



Программирование интерфейса приложений

Занятие №2



Дмитрий Тараев

- Отметиться!
- На прошлом занятии:
 - Организационная часть
 - Введение в архитектуру iOS
- **Сегодня**
 - База для написания приложений
 - Крэш-курс по ООП
- Оставить отзыв (после занятия)

Домашние задания - ДЗ 0



- Предложено:
 - 9 идей
 - 10 человек что-то прокомментировало
 - 13 человек заполнило свои
- Проекты:
 - Docker - клиент для удалённого управления Docker контейнерами
 - Instacontrol
 - Запись в различные салоны (парикмахерские, тату, красоты и проч.)
 - BMSTUNavigation - навигация по ГЗ МГТУ
 - HealthyFoodDiaries - индекс массы и выбор плана питания
 - studlib - личная студенческая библиотека с материалами

Домашние задания



1. **ДЗ 0 (срок 11.03.2018 12:00) - максимум 5 баллов**
 - Разбиться на группы, придумать тему, создать репозиторий

2. **ДЗ 1 (срок 18.03.2018 12:00) - максимум 5 баллов**
 - Базовая функциональность

3. **ДЗ 2 (срок 08.04.2018 12:00) - максимум 10 баллов**
 - Использование списков элементов

4. **ДЗ 3 (срок 29.04.2018 12:00) - максимум 15 баллов**
 - Работа с сетью и локальной базой данных

5. **ДЗ 4 (срок 13.05.2018 12:00) - максимум 10 баллов**
 - Работа с дополнительными API
 - карта
 - камера / фотогалерея

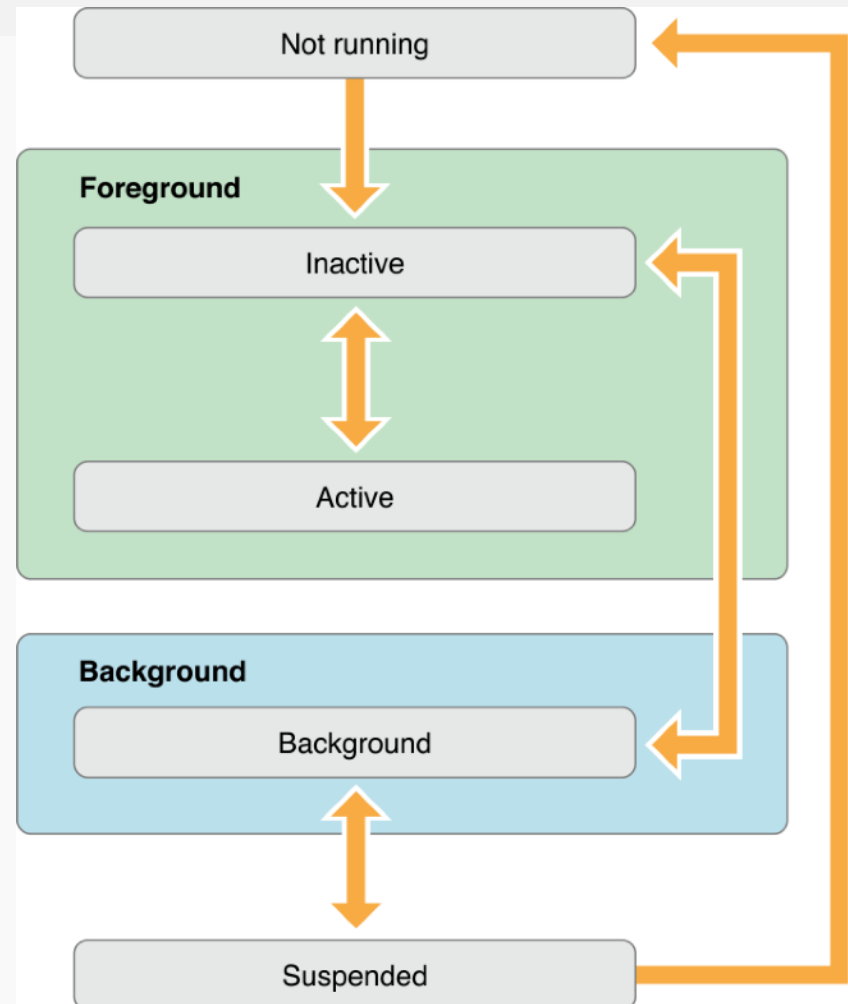
- Жизненный цикл приложения
- Жизненный цикл экрана
- Контейнеры
 - UITabBarController
 - UINavigationController
- View
 - Метки, картинки
 - Поля ввода, кнопки
 - Всё остальное
- Раскладка
 - Storyboard / xib
 - Segues

