



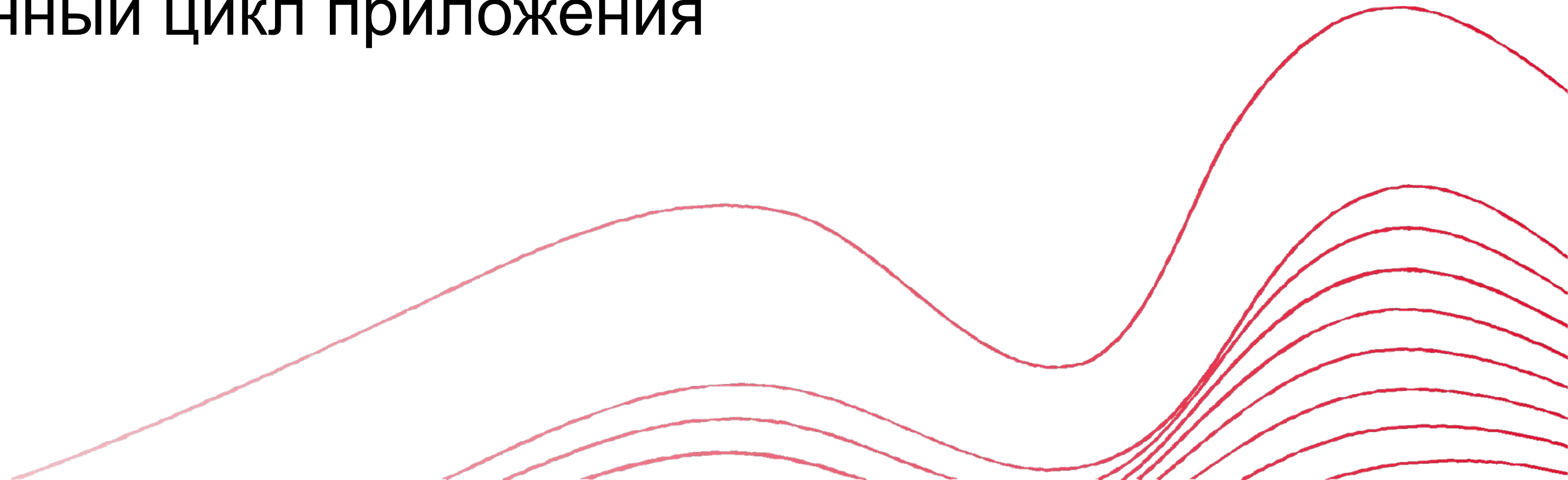
Лекция 6





Agenda

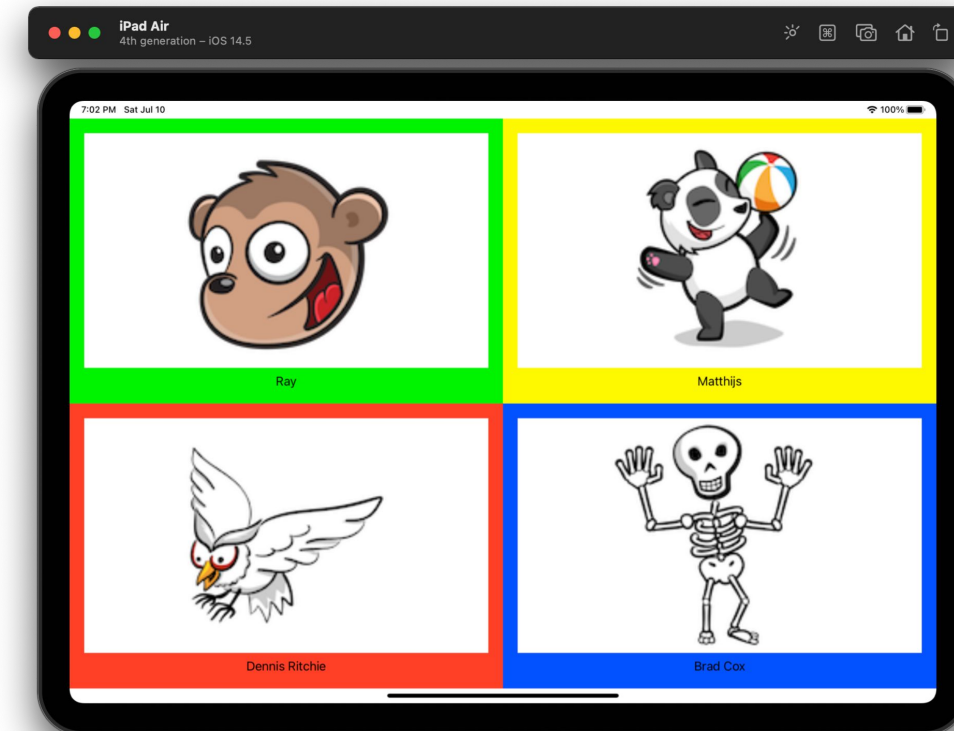
- Constraints
- UITableView, Custom cells, Navigation
- Жизненный цикл приложения





