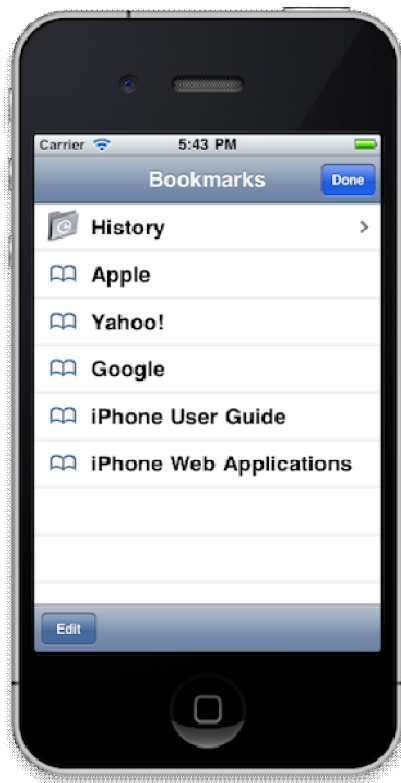


Разработка iOS приложений

Лекция 5.

Стандартные ViewController'ы

UITableViewController



UITableViewController



UITableViewController



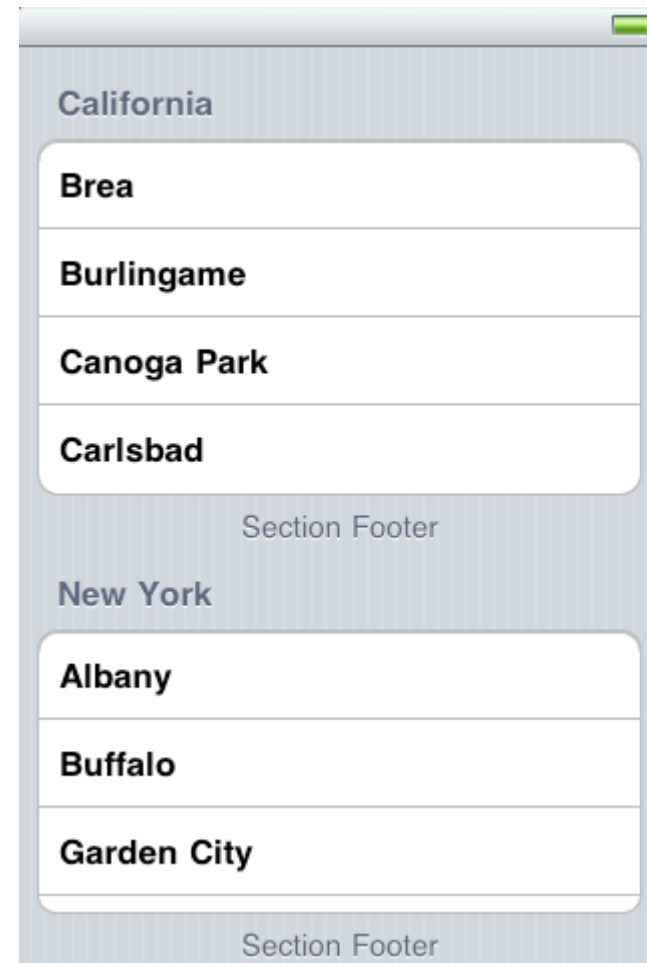
UITableViewController



UITableViewStyle

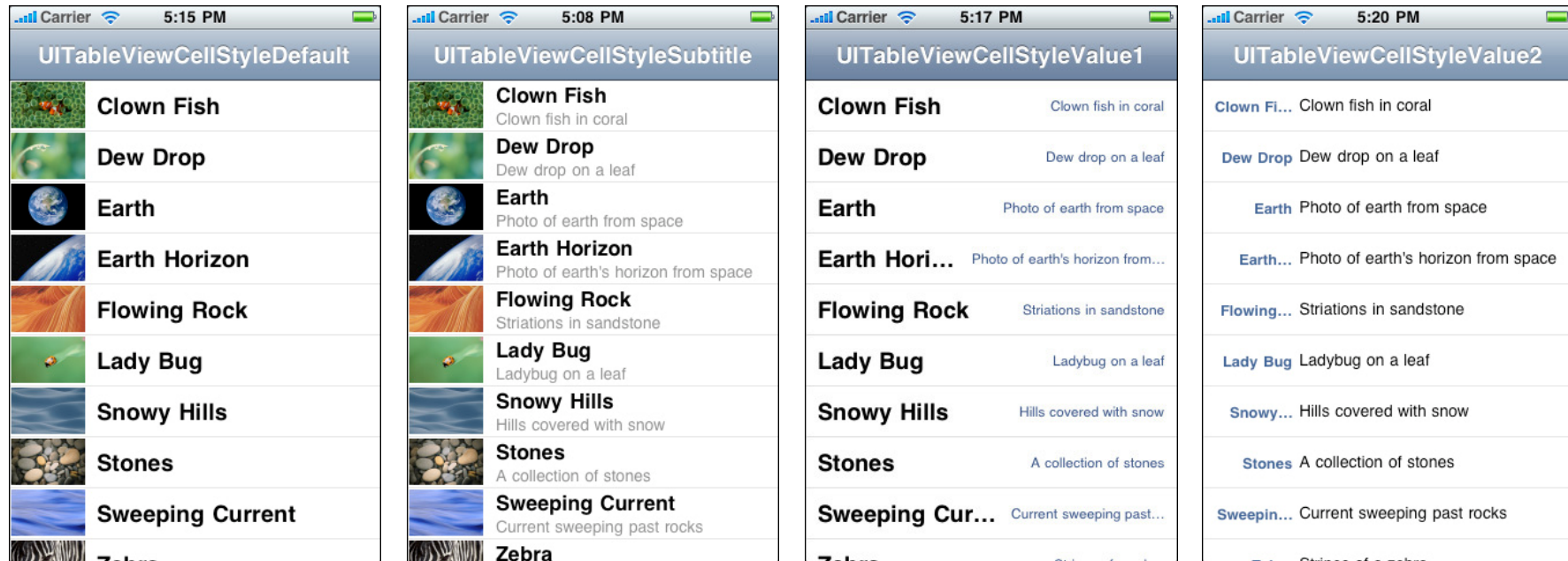


