

# Программирование под iOS

Лекция 6

**Бесараб Иван**

Жизненный цикл UIViewController, приложения с  
несколькими MVC

Просьба отключить мобильные телефоны



# Содержание лекции

- методы `ViewController`a`
- последовательность методов и варианты развития событий)



# Содержание лекции

- UINavigationController
- UITabBarController
- segue
- singleton



# UIViewController lifecycle

- методы которые вызываются у ViewController во время работы приложения:
- все начинается с создания  
большинство создаются из storyboard
- после создания на каждом из этапов вызываются соответствующие методы при:  
связывание outlet`ов  
появление и исчезновение с экрана  
изменение геометрии (переворот устройства)  
ситуации с нехваткой памяти

не забудьте про **super**



# UIViewController lifecycle

- при создании VC вызывается метод
- `init(nibName: String,  
bundle: NSBundle)`
- `awakeFromNib`
- из обоих этих методов необходимо вызвать метод setup
- outlets are not set!



# UIViewController lifecycle

- `viewDidLoad()`
- лучшее место для инициализации (outlet`ы установлены)
- вызывается до появления VC на экране
- вызывается только один раз при инициализации VC
- границы и размеры элементов не установлены  
не писать код который работает с отображением,  
размерами и т. д.



# UIViewController lifecycle

- появление view на экране
- `viewWill/DidAppear(animated: bool)`
- лучшее место для синхронизации с model
- вызывается до появления VC на экране
- вызывается множество раз до появления VC
- границы и размеры элементов установлены



# UIViewController lifecycle

- изменение геометрии при поворотах
- `viewWill/DidLayoutSubviews(animated: bool)`
- лучшее место для синхронизации перерисовки UI для нужной ориентации

When a rotation occurs for a visible view controller, the `willRotateToInterfaceOrientation:duration:`, `willAnimateRotationToInterfaceOrientation:duration:`, and `didRotateFromInterfaceOrientation:` methods are called during the rotation. The `viewWillLayoutSubviews` method is also called



# UIViewController lifecycle

- уход view с экрана
- `viewWill/DidDisappear(animated: bool)`
- лучшее место для синхронизации с model
- сохранить текущее состояние
- вызывается перед уходом VC с экрана
- перестать кушать ресурсы (анимация, аудио, видео)



# UIViewController lifecycle

- нехватка памяти (ОЗУ) на устройстве
- `didReceiveMemoryWarning()`
- место для присвоения strong указателям на возобновляемые ресурсы nil которые в не используются в данный момент



