



Списки (UITableView и UICollectionView)

Занятие №4



Дмитрий Тараев

- Отметиться!
- Повторение
- UIScrollView
- UITableView
 - UITableView data source & delegate
- UICollectionView
 - data source & delegate
 - UICollectionViewLayout
- Демо

- Про Objective-C
 - посмотрели пример, переписанный на нём
- MVC
 - Model
 - View
 - Controller
- Другие архитектурные паттерны
- Способы коммуникации между слоями
 - рассмотрели не все, только delegate

Домашние задания - ДЗ 0 (результаты)



- Начато:
 - 19 проектов
- Заявлено:
 - 39 участников

Домашние задания



1. **ДЗ 0** (срок **11.03.2018 12:00**) - максимум **5 баллов**
 - Разбиться на группы, придумать тему, создать репозиторий

2. **ДЗ 1** (срок **18.03.2018 12:00**) - максимум **5 баллов**
 - *Базовая функциональность*

3. **ДЗ 2** (срок **08.04.2018 12:00**) - максимум **10 баллов**
 - *Использование списков элементов*

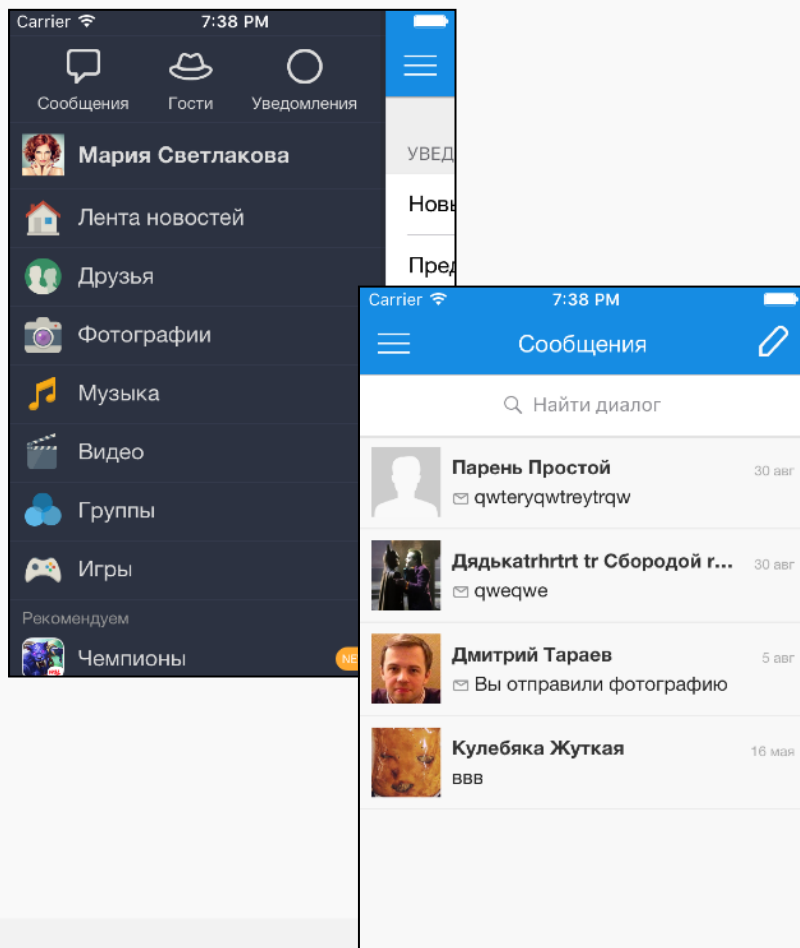
4. **ДЗ 3** (срок **29.04.2018 12:00**) - максимум **15 баллов**
 - *Работа с сетью и локальной базой данных*

5. **ДЗ 4** (срок **13.05.2018 12:00**) - максимум **10 баллов**
 - *Работа с дополнительными API*
 - *карта*
 - *камера / фотогалерея*

