

# Программирование интерфейсов

Занятие №2



Евстратов Геннадий

# Привет



- Не забываем отметить
- Поставить отзыв после занятия



- Введение
- Жизненный цикл приложения
- Жизненный цикл экрана
- Контейнеры
  - UITabBarController
  - UINavigationController
- View
  - Метки, картинки
  - Поля ввода, кнопки
  - Всё остальное
- Раскладка

# План лекции

---



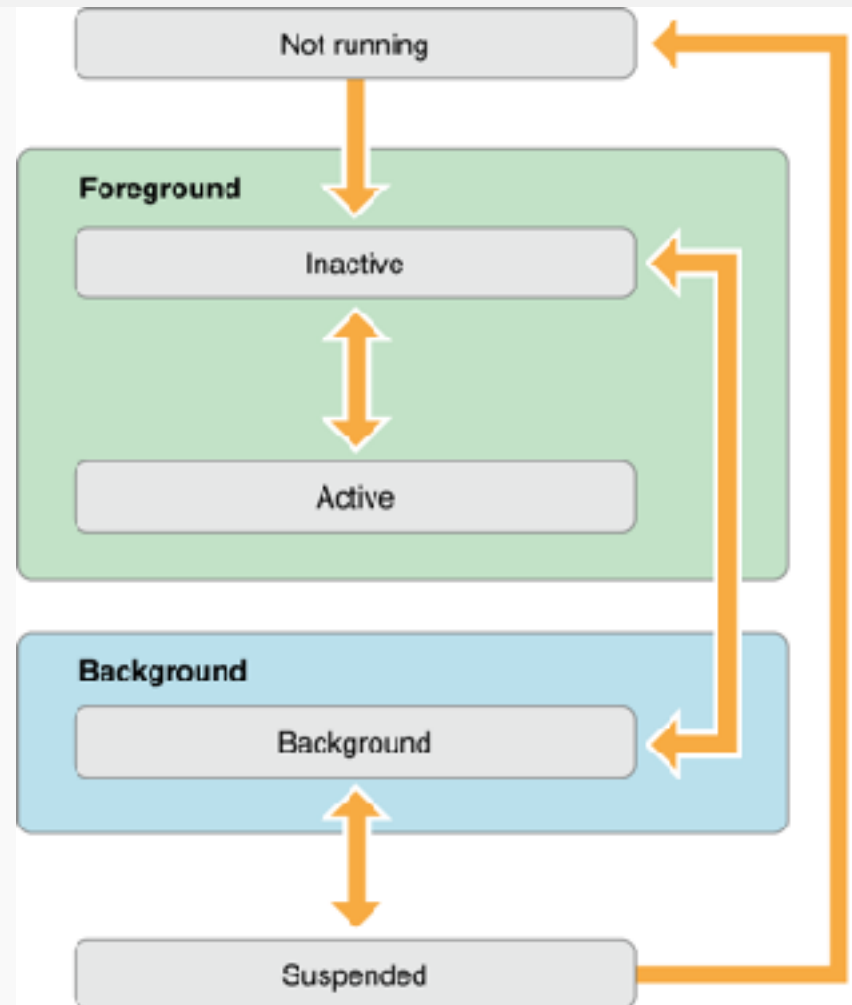
- Storyboard / xib
- Segues

# Жизненный цикл приложения

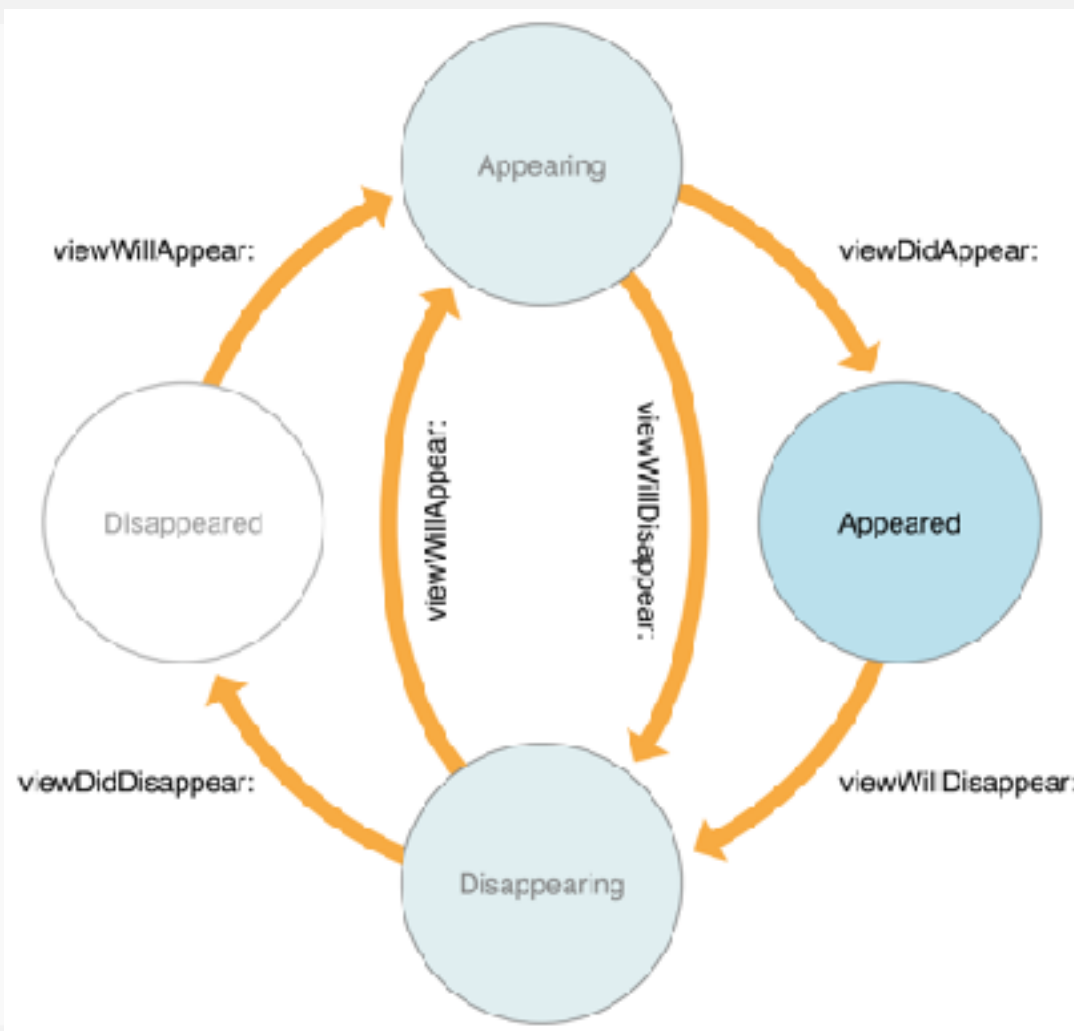


При переходах между состояниями вызываются методы AppDelegate.

Пока нас интересует только `application:didFinishLaunchingWithOptions`



# Жизненный цикл экрана (контроллера)



## UITabBarController

- Переключение между основными разделами приложения
- пример

## UINavigationController

- Стек view controllers
- Пожалуй, наиболее часто используемый контейнер
- пример





## Общие свойства views/controls

- isHidden
- isEnabled
- alpha
- transform

## Статичные views

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

## Поля ввода

- Однострочные
- Многострочные
- Типы клавиатур
- Ввод пароля



## Кнопки (UIButton)

- Тексты, картинки, тексты + картинки
- (секрет!) Всё можно сделать кнопкой



- UIView
- UIScrollView
- UIImageView
- UILabel
- UITextField
- UITextView
- UINavigationBar
- UINavigationController
- UITabBarController
- UIViewController
- UIPickerView
- UIProgressView
- UISlider
- UIPageControl
- UIStepper
- UISwitch

Сложные view — UITableView, UICollectionView, UIScrollView будут темой отдельной лекции



Как это работает

- `UIView.layoutSubviews`
  - Реализация по-умолчанию
  - Как и когда переопределять

## Autolayout

- Линейные неравенства
- Design-time constraints
- Что делать, если система не решается
  - Приоритеты условий
- Размеры элементов в системе
  - Hugging и compression resistance
  - Intrinsic size

## Autolayout

- Линейные неравенства
- Design-time constraints
- Что делать, если система не решается
  - Приоритеты условий
- Размеры элементов в системе
  - Hugging и compression resistance
  - Intrinsic size





## Storyboards

- 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\*



## XIBs

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



## Ручное создание интерфейса

- Сложно
- Многословно
- Иногда приходится

# Storyboards & XIBs

---



Что же делать?

- Разбивать сториборды на части
- Что-то реально переиспользуемое держать в xib
  - Или делать вообще кодом

Переходы между экранами в storyboards называются  
Segue (и не только переходы)

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



- цикл жизни приложения [https://developer.apple.com/library/content/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/TheAppLifeCycle/TheAppLifeCycle.html#//apple\\_ref/doc/uid/TP40007072-CH2-SW3](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](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>