

Урок номер 5. Массивы. Задание, обсуждение, вопросы.

Итак домашнее задание по массивам, ну и заодно повторим все пройденное:

Уровень ученик:

(просто повторение того что увидел)

1. Создать класс "человек", который будет содержать в себе базовые качества - "имя", "рост", "вес", "пол", а также будет иметь метод "передвижение".
2. Создать дочерние классы (наследники) "велосипедист", "бегун", "пловец" и переопределить метод "передвижение" в каждом из этих классов.
3. При старте программы создать по одному объекту каждого класса и объединить их в массив.
4. В цикле пройти по всем элементам массива и вывести на экран все характеристики каждого объекта (имя, рост и тд) и у каждого вызвать метод "передвижение".

Если получилось пройти уровень ученик, то можно попробовать уровень студент:

(небольшая импровизация + дополнительная мозговая деятельность)

5. Создать еще одного наследника от класса человек, добавить в него пару новых свойств, добавить в массив и в цикле вывода вывести его свойства как дополнение к свойствам человека
6. Метод "передвижение" реализовать таким образом, чтобы перед тем как выполнить свое собственное передвижение, он выполнял передвижение реализованное в классе человек
7. Массив выводить в обратном порядке.

Уровень Мастер:

(простор для фантазии + больше мозговой деятельности + больше практики)

8. Создать класс "животное" (не наследник класса человек!) со своими собственными базовыми свойствами (отличными от человеческих) и методом "передвижение".
9. Унаследоваться от него и создать пару других классов с переопределенным передвижением.
10. Объединить всех людей и животных в один массив.
11. В цикле выводить тип объекта (человек или животное) перед тем как выводить его свойства и вызывать метод

Уровень Звезда:

(разобраться с тем чего Леша не объяснял! + реальная задача!)

12. Поместить всех людей в один массив, а животных в другой массив (количество людей и животных должно быть разное)
13. В одном цикле выводить сначала человека а потом животное, доставая данные поочередно из двух разных массивов, если в одном из массивов объектов больше, то в конце должны выводиться только объекты этого массива (так как других уже нет)

Уровень Супермен!

(кодинг за гранью возможного! + вероятность успеха практически 0! + реальный вызов!)

14. Соединить животных и людей в одном массиве.
15. Используя нужный метод класса NSArray отсортировать массив (как результат будет другой массив).
16. Сортировать так: сначала люди, а потом животные, люди отсортированы по имени, а животные по кличкам
17. Реально требует разобраться с сортировкой самому, тяжело, но достойно уважения

Урок номер 6. Типы данных. Задание, обсуждение, вопросы.

Задание:

1. Я очень хочу чтобы вы попрактиковались создавать и использовать enum списки. Они ОЧЕНЬ распространены, они делают код красивее и вносят дополнительную информацию. Это просто очень хорошая практика их использовать! Практикуйтесь! Создайте кучу разных пропертей под разные

энумы. Постарайтесь усвоить это сразу.

2. Надо попрактиковаться со структурами. Например такое небольшое задание: на поле 10x10 рандомно создайте точек (разберитесь как рандомно генерировать цифры, подсказка - функция `arc4random()`) и проверяйте какая из точек попадает в квадрат размером 3x3 в самом центре поля. Грубо говоря надо определить какие из точек в массиве попадают в центр и вывести их на печать. Пробуйте и задавайте вопросы.

3. Разберитесь с оболочками `NSNumber` и `NSValue`. На самом деле тут все предельно просто, но вы должны понять саму суть.

Урок номер 7. Протоколы. Задание, вопросы, обсуждение.

Итак друзья, тема очень важная, надо практиковаться. Очень важно все понять, особенно суть. Если есть вопросы - обязательно задавайте!

Задание такое:

1. Реализуйте протоколы `Jumpers`, `Swimmers`, `Runners`. Каждый протокол должен иметь пару свойств, например скорость или максимальная высота прыжка и пару методов, например прыгать или плыть, нырять и тд. Включите фантазию!!!

2. Часть методов делайте обязательными (например у пловцов обязательный метод это плыть а у бегунов - бежать, логично?), а другую часть не обязательными.

3. Используйте классы из задания про массивы, там где у вас люди и животные. Например у кенгуру и у особых спортсменов реализуйте протокол `Jumpers`, тоже самое с другими протоколами. Пусть каждый протокол будет реализован как некоторыми классами животных, так и некоторыми классами людей.

4. Положите всех в один массив. Тех кто прыгает заставьте прыгать и рассказывать про свои свойства. Причем у некоторых объектов пусть будут одни необязательные свойства или методы, а у других другие. Тот кто бежит пусть бежит, а тот кто плавает, пусть плавает. Тот кто ничего не делает должен быть выведен на экран как лодырь :)

5. Сделайте так чтобы один какой-то класс мог и бегать и прыгать и плавать и посмотрите как он себя поведет в цикле.

Урок номер 8. NSDictionary. Задание, обсуждение, вопросы.

Урок на самом деле не большой и я думаю после опыта с массивами весь смысл дикшиари должен быть хорошо понятен. Но на всякий случай давайте сделаем небольшую практику.

Уровень Ученик

1. Создайте класс студент со свойствами имя, фамилия и фраза приветствия.
2. Создайте 10 - 15 объектов этого класса.
3. Теперь мы создадим дикшиари типо школьный журнал, где ключ будет фамилия + имя, а значение сам студент.
4. Распечатайте дикшиари

Уровень Студент.

4. В цикле пройдемся по всем ключам в дикшиари и распечатаем имя и фамилию каждого студента + его фразу приветствия.

Уровень Мастер.

5. Чтобы сделать тоже самое но по какому-то порядку, отсортируйте массив ключей по возрастанию и выведите приветствие каждого студента из дикшиари, но уже по отсортированному списку.

