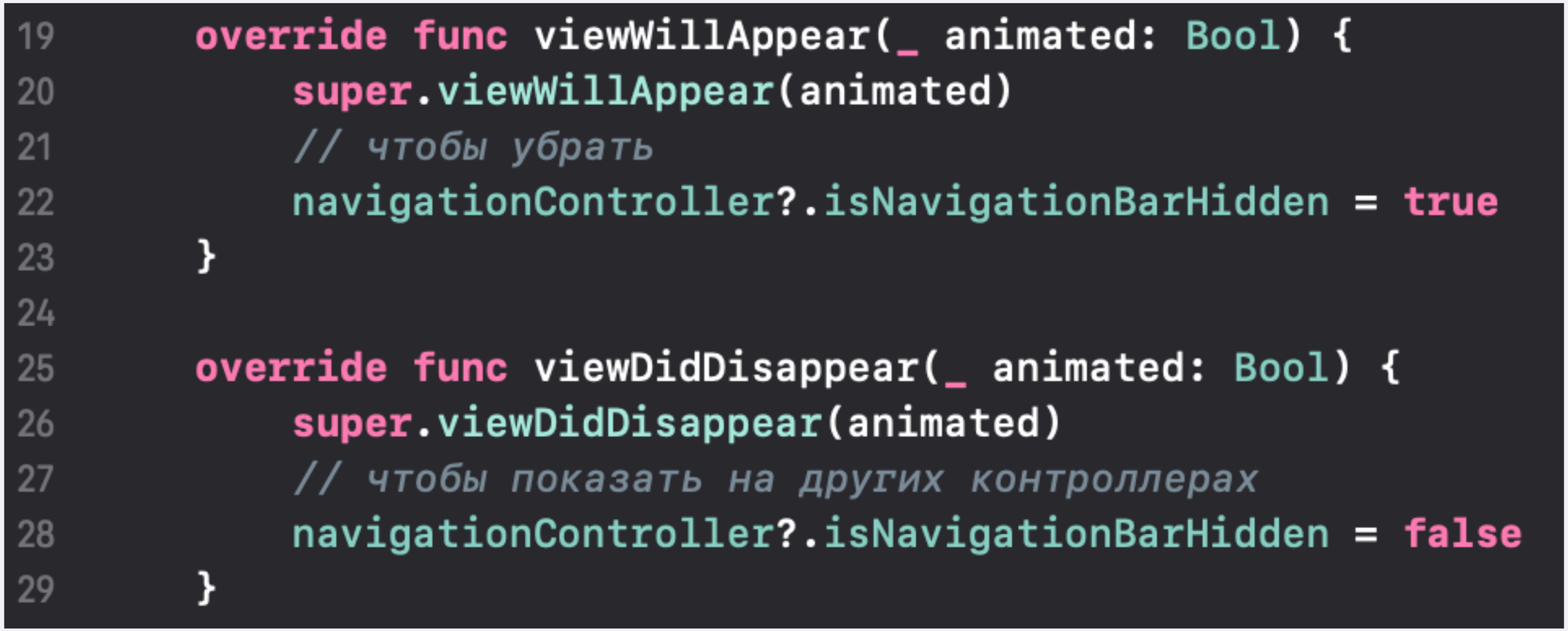
UINavigationController

- Объект контроллера навигации управляет своими дочерними контроллерами представления, используя упорядоченный массив, известный как стек навигации.
- Первый контроллер представления в массиве является корневым контроллером представления и представляет основание стека.
- Последний контроллер представления в массиве является самым верхним элементом в стеке и представляет контроллер представления, отображаемый в данный момент.

UINavigationControllerDelegate

- Используйте делегат контроллера навигации (пользовательский объект, который реализует этот протокол), чтобы изменить поведение, когда контроллер представления добавляется или извлекается из стека навигации объекта UINavigationController.