Constraints

- Align constraints
- Отступы
- Пропорциональные размеры
- Конфликты и приоритеты констрейнтов
- Констрейнты в коде



Создание констрейнтов в коде:

<https://www.hackingwithswift.com/articles/140/the-auto-layout-cheat-sheet>

Эффективный способ находить сломанные констрейнты

<https://medium.com/ios-os-x-development/dynamically-modify-ui-via-lldb-expression-1b354254e1dd>

Большая и подробная статья от Рэя об autolayout:

<https://www.raywenderlich.com/811496-auto-layout-tutorial-in-ios-getting-started>



UITableView

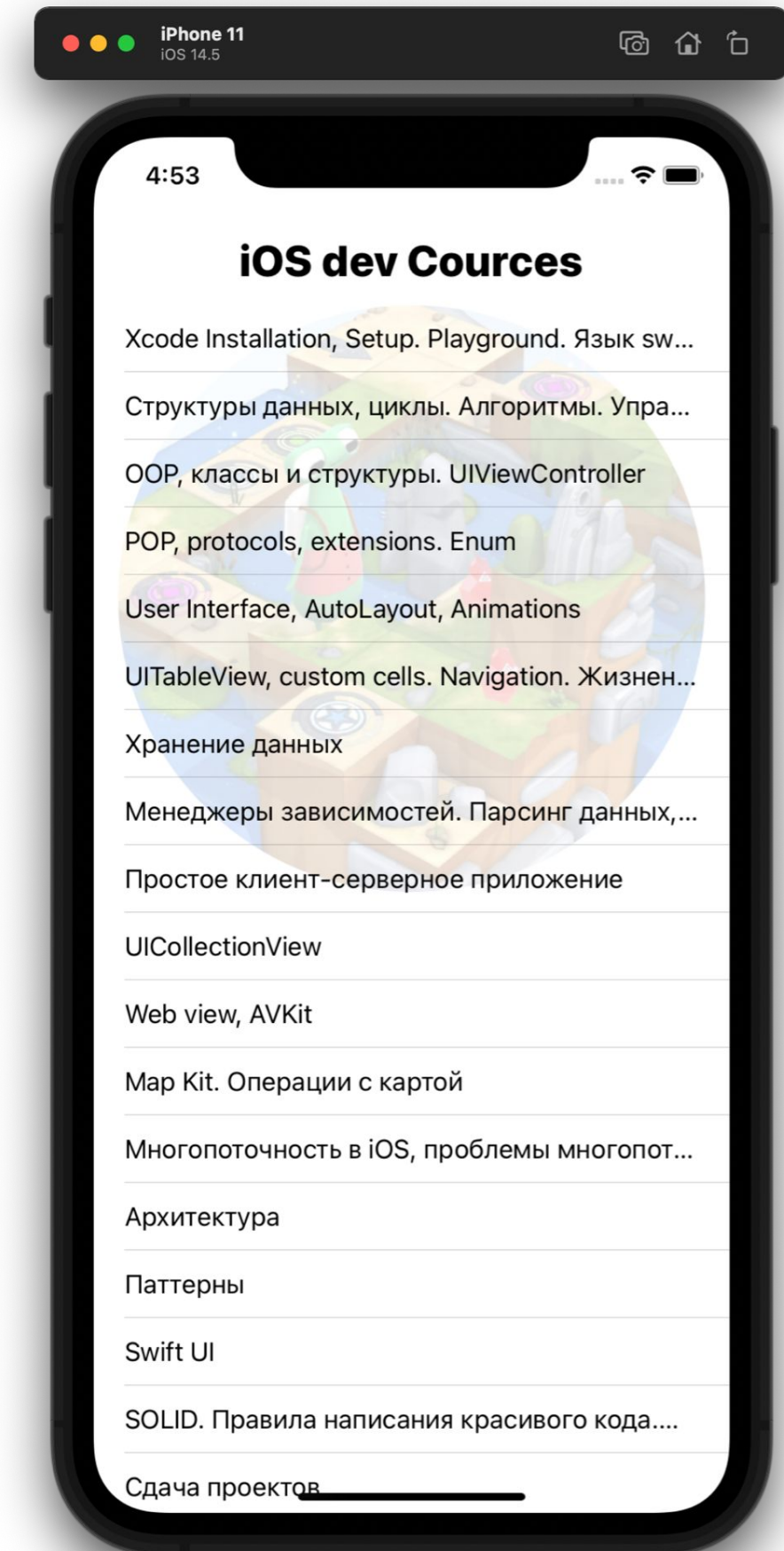
- UITableViewDelegate
- UITableViewDataSource
- Создаем таблицу со списком тем курса
- Создаем кастомные ячейки для таблицы
- Создаем переход по ячейке на новый экран