Plain



Grouped

UITableViewCellStyle



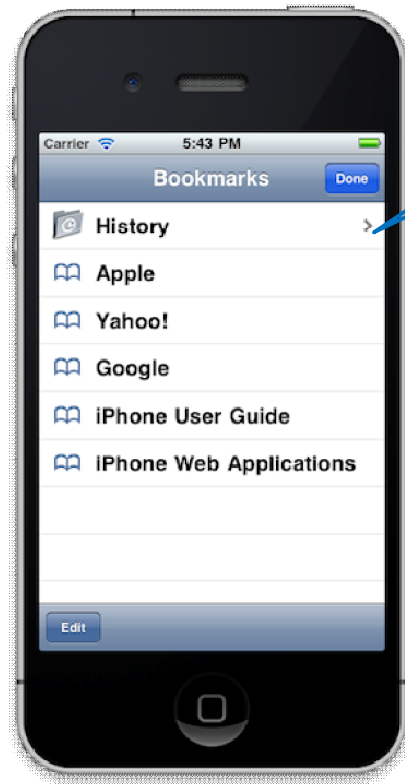
Default

Subtitle

Value1

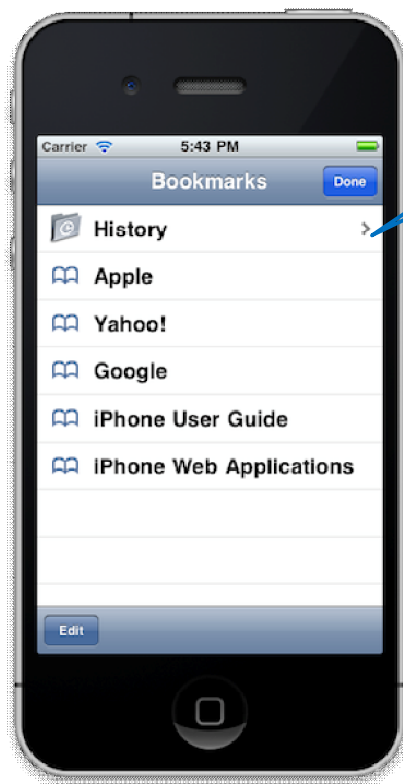
Value2

AccessoryView



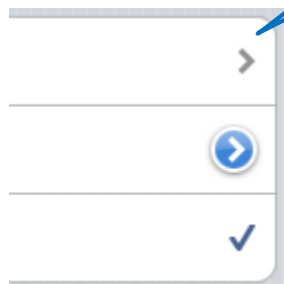
Accessory

AccessoryView

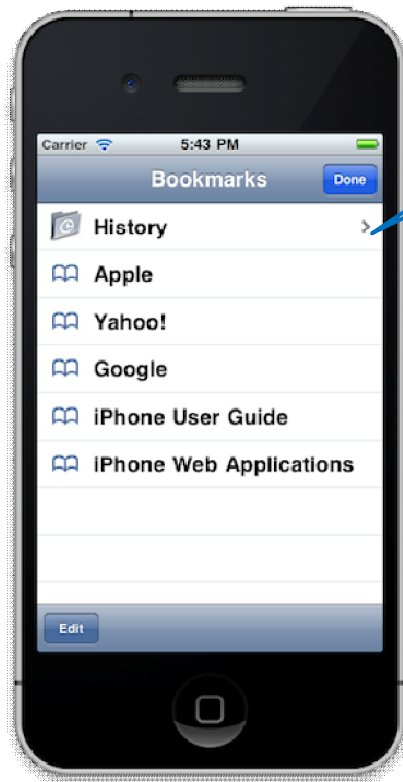