Супермена тут больше нет :)

Урок номер 9. Делегаты. Задание, обсуждение, вопросы.

Итак вот мы и добрались до делегатов. Нам нужно очень хорошо их понять, так как они используются везде и всегда. Ничего сложного в этом нет. Делегат это просто ссылка на того, кто будет делать определенные действия, описанные протоколом. Надо закрепить.

Ученик:

1. Создать пару пациентов и доктора по тому же принципу что и в видео. (Доктор делегат у пациентов)
2. У пациента пусть будет температура и другие симптомы, по которым доктор может принимать решение.
3. У пациента сделайте метод типа стало хуже и пусть когда станет хуже, то он идет к доктору
4. Всех пациентов объедините в массив и в цикле вызовите метод "стало хуже".
5. Доктор должен лечить каждого согласно симптомам.

Студент:

6. Создайте другой класс доктора, не наследника от первого доктора, например друг :)
7. этот друг должен лечить своих пациентов своими собственными методами и короче плохой он доктор
8. пусть кто-то ходит к врачу, а кто-то к нему
9. создайте пару разных объектов класса друг и пусть они лечат своих пациентов (чтобы понять что делегат это не класс, а объект)

Мастер:

10. Создайте список частей тела в делегате пациента (голова, живот, нога и тд) и когда пациент приходит к врачу, пусть говорит что болит.
11. Доктор должен принимать во внимание что болит
12. Создайте у доктора метод "рапорт". Пусть в конце дня, когда все уже пожаловались достаточно, доктор составит рапорт (выдаст имена) тех у кого болит голова, потом тех у кого болел живот и тд

Супермен

13. Создайте в классе пациента проперти - оценка доктору.
14. Когда доктор вам назначает лечение некоторые пациенты должны стать недовольны.
15. В конце дня после того как все лечение будет сделано и доктор напишет рапорт, надо пройтись по пациентам и всем недовольным поменять доктора.
16. Начать новый день и убедиться что Недовольные пациенты таки меняли доктора.

Урок номер 10. Нотификации. Задания, обсуждения, вопросы.

Итак друзья, эта тема весьма интересная, применяется она довольно часто и самое главное понять из урока сам принцип, для чего оно надо и неужели нельзя обойтись без него :)

Задание:

Ученик.

1. Используя мой код создать свой проект со своими классами и понять как оно работает
2. Добавить классы "пенсионер" и "бизнесмен"
3. Доктор следит за изменением зарплаты, бизнесмен должен следить за изменением налогов, а пенсионер за изменением пенсий.
4. Все классы: Доктор, Пенсионер и Бизнесмен должны следить также и за средней ценой на товар.

Умничка.

5. Каждый из классов следит за инфляцией (процентный рост средней цены на товар) и оценивает свой потенциальный доход по отношению к инфляции, например размер пенсии к возможности купить продукты у пенсионера, либо новая прибыль за вычетом налогов у бизнесмена также по отношению к ценам на товар.

6. Не секрет что изменение этих характеристик оказывает разное влияние на разные слои населения, поэтому пусть в зависимости от уровня покупательской способности каждый класс выдает свои перлы.

Мастер.

7. Подпишите классы на нотификацию ухода приложения в бекграунд, чтобы когда нажимается кнопка HOME и приложение сворачивается, каждый объект заявлял о том что он идет спать

8. Тоже самое сделать для случая, когда приложение возвращается из свернутого состояния

Супермен

9. Создайте свой класс аналог апп делегату, только он не делегат приложения, но слушатель тех же самых нотификаций, какие методы есть у делегата. Грубо говоря у вашего класса должны быть те же методы, что и у апп делегата, но вызываться они должны путем нотификаций

10. Добавьте НСЛоги в каждый метод апп делегата и своего класса

11. Сворачивая и разворачивая приложение добивайтесь вызовов определенных методов делегата и тех же нотификаций и посмотрите что вызывается раньше - метод делегата или нотификация :)

Урок номер 12. Блоки. Задание, обсуждение, вопросы.

Итак друзья, надеюсь вам понравился урок и вы даже досмотрели до конца :)

По поводу задания:

Ученик. Фактически это повторить первую половину.

1. В апп делегате создайте блок с без возвращаемой переменной и без параметров и вызовите его.
2. Создайте блоки с параметрами и передайте туда строку, которую выведете на экран в последствии.
3. Если вы не определили тип данных для ваших блоков, то сделайте это сейчас и используйте их
4. Создайте метод который принимает блок и вызывает его и вызовите этот метод.

Студент.

5. Создайте класс студент с проперти имя и фамилия.
6. Создайте в аппделегае 10 разных студентов, пусть у парочки будут одинаковые фамилии.
7. Поместите всех в массив.
8. Используя соответствующий метод сортировки массива (с блоком) отсортируйте массив студентов сначала по фамилии, а если они одинаковы то по имени.

Мастер.

9. Задание из видео. Из урока о делегатах. У пациентов удалите протокол делегат и соответствующее проперти.
10. Добавьте метод принимающий блок когда им станет плохо.
11. Блок должен передавать указатель на самого студента ну и на те параметры, которые были в задании по делегатам.
12. Теперь когда пациентам поплохеет, они должны вызывать блок, а в блоке нужно принимать решения что делать (доктор не нужен делайте все в апп делегате)

Супермен

13. Познайте истинное предназначение блоков :) Пусть пациентам становится плохо не тогда когда вы их вызываете в цикле(это убрать), а через случайное время 5-15 секунд после создания (используйте специальный метод из урока по селекторам в ините пациента).

14. не забудьте массив пациентов сделать проперти аппделегата, а то все помрут по выходе из функции так и не дождавшись :)

Урок номер 13. Многопоточие. Задание, обсуждение, вопросы.

Я честно говоря не знаю когда вы будете напрямую использовать NSThread, я рекомендую всегда, когда надо делать что-то в бэкграунде, то использовать либо GCD (Grand Central Dispatch) либо NSOperation. Так что задание будет соответствующее :)

Ученик.

