

Привет, друзья! Перед новым годом я хочу сделать небольшой подарок всем, кто только делает первые шаги в замечательной профессии iOS разработчика :)

В первую очередь я хочу от всей души поблагодарить Алексея Скутаренко за его труд. Сложно представить, насколько длиннее был бы мой путь от начала изучения языка до устройства на работу, не найди я в свое время его курсы. Уроки с домашними заданиями - это, безусловно, первое, что нужно сделать всем будущим разработчикам! Но после их окончания у меня, как человека, никогда не работавшего в этой сфере, все равно оставалась масса вопросов именно про устройство на работу. Думаю они есть и у многих из вас. Время от времени находя ответы на свои вопросы, я собирал их в одном месте, и, вспоминая, как долго и трудно их было искать, не имея знакомых в этой сфере, я хочу надеяться, что кому-то из вас будет легче на данном этапе после прочтения топики.

Для удобства, я разделю вопрос устройства на работу на следующие вопросы, пусть и не по порядку:

- Вопросы на собеседование
- Тестовые задания
- FAQ по общим вопросам

Вопросы на собеседование.

Собеседование/техническое интервью - этап, который был и будет всегда, и который по уровню стресса, пожалуй, самый большой в этом списке. Никто никогда не знает, о чем спросят на собеседовании, но поверьте на слово, все темы, которые будут озвучены в списках вопросов вам пригодятся. Рано или поздно, но пригодятся, поэтому изучить, ознакомиться с ними сейчас - совсем не рано, если вам это интересно и вы чувствуете, что готовы. Если вы почитаете комментарии топиков с вопросами ниже - увидите, что абсолютно на все вопросы мало кто ответит даже из уже работающих в этой сфере людей, а причина этому проста: у всех своя специфика, сложно представить место работы, где один человек занимался бы сразу всем, на что способны платформа и язык. Поэтому не бойтесь чего-то не знать сейчас, бойтесь не попробовать и не узнать! Я бы сравнил эти вопросы с экзаменом в институте, когда вопросы выдаются студентам как отправную точку для подготовки. Многое из этого вы уже проходили, что-то гуглится за пару минут, а что-то просто полезно будет усвоить. Просто садитесь и разбирайте по несколько вопросов, когда выдастся свободная минутка, это самый лучший способ, который увеличит ваши шансы трудоустройства в несколько раз.

Вопросы в основном теоретические, что-то есть и по практике, но обычно практика это тестовые задания. Ответы вместе с вопросом писать было бы только во вред вам, потому что при вашем индивидуальном поиске вы найдете гораздо больше и актуальнее информации, чем просто прочитав ответ, поверьте. Да и к тому же поиск ответов - это повседневная задача разработчика, поэтому тренируйтесь и привыкайте :)

Самое главное, что я хочу вам сказать - поймите, что даже несмотря на то, что курс уже пройден - вы все еще учитесь, более того, вы все еще в начале большого пути. Поэтому будьте готовы узнавать и учиться новому всегда, каждый день, каждую минуту, каждый вопрос из списков ниже, если придется.

Ну а для базы вопросов, вот ссылки из просторов интернета и наших участников группы:

1. <https://gist.github.com/arthurigberdin/3a078212ee3f347a4ff5>
2. https://vk.com/wall-58860049_52158
3. https://vk.com/topic-58860049_32550138

Тестовые задания

«Уроки уроками, но что же реально просят сделать компании?», - думал я. А реально - это же и просят, друзья! Сделав домашние задания всех уроков по всем уровням, можете считать, что тестовое задание у вас в кармане, ведь вы с легкостью, в спокойной обстановке, сядете и выполните его, также как и 40 с лишним раз до этого :) Самое главное - пишите красиво и качественно, как только вы умеете. Ведь задания нужны в первую очередь, чтобы посмотреть на ваш код. Ну и ниже несколько тестовых заданий, которые накопились у меня, для тех кто не уверен в своих силах и хочет потренироваться. Уровень сложности у них конечно отличается, может это связано со срочностью закрытия вакансии, может с годом поиска, может с уровнем компании, не знаю. Но требуемые технологии очень похожи. Часть заданий взята из этой группы, к сожалению ссылок на темы не сохранилось:

Задание 1. Таблица сотрудников.

