

iOS-разработчик. Базовый курс

Научитесь профессионально разрабатывать iOS-приложения и подготовитесь к собеседованиям в компаниях-партнерах

Длительность курса: 144 академических часа

1 Основы языка Swift

- | | | |
|---|--|---|
| 1 | Обзор по iOS платформе и языку Swift. Начинаем программировать. | <ul style="list-style-type: none">- поймем как можно применить полученные на курсе знания- разберем линейку устройств от Apple под которые можно писать программы- установим IDE- разберем типы данных в Swift- научимся использовать переменные, и выводить значения на консоль. |
|---|--|---|
-

2 Базовые конструкции языка

playground. Типы данных. Коллекции. Управление потоком. Функции.

Познакомимся со основными типами данных, операторами, коллекциями и операторами управления потоком и функциями.

Научимся правильно их использовать.

Узнаем, когда выбирать тот или иной тип данных, коллекцию или оператор.

Решим простую задачу fizzbuzz.

3 Особенности Swift

optional. Result. Перечисления. Кортежи.
Особенности объявления функций. Расширенные возможности операторов if/switch/for.
Познакомимся с ключевыми особенностями языка.
Рассмотрим дополнительные типы данных
перечисление и кортеж. А также линтером.
Научимся писать не в Swift-стиле, узнаем, как
менять стиль в зависимости от задачи. Научимся
правильно работать с опциональными типами.
Узнаем, как строить контракт функции.
Разработаем простой консольный калькулятор.

Домашние задания

1 Консольная программа на закрепление

Напишите консольную программу, которая
принимает от пользователя два параметра:
букву, число.

Вывести в консоль список всех фамилий,
начинающихся на данную букву, если их
владелец имеет пол, соответствующий
введённому числу. Источником является
массив, подготовленный заранее.

2 Консольная программа про работника

Работник работает по сменному графику 2/2.
Дневная смена, ночная смена, выходной,
выходной. Напишите программу,
определяющую соответствие даты и смены
работника.

Пользователь должен ввести в консоли,
какая смена была 1-го января, а затем может
вводить месяц и день для получения
результата.

4 Структуры и классы

структура. Класс. Метод. Свойство.
Наследование. Модификаторы доступа.
Инициализация. Деинициализация.
Познакомимся со своими типами данных, структурами и классами. Их особенностями и устройством.
Научимся правильно их использовать.
Узнаем, когда выбирать структуру, а когда класс.

5 Модель управления памятью

стек. Куча. ARC. GC. Цикл удержания.
Научимся использовать стек и кучу.
Узнаем, как Swift управляет памятью.
Научимся избегать утечек памяти.

Домашние задания

1 Программа, моделирующая фабрику по производству строительных материалов

Напишите программу, которая моделирует фабрику по производству строительных материалов.

Опишите в программе два класса: Человек (ФИО, возраст, паспорт), Паспорт (Серия, Номер, дата выдачи, Человек) и инициализируйте их.

6 Протоколы и расширения

протокол, расширение, диспетчеризация, миксины.
Узнаем о том, как описывать интерфейс типа без реализации и реализацию без типа.
Научимся правильно строить структуру своих типов и добавлять расширенные возможности в стандартные типы.

7 **Generic**

generic-классы, функции и протоколы.

Познакомимся с дженериками.

Узнаем о практиках обобщённого программирования в Swift.

Научимся использовать компилятор для проверки наших утверждений на стадии компиляции.

Домашние задания

- 1 Напишите собственный тип данных стек, подходящий для хранения любых типов
 - 2 Напишите функцию для сложения любых двух числовых аргументов
-

8 **Функциональное программирование**

функторы, монады, map, flatMap, reduce, filter.

Познакомимся с концепцией функционального программирования.

Узнаем об основных инструментах функционального программирования.

Научимся правильно использовать функциональные инструменты, предоставляемые языком.

1 Знакомство с основными компонентами

xcode. UIApplication. UIWindow. UIView. UIViewController. MVC.

Познакомимся с основными компонентами построения приложений под iOS, средой разработки и инструментами управления ресурсами.

Узнаем, что из себя представляет iOS-приложение.

Как происходит его запуск и отображение на экране устройства.

Научимся правильно использовать компоненты и создавать приложения.

Напишем простой пример "hello world".

2 UIView

UIView. UIButton. UIControl. UIResponder. UIControlEvent.

Познакомимся с основой графического интерфейса, классом UIView и его производными.

Узнаем, какие возможности предоставляет нам UIKit для отображения информации.

Научимся создавать свои UI-компоненты.