1. Создайте класс студент. У него должен быть метод - угадать ответ :)
2. В метод передается случайное целое число например в диапазоне от 0 - 100 (или больше) и сам диапазон, чтобы студент знал где угадывать
3. Студент генерирует случайное число в том же диапазоне пока оно не будет искомым
4. Весь процесс угадывания реализуется в потоке в классе студент
5. Когда студент досчитал то пусть пишет в NSLog
6. Создайте 5 студентов и дайте им одну и ту же задачу и посмотрите кто справился с ней лучше

Студент.

7. Задача та же, но вместе с условием передавайте студенту блок, в котором вы и объявите результаты
8. Блок должен определяться в том же классе, где и определялись студенты
9. Блок должен быть вызван на главном потоке

Мастер.

10. Создать приватный метод класса (да да, приватный метод да еще и с плюсом), который будет возвращать статическую (то есть одну на все объекты класса студент) `dispatch_queue_t`, которая инициализируется при первом обращении к этому методу.
11. Лучше в этом методе реализовать блок `dispatch_once`, ищите в инете как и зачем :) А что, программист всегда что-то ищет и хороший программист всегда находит.
12. Все студенты должны выполнять свои процессы в этой queue и она должна быть `CONCURRENT`, типа все блоки одновременно выполняются

Супермен.

13. Добавьте еще один класс студента, который делает все тоже самое что вы реализовали до этого, только вместо GCD он использует NSOperation и NSOperationQueue. Вообще вынос мозга в самостоятельной работе :)
14. Все сделавшие Мастера и Супермена и с красивым кодом получают отдельный огромный РЕСПЕКТ, так как они это на самом деле заслуживают.

Урок номер 14. Строки. Задания, обсуждения, вопросы.

Да, тема работы с текстом никогда не бывает веселой и забавной, но, как ни крути, надо это знать.

Ученик.

1. Возьмите текст из моего исходника и создайте на его основе строку, с которой будете работать.
2. Выведите на экран сначала первую половину текста, а потом вторую.
3. Создайте массив, каждый элемент которого является предложением данного текста.

Студент.

4. Используя массив из пункта 3 создайте новый массив, элементы которого это предложения, содержащие слово NSString. Нужно будет использовать поиск и мутабл массивы.

Мастер.

5. Создайте новый массив на основе массива из пункта 4, но слова в предложении пусть идут наоборот, первое слово будет в конце, а последнее вначале :)

Супермен.

6. В оригинальном тексте каждое слово должно начинаться с маленькой буквы, а заканчиваться на заглавную.

Урок номер 15. Битовые операции. Задание, обсуждение, вопросы

Это довольно нужный материал, так что надо попрактиковаться хорошенько.

Ученик.

1. Повторите мой код и создайте класс студент с соответствующим набором предметов
2. В цикле создайте 10 студентов и добавьте их в массив. Используйте мутабл массив
3. У каждого рандомно установите предметы

Студент

4. В новом цикле пройдитесь по студентам и разделите их уже на два массива - технари и гуманитарии.
5. Также посчитайте количество тех кто учит программирование

Мастер.

6. Если студенты выбрали биологию, то отмените ее у них и выведите сообщение в лог, предмет отменен
7. Тут придется разобраться как сбросить бит, включите логику :)

Супермен.

8. Сгенерируйте случайный интeджер в диапазоне от 0 до максимума.
9. Используя цикл и битовые операции (и возможно NSMutableString) выведите это число на экран в двоичном виде

Урок номер 16. Дата и время. Задание, обсуждение, вопросы

Немного практики с датами нам не помешает, так что вот задание.
(немного повышу градус сложности)

Ученик.

1. Создайте класс студент у которого будет проперти dateOfBirth (дата рождения), которая собственно NSDate.
2. Инициализируйте NSMutableArray и в цикле создайте 30 студентов.
3. Каждому установите дату рождения. Возраст рандомный от 16 до 50 лет.
4. В другом цикле пройдитесь по всему массиву и выведите день рождения каждого студента в адекватном формате.

Студент.

5. Отсортируйте массив студентов по дате рождения, начиная от самого юного.

6. Используя массивы имен и фамилий (по 5-10 имен и фамилий), каждому студенту установите случайное имя и случайную фамилию.

7. Выведите отсортированных студентов: Имя, Фамилия, год рождения

Мастер.

10. Создайте таймер в апп делегате, который отсчитывает один день за пол секунды.

11. Когда таймер доходит до дня рождения любого их студентов - поздравлять его с днем рождения.

12. Выведите на экран разницу в возрасте между самым молодым и самым старым студентом (количество лет, месяцев, недель и дней)

Супермен.

13. Выведите на экран день недели, для каждого первого дня каждого месяца в текущем году (от начала до конца)

14. Выведите дату (число и месяц) для каждого воскресенья в текущем году (от начала до конца)

15. Выведите количество рабочих дней для каждого месяца в текущем году (от начала до конца)

Урок номер 19. UIView Геометрия. Задание, вопросы, обсуждение.

Итак наконец-то мы начали изучать графику :) Надо тренироваться! Повышаем градус заданий :)

Ученик

1. В цикле создавайте квадратные UIView с черным фоном и расположите их в виде шахматной доски

2. доска должна иметь столько клеток, как и настоящая шахматная

Студент

3. Доска должна быть вписана в максимально возможный квадрат, т.е. либо бока, либо верх или низ должны касаться границ экрана

4. Применяя соответствующие маски сделайте так, чтобы когда устройство меняет ориентацию, то все клетки растягивались соответственно и ничего не вылетало за пределы экрана.