Accessory

При нажатии на ячейку
будет переход

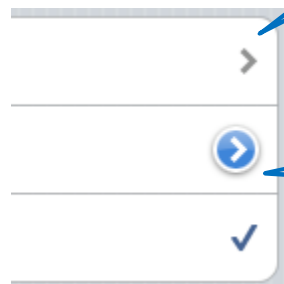


AccessoryView



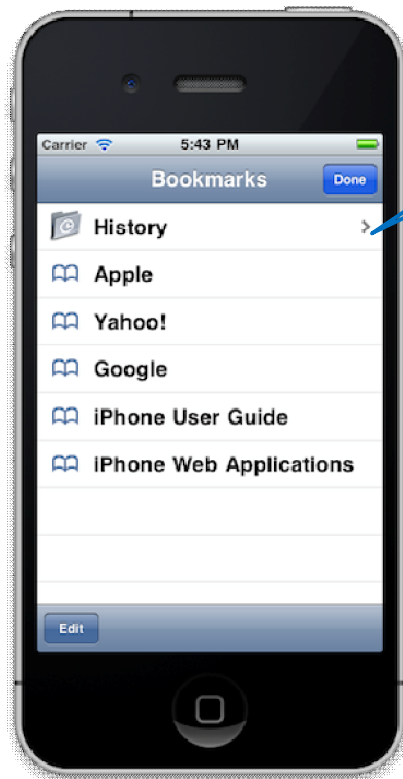
Accessory

При нажатии на ячейку
будет переход



При нажатии на СТРЕЛОЧКУ
будет отображена
дополнительная информация

AccessoryView