- Необходимо создать приложение для iPhone, основанное на UINavigationController;
- На главном ViewController'e разместить таблицу (UITableView), кол-во строк на ваш выбор. Каждая строка должна содержать название вида «Поручение № n», где n-номер строки, и описание поручения вида «Описание поручения № n»;
- Сделать обработку нажатия на каждую строку таблицы, после которого будет происходить анимированный переход на другой ViewController, где будет отображаться заголовок взятый из названия строки по которой мы перешли и описание, в рабочей области ViewController. Так же во втором ViewController нужно предусмотреть кнопку Назад, для возврата обратно, в список. Кнопка должна являться частью UINavigationController (UINavigationControllerItem);

Огромным преимуществом будет, если у вас получится разработать данное приложение без использования Interface Builder.

Комментарий: Задание по кор дате было труднее, правда ведь?

Задание 2. Лекарства из кор даты.

Нужно сделать следующее:

- Подключить к проекту базу SQLite, данные брать из базы.
- Нужно выводить список препаратов сортированных по названию с форматированными свойствами и мелкой картинкой;
- При тапе на ячейку препарата, падаем в список его свойств, сначала большую картинку а также, выводим название свойства + текст свойства.

Комментарий: По сути тоже самое, только в с другими данными

Задание 3. Мобильный клиент для GitHub

В качестве тестового задания надо написать несложный клиент для сервиса GitHub с использованием REST API (<http://developer.github.com/v3/>)

Необходимые функции:

- Авторизация по логину/паролю: при запуске приложения пользователь должен ввести свой логин и пароль от GitHub. После успешной авторизации логин и пароль должны запоминаться и использоваться при последующих запусках. (Авторизация через OAuth будет плюсом).
- Список репозиториев авторизованного пользователя (в табличном виде)
- Общая информация о репозитории: название, описание, автор, аватарка автора, количество forks и watches
- Список коммитов в репозитории в табличном виде: hash, короткий commit message, автор, дата

Приложение должно работать на устройствах под управлением iOS4+, соответствовать HIG и не иметь в коде проблем, препятствующих выкладке в AppStore (использование

закрытых API, утечек, крашей, зависаний и т.д.). Можно использовать любые сторонние библиотеки (если они сами не имеют вышеупомянутых проблем). При отсутствии сети или ошибках сервера пользователь должен получать внятное сообщение об ошибке. Локальное кеширование данных не обязательно, но будет плюсом. Структура навигации, дизайн экранов, и дополнительные функции остаются на усмотрение разработчика.

Комментарий: а вот и ДЗ по API, осталось лишь изучить другую документацию (которая тоже очень хороша)

Задание 4. Мобильный клиент flickr

Надо реализовать приложение для просмотра картинок из Flickr Public Feed.

Программный интерфейс Flickr Public Feed описан тут:

https://www.flickr.com/services/feeds/docs/photos_public

Страница представляет из себя список фотографий с датой создания, названием и списком тегов.

Должна быть возможность задавать список тегов для поиска фотографий, сортировать фотографии по дате создания и названию.

При нажатии на фото оно должно открываться на весь экран.

Комментарий: просто еще одно новое API, от предыдущего задания отличается разве что работой с фото

Задание 5. Вывод видеоизображения камеры

Нужно сделать простое приложение, которое по нажатию кнопки начинает вывод видеоизображения из камеры.

Комментарий: тут в группе уже смеялись на тему данного задания, т.к. составлено оно не совсем корректно, и сделать на самом деле просят с использованием AVFoundation, хотя именно то, что написали, очень легко реализовать и без этого фреймворка :) Я бы не отнес это задание совсем к начальному уровню, но хочу сказать, что если вам интересна тема работы с аудио/видео - то изучить необходимую информацию с нуля и выполнить требуемое реально за пару дней (утверждаю это, исходя из своего опыта)

Задание 6. Работа со строками

Для iOS подразумевается выполнение на C/C++ или ObjC.