Напишем простой калькулятор.

Домашние задания

1 Напишите приложение с формой ввода паспортных данных

2 Напиши свой компонент, кнопку с изображением посередине и надписью под ней

3 UINavigationController

UIViewController. UITabBarController.
UINavigationController. UIContainerController-
переходы.

Познакомимся с классом, предназначенным для
управления UIView.

Узнаем, какие виды контроллеров существуют.

Научимся строить навигацию в приложении.

Напишем приложение с несколькими вкладками.

4 View Layout

auto Layout. UIStackView. NSLayoutConstraint. Size
Classes. Intrinsic content size.

Познакомимся со средствами размещения элементов
интерфейса.

Узнаем, какие способы позиционирования выбирать
под разные задачи.

Научимся делать адаптивную верстку.

Напишем приложение с формой ввода
логина/пароля.

Домашние задания

1 Создание Todo List

Написать простой todo list:

1) на обычном контроллере разместите поле
ввода текста и кнопку ""Добавить"";

2) вертикальный стек вью;

3) создайте свой собственный UI элемент —
строку с надписью и кнопкой "Удалить" (справа);

4) по нажатию на кнопку ""Добавить"" текст из
поля ввода переносится, как строка в стек вью;

5) при изменении ориентации экран должен
адекватно реагировать на изменения.

5 UITableView

UITableView. UITableViewCell. UITableViewDataSource. UITableViewDelegate.

Познакомимся с компонентом для отображения таблиц UITableView.

Узнаем, как правильно отображать разные типы данных в одном списке.

Научимся отображать большие списки данных.

Напишем каталог городов с поиском.

6 UICollectionView

UICollectionView. UICollectionViewCell.

UICollectionViewDataSource. UICollectionViewDelegate.

UICollectionViewLayout.

Познакомимся с компонентом для отображения массивов данных UICollectionView.

Узнаем, как создавать свои варианты отображения.

Научимся отображать любые коллекции данных.

Напишем галерею изображений.

Домашние задания

- 1 Написать простой todo list с использованием таблицы

Написать простой todo list с использованием таблицы:

- 1) на обычном контроллере разместите поле ввода текста и кнопку "Добавить";
 - 2) используем таблицу (UITableViewController);
 - 3) по нажатию на кнопку ""Добавить"" текст из поля ввода переносится, как строка в таблицу;
 - 4) используйте стандартную ячейку UITableViewCell;
 - 5) добавьте удаление по свайпу влево (стандартное).
-

7 UIAnimation

UIAnimation. CAAnimation. UIViewPropertyAnimator.

Познакомимся с анимациями.

Узнаем о различных видах анимаций.

Научимся создавать динамические интерфейсы.

Напишем приложение с карточками.

8 UIApplication (уведомления, фоновое выполнение)

UIApplication. Жизненный цикл приложения. Фоновое выполнение, UIApplicationDelegate.

Подробнее узнаем о сером кардинале нашего приложения UIApplication.

Узнаем о жизненном цикле приложения, работе в фоне и других возможностях.

Научимся правильно организовывать код AppDelegate.

Напишем приложение с защитой от несанкционированного доступа.

Домашние задания

- 1 Реализуйте гамбургер-меню с анимацией появления типа "пружина"

Реализуйте гамбургер-меню, с анимацией появления типа "пружина":

- 1) создайте 4 контроллера;
 - 2) разместите на каждом из них кнопку вызова меню. Кнопка должна быть простой по центру контроллеров;
 - 3) при нажатии на кнопку должно появляться меню. Выезжать с пружинной анимацией с левой стороны экрана;
 - 4) меню — это UIView с 4 пунктами меню. Реализация любая (кнопки, кнопки в UINavigationController, UITableView);
 - 5) при выборе пункта мы должны переходить на другой контроллер. Переход выполняется подменой rootViewController у UIWindow. Меню при этом просто пропадает вместе с контроллером;
 - 6) меню должно иметь кнопку "закрыть", по нажатию на которую оно закрывается с пружинной анимацией;
 - 7) если пользователь ничего не выбрал, меню должно закрываться спустя 30 секунд, даже если приложение свернуто.
-

9 **UIStoryboard**

uiStoryboard. UIStoryboardSegue. Xib.

Познакомимся с графическим редактором интерфейса.

Научимся создавать интерфейс без кода.

Узнаем область применения StoryBoard.

Напишем приложение с несколькими вкладками с использованием StoryBoard.

10 **Паттерны навигации**

router. Coordinator.

Познакомимся с паттернами навигации.

Научимся правильно организовывать навигацию в приложении.