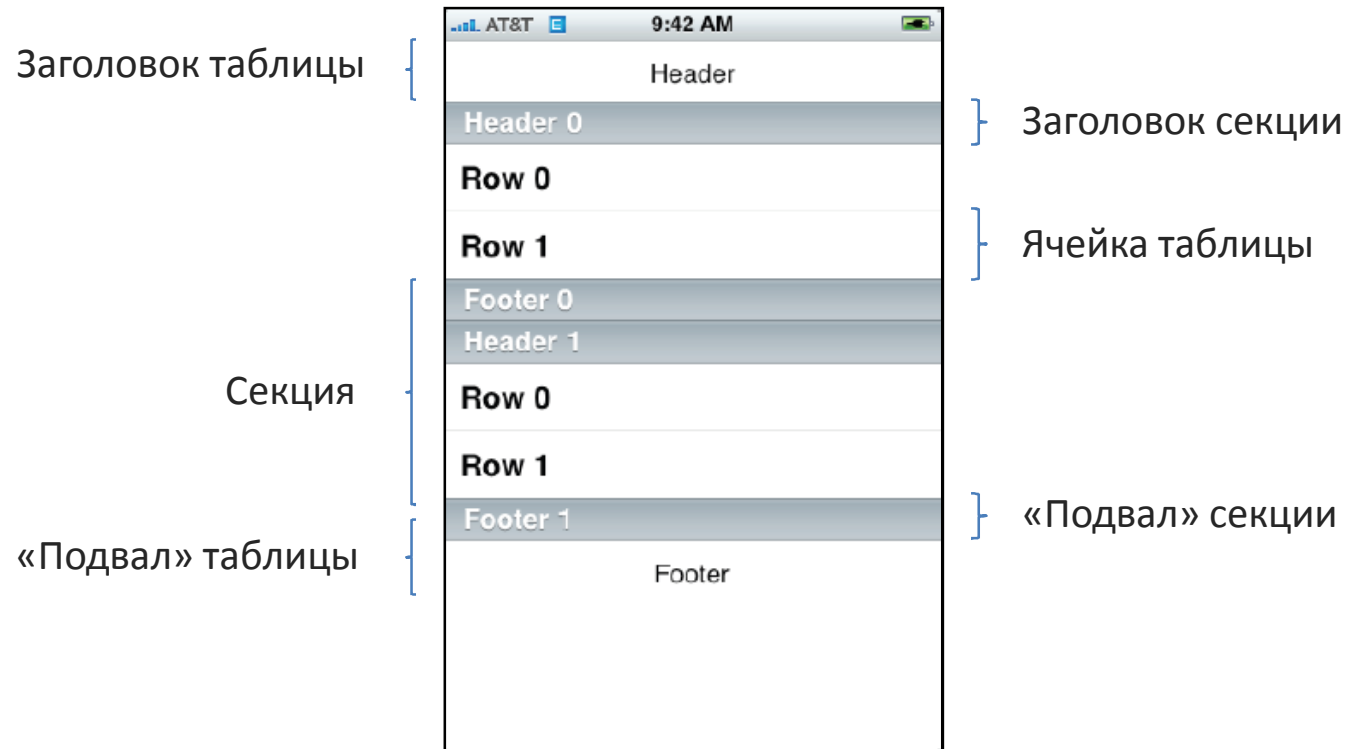
Accessory

При нажатии на ячейку
будет переход

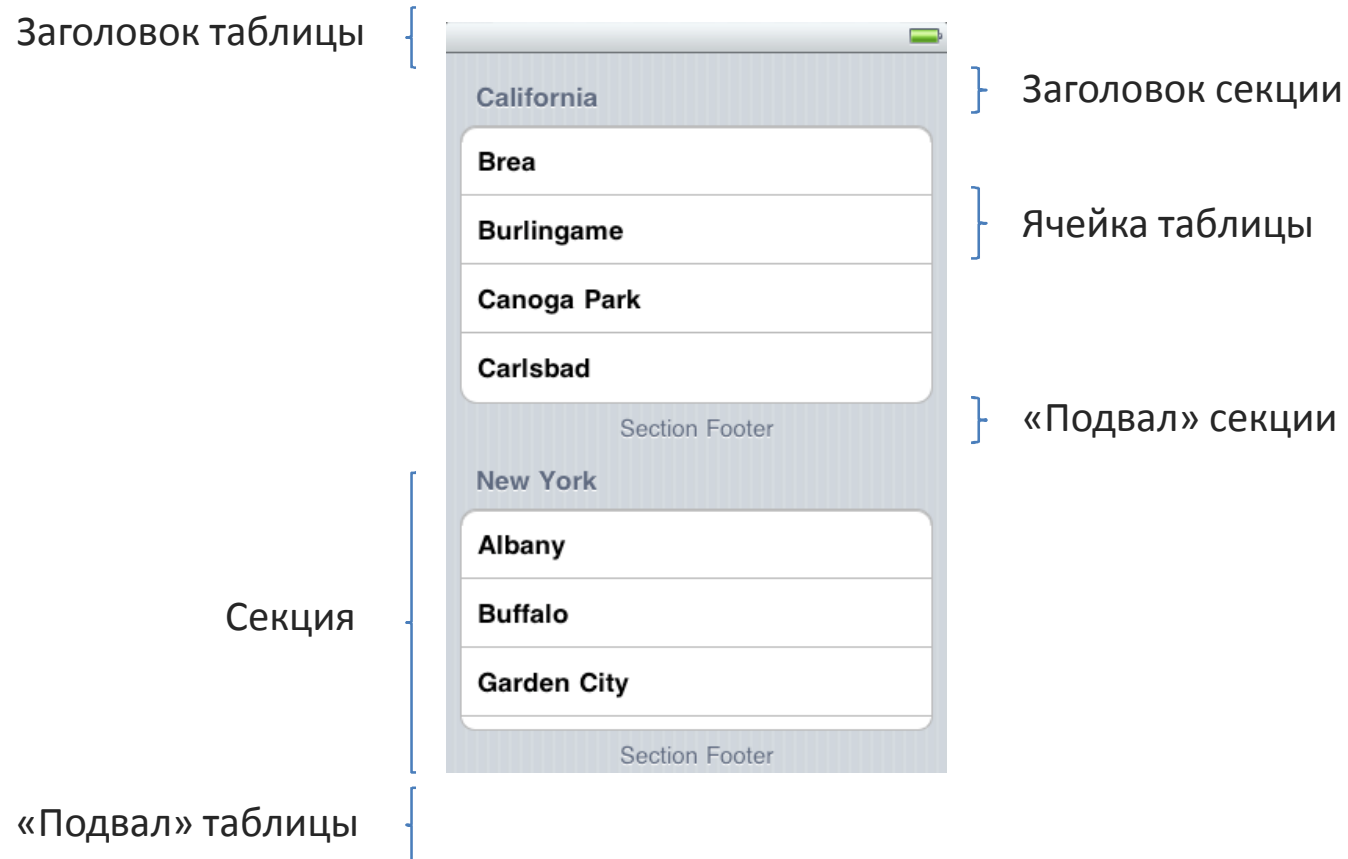
При нажатии на СТРЕЛОЧКУ
будет отображена
дополнительная информация

Просто галочка

Элементы TableView (Plain)



Элементы TableView (Grouped)



Протоколы UITableView

- `id< UITableViewDataSource > dataSource`

Используется для получения данных и разметки

- `numberOfSectionsInTableView:`
- `tableView:numberOfRowsInSection:`
- `tableView:cellForRowAtIndexPath:`

- `id< UITableViewDelegate > delegate`

Используется для обработки событий (нажатий, переходов, ...)