Мастер

5. При повороте устройства все черные клетки должны менять цвет :)

6. Сделайте так, чтобы доска при поворотах всегда строго находилась по центру

Супермен

8. Поставьте белые и красные шашки (квадратные выюхи) так как они стоят на доске. Они должны быть сабвыюхами главной выюхи (у них и у клеток один супервью)

9. После каждого переворота шашки должны быть перетасованы используя соответствующие методы иерархии UIView

Урок номер 20. UIView Аутлеты. Задание, вопросы, обсуждение.

На самом деле урок очень простой, единственно что я рекомендую повторить прошлое домашнее задание, но используя аутлеты. Итак:

ВСЕ ЗАДАНИЯ ДОЛЖНЫ БЫТЬ ВЫПОЛНЕНЫ ТОЛЬКО ИСПОЛЬЗУЯ АУТЛЕТЫ.

МЕТОД subviews НЕ ИСПОЛЬЗУЙТЕ, ВМЕСТО НЕГО ИСПОЛЬЗУЙТЕ АУТЛЕТ КОЛЛЕКЦИИ

Ученик

1. В цикле создавайте квадратные UIView с черным фоном и расположите их в виде шахматной доски

2. доска должна иметь столько клеток, как и настоящая шахматная

Студент

3. Доска должна быть вписана в максимально возможный квадрат, т.е. либо бока, либо верх или низ должны касаться границ экрана
4. Применяя соответствующие маски сделайте так, чтобы когда устройство меняет ориентацию, то все клетки растягивались соответственно и ничего не вылетало за пределы экрана.

Мастер

5. При повороте устройства все черные клетки должны менять цвет :)
6. Сделайте так, чтобы доска при поворотах всегда строго находилась по центру

Супермен

8. Поставьте белые и красные шашки (квадратные выюхи) так как они стоят на доске. Они должны быть сабвыюхами главной выюхи (у них и у клеток один супервью)
9. После каждого переворота шашки должны быть перетасованы используя соответствующие методы иерархии UIView

Урок номер 21. UIView Animations. Задание, обсуждение, вопросы

Вот и добрались до интересного! Надо немного поиграться чтобы усвоить материал :)

Ученик.

1. Создайте 4 выюхи у левого края ипада.
2. Ваша задача всех передвинуть горизонтально по прямой за одно и тоже время
3. Для каждой выюхи используйте свою интерполяцию (EasyInOut, EasyIn и т.д.). Это для того, чтобы вы увидели разницу своими собственными глазами :)
4. Добавьте реверсивную анимацию и бесконечные повторения
5. добавьте смену цвета на случайный

Студент

5. Добавьте еще четыре квадратные выюхи по углам - красную, желтую, зеленую и синюю
6. За одинаковое время и при одинаковой интерполяции двигайте их всех случайно, либо по, либо против часовой стрелки в другой угол.
7. Когда анимация закончится повторите все опять: выберите направление и передвиньте всех :)
8. Выюха должна принимать в новом углу цвет той выюхи, что была здесь до него ;)

Мастер

8. Нарисуйте несколько анимационных картинок человечка, который ходит.
9. Добавьте несколько человечков на эту композицию и заставьте их ходить

Супермена на этот раз нет, ничего сверхъестественного не смог придумать :(

Урок номер 22. Touches. Simple Drag and Drop. Задание, обсуждение, вопросы

Вот такой вот урок по тачам. Решил жесты и тачи в один урок не объединять, а относительно простой функционал тачей решил дополнить практическим примером :)

Уровень супермен (остальных уровней не будет)

1. Создайте шахматное поле (8x8), используйте черные сабвыюхи
2. Добавьте белые и красные шашки на черные клетки (используйте начальное расположение в шашках)
3. Реализуйте механизм драг'н'дроп подобно тому, что я сделал в примере, но с условиями:
4. Шашки должны ставать в центр черных клеток.
5. Даже если я отпустил шашку над центром белой клетки - она должна переместиться в центр ближайшей к отпусканию черной клетки.

6. Шашки не могут становиться друг на друга
7. Шашки не могут быть поставлены за пределы поля.

Вот такое вот веселое практическое задание :)

Урок номер 23. Gestures. Задание, обсуждение, вопросы

Собственно урок был долгим, но я показал все что вам нужно для создания классных интерфейсов.

Ученик

1. Добавьте квадратную картинку на вьюху вашего контроллера
2. Если хотите, можете сделать ее анимированной

Студент

3. По тачу анимационно передвигайте картинку со ее позиции в позицию тача
4. Если я вдруг делаю тач во время анимации, то картинка должна двигаться в новую точку без рывка (как будто она едет себе и все)

Мастер

5. Если я делаю свайп вправо, то давайте картинке анимацию поворота по часовой стрелке на 360 градусов
6. То же самое для свайпа влево, только анимация должна быть против часовой (не забудьте остановить предыдущее кручение)
7. По двойному тапу двух пальцев останавливайте анимацию

Супермен

8. Добавьте возможность зумить и отдалять картинку используя пинч
9. Добавьте возможность поворачивать картинку используя ротейшн

Урок номер 24. UIView Drawings. Задание, вопросы, обсуждения

Вот такой вот базовый урок получился. Эта тема - поле непаханое. Вы можете, если захотите, поискать кучу гайдов и примеров по этому материалу. Можно делать классные вещи. Но я не хочу на этом на долго останавливаться, нам нужно идти дальше :)

Ученик.

1. Нарисуйте пятиконечную звезду :)
2. Добавьте окружности на концах звезды
3. Соедините окружности линиями

Студент.

4. Закрасьте звезду любым цветом цветом оО
5. При каждой перерисовке рисуйте пять таких звезд (только мелких) в рандомных точках экрана

Мастер

6. После того как вы попрактиковались со звездами нарисуйте что-то свое, проявите творчество :)

Супермен

7. Сделайте простую рисовалку - веду пальцем по экрану и рисую :)

Урок номер 25. Кнопки. Задание, обсуждение, вопросы.