Жизненный цикл приложения

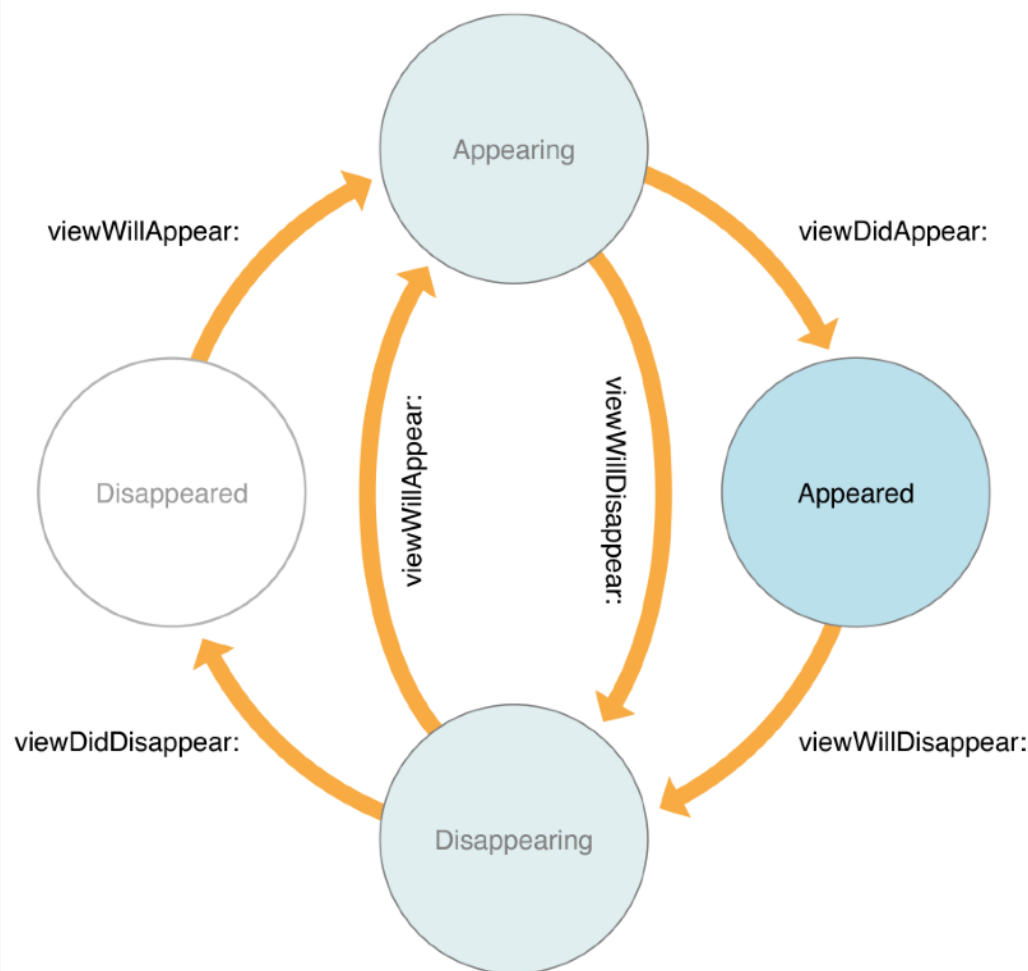


При переходах между состояниями вызываются методы AppDelegate.

Пока нас интересует только `application:didFinishLaunchingWithOptions`



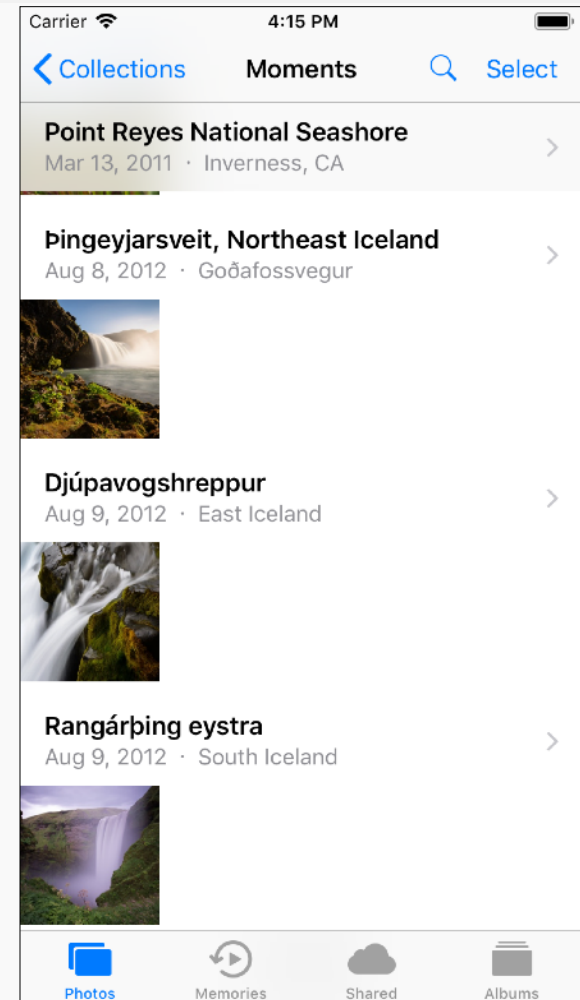
Жизненный цикл экрана (контроллера)



