

Objective-C

Курс лекций и семинаров для студентов,
желающих научиться программировать под iPhone

Осень-Зима 2013

Лекция №3

Инструменты: IDE, фреймворки, технологии

Автор: Дмитрий Волков, iPhone Developer, Sibers

Sibers®

Что вы узнаете сегодня?

1. Доступные IDE: xCode и AppCode
2. xCode: Interface Builder
3. Фреймворки и технологии: Cocoa Touch, Media Layer, Core Services, Core OS
4. Ресурсы и документация

Доступные IDE

Xcode



AppCode



Xcode

- ▶ Основная IDE от Apple
- ▶ Является бесплатной
- ▶ Когда-то была простым редактором кода

На данный момент включает в себя:

- ▶ Редактор кода
- ▶ Встроенный браузер документации
- ▶ GUI для систем контроля версий (Git, SVN)
- ▶ Interface Builder

Достоинства и недостатки

Плюсы

- ▶ Бесплатна
- ▶ Apple довольно часто выпускает обновления
- ▶ Так же поддерживает разработку на C / C++
- ▶ Подсветка синтаксиса, автодополнение кода
- ▶ Мощнейшие инструменты для измерения производительности

Минусы

- ▶ Не очень стабильна
- ▶ Новые обновления обычно что-то ломают
- ▶ Поддержка систем контроля версий очень нестабильна

Interface Builder



Interface Builder

- ▶ Основной инструмент создания UI
 - До версии Xcode 4.0 являлся отдельным приложением. В данный момент интегрирован в Xcode.
- ▶ WYSIWYG-редактор (в 90% случаев)
- ▶ Позволяет создавать UI без единой строчки кода
- ▶ Позволяет связывать элементы UI с объектами в коде
- ▶ До версии Xcode 5.0 имел довольно непредсказуемый формат файлов
- ▶ Довольно своенравная поддержка Autolayout

Interface Builder

- ▶ Экраны приложения создаются в визуальном редакторе, который описывает структуру UI посредством XML в файлах с расширением `.xib`.
- ▶ Позволяет как настраивать базовые параметры элементов UI (текст, цвет и тд.), так и задавать динамические параметры, недоступные из редактора.
- ▶ Начиная с Xcode 5 позволяет просматривать то, как будет выглядеть UI на iOS 7 и iOS 6
- ▶ Начиная с iOS 5.0 позволяет создавать не только отдельные экраны, но и описывать путь работы программы посредством нового формата — Storyboard.

Фреймворки и технологии

Cocoa Touch

Media Layer

Core Services

Core OS

Фреймворки и технологии

Сосоа (Mac OS X)	Сосоа Touch (iOS)
Foundation Kit Application Kit Framework	Foundation Kit Framework UIKit Framework Game Kit Framework <i>Game Center</i> iAd Framework MapKit Framework

AppKit – UI фреймворк для разработки приложений для Mac OS

UIKit – UI фреймворк для разработки приложений для iOS

Foundation – фреймворк, содержащий базовые элементы для разработки: коллекции, классы для управления строками, датами, потоками и синхронизацией доступа к данным, а так же многие другие.

Cocoa Touch: Foundation



Cocoa Touch — Foundation

- ▶ Фреймворк, содержащий «модельные» объекты.
- ▶ Семейство NS* классов.
- ▶ Многие классы являются toll-free bridged с «классами» из Core foundation - процедурный C-based API.
- ▶ Содержит как «примитивные» объектные типы — строки, даты, объектные обертки для числовых типов, так и коллекции для управления группами объектов.

Cocoa Touch — Foundation

2 базовых класса, от которых наследуются объекты:

- ▶ *NSObject* — основной базовый класс. Реализует методы создания объектов и логики получения и обработки сообщений.
- ▶ Протокол <*NSObject*> — протокол, реализуемый классом *NSObject*. Определяет методы управления памятью и получения сведений о классе объекта.
- ▶ *NSProxy* — основной базовый класс. Сам не умеет ничего, но интерфейс схож с *NSObject*. Может использоваться для реализации множественного наследования либо перехвата сообщений, посылаемых конкретным объектам.

Cocoa Touch — Foundation

Архитектура фреймворка:

- ▶ Одной из многих особенностей данного фреймворка является повсеместное использование паттерна “Class Cluster” aka “Abstract Factory”.
- ▶ В Foundation для каждой определенной задачи существует некий специализированный приватный класс-наследник, который и выполняет всю работу.