- `tableView:didSelectRowAtIndexPath:`

tableView:cellForRowAtIndexPath

```
-( UITableViewCell* ) tableView: ( UITableView* ) tableView
    didSelectRowAtIndexPath: ( NSIndexPath* ) indexPath
{
    static NSString* cellStyle = @"cellStyle";

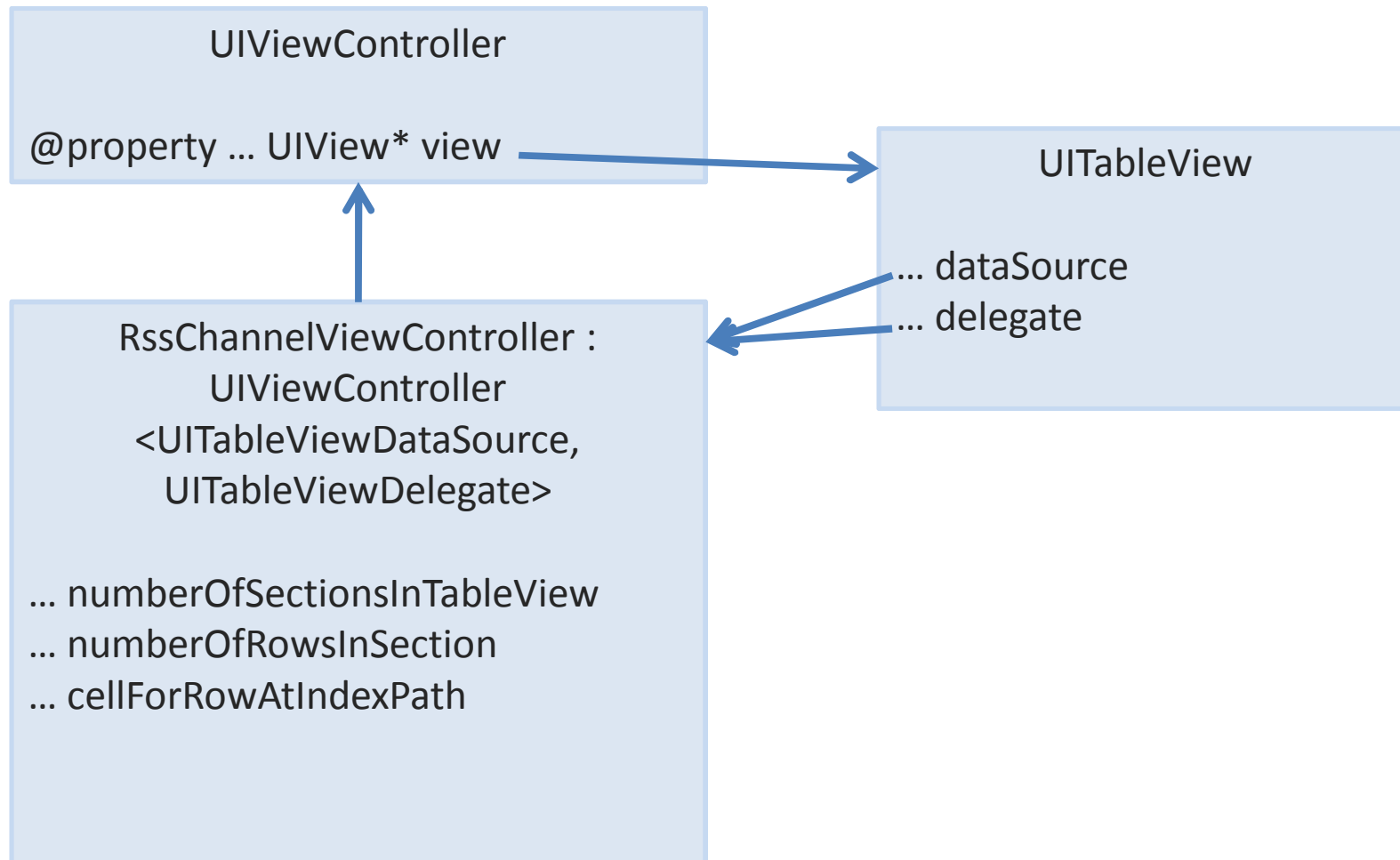
    UITableViewCell* cell = [ tableView