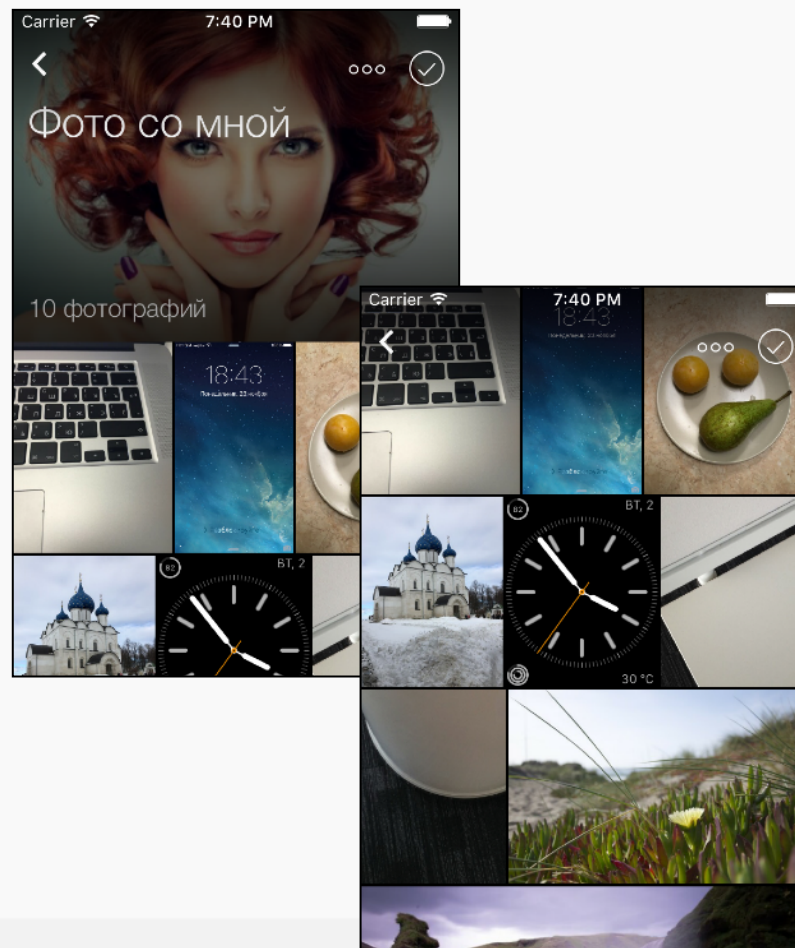
Примеры списков



UITableView



UICollectionView



UIScrollView 1/3

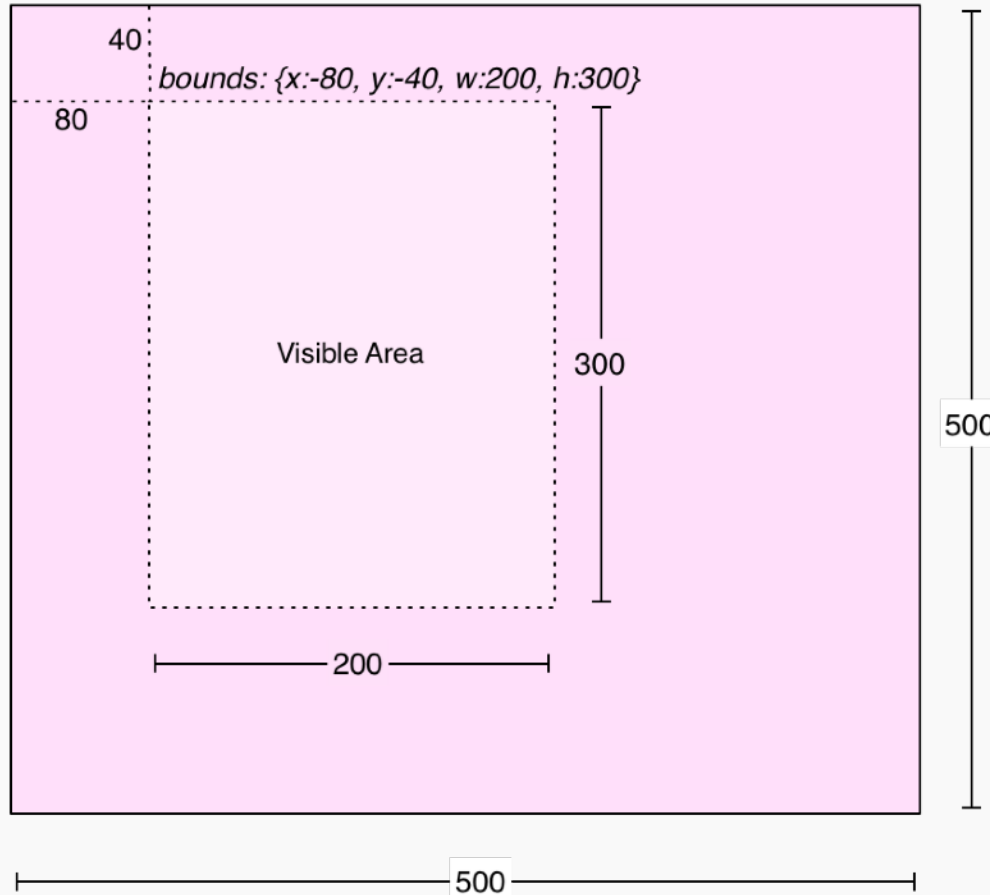


- Для контента, который не помещается на экране
- Особенности
 - масштабирование (zooming)
 - перетаскивание (panning)
- Координаты
 - frame
 - contentView
