курс | development | 12 недель

iOS: разработка приложений с 0

r_d

iOS программа курса 20 занятий

1

Swift: начало

2

ООП: основы

3

Создание iOS-прило-
жения в Xcode

4

Создание
интерфейса iOS-
приложения

5

Динамические
интерфейсы, часть 1

6

Динамические
интерфейсы, часть 2

iOS программа курса 20 занятий

7 Динамические интерфейсы, часть 3

8 Навигация в приложении, часть 1

9 Навигация в приложении, часть 2

10 Анимации в iOS

11 Работа с памятью в iOS

12 Многозадачность в iOS, часть 1

iOS программа курса 20 занятий

13
Многозадачность в iOS, часть 2

14
Дебаг iOS-приложения

15
Тестирование

16
Хранение данных в приложении

17
Работа с сетью в приложении

18
Сборка приложения

iOS программа курса 20 занятий

19

Современные архитектуры для iOS приложений

20

Защита курсовых проектов

Динамические интерфейсы. Часть 3. Финал.

- Что такое UICollectionView и зачем нам еще одна таблица.
- UICollectionViewCell — владение контентом в collection view.
- UICollectionViewDataSource — задаём наши данные.
- Что такое Layouts для collection view.
- Demo  Создание экрана Library приложения Apple Music.

r_d

Что такое UICollectionView и зачем нам еще одна таблица

- UICollectionView — это объект, который (как и UITableView) управляет упорядоченным набором данных и показывает эти элементы на экране, используя индивидуальный лэйяут (custom layout).
- UICollectionView может заменить UITableView и с помощью такого же layout, как у таблицы (1 колонка, много строк) использоваться при создании сложных интерфейсов.

Что такое UICollectionView и зачем нам еще одна таблица

- UICollectionView — наследник UIScrollView (как и UITableView).
- Скролл пальцем в UICollectionView может быть как по-вертикали, так и по-горизонтали. По-факту, это определяется тем расположением элементов на экране, которое вы зададите через custom layout объект.

Что такое UICollectionView и зачем нам еще одна таблица

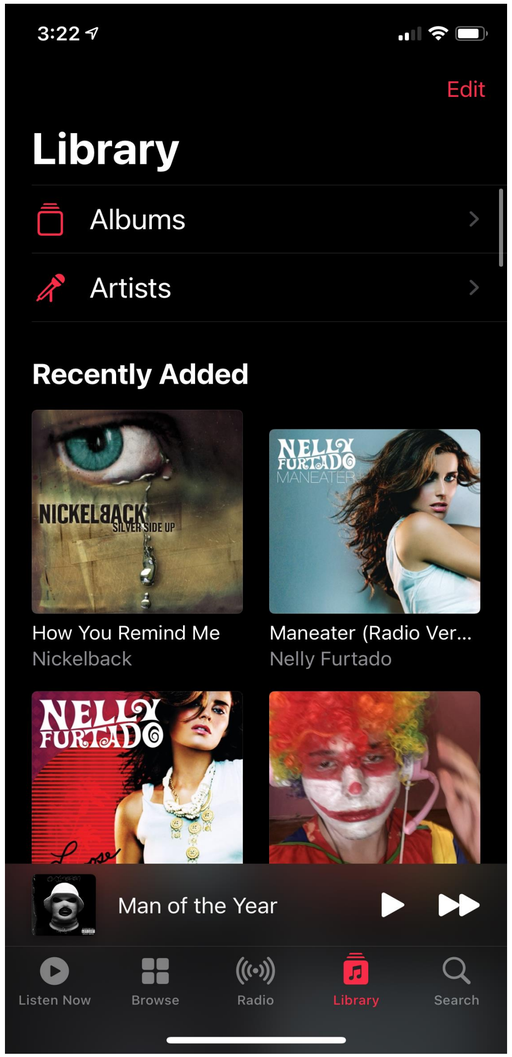
- Когда вы используете UICollectionView, ваша основная задача — управление контентом, т.е. данными, которые UICollectionView должна отобразить на экране.
- Объект, который предоставляет необходимые данные UICollectionView должен конформить протокол UICollectionViewDataSource.
- Внутри класса UICollectionView хранятся переменные dataSource и delegate, логика каждой из которых дублирует UITableView.

UICollectionViewCell — владение контентом в collection view

- Как и UITableView объект класса UICollectionView не управляет своим контентом самостоятельно.
- Каждый кусочек информации в UICollectionView, как и в UITableView, управляется (management) и располагается на экране (layout) внутри специальных ячеек, объектов класса UICollectionViewCell.
- Объект, который конформит протокол UICollectionViewDataSource, предоставляет, нужным образом конфигурирует ячейки и предоставляет каждой ячейке соответствующие данные.

UICollectionViewCell — владение контентом в collection view

- Неплохой пример того, где нужно использовать UICollectionView в iOS приложении — приложение Apple Music.
- Здесь несколько видов UICollectionViewCell: под заголовком Library и Recently Added.
- Скролл вертикальный, на весь экран.
- Ячейки в секции Recently Added расположены в 2 строки.
- Текст Recently Added также расположен внутри section header ячейки, и показывается внутри supplementary view*.



UICollectionViewDataSource — задаём наши данные

- Какие особенности есть у UICollectionViewDataSource при сравнении с UITableViewDataSource?
- Основное отличие — начиная с iOS 13 Apple предоставляет реализованный класс UICollectionViewDiffableDataSource.

