Задания курса «Разработка под iOS. Начинаем»

Задание 1. Note

1. Создайте файл Note.swift. В нём вам предстоит реализовать первый код для будущего приложения заметок — структуру Note. По ходу курса файлов будет становиться больше, и вы соберёте из них полноценный проект.
2. Реализуйте заметку в виде структуры Note. Структура должна удовлетворять следующим условиям:
    - Она иммутабельна. Это поможет вам легко обеспечить потокобезопасность и избежать проблем в дальнейшем.
    - Содержит уникальный идентификатор uid. Если не задан пользователем, генерируется с помощью UUID().uuidString. Это позволит легко сравнивать два экземпляра одной заметки и облегчить дедупликацию.
    - Содержит обязательные строковые поля — title и content. Без заголовка и текста не обойтись.
    - Содержит цвет заметки — color. Пользователь сможет создать заметку любого цвета. Если он его не задал, по умолчанию ставим белый (класс UIColor из UIKit). Так пользователи смогут структурировать заметки по темам с помощью цветовой раскраски.
    - Содержит обязательное поле важность (importance). Должно быть enum с тремя значениями: «неважная», «обычная» и «важная».Так пользователь сможет расставлять заметкам приоритет.
    - Содержит необязательное поле «Дата самоуничтожения» (selfDestructionDate) типа Date.

Задание 2. JSON Extension

Вам предстоит реализовать расширение для работы с json файлами. Оценивать задания будут студенты и преподаватели по заданным критериям.

Инструкция по выполнению задания

1. Создайте файл NoteExtension.swift. В нём вам предстоит реализовать код расширения.
2. Реализуйте расширение структуры Note, которое:

- Содержит функцию для разбора json: `static func parse(json: [String: Any]) -> Note?`.
- Содержит вычислимое свойство для формирования json: `var json: [String: Any]`.
- Если цвет НЕ белый, сохраняет его в json.
- Если важность «обычная», НЕ сохраняет её в json.
- UIColor, enum, Date сохраняет в json НЕ в виде сложных объектов. То есть допустимы любые скалярные типы (Int, Double, …), строки, массивы и словари.

Задание 3. FileNotebook

Вам предстоит реализовать записную книжку из ваших заметок. Оценивать задания будут студенты и преподаватели по заданным критериям.

Инструкция по выполнению задания

1. Создайте файл FileNotebook.swift. В нём вам предстоит реализовать записную книжку.
2. Записная книжка должна удовлетворять следующим условиям:
   - Объявлена как класс: `class FileNotebook`.
   - Содержит закрытую для внешнего изменения, но открытую для получения коллекцию Note.
   - Содержит функцию добавления новой заметки: `public func add(_ note: Note)`.
   - Содержит функцию удаления заметки на основе uid: `public func remove(with uid: String)`.
   - Содержит функцию сохранения всей записной книжки в файл: `public func saveToFile()`, сигнатура дана для примера.
   - Содержит функцию загрузки записной книжки из файла: `public func loadFromFile()`, сигнатура дана для примера.

Задание 4. Добавляем launch screen и иконку

Задайте вашему приложению внешнее лицо:)

Инструкция по выполнению задания

1. Добавьте к проекту иконку. Вы можете взять любую картинку из интернета или нарисовать её самостоятельно. У платформы есть ограничение на загружаемый файл в 5Мб. Пожалуйста, не добавляйте слишком большие картинки.
2. Обратите внимание на размеры. Должны быть все размеры, в том числе иконка для AppStore. Для подготовки картинок вы можете использовать любой онлайн-сервис, который найдёте в сети.
3. Нужна ли иконка для iPad? На ваше усмотрение. Если в настройках проекта указать, что поддерживается только iPhone, иконку для iPad добавлять не нужно. Но если вы планируете поддерживать iPad и это отмечено в настройках проекта, то отсутствие иконки для iPad будет ошибкой.
4. Убедитесь, что при загрузке приложения у иконки нет серых углов, добавлен launch screen, который адаптируется под разные размеры экранов и отличается от дефолтного (добавленного в проект).