 - .contentSize,.contentOffset, contentInset

UIScrollView 2/3



`contentOffset: {x:80, y:40}`
`contentSize: {w:500, h:500}`



- bounds
- contentOffset
- contentSize
- contentInset

<https://www.objc.io/issues/3-views/scroll-view/>

UIScrollView 3/3



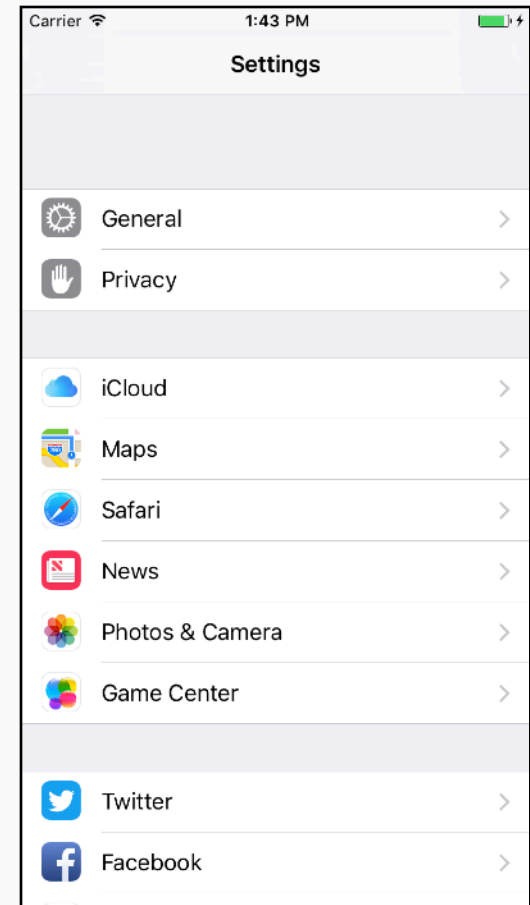
важно!