Сосоа Touch — Foundation: Управление памятью

В Foundation и многих других фреймворках iOS и Mac OS приняты следующие соглашения по именованию методов, создающих объекты:

1. Методы классов всегда возвращают *autorelease*-объекты, например:

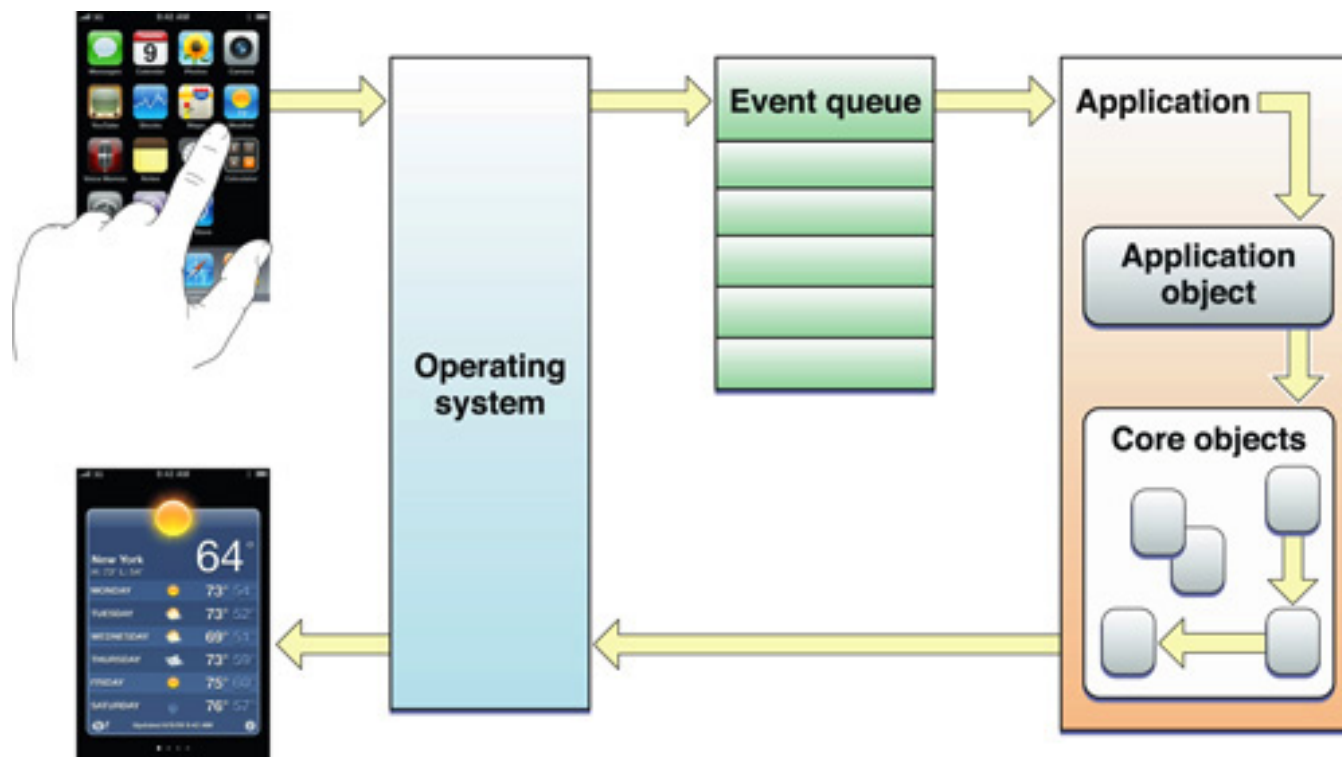
```
[NSArray array]; [NSDate date]; [NSData dataWithBytes:ptr length:42];
```

2. Если для инициализации класса требуется какой-либо объект или начальные условия, создается специализированный *init* метод объекта:

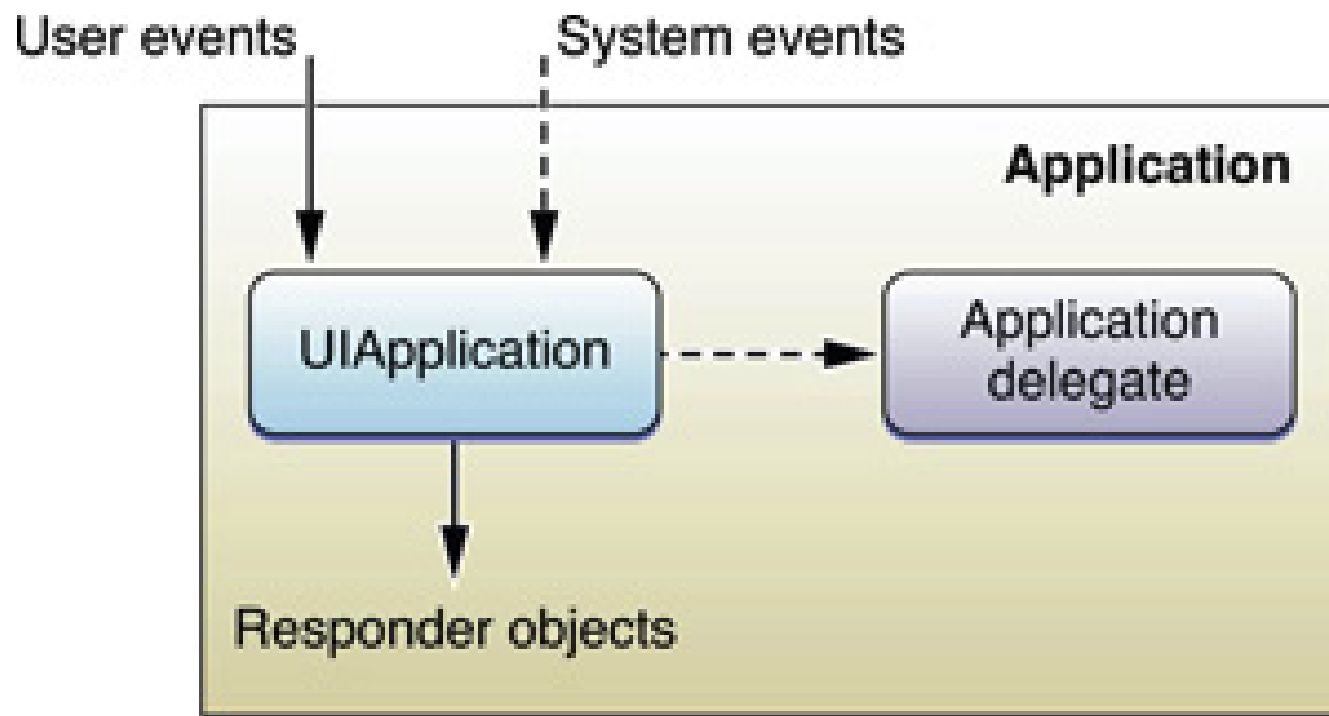
```
[[NSArray alloc] initWithArray:array]; [[NSString alloc] initWithData:data];
```

Объектам, созданным сообщениями *alloc-init* требуется явно послать сообщение *release*.