Во входном файле input.txt в первой строке через пробел записаны два целых числа $M, N \in [0, 10000]$. В следующих M строках записано по N символов - матрица размера $M \times N$ (формат см. в примере). Необходимо написать консольное приложение, выводящее символы из матрицы, из которых составляется строка "HelloWorld" (регистр символа при составлении слова значения не имеет), и их индексы в соответствующем порядке в столбик через точку с запятой в следующем виде: "СИМВОЛ - (m, n);" либо вывести "Impossible", если это невозможно сделать. Один элемент матрицы при составлении строки можно использовать только один раз. Корректность входных данных гарантируется.

Пример:

input.txt:

```
3 4
nOe1
```

Two2

Trip

output.txt (один из возможных):

```
o - (1, 2);  
n - (0, 0);  
e - (0, 2);  
T - (1, 0);  
w - (1, 1);  
O - (0, 1);  
T - (2, 0);  
r - (2, 1);  
i - (2, 2);  
p - (2, 3);
```

Комментарий: Наверное по ключевому слову вы уже догадались что за компания, довольно известная, задание интересное, задание выполнимое. Заодно повторите работу со строками :)

Задание 7. Тестовое задание для школы мобильной разработки Яндекс

Приложение, отображающее дерево категорий товаров и услуг Яндекс.Денег

Задачи

Запрашивать дерево категорий товаров и услуг по адресу <https://money.yandex.ru/api/categories-list>

Сохранять в SQLite или другую локальную реляционную базу данных

Отображать дерево на экране устройства

При повторном запуске приложения дерево категорий товаров и услуг должно

загружаться из локальной базы данных

Иметь UI-контрол инициирующий повторную загрузку и сохранение дерева с сервера

Комментарий: CoreData + API , соедините два ДЗ, и вы уже достойны школы яндекса

Задание 8. Переводчик (Core Data + Rest API)

Тестовое задание на позицию Junior iOS Developer.

1. Ознакомьтесь с текстом ТЗ и оцените время, которое вам необходимо на написание приложения. Оценку необходимо составить в гугл доке, с указанием в часах (сколько вам необходимо на создание тех или иных компонентов). После чего, приступайте к заданию.

2. Проект необходимо разместить на бесплатном гит репозитории (можно гитхаб).

3. Выслать оценку времени нам на ознакомление и доступ к гит репозиторию (с правом просмотра) и приступать к выполнению.

4. Минимальная версия iOS 7.0

5. Тема приложения темная с NavigationBar.

6. Словарь - приложение из нескольких экранов.

При входе в приложение открывается экран со списком ранее переводимых слов и полем для поиска слова.

При вводе буквы в строку поиска необходимо обновлять список из локальной базы данных (оставлять в нем только те слова, которые соответствуют введенному тексту).

Если в списке нет элементов, то необходимо провести поиск при помощи любого бесплатного сервиса переводчика, к примеру (<http://api.mymemory.translated.net/get?q=HelloWorld..lit>). Положительный ответ с сервиса необходимо положить в локальную БД и

обновить список. Если поиск на сервисе не дал результата - уведомить пользователя об этом.

В `NavigationBar` показывать бесконечный прогресс во время выполнения запроса и сохранения ответа.

При клике на слово в списке - открывается окно где указывается какое слово переводится и какой перевод. (Сделать кастомную кнопку для возврата в предыдущее окно в `NavigationBar`)

Язык перевода определять автоматически, если ввод латиницы - значит с `en`->`ru`, если ввод кириллицы, то с `ru` на `en`.

Использовать `AFNetworking` + `RestKit` + `MagicalRecord` + `Autolayout`.

7. Укажите финальное время, потраченное на реализацию задачи.

Комментарий: Отличное ТЗ, подробное и четкое. Наверное, самое серьезное и долгое задание из этого списка, но про то, что здесь опять в основном работа с `Core Data` + `API` наверное даже говорить не стоит. Ну и немного фантазии с кастомизацией.

Еще раз поясню насчет заданий, друзья. Грань между «Нереально» и «Элементарно» очень узкая, поэтому если задания кажутся сложными - значит вы просто не усвоили материал/не выполнили все ДЗ по урокам курса. Просто повторите еще раз уроки по БД и работе с сетью (строкам, таблицам, навигации, поиску, _вставьте нужное сами_). (исключение только задание 5, про которое я сказал в комментарии)

FAQ по разным вопросам

Q: Стоит ли учить столько сухой теории, ведь разработка это практика?