dequeueReusableCellWithIdentifier: cellStyle ];

    if ( nil == cell )
    {
        cell = [ [ [ UITableViewCell alloc ] initWithStyle:
UITableViewCellStyleSubtitle reuseIdentifier: cellStyle ]
autorelease ];
    }

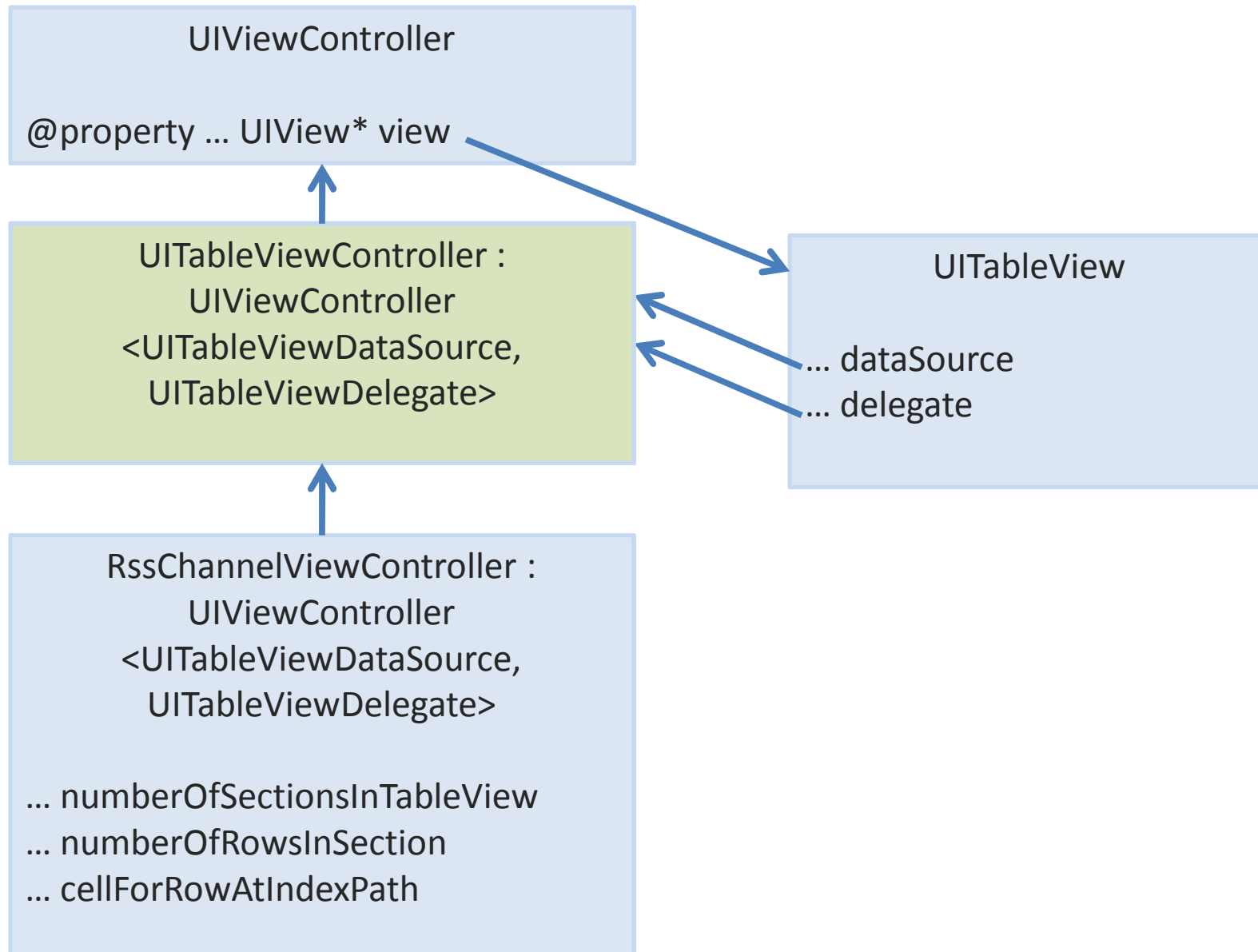
    cell.textLabel.text = @"something";

    return cell;
}
```