Сocoa Touch — Foundation: Жизненный цикл приложения

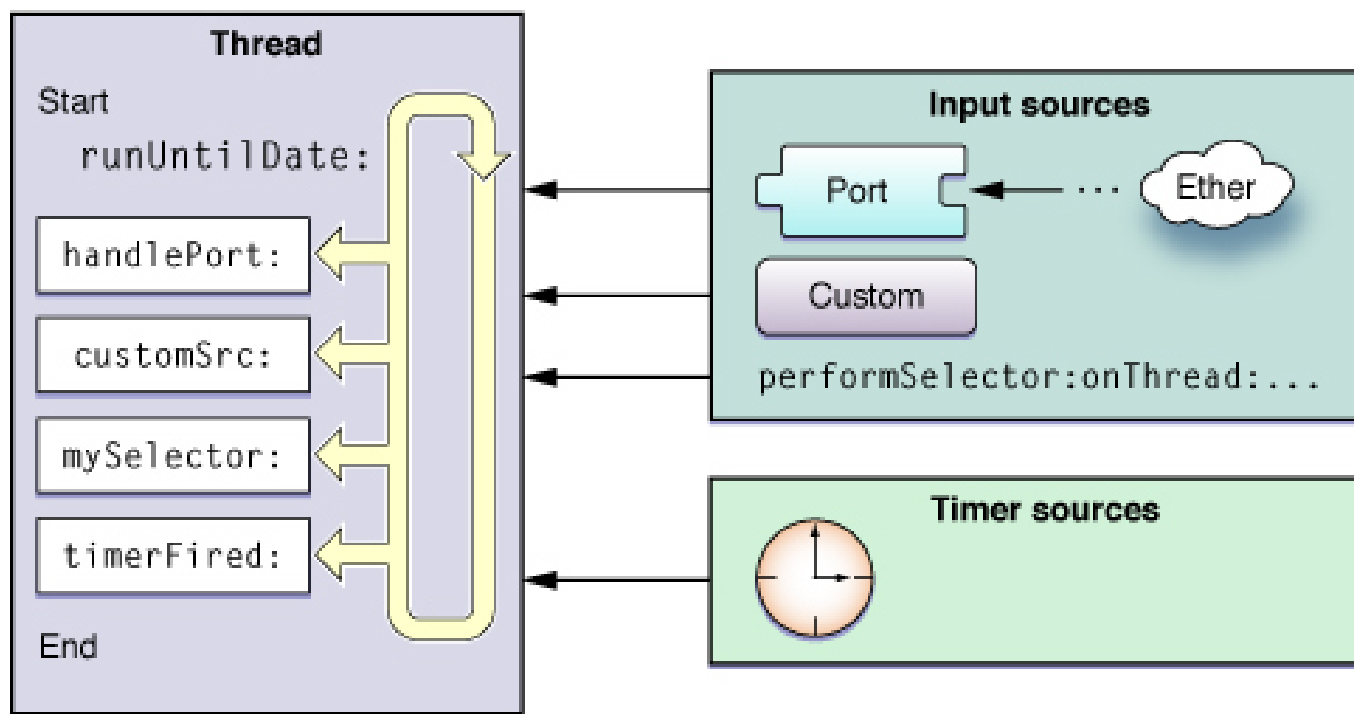


Сocoa Touch — Foundation: Жизненный цикл приложения

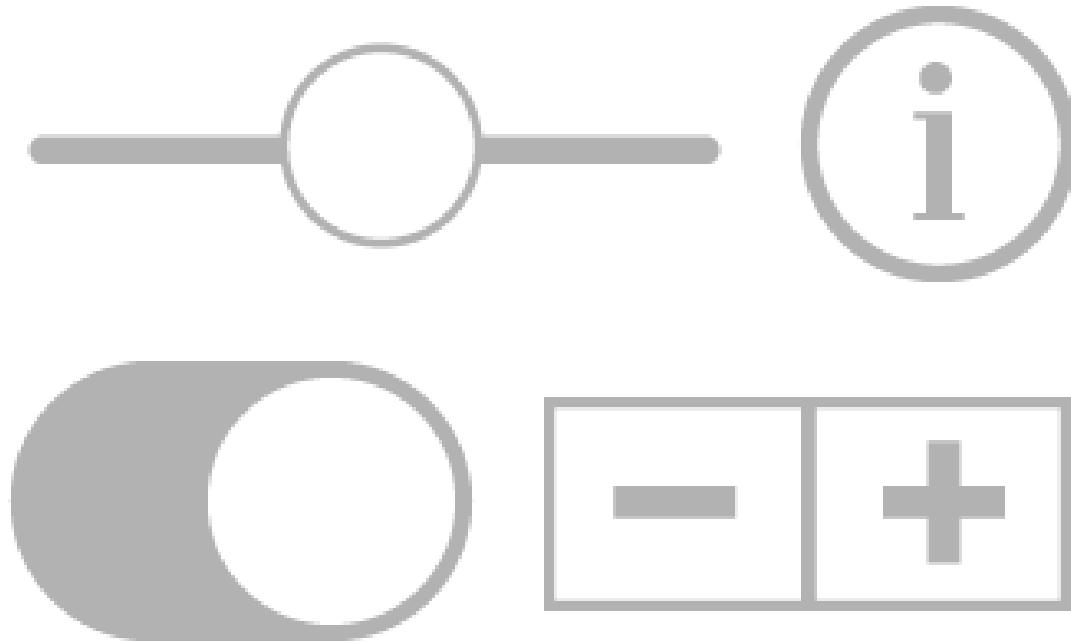


Сocoa Touch – Foundation: Жизненный цикл приложения

- ▶ Приложения в iOS и Mac OS X работают в Run Loop режиме
- ▶ Run Loop – бесконечный цикл, в котором крутится поток в приложении.