# UINavigationController

- Пример - календарь
- Year View
- Month View
- Day View

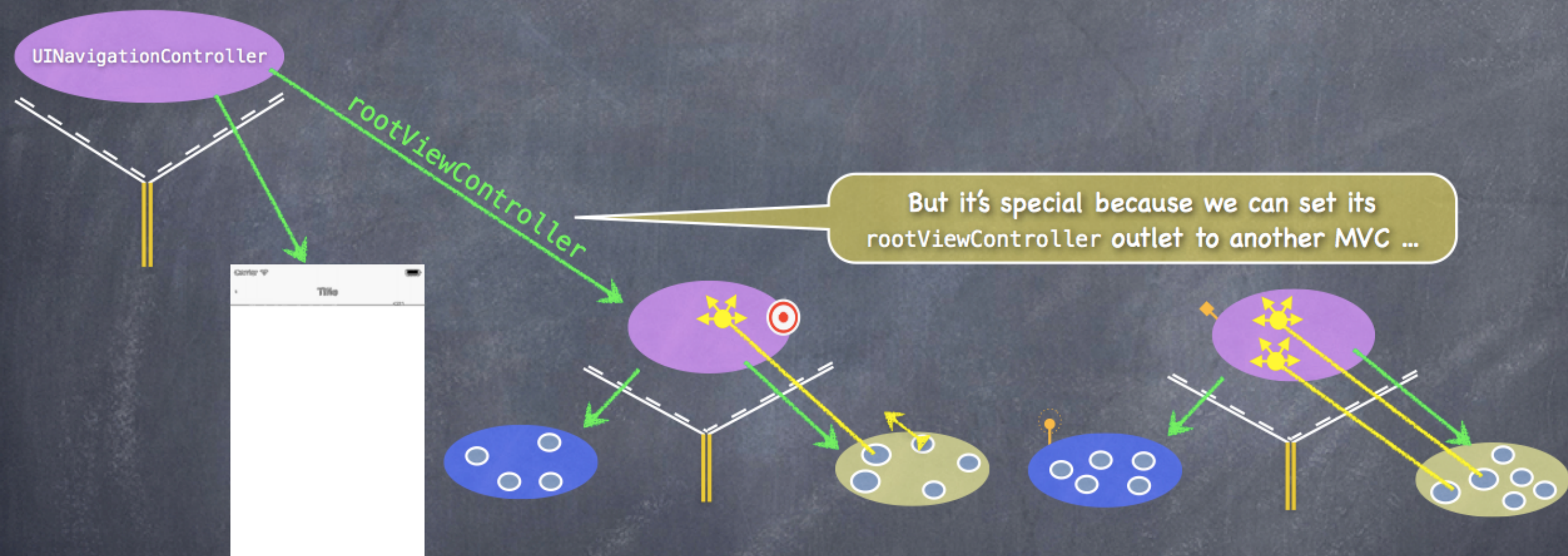


# UINavigationController

- Верхний бар - `navigationItem`
- Надпись - `title`
- Кнопка справа от `title` - `navigationItem.rightBarButtonItem`
- Кнопка Back - появляется автоматически
- Нижний бар - `toolbarItems`