Вот друзья, обстановка становится все веселее, раньше мы с вами только в консоль смотрели, а теперь уже потихоньку приложения собираем :) Задание будет тоже, что я говорил в уроке, а именно - сделать калькулятор :)

Ученик.

1. Создайте и разместите кнопки цифр и операторов, создайте нужные акшины
2. Добавьте лейбл и создайте соответствующий проперти
3. По нажатию на кнопку на индикатор должна выводиться либо цифра, либо оператор (никаких вычислений - выводим просто значения кнопок)

Студент.

4. Наведите более / менее красоту, можете насоздавать линий (вьюхи с малой шириной)б ставить бэкграундыб используйте картинки если надо и тд. - проявите творчество :)
5. Сделайте так, чтобы когда нажимаешь на цифровую кнопку, то число росло
6. Сделайте кнопку сброса

Мастер.

7. Сохраняйте вводимое число
8. Первую операцию тоже надо сохранить, пока не ввели второе число (используйте енумы!)
9. Когда нажимаешь равно то первое число выполняет операцию над вторым и выводится результат
10. После того, как результат вывелся на экран, он автоматом становится первым числом и к нему уже можно прибавлять, вычитать и т.д. - как в калькуляторе

Супермен

11. Добавьте точку, теперь можно вводить и дробное число, точка должна работать также, как и точка на обычном калькуляторе (имею в виду, что двух точек в числе быть не может в случае повторного нажатия)
12. С дробным вводом будет посложнее - задание для настоящих суперменов :)

Урок номер 26. Controls. Задание, обсуждение вопросы.

Такой вот на мой взгляд получился простенький урок, но все равно нужно практиковаться.

Будем делать небольшое приложение:

Мы будем применять анимации к тестируемой вьюхе

Ученик.

1. Расположите тестируемую вьюху на верхней половине экрана
2. На нижней половине создайте 3 свича: Rotation, Scale, Translation. По умолчанию все 3 выключены
3. Также создайте слайдер скорость, со значениями от 0.5 до 2, стартовое значение 1
4. Создайте соответствующие проперти для свитчей и слайдера, а также методы для события valueChanged

Студент.

5. Добавьте сегментед контрол с тремя разными сегментами
6. Они должны соответствовать трем разным картинкам, которые вы должны добавить
7. Когда переключаю сегменты, то картинка должна меняться на соответствующую

Мастер.

8. Как только мы включаем один из свичей, наша вьюха должна начать соответствующую анимацию

(либо поворот, либо скейл, либо перенос). Используйте свойство transform из урока об анимациях 9. Так же следует помнить, что если вы переключили свич, но какойто другой включен одновременно с ним, то вы должны делать уже двойную анимацию. Например и увеличение и поворот одновременно (из урока про анимации)

10. Анимации должны быть бесконечно повторяющимися, единственное что их может остановить, так это когда все три свича выключены

Супермен.

11. Добавляем использование слайдера. Слайдер регулирует скорость. То есть когда значение на 0.5, то скорость анимаций должна быть в два раза медленнее, а если на 2, то в два раза быстрее обычной.

12. Попробуйте сделать так, чтобы когда двигался слайдер, то анимация обновлялась без прерывания, а скорость изменялась в соответствующую сторону.

Урок номер 27 - 28. UITextField. Задание, обсуждение, вопросы

Итак начинаем текстовые поля. Напоминаю, этот элемент очень распространен и не очень простой, поэтому уделите ему должное внимание. Начнем с простого, а после урока номер 28 я добавлю задания уровней повыше.

Создаем форму для регистрации студента.

Ученик.

1. Создайте поля (и лейблы напротив как в уроке): имя, фамилия, логин, пароль, возраст, телефон, имейл адрес.
2. Установите правильные виды клавиатур для каждого текстового поля.
3. У всех текстовых полей кроме последнего кнопка ретерн должна быть NEXT, а у последнего DONE.
4. Осуществите переход по кнопкам NEXT и уберите клавишу с экрана кнопкой DONE.
5. Каждое поле при нажатии должно иметь кнопку очистки

Студент

6. Совет, чтобы осуществить переход по NEXT без проверки тонны пропертей, попробуйте использовать UIOutletCollection
7. Создайте соответствующие каждому текстовому полю UILabel чтобы выводить туда информацию из текстовых полей. Сделайте их мелкими и другого цвета.
8. По изменению текста (даже буквы) обновляйте эти лейблы (не забудьте про CLEAR button)

Мастер

9. Для поля ввода телефона используйте мой код из видео, можете поместить его в какой-то отдельный метод если надо
10. Этот код должен работать только для поля телефона и не для какого другого

Супермен

11. Для поля ввода имейла ограничте ввод символов недопустимых в адресе
12. Более того, символ "@" может быть введен только 1 раз
13. установите разумное ограничение для этого поля

Урок номер 29. UITableView Static Cells. Задание, обсуждение, вопросы

Итак, мы начали мою любимую тему о UITableView. Это очень важная тема, поэтому практиковаться очень важно. Заодно нужно добить контролы.

Задание:

Ученик-Студент-Мастер-Супермен:

1. Сделайте задания из уроков 27 и 28 используя статические ячейки.
2. Добавьте возможность сохранять результат.
3. Добавьте новые секции и ячейки с прочим контролами (свичи, сегменты, слайдеры) чтобы проработать материал

Урок номер 30. UITableView Dynamic Cells. Задание, обсуждение, вопросы

Вот теперь начинается самое интересное! Нужна практика, много практики.

Ученик.

1. Создайте таблицу которая содержит 1000 разных ячеек.
2. для каждой ячейки генерируйте рандомный цвет.
3. RGB данные пишите в тексте ячейки, например: RGB(10,20,255)
4. Также раскрашивайте текст ячейки этим цветом.

Студент.