Контейнеры: UITabBarController



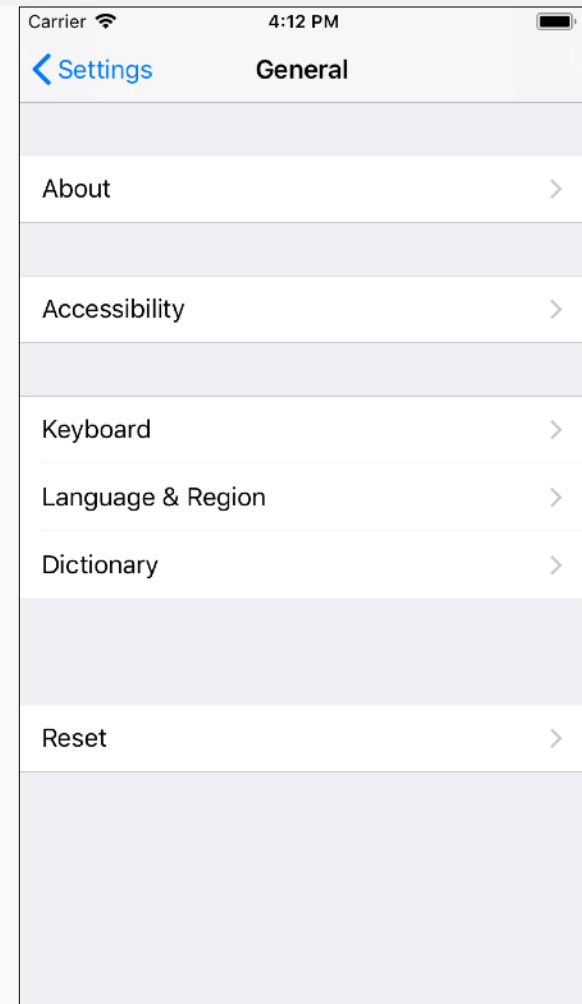
- Переключение между основными разделами приложения
- Пример:



Контейнеры: UINavigationController



- Стек view controllers
- Пожалуй, наиболее часто используемый контейнер
- Пример:



Общие свойства views/controls:

- isHidden
- isEnabled
- alpha
- transform

Views: Статичные views



- UILabel – отображение текста
 - Шрифты, цвета
 - Несколько строк
- UIImageView – отображение картинок
 - .contentMode
 - Highlighted
 - Простые анимации

Views: Поля ввода



- Однострочные
 - UITextField
- Многострочные
 - UITextView
- Типы клавиатур
- Ввод пароля
 - Secure Text Entry (y text field)

- ✓ Default
 - ASCII Capable
 - Numbers and Punctuation
 - URL
 - Number Pad
 - Phone Pad
 - Name Phone Pad
 - E-mail Address
 - Decimal Pad
 - Twitter
 - Web Search
 - ASCII Capable Number Pad

Views: Кнопки (UIButton)



- Тексты
- Картинки,
- Тексты + картинки
- Стили / подложки
- Состояния
 - normal
 - highlighted
 - selected

Views: IBOutlet & IBAction



- Связь Interface Builder'a (IB) и кода
 - т.е. какие части кода доступны IB
- IBOutlet
 - ссылка на какой-то элемент из IB
- IBAction
 - обработчик, который вызывается при определённом действии

- UIView
- UIScrollView
- UIImageView
- UILabel
- UITextField
- UITextView
- UIProgressView
- UISlider
- UIPageControl
- UIStepper
- UISwitch

Сложные view — UITableView, UICollectionView
(наследники UIScrollView)

Будут темой отдельной лекции

Раскладка (layout): Как это работает



- UIView.layoutSubviews
 - Реализация по-умолчанию
 - Как и когда переопределять
- Auto layout

- Линейные неравенства
- Design-time constraints
- Что делать, если система не решается
 - Приоритеты условий
 - Неравенства
 - Система сама решает
- Размеры элементов в системе
 - Hugging и compression resistance
 - hugging resistance - контент не хочет расти
 - compression resistance - контент не хочет сжиматься
 - Чем выше приоритет, тем это сильнее выражено
- Intrinsic size