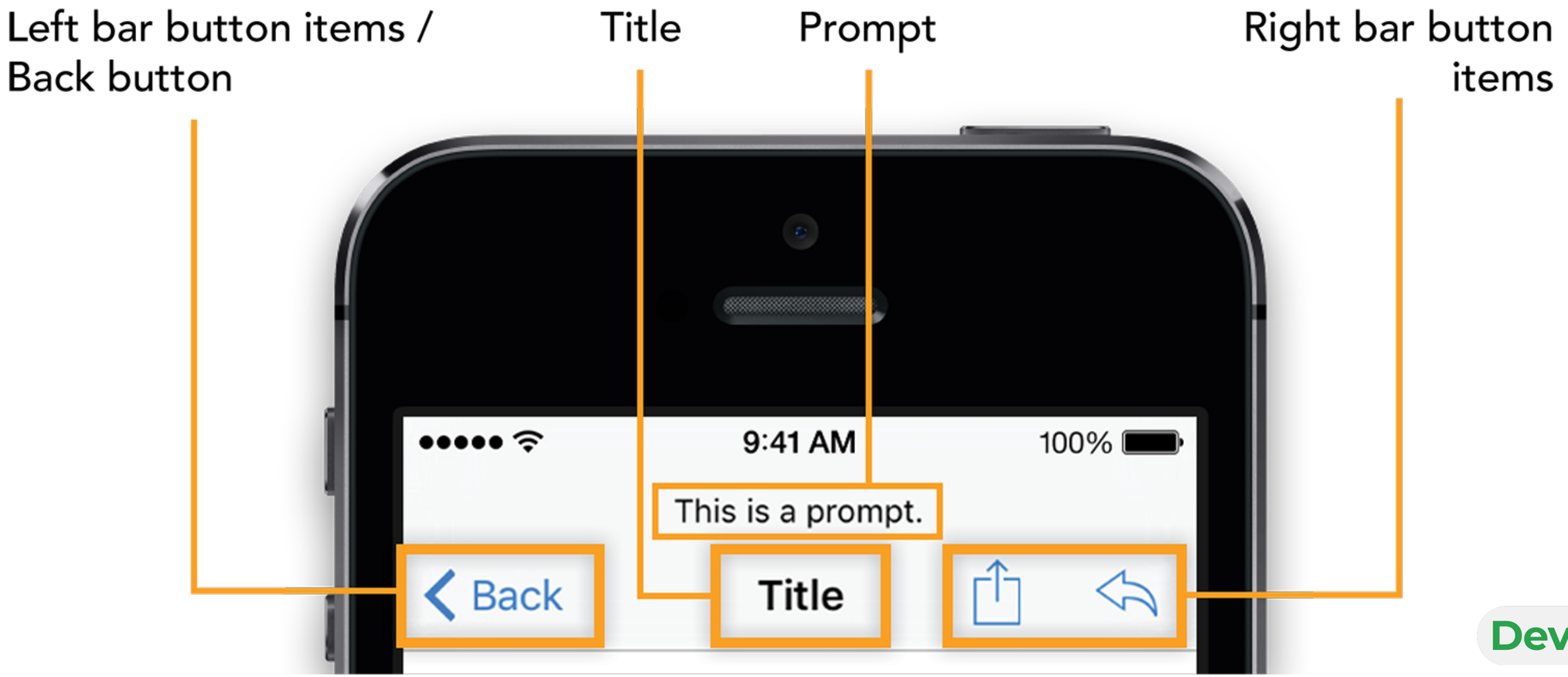
.hidden



```
19      override func viewWillAppear(_ animated: Bool) {
20          super.viewWillAppear(animated)
21          // чтобы убрать
22          navigationController?.isNavigationBarHidden = true
23      }
24
25      override func viewDidDisappear(_ animated: Bool) {
26          super.viewDidDisappear(animated)
27          // чтобы показать на других контроллерах
28          navigationController?.isNavigationBarHidden = false
29      }
```

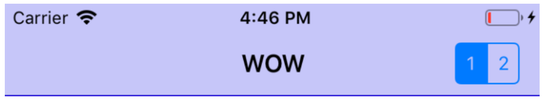
Devteam

UINavigationController.NavigationBar

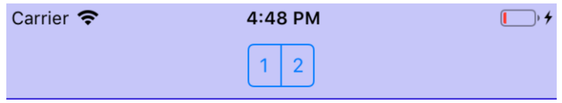


Devteam

UINavigationController.NavigationBar




```
let segmentedControl = UISegmentedControl(items: ["1", "2"])
let segmentBarItem = UIBarButtonItem(customView: segmentedControl)
navigationItem.rightBarButtonItem = segmentBarItem
```




```
let segmentedControl = UISegmentedControl(items: ["1", "2"])
self.navigationItem.titleView = segmentedControl
```


```
let segmentBarItem = UIBarButtonItem(customView: segmentedControl)
navigationItem.leftBarButtonItem = segmentBarItem
```

Let’s Code

Devteam

Обработка изменений ориентации

`viewWillTransition(to:with:)`

Devteam

Домашнее задание

- Реализовать кнопку share (UIActivityViewController)
- Исключить PostToFlickr, PostToVimeo, SaveToCameraRoll
- Обработать остальные
- Реализовать 1 любую свою UIActivity

Devteam

Полезные ссылки

- https://developer.apple.com/documentation/uikit/views_and_controls/uikit_catalog_creating_and_customizing_views_and_controls
- https://habr.com/en/post/416147/
  https://medium.com/@stasost/ios-root-controller-navigation
- https://developer.apple.com/library/archive/featuredarticles/ViewControllerPGforiPhoneOS/ImplementingaContainerViewController.html
- https://www.youtube.com/watch?v=ZKwZjxN9ka4 - свои контейнеры
- https://developer.apple.com/documentation/uikit/view_controllers

Devteam

Полезные ссылки

- https://www.raywenderlich.com/170144/custom-uiviewcontroller-transitions-getting-started
- https://developer.apple.com/documentation/uikit/uinavigationcontroller/customizing_your_app_s_navigation_bar
- https://developer.apple.com/documentation/uikit/uiviewcontroller

Обратная связь

Devteam

Вопросы?

Devteam

Спасибо 💚

Devteam