5. Теперь создайте класс, который содержит цвет и нейм.
6. В viewDidLoad сгенерируйте 1000 объектов такого класса по принципу из ученика
7. Положите их в массив и отобразите в таблице
8. В этом случае когда вы будете листать назад вы увидите те же ячейки, что там и были, а не новые рандомные

Мастер.

9. Возвращаемся к студентам. Сгенерируйте 20-30 разных студентов.
10. В таблице создавайте не дефолт ячейку а Value1. В этом случае у вас появится еще одна UILabel - detailLabel.
11. В.textLabel пишите имя и фамилию студента, а в detailLabel его средний бал.
12. Если средний бал низкий - окрашивайте имя студента в красный цвет
13. Отсортируйте студентов в алфавитном порядке и отобразите в таблице

Супермен.

14. Средний бал для студентов ставьте рандомно от 2 до 5
15. После того, как вы сгенерировали 30 студентов вам надо их разбить на группы: отличники, хорошисты, троечники и двоечники
16. Каждая группа это секция с соответствующим названием.
17. Студенты внутри своих групп должны быть в алфавитном порядке
18. Отобразите группы студентов с их оценками в таблице.

Mission Impossible!

19. Добавьте к супермену еще одну секцию, в которой вы отобразите 10 моделей цветов из задания Студент.
20. Помните, это должно быть 2 разных типа ячеек Value1 для студентов и Default для цветов.

Урок номер 31 - 32. UITableView Editing. Задание, обсуждение, вопросы

Итак редактируем наши таблицы.

Задание.

Ученик

1. Создайте контроллер с таблицей в коде без сторибордов.

2. Заполните таблицу данными на свое усмотрение
3. Объедините данные в группы (секции)

Студент.

4. Реализуйте механизм перемещения данных между рядами и секциями
5. Вы должны четко понимать что и как работает и в какой последовательности поэтому повторяйте задание пока вы полностью не освоите этот механизм

Мастер.

6. Реализуйте удаление рядов

Супермен

7. Реализуйте механзм добавления секций и рядов на ваше усмотрение

Урок номер 33 - 34. UITableView Navigation. Задание, обсуждение, вопросы

Я рекомендую вам немного задержаться на этом уроке и снова попрактиковать таблицы. Также будет довольно таки неплохо если вы углубитесь в `NSFileManager`

Ученик.

1. Добавьте возможность создавать директории
2. Добавьте возможность удалять файлы и папки

Студент

3. Сортируйте файлы и папки, сверху должны быть папки, снизу файлы, сортировка по имени
4. Не показывайте скрытые файлы

Мастер

5. В `detailedTextField` каждой ячейки файла, выводите размер файла.

Супермен

6. Тем же способом выводите размер папки (размер папки придется считать рекурсивно :))

Урок номер 35. UITableView Search. Задание, обсуждение, вопросы

В этом уроке мы рассмотрели как один массив данных можно преобразовать в другой, более упорядоченный, да еще и с фильтрами. Нужно сделать пару практических упражнений.

Ученик.

1. Создайте класс студента. У него должны быть свойства: имя, фамилия и год рождения.
2. Генерируйте случайное количество студентов и отобразите их в вашей таблице. (слева имя и фамилия, а справа дата рождения)

Студент.

3. Сгруппируйте студентов по секциям месяцев рождения, то есть все кто родился в январе в одной секции, а если в феврале никто не родился, то и секции такой нет.
4. Внутри секции студенты должны быть отсортированы по имени по алфавиту, а если имена одинаковы, то и по фамилии (подсказка, лучше отсортировать массив вначале по 3 всем параметрам: дата, имя и фамилия)
5. Добавьте индекс бар для быстрого перехода по секциям

Мастер.

6. Добавьте серчбар как в видео, чтобы кнопка кенсел анимировано добавлялась/уезжала и тд
7. Фильтруйте студентов каждый раз, когда вводится новая буква, причем совпадения ищите как в имени так и в фамилии

Супермен

8. Добавьте к серчбару сегментед контрол с тайтлами: Год рождения, Имя, фамилия (по умолчанию включен год рождения)
9. Когда пользователь переключает сегментед контрол, то секции меняются на соответствующие. Например если выбран контрол с именем, то студенты должны быть отсортированы по имя-фамилия-дата, и должны быть собраны в секции, соответствующие первой букве имени.
10. То же самое и для фамилий, фильтр = фамилия-дата-имя
11. если выбрана дата, то все должно отсортироваться как в начале.

Урок номер 36. UIPopoverController. Задание, обсуждение, вопросы

Поповеры надо очень хорошо отпрактиковать! Так же я рекомендую разобраться с тем, как поповер убирается из памяти (было в видео).

Учтите, на айфоне поповеров нет, поэтому для айфона вместо поповеров используйте модальные контроллеры с разными транзишинами если надо.

Ученик

1. Создайте универсальное приложение (айпад / айфон)
2. Первый контроллер должен быть статической таблицей с навигейшн баром
3. В правом углу на навигейшине должна быть кнопка инфо, если на нее нажать, то вылезит поповер с объяснением, что это такое за приложение :)

Студент

4. В таблице создайте классические ячейки:
имя + текстфилд
фамилия + текстфилд
пол + сегментед контрол (мужской/женский)
Дата рождения + текстфилд
Образование + текстфилд

5. с первыми тремя ячейками все понятно, а вот дальше самое интересное

Мастер

6. При нажатии на текст филд с датой рождения текст филду должно быть запрещено входить в режим редактирования, а вместо этого из него должен появиться поповер с UIDatePicker. При изменении даты, содержимое текст филда должно меняться (то есть мы не мучаем юзера форматами ввода, мы просто даем ему барабан с датами и предлагаем выбрать самому)
7. Подсказка. Вам надо сделать контроллер с дейт пикером, а дейт пикер это наследник от UIControl, то есть у него есть акшин valueChanged или типо того. У контроллера нужно создать проперти делегат, по которому мы будем отправлять данные, полученные с барабана. То есть по простому: контроллер следит за барабаном и отправляет изменения своему делегату. Не забудьте установить делегат перед созданием поповера.

