



Вводное занятие

Лекция №1

An arrow originates from the bottom right of the "Лекция №1" text and points towards the top left of the authors' box.

Андрей Решетников, Артур Сардарян,
Александр Каримов

О себе: Артур Сардарян



- Выпускник МГТУ им. Баумана (БМТ1)
- Окончил курс iOS в Технопарке
- Работаю в проекте Юла
- Опыт разработки 2D-игр в SpriteKit (just a little bit)
- Бизнес-приложения на аутсорсе
- Когда-то писал чат-ботов на Python, поднимал всякие бэкенды

Организационная часть



- Отметиться - важно
- О чем пойдет речь в сегодняшнем занятии
 - О себе
 - О целях курса
 - Структура курса
 - Что нужно выполнить в течение курса
 - Требования к проекту
 - Примеры проектов
 - Как будет построен процесс
 - О iOS и разработке под iOS
 - Интерактив
- Оставить отзыв (после занятия)

Содержание



-
- Разработка **неигровых** приложений
 - Правильный код на swift
 - Основы iOS SDK: базовые классы и создание UI
 - Работа с HTTP API
 - Хранение данных
 - Интеграция со сторонними библиотеками
 - И многое другое...

Что нужно выполнить в течении курса



- 3 контрольные (5, 5, 10 баллов)
- 2 контроля проекта (5, 10 баллов)
- Курсовой проект (90 баллов)

Баллы и оценки



Количество баллов	Итоговая оценка студента
До 60	Удовлетворительно
До 80	Хорошо
С 90	Отлично

Часто задаваемые вопросы



- Что делать, если у меня в это время лекции/семинары/лабы/тренировки/надо забрать ребенка из садика/ и тд?
- Что делать, если у меня нет Mac?
- Еще вопросы?

Требования и материалы



- Необходим MacBook с установленным Xcode
 - Виртуалка, хакинтош
- Базовое знание Git
- Знание ООП
- iPhone/iPad, иначе симулятор

Проекты из прошлых выпусков



- Приложение для сервиса каршеринга
- Приложение для записи в салоны красоты
- Путеводитель по МГТУ
- Помощник по упражнениям для фитнеса
- Органайзер для студентов, с расписанием, домашним заданием и пр. (StudentsApp)
- Клиент для сайта 500px
- Клиент для электронной очереди (eQueue)
- Клиент для GitHub (GitLit)
- Приложение для изучения английских слов (RemineMe)
- Приложение-гид по городам со списком достопримечательностей
- Лента Instagram в виде ленты VK
- 2 приложения для чтения книг
- Клиент агрегатора распродаж одежды и обуви
- Приложения для выбора фильма
- и многое другое

Оценивание проекта



- качество проработки деталей,
- качество кода,
- стабильность работы,
- качество презентации проекта на защите

- Проект должен собираться и запускаться без ошибок
 - на всякий случай пробовать заново клонировать из репозитория и удостовериться, что проект собирается и запускается
- Тестовый пользователь с тестовыми данными
- Функциональность
- Дизайн
- Навигация

- Есть знания/опыт
 - опыт iOS разработки на Swift/Objective-C
 - опыт Android разработки
 - опыт в других языках C/C++/Python/Java/JS/
PHP
 - опыт в Pascal
- Мало или нет опыта программирования вообще

Ожидания от курса: Цитаты



- Освоение процесса разработки ios-приложений от начала до релиза
- К сожалению, программа университета сильно устарела и дает лишь базовые знания о технологиях.
- Вообще большие ожидания, хочется попробовать себя в работе в команде по созданию общего проекта. Освоить разработку мобильных приложений, понять как грамотно и правильно делать. Бывает, что вижу некоторые проблемы в приложениях и хочется научиться их исправлять и впоследствии не делать. Мои друзья очень рекомендовали, я давно хотел, а сейчас, послушав отзывы, решил что тоже нужно
- Честно признаться я допустил ошибку и хотел подать на курс "разработка приложений на android", но поскольку я опоздал сменить решение хотел бы обучиться на данном направлении и потому скорее как для общего развития
- Также для меня очень важно знакомство со специалистами, у которых я смогу перенять многолетний опыт, узнать самые новейшие новости в сфере IT и спросить интересующий меня вопрос по направлению курса.
- Поэтому, на курсах технопарка мне бы хотелось получить реальный опыт создания программных продуктов для реальных пользователей(а не для преподавателей, которые принимают лабораторные в университете)

Ожидания от курса: Реалии



- командная работа
- работа над реальным проектом
- упорядочение обрывочных знаний
- углубление знаний

Не будет:

- разработка игр

- По идеям:
 - уже есть проект
 - есть только идея
 - нет даже идеи
 - посмотреть открытые API
 - присоединиться к существующей команде
- По командам
 - есть
 - нет

