OTUS

Специальная цена

iOS Developer. Basic

(Обновленный)

Длительность курса: 138 академических часов

1 Язык Swift 5.2

1	**Playground, типы данных, кортежи, опционалы**	написать код в Playground; разбирать основные типы данных.
2	**Коллекции (массивы, словари, множества)**	познакомиться с коллекциями Swift; детально проанализировать особенности коллекций; познакомиться с готовыми методами работы с коллекциями: sorted, count и прочие.
3	**Циклы, ветвление**	объяснить основные методы управления циклом выполнения программы, условия остановки и смены логики дальнейшего выполнения; проанализировать циклы for in, while do, switch, guard else и их применить.
4	**Функции, замыкания**	познакомиться с функциями высшего порядка (замыканиями); работать с функциями; объяснить возможности функции языка; подробно рассмотреть замыкания, их особенности и применение.
5	**TDD, ООП, POP**	проанализировать принципы разработки через тестирование, основы ООП в SWIFT и познакомиться с протоко - ориентированным программированием; рассмотреть ООП, POP; объяснить разницу и особенности;

		проанализировать основы TDD и на практике начнем использовать его в каждодневной работе.
6	**Классы**	объяснить объект class, свойства, методы, инициализаторы; создать экземпляры класса.
7	**Структуры**	объяснить объект struct, свойства, методы, инициализаторы; создать экземпляр структуры.
8	**Перечисления**	объяснить возможности перечислений, виды, особенности.

Домашние задания

1 Написать основные отличия между объектами в ООП

Цель: Написать основные отличия между объектами в ООП. Придумать пример полиморфизма, и реализовать его. Создать очередь выполнения используя замыкания.

1. Используя замыкания, создать последовательность, которая будет с задержкой в секунду, выводить на печать в консоле print("1"), print("2"), print("3"). Для задержки можно использовать функцию sleep(1).

2 Интерфейс

1	**XCode, Storyboard, объекты UI, создание программно объектов, XIB**	познакомиться со средой разработки, средой графического построения интерфейсов, основами визуальных блоков программно и через графический интерфейс; проанализировать основы автопозиционирования объектов; создать контроллер авторизации, все элементы пропишем программно.
2	**TableView, CollectionView**	познакомиться с табличными представлениями; объяснить табличные представления; создать окно друзей на XIB.
3	**Переходы**	объяснить переходы в приложении; объяснить segue, его виды, настройки; создать переход из авторизации в окно друзей; вручную прописать переход в окно друзей из авторизации.
4	**Навигация в приложении**	разобрать основы навигации; объяснить NavigationController, его виды, настройки; создать все контроллеры приложения и связать их навигацией. Домашние задания 1 Реализовать все экраны приложения Цель: Реализовать все экраны приложения. В табличных представлениях использовать динамическую подгрузку. Все шрифты, цвета прописать в расширениях. Все ячейки постороить на XIB. Все должно быть покрыто UITest.

3 Хранение данных

1	**FileManager, UserDefaults**	рассмотреть две формы хранения информации в файловой системе; написать хранение логина и пароля в файловой системе.
2	**CoreData**	объяснить графовую платформу хранения; реализовать хранение друзей в CoreData.
3	**Realm**	познакомиться с библиотекой REALM реализовать хранение новостей в REALM.
4	**Firebase**	познакомиться с облачным хранилищем; сохранить в Firebase данные из личного кабинета.

Домашние задания

1 Реализовать хранение данных приложения

Цель: Реализовать хранение данных приложения следующим образом:
Личные данные хранить в файловой системе.
Данных для коллекций в CoreData, остальное в REALM.
Используя Fairbase добавить возможность удаленно менять цвета фона, отдельных кнопок.