- Весь контент находится в памяти
- Применение
 - фотогалерея
 - витрина книг в iBooks

UITableView 1/10



- UITableView
 - static / dynamic
 - plain / grouped
 - разбита на секции? да / нет
 - разные виды ячеек



- Классы:
 - UITableView
 - UITableViewCell
- Протоколы:
 - UITableViewDataSource
 - UITableViewDelegate

- Создание списка:
 - с помощью Interface Builder
 - storyboard
 - отдельный xib-файл
 - с помощью кода

- Data source & delegate (протоколы)
 - Data source - «что» показывается
 - Delegate - «как» показывается
- Паттерн «делегат»
 - объект, который удовлетворяет указанному протоколу, т.е. реализует объявленные методы
 - другой объект в нужные моменты вызывает эти методы у своего делегата
- Data Source (по сути, это - подвид делегата)
 - примерно то же, но эти методы возвращают данные, то есть объект, реализующий протокол, является источником данных

UITableViewController vs. UIViewController



- UITableViewController



- автоматически назначается dataSource и delegate
- уже есть property (для него она self.view)
*@property (nonatomic, strong) UITableView *tableView;*
- есть «из коробки» pullToRefresh
*@property (strong) UIRefreshControl *refreshControl;*



- ограничивает в реализации сложного дизайна экрана

- Data source

- сколько секций

*func **numberOfSections**(in tableView: UITableView) -> Int*

- сколько ячеек в секции

*func tableView(_ tableView: UITableView, **numberOfRowsInSection** section: Int) -> Int*

- какая должна быть ячейка

*func tableView(_ tableView: UITableView, **cellForRowAt** indexPath: IndexPath) -> UITableViewCell*

- Data source (другое)
 - UIView для header'а и footer'а
 - режим редактирования
 - удаление ячеек
 - вставка ячеек
 - перетаскивание ячеек



-tableView:cellForRowAtIndexPath:



```
1. - (UITableViewCell *)tableView:(UITableView *)tableView
   cellForRowAtIndexPath:(NSIndexPath *)indexPath {
2.     UITableViewCell *cell = [tableView
   dequeueReusableCellWithIdentifier:@"CellIdentifier"
   forIndexPath:indexPath];
3.     cell.textLabel.text = _cellsDataArray[indexPath.row];
4.     return cell;
5. }
```

- NSIndexPath
 - section
 - row
- cell identifier
 - или в коде, или в IB

UITableView 7/10: Self-Sizing Cells



- iOS 8+
 - Свойство `rowHeight` установить `UITableViewAutomaticDimension`
 - Задать `estimatedRowHeight`
 - для приблизительного расчёта высоты `contentSize`
 - фактическая высота рассчитывается по мере появления ячеек на экране
- iOS 7
 - всё сложнее
 - очень подробно: <https://stackoverflow.com/a/18746930>

- Delegate

- была выбрана ячейка

- func tableView(_ tableView: UITableView, **didSelectRowAt** indexPath: IndexPath)*

- нажатие на accessoryButton

- func tableView(_ tableView: UITableView, **accessoryButtonTappedForRowWith** indexPath: IndexPath)*

- методы will / did

- func tableView(_ tableView: UITableView, **willDisplay** cell: UITableViewCell, **forRowAt** indexPath: IndexPath)*

- название для header'а и footer'а

- func tableView(_ tableView: UITableView, **titleForHeaderInSection** section: Int) -> String?*