Супермен

8. Тоже самое сделать с образованием. Образование это список типа, неполное среднее, среднее, неполное высшее, высшее и тд, то есть если делать в сегментед контролах, то не поместится.
9. Когда нажимаем на образование, появляется поповер с контроллером и таблицей. Причем, выбранное образование должно быть отмечено чекбоксом. (кстати выбранная дата рождения в мастере тоже должна стоять по умолчанию в новом поповере)
10. У этого контроллера тоже должен быть делегат. По нажатию на ячейку вы должны изменить текущий выбор на новый (поменять чекбокс) и передать сообщение делегату, после чего тот должен изменить содержимое текст филда

Вот такое вот задание, а что вы хотели, это вам не хиханьки хахоньки :) Работать надо, много причем, детский сад закончился :)

Урок номер 37 - 38. MapKit. Задание, обсуждение, вопросы

Итак, друзья, в коротеньких двух уроках мы разобрались с картами. Конечно весь функционал карт вам вряд ли пригодиться, но зато практика никогда не бывает лишней, а особенно практика использования нового фреймворка! Так что готовьтесь к заданию :)

Давайте раз уж мы часто говорим о студентах, то будем играть и с ними.

Ученик.

1. Создайте массив из 10 - 30 случайных студентов, прямо как раньше, только в этот раз пусть у них наряду с именем и фамилией будет еще и координата. Можете использовать структуру координаты, а можете просто два дабла - лонгитюд и латитюд.

2. Координату генерируйте так, установите центр например в вашем городе и просто генерируйте небольшие отрицательные либо положительные числа, чтобы случайно получалась координата от центра в пределах установленного радиуса.

(Ну а если совсем не получается генерировать координату, то просто ставьте им заготовленные координаты - это не главное)

3. После того, как вы сгенерировали своих студентов, покажите их всех на карте, причем в титуле пусть будет Имя и Фамилия а в сабтитуле год рождения. Можете для каждого студента создать свою аннотацию, а можете студентов подписать на протокол аннотаций и добавить их на карту напрямую - как хотите :)

Студент.

4. Добавьте кнопку, которая покажет всех студентов на экране.

5. Вместо пинов на карте используйте свои кастомные картинки, причем если студент мужского пола, то картинка должна быть одна, а для девушек другая.

Мастер

6. У каждого колаута (этого облачка над пином) сделайте кнопку информации справа так, что когда я на нее нажимаю вылезает поповер, в котором в виде статической таблицы находится имя и фамилия студента, год рождения, пол и самое главное адрес.

7. В случае если это телефон, то вместо поповера контроллер должен вылезать модально.

Супермен

8. Создайте аннотацию для места встречи и показывайте его на карте новой соответствующей

картинкой

9. Место встречи можно перемещать по карте, а студентов нет

10. Когда место встречи бросается на карту, то вокруг него надо рисовать 3 круга, с радиусами 5 км, 10 км и 15 км (используйте оверлеи)

11. На какой-то полупрозрачной вьюхе в одном из углов вам надо показать сколько студентов попадают в какой круг. Суть такая, чем дальше студент живет, тем меньше вероятность что он придет на встречу. Расстояние от студента до места встречи рассчитывайте используя функцию для расчета расстояния между точками, поищите ее в фреймворке :)

12. Сделайте на навигейшине кнопочку, по нажатию на которую, от рандомных студентов до нее будут проложены маршруты (типо студенты идут на сходку), притом вероятности генератора разные, зависит от круга, в котором они находятся, если он близко, то 90%, а если далеко - то 10%

Сложно, но, надеюсь, интересно :)

Урок номер 39. UIWebView. Задание, вопросы, обсуждения

Даже не знаю какое дать вам задание по этому уроку. Это не настолько критический элемент, он не так часто встречается в приложениях, а если действительно нужен, то там как правило и javascript и все такое. Так что даже не знаю что вам дать :)

Задание следующее:

1. Сделайте один контроллер с таблицей, в ней две секции: pdf и url
2. Присоедините к проекту парочку pdf файлов, их имена должны быть в таблице
3. Добавьте парочку web сайтов во вторую секцию
4. Когда я нажимаю на ячейку, то через пуш навигейшн должен отобразиться либо pdf либо web
5. Надеюсь понятно что для загрузки того либо другого мы используем один и тот же контроллер с UIWebView и инициализируем его нужным NSURL
6. На веб вью должна быть крутилка, а в навигейшине кнопки назад и вперед

Урок номер 40. CoreData Intro. KVC + KVO. Задание, вопросы, обсуждения

Вот эта тема очень интересная, так что задания нужно выполнить обязательно :)

Ученик.

1. Создайте класс студента с пропертиками firstName, lastName, dateOfBirth, gender, grade
2. Также создайте статическую таблицу куда все эти данные выводятся и где их можно менять (с текстом филдами, сегментированными контролами и тд)
3. Пропертисы вашего студента меняйте тем же образом что и в предыдущих уроках (через делегаты и акшины)

Студент.

4. Повесьте обсервера на все пропертисы студента и выводите в консоль каждый раз, когда проперти меняется
5. Также сделайте метод "сброс", который сбрасывает все пропертисы, а в самом методе не используйте сеттеры, сделайте все через айвары, но сделайте так, чтобы обсервер узнал когда и что меняется. (типо как в уроке)

Мастер.