A: Стоит! Теория это фундамент в первую очередь. Она помогает выстроить понимание работы платформы, языка, вообще компьютеров в целом. Имея представление обо всем этом вы будете допускать гораздо меньше ошибок во время практики и не задумываясь выбирать правильные решения. Поэтому теория не только фундамент, но и инвестиция. Инвестиция в более легкий и прозрачный путь вашего развития в будущем.

Q: Сколько времени дается на тестовое задание?

A: Везде по-разному, в среднем от двух дней до недели.

Q: Сколько времени ждать ответа после отправки тестового задания?

A: Также, от дня до недели. Поймите, что у разработчиков кроме просмотра вашего тестового задания есть еще и его реальное, у которого есть срок и которое нужно выполнять, поэтому если не отвечают сразу - запаситесь терпением и подождите чуть дольше.

Q: Комментарии в `Objective-C`? Нужны ли они? На каком языке?

A: Нужны, но в меру. Как вы уже заметили, `Objective-C` сам по себе говорящий язык, с его названиями методов и переменных, из которых уже можно все понять. Просто старайтесь писать говорящие названия для своих классов, переменных, методов, и тогда комментировать вам останется только те участки, где есть сложная логика или прочие неочевидные моменты/решения. В идеале, на английском языке. Да, и названия ваших переменных тоже на английском. Учите, если не знаете :)

Q: А зачем мне английский, если я не собираюсь работать в англоговорящих компаниях?

A: Хорошая попытка, но нет, английский для разработки нужен. По началу вы вполне обойдетесь и без него, но со временем нужные ресурсы находить будет сложнее и сложнее, тогда как на английском их гораздо, гораздо больше! Вы буквально откроете для себя новые горизонты, новые материки источников знаний, поэтому хоть немного, но уделяйте время на его изучение тоже.

Q: А почему Objective-C, ведь да дворе Swift 2.1?

A: Да, Swift крут, он в тренде, на нем все конференции и tutorиалы. Учите его. Ведь рано или поздно, но познакомиться с ним придется. Тоже самое и в обратную сторону: и Objective-C изучить придется, если вы начали со Swift, потому что все еще очень много реальных проектов написано на Objective-C, а чтобы их поддерживать - вам придется знать их язык.

Q: Глядя на мой код, разработчики буду смеяться над его архитектурой, как ее улучшить?

A: Не будут, над кодом смеяться грешно! :) Все ведь когда-то писали также, не волнуйтесь. Главное показать в коде ваши накопленные знания, не стоит пытаться объять сразу всё. Ну а если вас все-таки волнует архитектура, изучите книгу, находящуюся в свободном доступе «Хрестоматия iOS паттернов. На всякий...». В ней на языке Objective-C кратко описаны с примерами все основные шаблоны проектирования, я в свое время был очень рад этой находке, надеюсь порадуетесь и вы :)

Q: В моем городе совсем нет вакансий iOS разработчиков, особенно уровня junior, но предложили удаленку/фриланс. Стоит ли соглашаться?

A: Если говорить о продуктивности вашего развития - нет, не стоит. Лучше работать за «копейки», но в офисе, и, что важно, в команде с более опытными разработчиками. Поверьте мне, за неделю работы со старшими вы узнаете больше, чем за месяц(/месяцы) фриланса. Я повторю вам в очередной раз: на данном этапе опыт для вас куда более ценен, чем деньги. По поводу отсутствия вакансий: очень продуктивный метод описал в группе один из участников, вот ознакомьтесь: https://vk.com/topic-58860049_32464193 Смысл в том, что не все работодатели размещают резюме именно для поиска junior'ов и их выращивания. Безусловно, больше рады будут человеку с опытом за спиной. Но это совсем не значит, что все полностью отказываются от варианта приема сотрудника без опыта. Рассылайте резюме везде, где вы примерно подходите по требованиям, не обращая внимания на требуемый опыт, и вам обязательно кто-нибудь ответит.

Q: Раз уж зашла речь о деньгах.. Я совершенно не знаю себе цены на рынке труда. Боюсь продешевить, сколько ставить зарплату в резюме?