UICollectionViewDataSource — задаём наши данные

- Объект класса UICollectionViewDiffableDataSource большую часть работы реализует за нас, нам лишь нужно правильно определить типы данных, с которыми UICollectionViewDiffableDataSource будет работать.

```
dataSource = UICollectionViewDiffableDataSource<Int, UUID>(collectionView: collectionView)
{
    (collectionView: UICollectionView, indexPath: IndexPath, itemIdentifier: UUID) ->
UICollectionViewCell? in
    // configure and return cell
}
```

UICollectionViewDataSource — задаём наши данные

- Если мы определяем логику data source самостоятельно, одним из кандидатов на то, кто будет конформить протокол UICollectionViewDataSource является UICollectionViewController.
- Мы создаём подкласс класса UICollectionViewController, где переопределяем методы протокола data source.

UICollectionViewDataSource — задаём наши данные

- В простейшем варианте кода это выглядело бы так:

```
import UIKit

final class MusicLibraryCollectionViewController: UICollectionViewController {
    private var tracks: [Track] = // initialize array of songs stored in the Music App

    // Data Source methods

    override func numberOfSections(in collectionView: UICollectionView) -> Int {
        return 1
    }

    override func collectionView(_ collectionView: UICollectionView,
numberOfItemsInSection section: Int) -> Int {
        return tracks.count
    }

    override func collectionView(_ collectionView: UICollectionView, cellForItemAt
indexPath: IndexPath) -> UICollectionViewCell {
        return UICollectionViewCell()
    }
}
```

UICollectionViewLayout

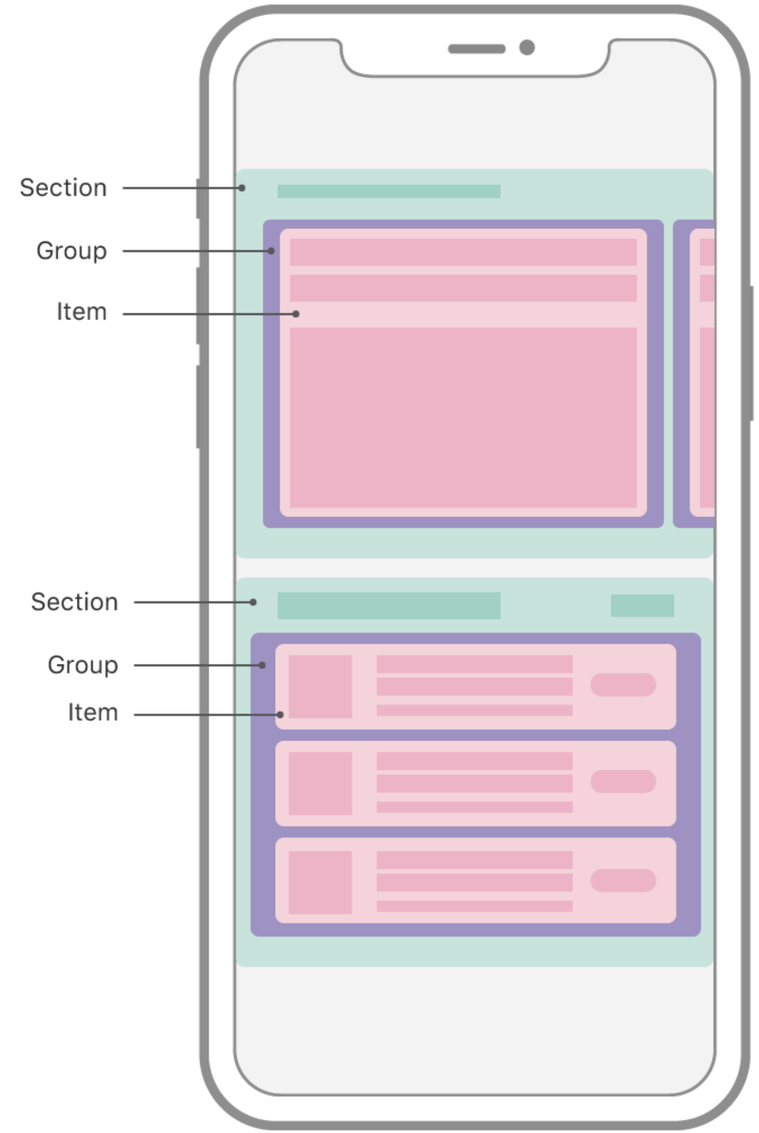
- Объект layout определяет визуальное расположение контента внутри всей collection view.
- Объект layout определяет расположение ячеек, supplementary views, декорированием views внутри границ collection view а также предоставляет всю необходимую информацию визуальной части collection view.
- Класс UICollectionViewLayout — абстрактный базовый класс, который можно использовать для создания своего собственного класса layout.

UICollectionViewLayout

- Кроме UICollectionViewLayout стоит также рассмотреть готовый класс от Apple — UICollectionViewCompositionalLayout.
- Класс UICollectionViewCompositionalLayout предоставляет работу с любым количеством секций внутри collection view, чтобы визуально разбить контент на отдельные группы.
- Каждая группа может располагать ячейки с контентом либо вертикально, либо горизонтально, либо в другом необходимом порядке.

UICollectionViewLayout

- Как выглядит такой layout при использовании готового UICollectionViewCompositionalLayout?



Demo Time 

спасибо

задавайте вопросы