- Firebase Realtime Database
- Handmade (собственными ручками)
- Public API

Требования к проекту



Общие требования:

- программа должна выполнять все http запросы параллельно, не блокируя основной поток приложения
- полученные данные должны записываться в базу данных
- программа должно работать offline, используя данные из базы
- должны быть доступны ручные обновления данных через pull-to-refresh или кнопку
- должна быть сделана обработка и показ ошибок (сетевые, работа с базой данных)

Требования к архитектуре и качеству кода:

- приветствуется использование MVVM, VIPER, но жестких требований нет - будет оцениваться отдельно дополнительными баллами

Требования к написанию Unit/UI тестов:

- приветствуется, но жестких требований нет - будет оцениваться отдельно дополнительными баллами

Требования к проекту



Требования к UI:

- отдельным экраном должна быть авторизация в сервис (там где требуется)
- основной экран должен содержать табличное представление данных (UITableView / UICollectionView)
- минимальное требование для представления отдельного элемента - картинка (аватар, эмблема), название и краткое описание
- дополнительный экран, открывающийся по нажатию на элемент в основном, должен содержать детальную информацию по выбранному элементу (аватар, название, подробное описание)
- на основном экране должна быть кнопка для перехода настройки, где можно конфигурировать отображение главного экрана - скрывать/показывать картинки и краткое описание

Выбор проекта



- Git

- <https://medium.com/@evgeny.rodionov/гит-это-система-контроля-версий-bd81b15e218d>
- <https://git-scm.com/book/ru/v2> — книга от авторов (рус.)
- <https://try.github.io/levels/1/challenges/1> — интерактивный tutorial (англ.)
- <https://youtu.be/IHyPMbSN5vU> — видео-введение от Технопарка

- swift/obj-c

- <https://github.com/matteocrippa/awesome-swift> — набор ссылок для swift
- <https://github.com/vsouza/awesome-ios> — набор ссылок для obj-c/swift

- 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.html
- про структуру проекта в Xcode: <https://habrahabr.ru/post/261907/>

Ссылки



- Каталоги АПИ
 - <https://www.programmableweb.com/apis/directory>
 - <https://github.com/toddmotto/public-apis>
- Бесплатные клиенты для git
 - Fork (<https://git-fork.com/>)
 - SourceTree (<https://www.sourcetreeapp.com/>)
 - GitKraken (<https://www.gitkraken.com/>)
 - GitUp (<https://gitup.co/>)
 - И консольные tig и magit



- Репозиторий
- Общий чат - <https://t.me/joinchat/EmNHgQx1Rw1GzLytJrhUDg>
- Блог курса

О себе: Андрей Решетников



- iOS разработчик проекта Почта
- Закончил магистратуру МФТИ
- Окончил курс iOS в Технотреке

О себе: Александр Каримов



- Руководитель группы iOS-разработки проекта Почта
- Опыт iOS разработки: 9 лет
- Общий опыт разработки: 15 лет
- В проекте Почта Mail.Ru: 6 лет

Вы научитесь:

- * программировать на языках Swift/Obj-C
- * самостоятельно создавать iOS-приложения
- * познакомитесь с полным циклом разработки iOS приложения - от идеи до выкладки в AppStore
- * приобретете навыки владения всеми инструментами, необходимыми для успешной разработки под iOS/MacOS - XCode, Interface Builder, Profilers, LLDB console
- * делать качественные приложения при помощи Unit/UI тестов

- выпущена в 2007 году, первоначально — для iPhone и iPod touch
- до 24 июня 2010 года — iPhone OS
- разработана на основе операционной системы OS X (с тех пор переименованной в macOS):
 - используется Unix-подобное ядро XNU
 - в основе микроядро Mach
 - BSD часть ядра реализует интерфейсы POSIX
- дистрибуция приложений - магазин приложений App Store, на текущий момент содержит содержит более 1,4 млн приложений
- устанавливается на:
 - iPhone
 - iPad
 - iPod
- обновление выходит каждый год, последняя версия оси - iOS13

- Был сделан в 2005 году
- Сейчас это самая популярная система хранения исходного кода
- Есть масса сервисов, которые предоставляют возможности делать git репозитории
 - GitHub
 - gitlab
 - bitbucket

git — основные понятия



- Репозиторий — набор файлов, которые находятся в системе
- Коммит — атомарный набор изменений в один или несколько файлов
- Ветка — именованное ответвление от основной серии коммитов

git — ветки



Ветки используются для

- параллельной разработки
- сохранения состояния
- упорядочивания кода

Когда мы хотим забрать изменения из нескольких веток в одну, то это называется **мёрж (merge)**

Желание совершить с кем-нибудь мёрж называется **пулл (или мёрж) реквест**

git — пулл реквест



При мёрдже часто возникаю **конфликты**. Это когда в ветках один и тот же файл был изменён в одном и том же месте.