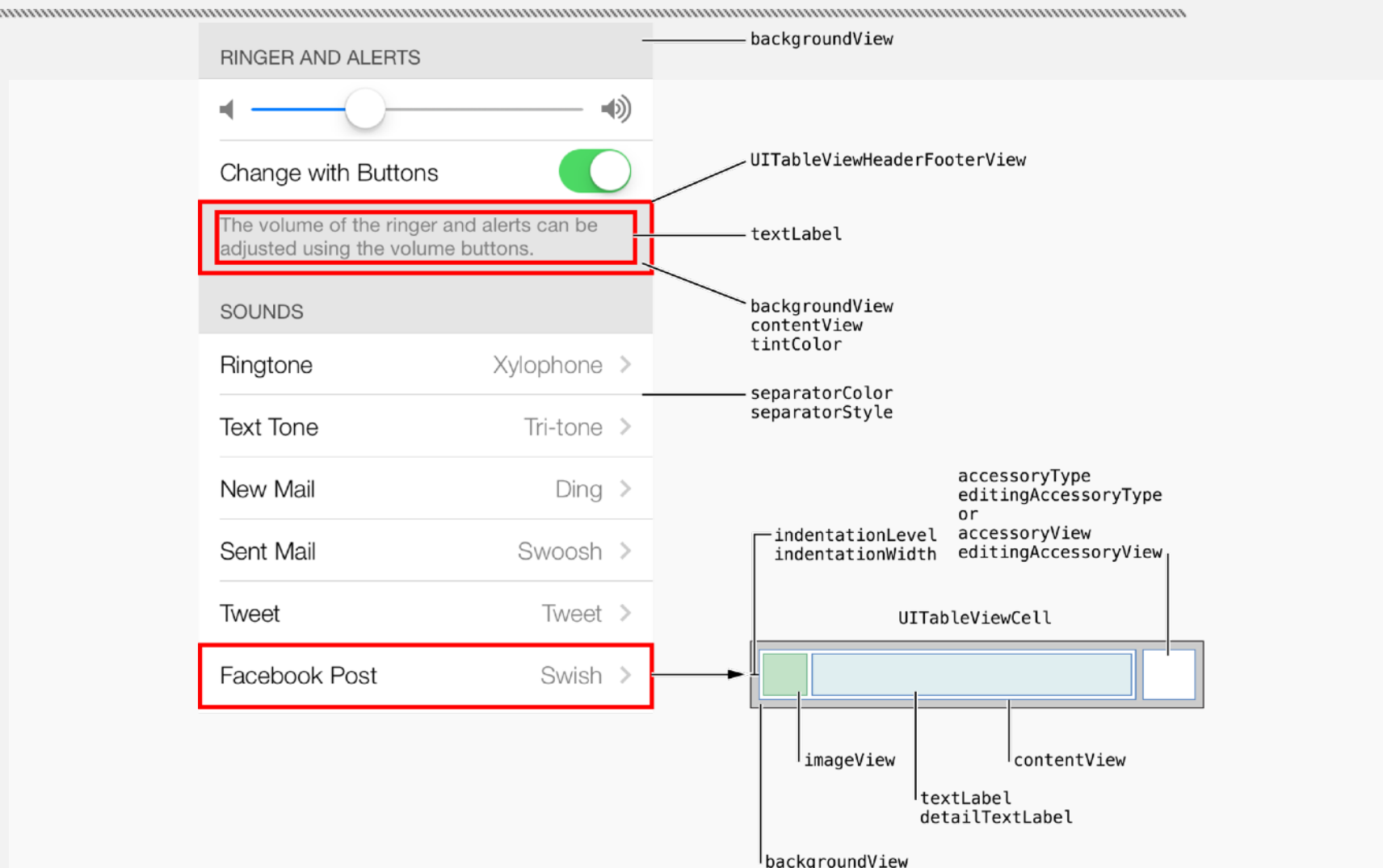
- высота ячейки (далее более подробно)

- func tableView(_ tableView: UITableView, **heightForRowAt** indexPath: IndexPath) -> CGFloat*