Тutorial, как создать самую простую таблицу:

<https://coderoad.ru/40220905/Создайте-UITableView-программно-в-Swift>

Более подробный tutorial:

<https://habr.com/ru/post/436390/>





Custom cells (кастомные ячейки)

- Добавляем Cocoa Touch файл в проект, наследник UITableViewCell. Устанавливаем галочку "Создать так же хіб файл".
- Не забываем добавить идентификатор ячейки - в коде, или в хіб файле в инспекторе (на правой панели).
- Настраиваем ячейку, добавляя нужные проперти и функцию которая будет сэтить (калька от set, дословно "задавать") данные в ячейку.
- В контроллере с таблицей мы регистрируем ячейку.
- В методе cellForRow создаем ячейку через объект таблицы и сэтим в него данные.

Статья о том, как создать custom UITableViewCell:

<https://programmingwithswift.com/create-a-custom-uitableviewcell-with-swift/>



Navigation

- Добавляем в сториборде UINavigationController.
- Перетаскивая через ПКМ на контроллер с таблицей, указываем, что он “root”.
- В том методе, где мы хотим выполнить переход, в коде:
 - Создаем объект сториборда
 - Создаем через сториборд объект контроллера
 - Выполняем переход, указывая в качестве цели созданный контроллер

Все способы передавать данные между контроллерами:

<https://learnappmaking.com/pass-data-between-view-controllers-swift-how-to/#forward-properties>



Жизненный цикл приложения

didFinishLaunchingWithOptions - приложение запущено

applicationWillResignActive - переключение на другое приложение, или сворачивание

applicationDidBecomeActive - приложение перешло в активное состояние

applicationDidEnterBackground - приложение перешло в бекграунд

applicationWillEnterForeground - переход с бекграунд в форграунд

applicationWillTerminate - пользователь закрыл (выгрузил) приложение

Статья на русском о жизненном цикле iOS приложения:

<https://proswift.ru/ios-application-lifecycle-ili-zhiznennyj-cikl-ios-prilozheniya/>

Видеоурок на 13 минут:

<https://swiftbook.ru/post/koposhilka/application-life-cycle/>



UITableView - практика

1. Создайте приложение с tableView, которая будет использовать нативные ячейки UITableViewCell(). А в качестве данных будет принимать array of strings, в котором будут названия месяцев. Приложение должно вывести список месяцев.

2. По нажатию на ячейки, открывайте второй контроллер, в котором будет 1 UILabel в центре экрана. Он должен отображать название месяца, который был выбран.