Для того, чтобы завершить слияние эти конфликты надо **разрешить** — выбрать то или иное изменение (или даже как-то их объединить эти изменения)

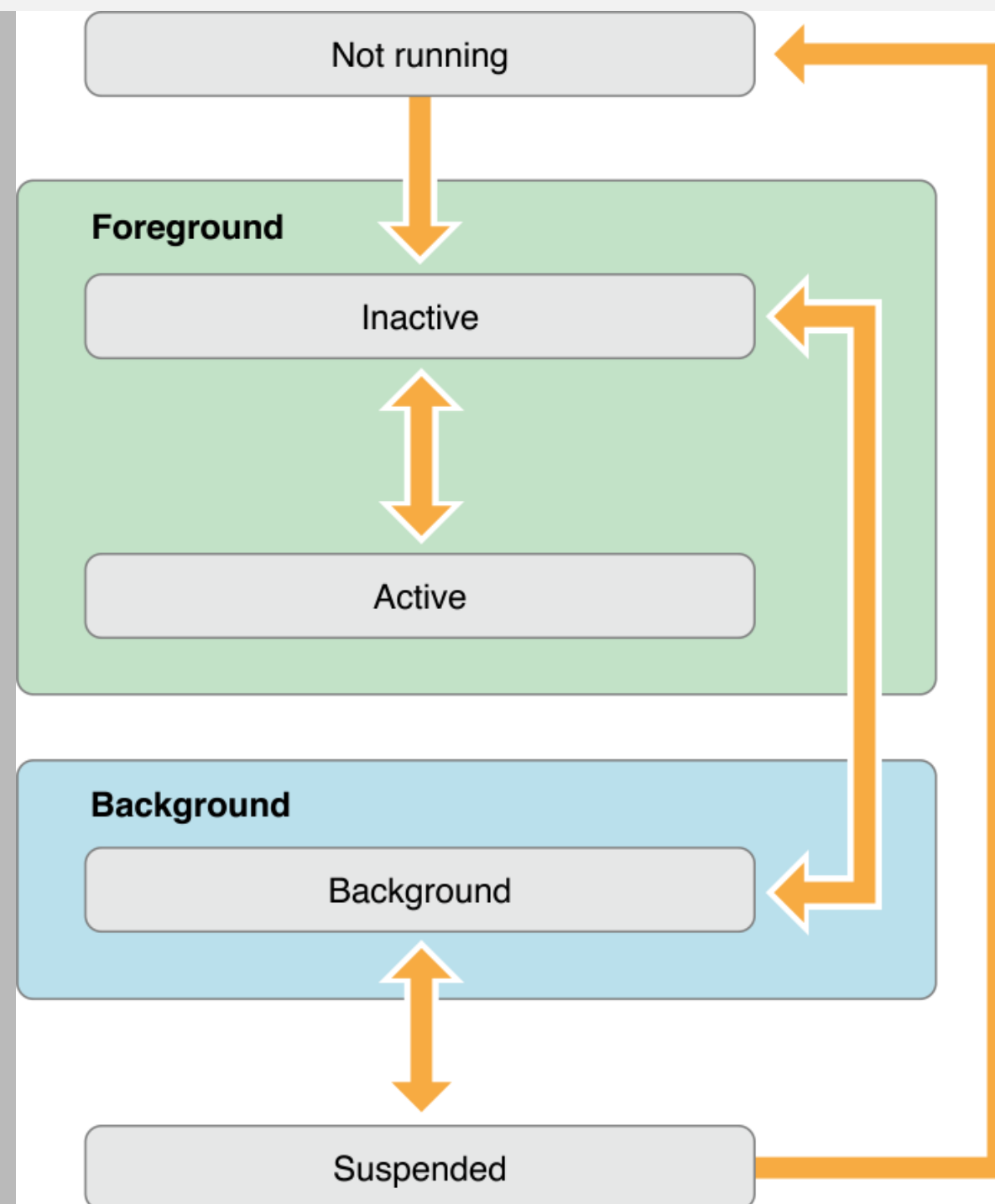
-
- Основная среда разработки
 - Текущая версия 11
 - Содержит в себе кучу всего
 - Редактирование интерфейсов
 - git клиент
 - КОНСОЛЬ

Жизненный цикл приложения



При переходах между состояниями вызываются методы AppDelegate.

Пока нас интересует только `application:didFinishLaunchingWithOptions`



Жизненный цикл экрана (контроллера)