Добавим координаторы в приложение с несколькими вкладками.

3 Репозитории данных

- 1 **URLSession** URLSession. URLSessionTask. URLRequest. URLResponse.
Научимся работать с http-запросами.
Узнаем, как отправлять и получать данные на сервер.
-

- 2 **Codable** decodable. Encodable.
Научимся обрабатывать ответ сервера с данными в формате json.
Узнаем, как правильно преобразовывать данный формат в объекты.

Домашние задания

- 1 Загрузка и отображение постов с использованием внешнего API

Загрузить список постов с сервиса <https://jsonplaceholder.typicode.com/posts> и отобразить в таблице.
Реализовать добавление нового поста с помощью api.

- 3 **Типы хранилищ** userDefaults. Keychain. Files. DB.
Познакомимся с основными средствами хранения информации.
Научимся правильно выбирать тип хранилища.
-

4 **Файловая система**

nSFileManager.

Подробнее узнаем о файловой системе и работе с файлами.

Научимся сохранять данные в файлах.

Домашние задания

- 1 Загрузить список фотографий с внешнего API и отобразить их

Загрузить список фотографий по url <https://jsonplaceholder.typicode.com/photos> и отобразить в таблице.

Реализовать кэширование изображений в файловой системе.

5 **Введение в CoreData**

core Data Stack. NSManagedObjectContext.

NSManagedObjectContext.

NSPersistentStoreCoordinator.

Узнаем о стандартном ORM от Apple для работе с sqlite.

Научимся проектировать модель хранилища, сохранять и получать объекты.

6 Offline first with CoreData

nSFetchResultsController.

Узнаем о подходе offlineFirst.

Научимся подстраивать интерфейс под данные в базе.

Домашние задания

- 1 Внедрить подход offline first в приложение с цитатами

Внедрить подход offline first в приложение с цитатами:

- 1) загрузить список постов с сервиса <https://jsonplaceholder.typicode.com/posts>;
 - 2) сохраните его в CoreData;
 - 3) организуйте показ постов из CoreData;
 - 4) при обновлении данных в CoreData таблица должна обновляться.
-

7 Продвинутые возможности CoreData

nSManagedObjectContext. Query Generations.

Познакомимся с вариантами настройки стека для работы с большими объемами данных, узнаем, как работать с неизменяемой версией хранилища.

- | | | |
|---|---------------------------------|---|
| 1 | Внедрение зависимостей | storyboard. Swinject. Factory.
Познакомимся с одним из основных способов уменьшения связанности.
Научимся гибко собирать классы в приложении.
Напишем список книг с возможностью добавить отзыв. |
| 2 | Тестирование | xCTest. URLProtocol. GTW. Expectations.
Познакомимся с фреймворком для написания автотестов XCTest.
Научимся покрывать бизнес-логику автотестами.
Протестируем приложение списка книг. |
| 3 | Многопоточность — Thread | thread. DispatchQoS.
Познакомимся с потоками.
Научимся выполнять код параллельно. |
| 4 | Многопоточность — GCD | dispatch Queues. Dispatch Groups. DispatchWorkItem. DispatchWorkItemFlags.
Познакомимся с инструментом абстракции от потоков GCD, научимся делить код на задачи и оптимизировать время их выполнения. |

Домашние задания

1 Задача на многопоточность

- 1) напишите потокобезопасный контейнер, для хранения любого типа (Int, String, Double);
- 2) создайте массив чисел от 0 до 9;
- 3) возведите каждый элемент массива в квадрат в отдельной параллельной задаче;
- 4) сложите результат.

5 **Многопоточность** **— Operation**

operation.

Познакомимся с инструментом абстракции от потоков Operation, научимся делить код на задачи и оптимизировать время их выполнения.

5 Swift UI + Combine

- 1 SwiftUI**
 - разберем устройство SwiftUI
 - научимся строить сложные интерфейсы при помощи SwiftUI
 - научимся интегрировать SwiftUI в существующее приложение и наоборот

- 2 Combine**
 - основы реактивного программирования
 - как реактивное программирование помогает нам в решении повседневных задач

- 3 Swift UI + Combine**
 - разберем как используя SwiftUI и Combine создавать сложные, production-ready приложения

1 **Работа над проектом: постановка задач**

2 **Консультация по проекту**

3 **Защита проектов**

Домашние задания

1 Сдача проекта

Цель: Сдать ссылку на репозиторий курсового проекта. В репозитории обязательно должен быть заполнен файл Readme.md с описанием проекта.

Необходимо сдать и защитить выпускной проект.