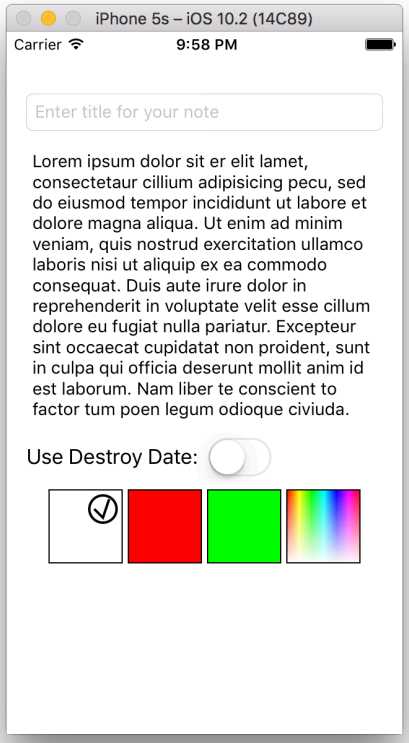
Дополнительные ссылки, которые помогут выполнить задание:

- Подробнее познакомиться с launch screen (или splash screen) и узнать рекомендации к его оформлению можно в human interface guidelines.
- Статья «Replacing Launch Images With Storyboards».
- Обратите внимание, что иконку необходимо оставлять квадратной. Это требования human interface guidlines. Необходимое скругление система добавит сама.

Задание 5. Экран редактирования

Вам предстоит создать экран редактирования заметки. Для этого можете использовать Storyboard или создать отдельный xib-файл.

Пример экрана:



Требования, которые необходимо реализовать:

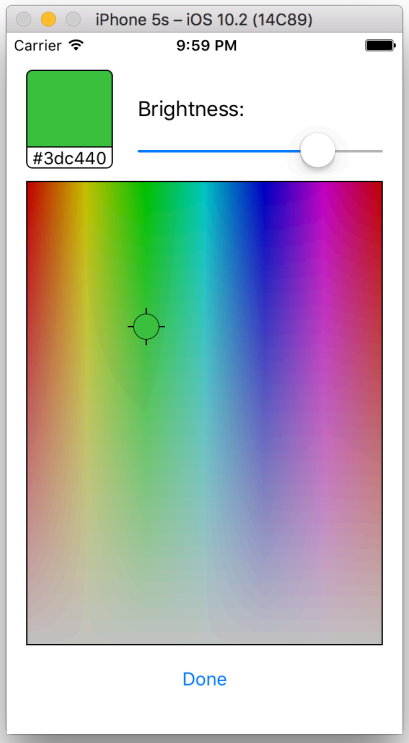
1. Все элементы адаптируются под любые размеры экрана. Для реализации используйте AutoLayout или учитывайте ширину при ручном расположении элементов. Вы можете проверить результат, поворачивая симулятор или устройство, а также в ассистенте в режиме Preview.
2. Интерфейс хорошо отображается на iPhone X. Помните про SafeArea. Проверьте результат на симуляторе iPhone X, обязательно поверните устройство.
3. Экран прокручивается, если содержимое не умещается по высоте. Проверить можно так: запустите приложение в симуляторе и начните редактировать любое поле ввода текста. Появившаяся клавиатура значительно ограничит доступную для отображения элементов высоту. Необходимый для реализации компонент в лекции мы не рассматривали, но в ней рассказали, где искать разные компоненты для разных задач. Вам нужно самостоятельно подобрать компонент и разобраться как с ним работать.
4. Высота поля для ввода текста заметки динамически меняется в зависимости от содержимого. У пустого поля задан минимальный размер.
5. Выбор цвета реализован в виде цветных квадратиков с чёрной рамкой. Текущий цвет помечается флажком. Флажок нарисуйте с помощью CoreGraphics.
6. В секции выбора цвета есть квадратик для выбора произвольного цвета и он удовлетворяет следующим требованиям:
   - Изначально выглядит как палитра цветов.
   - По долгому нажатию на квадратик открывается экран с представлением выбора цвета (компоненты ColorPicker-а). Пример окна смотрите ниже.
   - После выбора цвета в ColorPicker-е квадратик на первом экране окрашивается в выбранный цвет и помечается флажком.
7. Представление выбора цвета (ColorPicker) обязательно выполнено в виде самостоятельного компонента. То есть компонента, файлы которого (.swift и, возможно, .xib) можно перенести в другой проект и он заработает без изменений в коде компонента. В нём не должно быть зависимостей от других объектов в текущем проекте. Создайте для него отдельный класс-

наследник UIView, как мы делали для GameFieldView. Компонент удовлетворяет следующим критериям:

- Выбор цвета осуществляется путём перемещения пальца по палитре.
- Элемент, указывающий на текущий цвет в палитре, залит в тот же цвет.
- Элемент, отображающий текущий цвет (в левом верхнем углу), имеет скругление углов. Радиус выберите сами.
- Выбранный цвет сохраняется при повторном заходе на экран ColorPicker.