забудьте про UI

6. Создайте несколько студентов и положите их в массив, но обсервер оставьте только на одном из

них

7. У студентов сделайте weak проперти "friend". Сделайте цепочку из нескольких студентов, чтобы один был другом второму, второй третьему, тот четвертому, а тот первому :)
8. Используя метод setValue: forKeyPath: начните с одного студента (не того, что с обсервером) и переходите на его друга, меняя ему проперти, потом из того же студента на друга его друга и тд (то есть путь для последнего студента получится очень длинный)
9. Убедитесь что на каком-то из друзей, когда меняется какой-то проперти, срабатывает ваш обсервер

Супермен

10. Добавьте побольше студентов
11. Используя операторы KVC создайте массив имен всех студентов
12. Определите самый поздний и самый ранний годы рождения
13. Определите сумму всех баллов студентов и средний бал всех студентов

Урок номер 41 - 44. CoreData. Задание, вопросы, обсуждения (Обновлено)

Вот CoreData и пройдена! Ведь не сложно совсем, правда?

Итак домашнее задание создаем локальный мини ГуруКрафтер!

Задание объемное, зато хорошо пройдемся по кордате :)

Ученик.

1. Делаем экран со списком юзеров. На панели навигации есть плюсики, который добавляет нового юзера. При добавлении либо редактировании юзера мы переходим на экран, где можем вводить его данные в динамической(!) таблице: имя, фамилия, почтовый ящик.
2. Юзеров можно добавлять, удалять и редактировать.
3. Добавляем UITabBarController к проекту! Приложение стартует с него и наш экран с юзерами это всего лишь один из табов. Разберитесь с этим контроллером самостоятельно. Делайте его в сториборде. Для каждого контроллера в табах делайте навигейшн, чтобы была у всех панелька навигации. То есть идет так ТабКонтроллер -> таб-> Навигейшн -> наш экран

Студент.

4. Добавьте экран с курсами. На этом экране вы можете добавить, редактировать и удалить курс. Также само как и в случае с юзерами у вас открывается контроллер редактирования. Также динамическая таблица
5. В первой секции идут поля "название курса", "предмет", "отрасль" и преподаватель (имя и фамилия).
6. Во второй секции идет список юзеров, которые подписаны на курс. Можно юзера удалить, тогда он не удаляется из юзеров - он удаляется из курса. Также есть ячейка добавить студентов. (например первая ячейка в секции)
7. если я нажимаю на ячейку студента, то перехожу к его профайлу.

Мастер

8. Если я нажимаю на ячейке добавить студентов, то мне выходит модальный контроллер либо поповер, содержащий список всех юзеров, причем юзеры которые выбрали этот курс имеют галочки. Тут я могу снимать студентов с курса либо добавлять на этот курс новых студентов.

9. Так же и для преподавателя: если нажать на ячейку с преподавателем - переходишь к экрану юзеров, но тут можно выбрать только одного на этот раз.

10. Если преподаватель выбран, то ячейка "преподаватель" на экране редактирования курса должна содержать его имя и фамилию, если нет - должен быть текст "выберите преподавателя"

11. Тоже самое сделайте на экране юзеров, мы ведь там сделали динамическую таблицу также. Добавьте секцию "курсы, которые ведет" и добавьте туда все его курсы. Также добавьте секцию "курсы, которые изучает". Если у студента нет курсов в какой-либо из этих секций - не показывайте секцию :)

Супермен.

12. Сделать все вышеперечисленное

13. Сделайте третий экран - преподаватели. На этом экране будет выводиться список всех преподавателей сгруппированных по "предмету" (например программирование). У каждого преподавателя видно количество курсов (просто цифра) и если нажать на преподавателя, то переходишь в его профайл.

Урок номер 45. Client Server APIs Basics. Задание, вопросы, обсуждения

Итак вот мы и затронули всеми долгожданную тему :) Давайте теперь попрактикуемся создавать эти интерфейсы. начнем с простых задач и я сюда буду добавлять задания по мере поступления новых уроков.

Кстати, я прекрасно понимаю что у контакта есть свой фреймворк, но ведь наша цель научиться работать с API.

Ученик

1. Используйте все способы из урока чтобы получить список друзей из контакте, в качестве дополнительных полей берите только маленькую картинку.

Студент

2. Когда я нажимаю на друга, должен пушем открыться новый контроллер со статической таблицей, в которой идет детальное описание друга и большая картинка. Для этого используйте метод "users.get". Не забудьте использовать id друга, по которому вы нажали. Запрос должен отправляться именно в этом контроллере, а не в первом.

Мастер

3. В описании друга добавьте две ячейки - subscriptions и followers. Если юзер нажимает на них, то переходит в соответствующий контроллер. В одном случае мы отображаем список подписок, а в другом список последователей.

Супермен

4. В описание юзера добавьте еще одну ячейку - wall. По нажатию на нее надо перейти на контроллер стены пользователя и отобразить стену :)

Не забывайте все API методы держать в синглтоне и всегда создавайте модели для приходящих данных.

Урок номер 46 - 47. Client Server APIs. Задание, вопросы, обсуждения

Я думаю мы закончили с клиент серверными приложениями, все что вам нужно для начала вы уже знаете, а все прочее вы поймете с практикой. Так что давайте немного практики. Давайте сделаем

приложение для нашей группы. Я приведу список функционала по разным уровням. Не используйте СДК контакта, делайте все как в уроках

Ученик:

1. Логин
2. Отображение стены группы (посты)
3. Отображение количества лайков и комментариев на каждом посте (только цифры)
4. Добавление своего текста

Студент

5. Отображение аватарки и имени и фамилии создателя поста, а также даты поста
6. По нажатию на аватарку юзера иметь возможность отправить ему личное сообщение
7. Как следствие нужно иметь возможность читать свои сообщения и отвечать на них

Мастер

6. По нажатию на ячейку поста переход на контроллер, где видно сам пост + все комменты и все лайки.
7. В том же контроллере иметь возможность комментировать данный пост и ставить лайки

Супермен

8. Иметь возможность загружать фотографии с камеры или из библиотеки в фотоальбом группы
9. Иметь возможность проигрывать видео урок