Storyboards & XIBs: Storyboard



- Pro
 - Наглядное представление переходов
 - Всё в одном месте
 - Меньше кода
- Con
 - Если интерфейс сложный, то быстро забиваются экранами
 - Стало можно разбивать на части
 - Сложнее делать переиспользуемые элементы*

Storyboards & XIBs: Xib



- Pros
 - Проще переиспользовать
 - Проще в командной разработке
- Cons
 - Их становится много, трудно найти нужный
 - Нет возможности задать визуально переходы между экранами

Storyboards & XIBs: Создание кодом



- Сложно
- Многословно
- Иногда приходится

- Что же делать?
- Применять «гибридный» подход:
 - Разбивать сториборды на части
 - Выделить нужные экраны
 - Editor -> Refactor to Storyboard...
 - Что-то реально переиспользуемое держать в xib
 - Или делать вообще кодом

Переходы между экранами в storyboards называются segue

Обработка segues

- `prepareForSegue` – подготовка к segue
- `shouldPerformSegue` — можно ли делать segue?
- `performSegue` — вызов segue из кода

Как передавать данные между контроллерами?

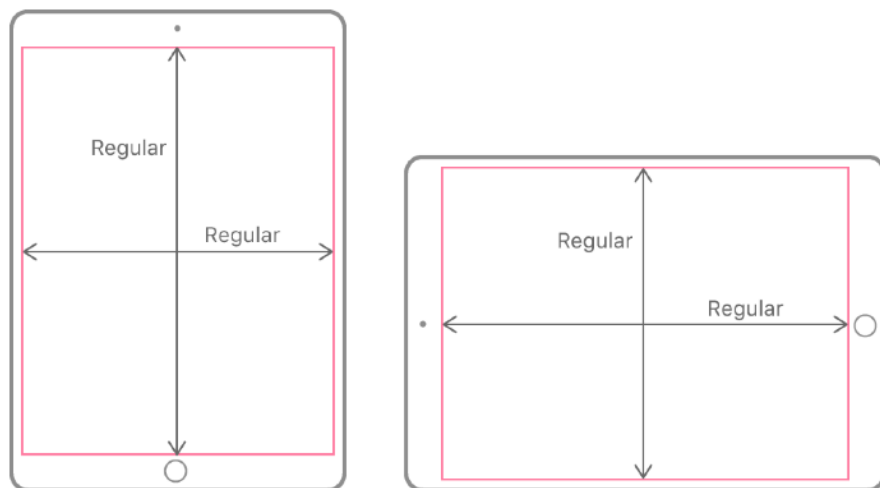
План

- Делаем свойства в том контроллере, который должен получить данные
- В `prepareForSegue` устанавливаем его
 - Или при инициализации этого контроллера для показа руками

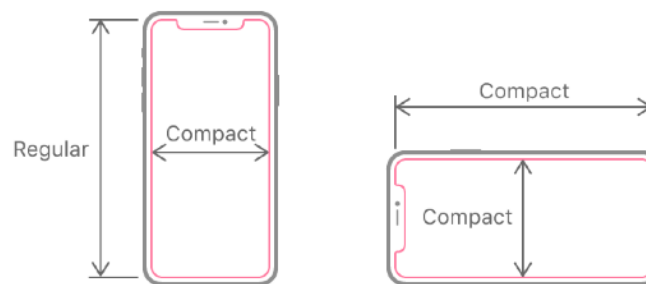
Size classes (адаптивная вёрстка)



- Vertical
 - Compact
 - Regular
- Horizontal
 - Compact
 - Regular



iPad size class



iPhone size class

<https://developer.apple.com/documentation/uikit/uikitcollection>

- Перегрузить функцию
 - override func viewWillTransition(to size: CGSize, with coordinator: UIViewControllerTransitionCoordinator) { ... }

или

- Нужно подписаться на уведомление
 - `NSNotification.Name.UIDeviceOrientationDidChange`
- `UIDevice.current.orientation`

Заключение



- **База для написания приложений**
 - Контейнеры
- **View**
- **Всё остальное**
 - auto layout
 - storyboard / xib / кодом
- **Следующая лекция**
 - MVC и другие архитектурные паттерны
- **Отзыв**
- **Вопросы?**
 - по лекции
 - по домашним заданиям

- Цикл жизни приложения
https://developer.apple.com/library/content/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/TheAppLifeCycle/TheAppLifeCycle.html#//apple_ref/doc/uid/TP40007072-CH2-SW3
- Цикл жизни экрана
http://developer.apple.com/library/ios/#documentation/uikit/reference/UIViewController_Class/Reference/Reference.html
- PureLayout
<https://github.com/PureLayout/PureLayout>
- Masonry / SnapKit
<https://github.com/SnapKit/Masonry>
<https://github.com/SnapKit/SnapKit>
- iOS Human Interface Guidelines
<https://developer.apple.com/ios/human-interface-guidelines/>