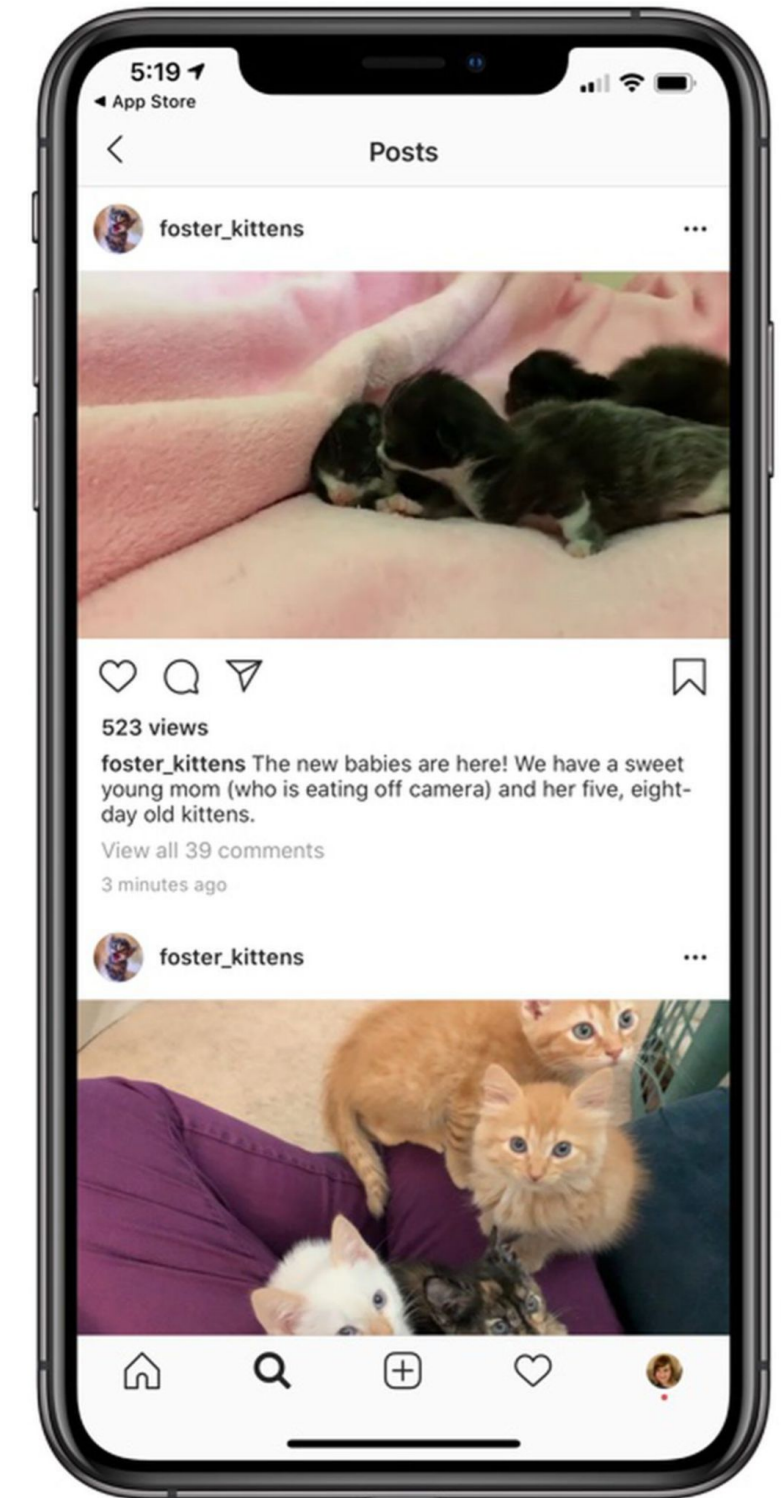


UITableView & custom cells - практика

2. Создайте приложение - копию инстаграмной ленты. Для этого нам нужно будет позаботиться о данных, это будет массив с 3мя экземплярами класса Post, который включает:

- avatarImageName: String
- postImageName: String
- viewsCount: Int
- description: String
- countOfComments: Int
- postedAt: String

На дизайне также есть иконки - троеточие сверху справа, сердечко, знак комментария, бумажный самолётик, знак закладки. Не стоит тратить время на поиск каждой иконки. Каждая из них - просто UIImageView, в которую можно засетить, как бекграунд, любой цвет.





UITableView & custom cells - практика

Конечный результат схематически можно изобразить следующим образом:

- Синяя область - **UILabel** (текст должен подтягиваться из объекта Post)
- Зеленая - **UIImageView** (иконки, можем задать прямо в xib файле)
- Желтая - **UIImageView** (картинки, должны подтягиваться из объекта Post)

Красным прямоугольником выделены границы 1 ячейки. Все, что выходит за рамки ячеек в приложение добавлять не нужно (UITabBar - панель с иконками внизу экрана, UINavigationController - область над таблицей).





Все ссылки:

Создание констрейнтов в коде:

<https://www.hackingwithswift.com/articles/140/the-auto-layout-cheat-sheet>

Эффективный способ находить сломанные констрейнты

<https://medium.com/ios-os-x-development/dynamically-modify-ui-via-lldb-expression-1b354254e1dd>

Большая и подробная статья от Рэя об autolayout:

<https://www.raywenderlich.com/811496-auto-layout-tutorial-in-ios-getting-started>

Тutorial, как создать самую простую таблицу:

<https://coderoad.ru/40220905/Создайте-UITableView-программно-в-Swift>

Более подробный tutorial:

<https://habr.com/ru/post/436390/>

Статья о том, как создать custom UITableViewCell:

<https://programmingwithswift.com/create-a-custom-uitableviewcell-with-swift/>

Все способы передавать данные между контроллерами:

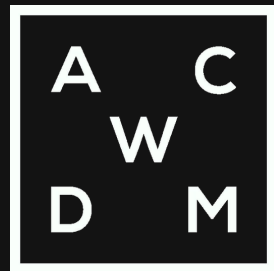
<https://learnappmaking.com/pass-data-between-view-controllers-swift-how-to/#forward-properties>

Статья на русском о жизненном цикле iOS приложения:

<https://proswift.ru/ios-application-lifecycle-ili-zhiznennyj-cikl-ios-prilozheniya/>

Видеоурок на 13 минут:

<https://swiftbook.ru/post/koposhilka/application-life-cycle/>



Web
Academy

CREATE YOUR IT FUTURE.

WITH WEB ACADEMY