8. При включении свитча Destroy Date должно появляться поле выбора даты — UIDatePicker, стандартный компонент. При выключении — пропадать.

Пример экрана с компонентом выбора цвета ColorPicker:

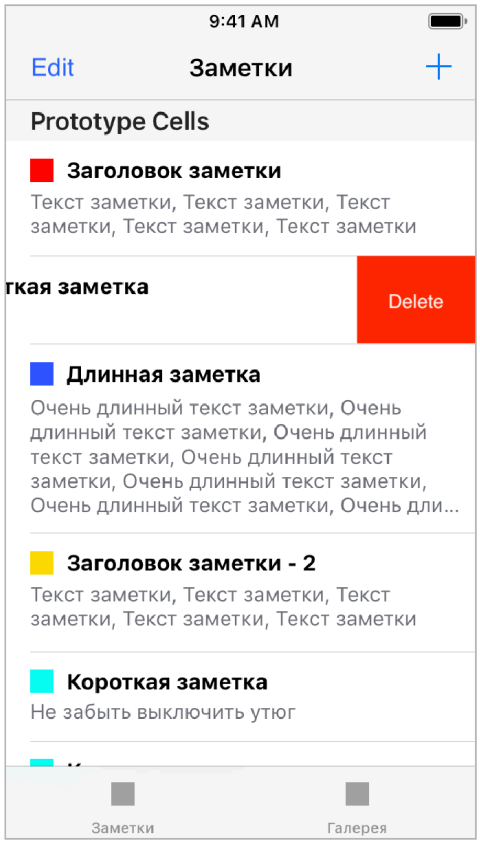


Задание 6. Список заметок

Вам предстоит создать список заметок. В основе приложения разместите UITabBarController с двумя закладками.

Требования:

- На первой закладке разместите UINavigationController. Первый экран в нём должен содержать таблицу (UITableView) со списком заметок. По нажатию на ячейку таблицы нужно переходить (Push) на экран редактирования заметки из предыдущего модуля.
- На первой закладке разместите кнопку «+» в navigation bar. Если нажать на кнопку, будет создана новая заметка и пользователь сразу попадает на экран её редактирования.
- После завершения редактирования пользователь закрывает экран и возвращается на список заметок. Созданная заметка должна добавиться на экран.
- Для удаления заметки добавьте кнопку «Редактировать» или «Edit» слева от названия. Она должна переключать режим редактирования таблицы (tableView.isEditing = true/false). Удаление заметки из списка осуществляется свайпом влево или нажатием на «Delete» на нужной ячейке.
- На второй закладке разместите галерею фото на ScrollView, созданную в предыдущей задаче. В галерее разместите минимум 5 фотографий.



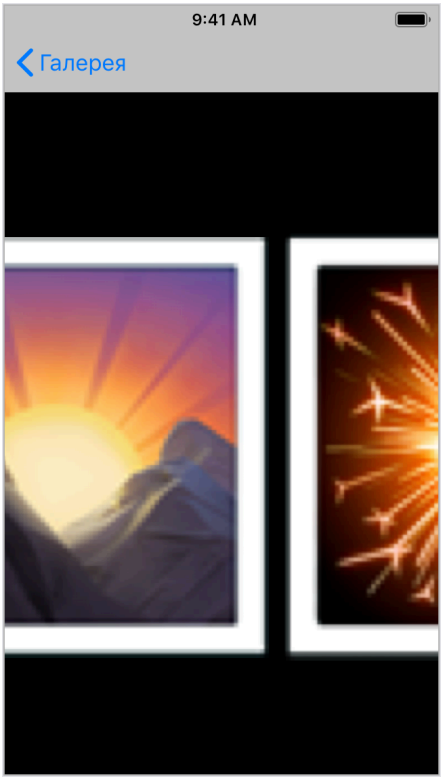
Усложняем 1:

- Сделайте высоту ячеек в таблице зависимой от контента так, чтобы вся заметка помещалась в ячейку таблицы, но не более пяти строк. В конце идёт троеточие.
- Перенесите ColorPicker, созданный в предыдущем модуле, в отдельный UIViewController и покажите его при помощи метода Push.

Усложняем 2:

- На второй закладке сделайте «фото-заметки». Разместите иконки фотографий с использованием UICollectionView.
- По кнопке «+» позвольте пользователю добавлять фотографии через UIImagePickerController.
- Тап по фото открывает (push) новый экран, в котором фото представлено в полный размер.
- Свайпом вправо-влево можно пролистывать фото вперёд/назад по всем фотографиям.


9:41 AM
Галерея
Заметки
Галерея


9:41 AM
Галерея