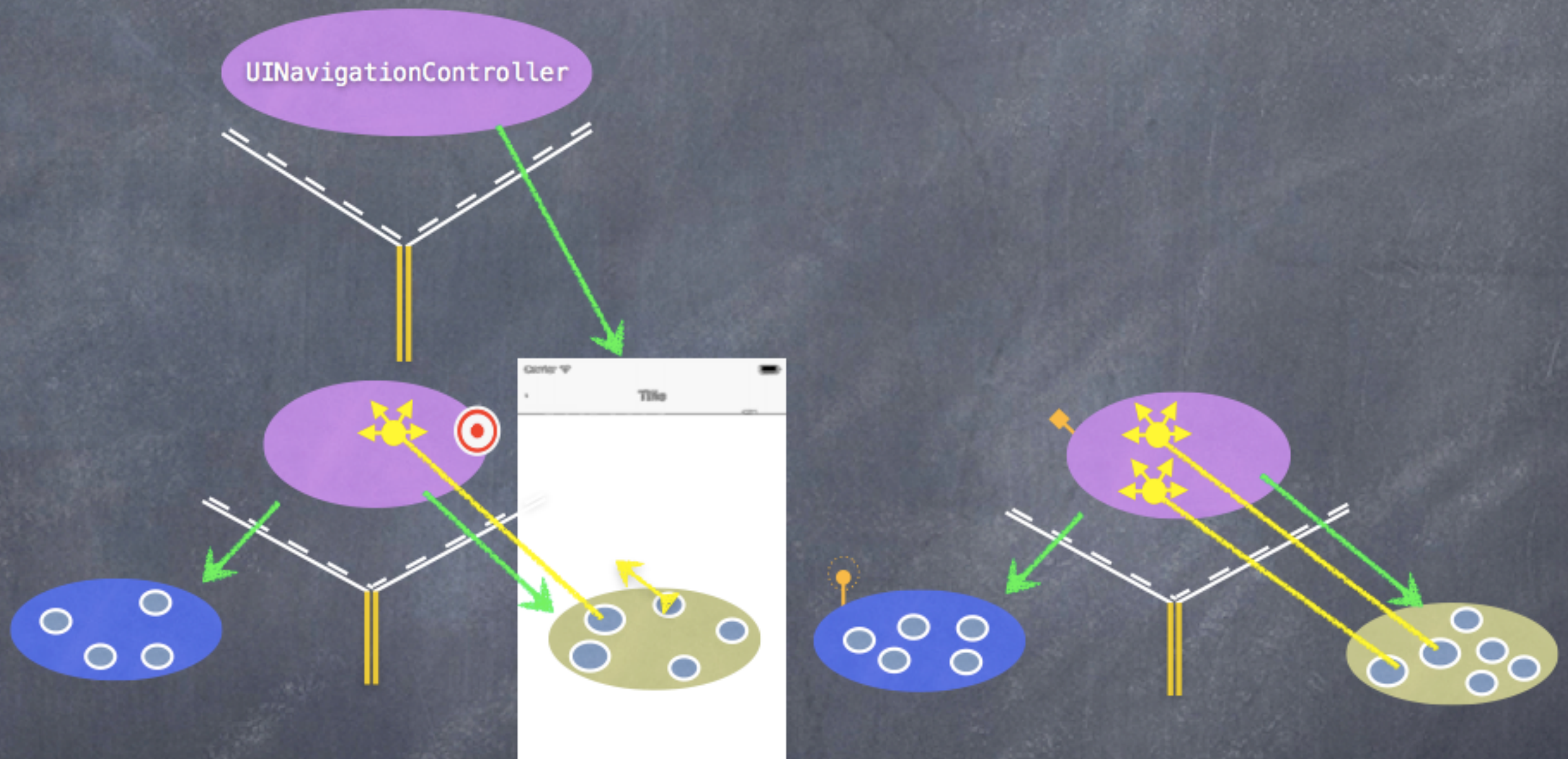


# Взаимодействие нескольких MVC



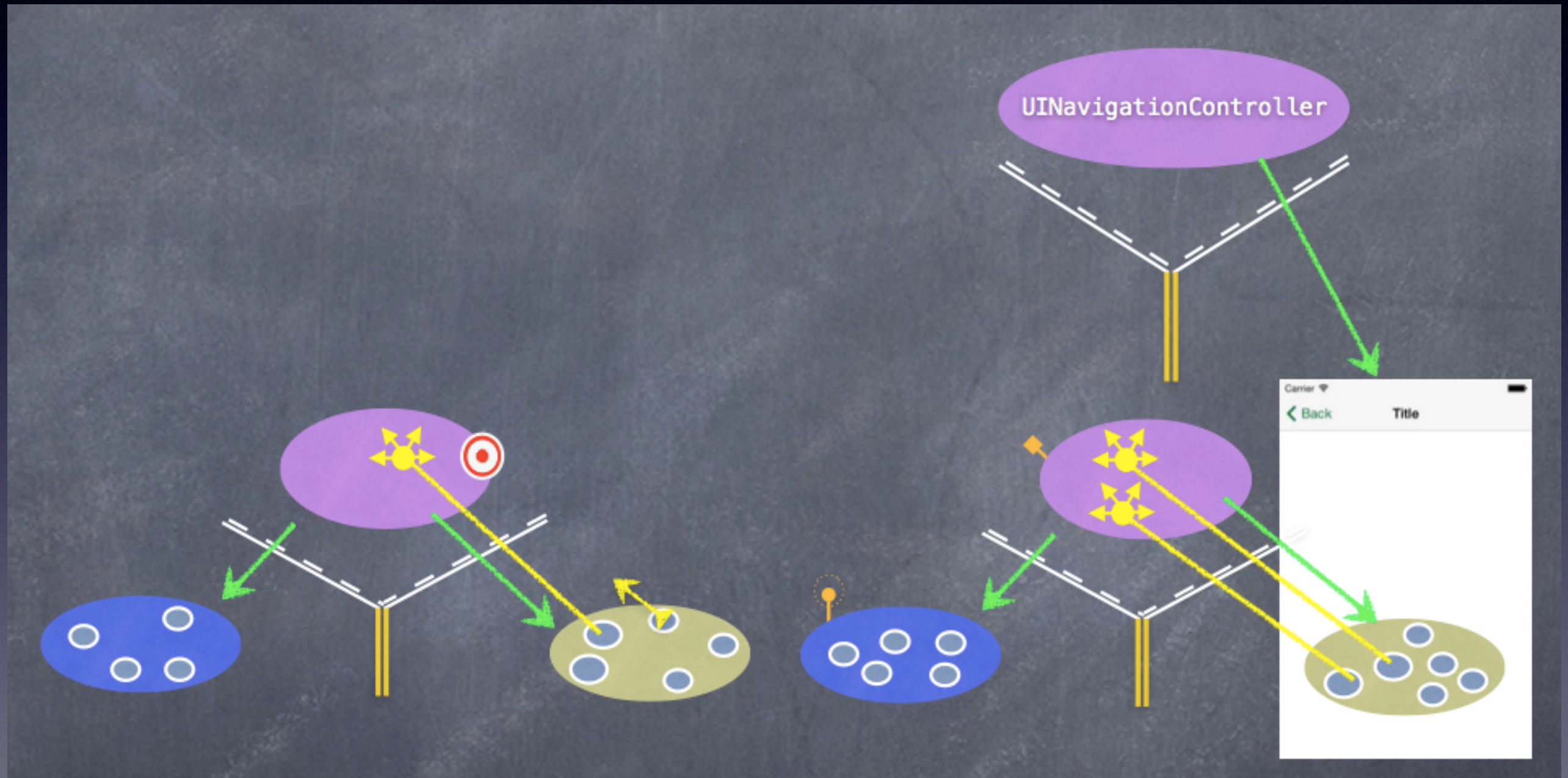


# Взаимодействие нескольких MVC





# Взаимодействие нескольких MVC





# Segues - переходы

- `popViewControllerAnimated(animated: bool)` -  
ВЫЗЫВАТЬ у `navigationController`
- **Вызов Segue из кода**  
`performSegue(identifier: String, sender: AnyObject?)` ВЫЗЫВАТЬ у `self`



# Segues - переходы

- Перед переходом у `UIViewController` вызывается
  - `prepareForSegue` это место где нужно передать данные в следующую MVC
- Метод
  - `shouldPerformSegueWithIdentifier` позволяет запретить переход



# NSNotification

- – addObserver(observer: AnyObject,  
                  selector: String,  
                  name:: String,  
                  object: AnyObject?)
- NotificationCenter.defaultCenter.  
          addObserver:self,  
                  selector:"someMethod:",  
                  name:nameOfStation,  
                  object:nil)



# NSNotification

имя станции (`const`)

метод который  
вызывается при  
получении извещения

- – `addObserver(observer: AnyObject,  
selector: String,  
name:: String,  
object: AnyObject?)`