- *estimatedHeightForRowAt*

- *высота для header'а и footer'а*

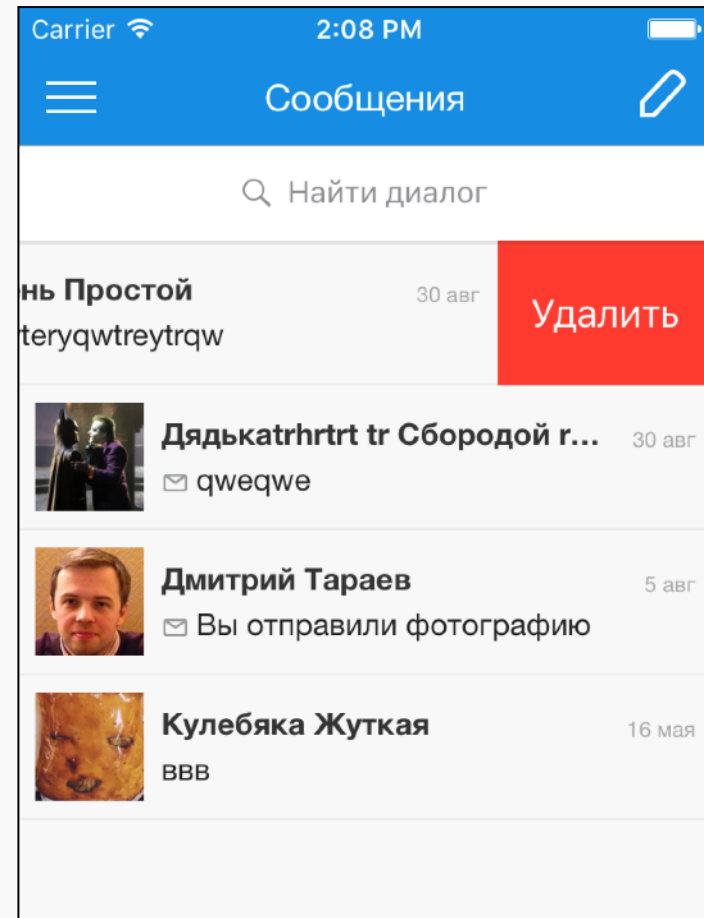
UITableView 9/10



UITableView 10/10



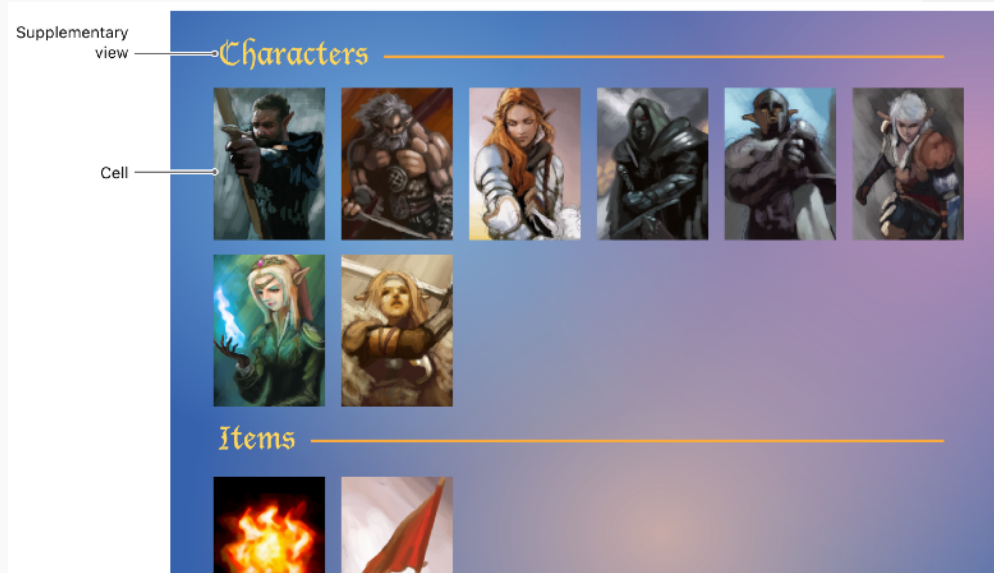
- Режим редактирования



UICollectionView 1/2



- UICollectionView
 - scroll direction
 - vertical / horizontal
 - layout
 - flow / custom
 - разбита на секции?
 - да / нет (header, footer)



UICollectionView 2/2



- Отличия от UITableView
 - NSIndexPath
 - section - item вместо section - row
 - Отдельный класс UICollectionViewLayout для пользовательского layout
 - UICollectionViewCell вместо UITableViewCell
 - UICollectionViewReusableView - для header и footer

1. UITableViewController

2. UIViewController с table view вручную

Заключение



- **UITableView & UICollectionView**
 - data source & delegate
- **Следующая лекция**
 - UIView
- **Отзыв**
- **Вопросы?**
 - по лекции
 - по домашним заданиям

- Подробнее про кастомный layout у collection view:
<https://www.objc.io/issues/3-views/collection-view-layouts/>