4 Сеть

1	**URL, URLRequest, URLSession, URLSessionDelegate**	рассмотреть основные классы работы с сетью; получить реальные данные из сети, реализуем авторизацию.
2	**API VK, Facebook**	проанализировать основы работы со сторонними API; рассмотреть все необходимые запросы для функционирования приложения.
3	**Создание сетевого слоя, ResultType**	создать свой сетевой слой с безопасным типом данных; создать свою фабрику запросов для всего приложения.
4	**Codable, DynamicJSON**	объяснить основы кодирования данных; использовать данные в разборе ответов сервера и познакомиться с библиотекой DynamicJSON; пасить данные; сохранить получаемые данные в приложении. Домашние задания 1 Создать сетевой слой для приложения, используя API Цель: Создать сетевой слой для приложения, используя API, интегрировать его со слоем хранения. Покрыть все UnitTest.

5 Управление потоками

1	**Thread, RunLoop**	разобрать и научиться работать с потоками и очередями.
2	**GCD**	проанализировать работу с очередями на более высоком уровне, используя GCD; написать плавную загрузку изображения в приложении.
3	**Operation, OperationQueue**	научиться управлять выполнением задач с объектах OparationQueue; создать очередь загрузки после авторизации. Домашние задания 1 Создать очередь загрузки в загрузочном контроллере Цель: Создать очередь загрузки в загрузочном контроллере, добавить плавную загрузку изображений. Все покрыть тестами.

6 Паттерны проектирования

1	**Порождающие паттерны**	познакомиться с порождающими паттернами.
2	**Структурные паттерны**	познакомиться со структурными паттернами.
3	**Поведенческие паттерны**	познакомиться со структурными паттернами. Домашние задания 1 На базе сетевого слоя и базы данных создать фабрику моделей Цель: На базе сетевого слоя и базы данных, создать фабрику моделей. Все покрыть тестами

7 Архитектуры приложений

1	**SOLID, SOA, MVC, MVVM**	проанализировать основы создания архитектуры приложения; переписать авторизацию по патерну MVVM.
2	**Протоколы, делегаты**	рассмотреть протоколы и шаблон делегат в создании архитектуры.
3	**CleanSwift, VIPER**	познакомиться с двумя архитектурами, используемыми в крупных проектах. переписать личный кабинет по патерну CleanSwift. Домашние задания 1 Контроллер авторизации переписать на CleanSwift Цель: Контроллер авторизации переписать на CleanSwift, остальное приложение на MVVM. Покрыть все тестами.

8 Обработка исключений и сбоев

1	**Xcode Instruments**	объяснить готовую среду тестирования приложения; искать утечки памяти, оптимизировать скорость работы приложения.
2	**Fabric, Crashlitics**	объяснить облачные средства аналитики и сбора исключений; подключить Fabric в приложение.

Домашние задания

1 Интегрировать в приложение Fabric, Crashlitics

Цель: Интегрировать в приложение Fabric, Crashlitics. Привести видео работы LEAKS.

9 Combine, SwiftUI

1	**Combine основы**	познакомиться с основами реактивного программирования SWIFT.
2	**SwiftUI**	познакомиться с функциональным созданием приложения.
3	**Окно авторизации на SwiftUI и Combine**	на примерах научиться использовать функциональное и реактивное программирование; создать еще одно окно авторизации на SwiftUI.
4	**Сеть на Combine**	создать очередь запросов на основе Combine; переписать сетевой запрос авторизации на Combine.

10 Проектный модуль. Сдача проектов

1 **Публикация в AppStore**

познакомиться с ITunesConnect, сертификатами, testflight; отправить приложение на публикацию.

Домашние задания

1 Создать отдельный проект

Цель: Создать отдельный проект.
В нем реализовать авторизацию на SwiftUI с учетом архитектуры основного приложения.
Покрыть тестами

Продемонстрировать опубликованный проект.

Проект должен включать:
авторизацию,
личный кабинет,
окно со списком друзей,
окно с личными данными друзей,
окно последних сообщений.

---

2 **Защита проектных работ**

защитить проект и получить рекомендации экспертов.

Домашние задания

1 Сдать ссылку на репозиторий курсового проекта. В репозитории обязательно должен быть заполнен файл Readme.md с описание проекта.