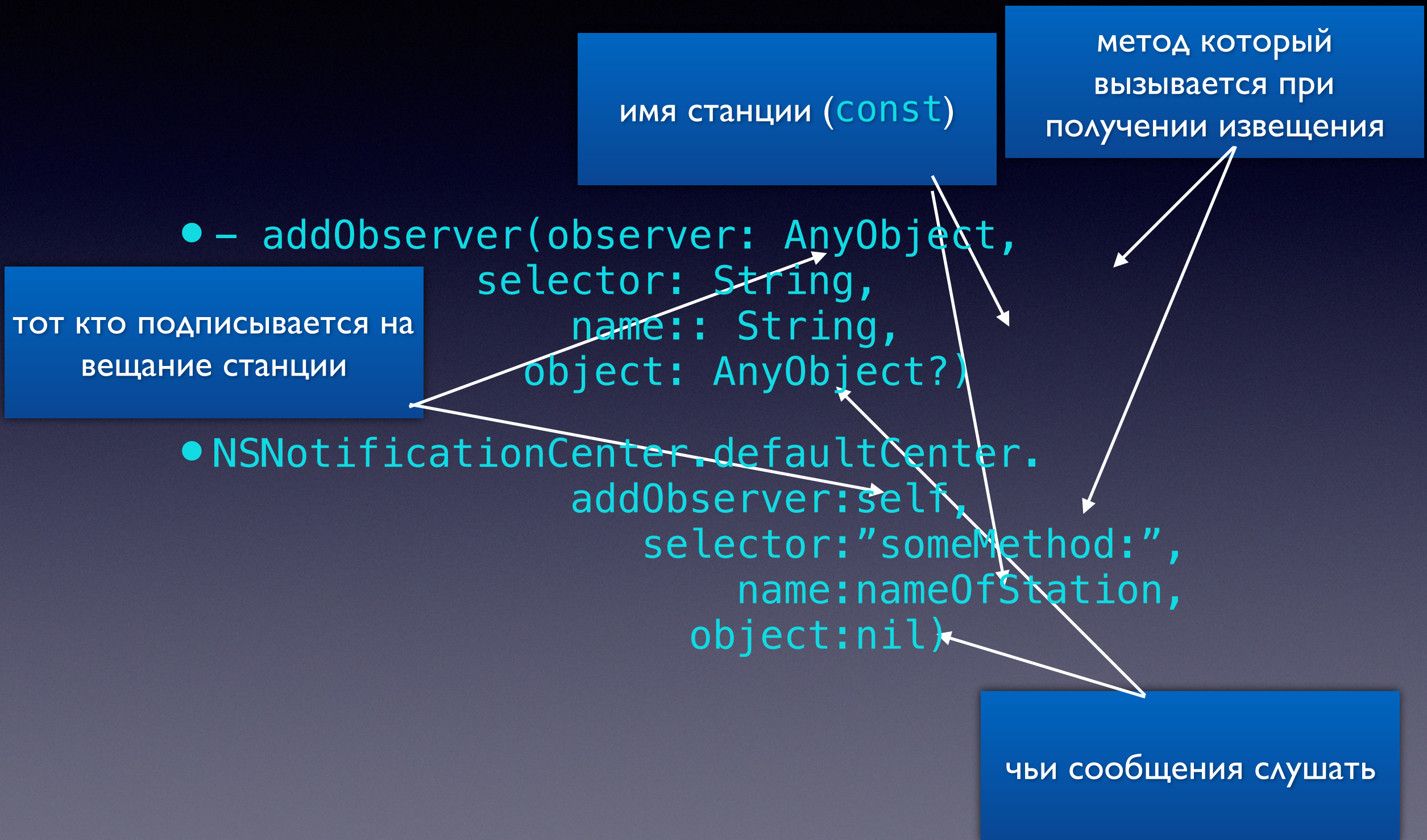
ТОТ КТО ПОДПИСЫВАЕТСЯ НА  
ВЕЩАНИЕ СТАНЦИИ

- `NSNotificationCenter defaultCenter.  
addObserver:self,  
selector:"someMethod:",  
name:nameOfStation,  
object:nil)`

чьи сообщения слушать



# NSNotification





# NSNotification

- отключение от вещателя ОБЯЗАТЕЛЬНО!
- unsafe retain - плохой указатель
- dealloc - последний шанс отключится от вещателя
- `NSNotificationCenter defaultCenter  
 .removeObserver(self)`
- `NSNotificationCenter defaultCenter  
 .removeObserver(self, name:  
 someName, object: nil)`



# NSNotification

- Создать нотификацию можно с помощью следующих методов:
- – `postNotification:`
- – `postNotificationName:object:`
- – `postNotificationName:object:userInfo:`