Cocoa Touch: UIKit



Cocoa Touch: UIKit

UIKit — основной UI фреймворк iOS.

- ▶ Семейство **UI*** классов.
- ▶ Содержит переиспользуемые и модифицируемые UI компоненты.
- ▶ Содержит множество готовых компонентов, заточенных под определенную задачу.
- ▶ Базовый класс — UIResponder. Обеспечивает обработку тач-сообщений, посылаемых системой.
- ▶ Базовый класс — UIView. Базовый класс, реализующий отображение UI компонентов: рисование, расположение на экране, реакция на действия пользователя. Наследник UIResponder.

Cocoa Touch: UIKit

- ▶ Содержит специализированные объекты - контроллеры.
- ▶ Базовый класс - *UIViewController*.
- ▶ Контроллеры управляют UIView-объектами.

- ▶ Существуют специализированные контроллеры, позволяющие управлять коллекциями контроллеров, а так же UI для доступа к ресурсам системы.

- ▶ Содержит специализированные классы-наследники UIView для управления коллекциями UIView.

Cocoa Touch: UIKit

- ▶ Содержит классы, реализующие логику обработки жестов - *UIGestureRecognizer*.
- ▶ Так же содержит специализированные подклассы *UIGestureRecognizer*, распознающие базовые широкоиспользуемые жесты - pinch / pan / swipe.
- ▶ Содержит расширения и категории для объектов из Foundation Kit для их более удобной интеграции в UI.

Media Layer

Media Layer

Media Layer – уровень абстракции, отвечающий за аудио и графическую составляющие системы.

- ▶ Содержит мощные технологии для работы с аудио, видео, создания анимаций UI и рисования.
- ▶ Как и большинство технологий Apple, имеются объектно-ориентированные фреймворки высокого уровня на Objective-C, так и низкоуровневые API на C.

Media Layer

Аудио:

- ▶ CoreAudio
- ▶ OpenAL
- ▶ AVFoundation

Видео:

- ▶ CoreVideo
- ▶ AVFoundation

Графика:

- ▶ QuartzCore
- ▶ CoreGraphics
- ▶ OpenGL ES

Анимация:

- ▶ CoreAnimation
- ▶ OpenGL ES

Core Services

Core Services

Уровень Core Services предоставляет необходимые сервисы и функции, используемые как системными приложениями, так и приложениями, создаваемыми разработчиками.

- ▶ Предоставляет доступ к аккаунтам пользователей в социальных сетях, адресной книге, информации о телефонной сети.
- ▶ Предоставляет низкоуровневые C API для работы с сетью, базами данных, XML, получения информации о локации пользователя и многое другое.
- ▶ CoreData — фреймворк для управления графами объектов и хранения их в базах данных.

Core OS

Фреймворки и технологии: Core OS

Core OS — низший из доступных разработчикам уровней системы. Здесь находятся системные вызовы FreeBSD и ядра Mach.

Сервисы, все еще предоставляемые системой на данном уровне:

- ▶ Accelerate Framework — фреймворк для обработки цифровых сигналов. Обработка изображений, аудио, преобразования Фурье и много другое. Full C.
- ▶ CoreBluetooth / ExternalAccessories Framework — фреймворки для взаимодействия с внешними устройствами посредством Bluetooth / dock-коннектора.
- ▶ Security — шифрование информации.

Ресурсы и документация

- ▶ <http://developer.apple.com/iphone>
- ▶ <http://www.raywenderlich.com>
- ▶ <http://www.nshipster.com>
- ▶ <http://idevblogs.com/>
- ▶ <http://www.imaladec.com/>
- ▶ <http://www.habrahabr.ru>

Заключение

1. Apple предоставляет множество фреймворков для разработки приложений
2. Благодаря четкому разделению различных уровней API, разработчики могут использовать как высокоуровневные возможности фреймворков и библиотек, так и перейти на более низкий уровень
3. Xcode, Interface builder, Instruments — основные инструменты разработчика для платформы iOS / Mac OSObjective-C — Объектно-ориентированный, предоставляющий богатые возможности для разработки приложений.

В следующей лекции:

Типы данных : Стандартные классы Foundation,
Особенности Исползования, Коллекции объектов.

Спасибо за внимание!

Жду ваших вопросов.