A: Думаю, не стоит объяснять, что все мы находимся в разных городах с разным уровнем зарплат. Поэтому и цена на наш труд относительно и очень колеблется от города к городу. В вакансиях на различных сайтах вы можете увидеть порой зарплату выше, чем есть на самом деле. Это делается для «переманивания» ценных специалистов. Поэтому не стоит торопиться и ставить такую зарплату в свое резюме, вы окажетесь «дорогим» разработчиком и не получите ни одного отклика на свое резюме. Я не могу ответить вам, какую оптимально ставить желаемую зарплату в вашем городе, но могу сказать про свой. Метка-ориентир: Москва, декабрь 2015 год, курс доллара - ровно 70 руб. Для самостоятельного Junior разработчика с наличием портфолио и хорошим уровнем теоретических знаний (как их получить см. 1 часть), с полным отсутствием коммерческого опыта разработки средняя зарплата составляет 40 000 - 60 000 рублей. (Но не стоит видеть только цифру 60 здесь, это высшая, на мой взгляд, планка, на которую можно претендовать только при хорошем везении и очень достойных знаниях с портфолио)

Q: Портфолио?

A: Портфолио - это ваши собственные проекты, доведенные вами от начала и до конца. Пока у вас нет возможности похвастаться реальным опытом в вашем резюме, но никто не запретит вам сделать что-то свое, чтобы поразить работодателя! :) Один-два проекта я вам советую сделать, это очень повысит шансы на трудоустройство, ведь наличие проекта показывает, что вы способны доводить дело самостоятельно от начала до конца, а это очень ценно качество в сотруднике для любого руководителя. Еще один момент, не поленитесь, и реализуйте свою идею, не выкладывайте в портфолио вашу вариацию домашнего задания из курса. Ведь существует вероятность, что до вас такое же уже присылали кандидаты, а если вакансия еще открыта, то какой из этого вывод? :) Конечно

это шутка, но если серьезно, то домашние задания слишком малы для портфолио, их вряд ли оценят. Все-таки портфолио нужно, чтобы впечатлить, поэтому будьте оригинальны и усердны.

Q: А нужно ли выкладывать свое приложение в App Store? У меня нет аккаунта разработчика и мне не хочется тратить 99\$ ради одного приложения для портфолио.

А: Нет, это совсем необязательно. Запишите видео с интерфейсом приложения, сделайте побольше скринов, выложите примеры кода на Github, работодатели прекрасно понимают, что не у всех есть возможность тратиться на этот аккаунт, и я повторюсь, для них главное увидеть, что вы умеете, ваш код. Но в то же время, приложение в App Store - это готовый продукт, который показывает вашу самостоятельность (а также опыт публикации приложений и общения с командой Apple Review). Поэтому опубликованные приложения всегда будут плюсом в ваше резюме. Если решитесь, достаточно одного-двух.

Q: Каким должно быть резюме? Стоит ли писать в него мой предыдущий опыт работы, не связанный с разработкой?

А: Не стоит, если ваш предыдущий опыт никак не поможет вам в разработке. Вместо этого лучше напишите каждый из ваших проектов, как место работы, опишите используемые технологии, какие фишки реализовали в нем, с чем интересным столкнулись.

Работодатель это оценит гораздо больше, чем чтение нескольких страниц описания опыта работы условного менеджера по продажам. Резюме должно быть кратким и лаконичным. Напишите список технологий, которые знаете, выложите ссылки на свои проекты, ссылки на ваш профиль GitHub, напишите о ваших сильных качествах и остановитесь на этом. Краткость - сестра таланта.

Q: Нужно ли высшее образование?

А: Желательно, но необязательно. Адекватные руководители всегда смотрят на знания, к тому же все понимают, что наличие диплома - совершенно не показатель знаний. Опять же, наличие диплома демонстрирует самостоятельность, способность доводить начатое до конца. С дипломом по данной специальности могут не спросить какие-то основы, подразумевая, что вы их знаете, ведь все это проходят в университетах. Но могут и спросить, поэтому диплом на этапе первого трудоустройства совсем не показатель. А вот что действительно показатель - так это ваша мотивация, желание развиваться и любовь к разработке (и Apple :)).

На этом всё, надеюсь, эта тема поможет начинающим свой путь ребятам поверить в собственные силы и сделать следующий, самый важный и, чего уж скрывать, самый тяжелый шаг. Потом будет гораздо легче! Удачи вам!