UITableViewController



UITableViewController



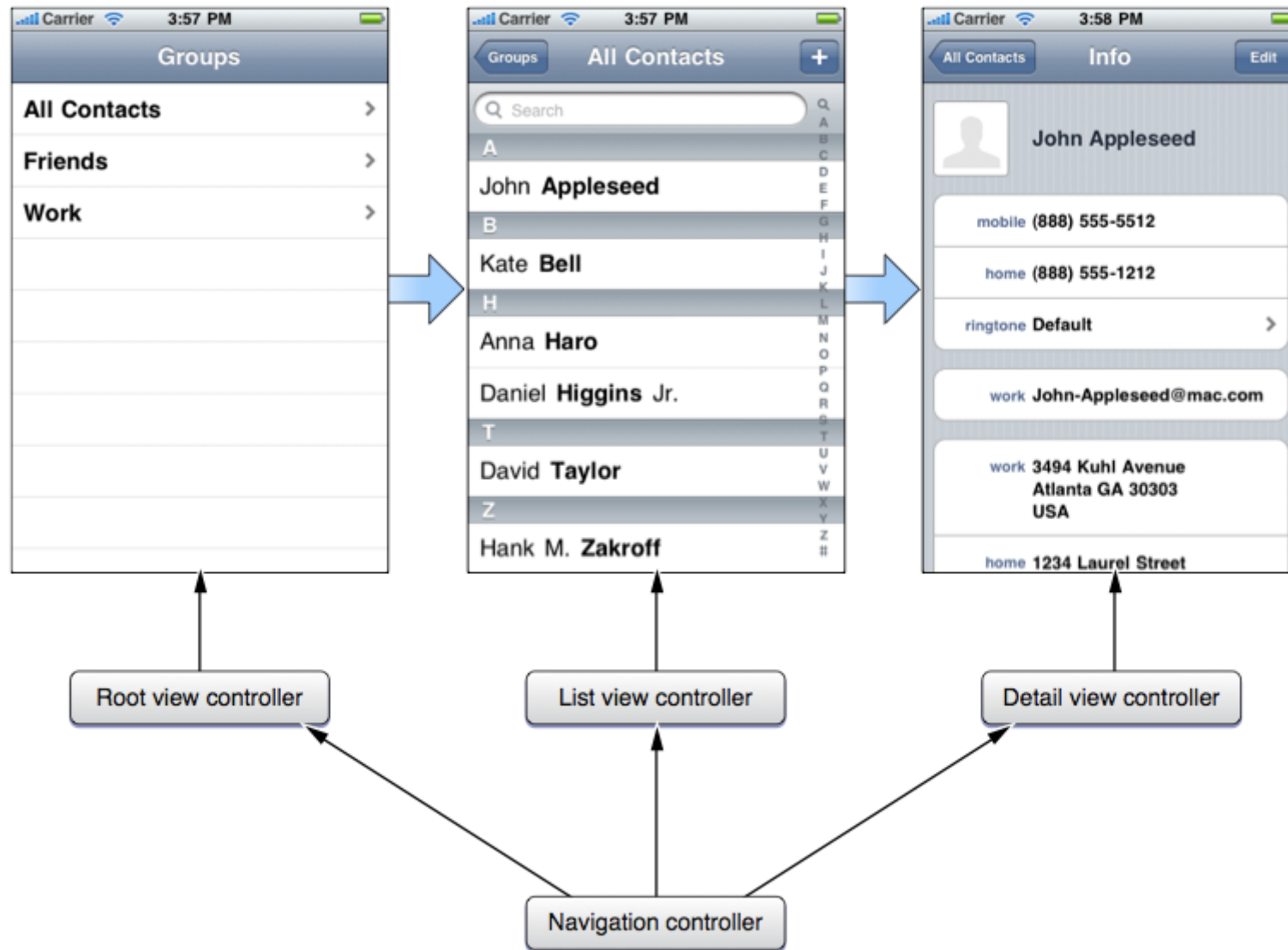
UITableViewController

- Очищает выделение перед отображением
- Моргает полосами прокрутки при завершении отображения
- Устанавливает размер UITableView
- И еще много полезных вспомогательных вещей

ОДНАКО!

Иногда бывает полезно сделать это руками самостоятельно. В основном по требованиям дизайна

UINavigationController



UINavigationController

- Добавляет сверху UINavigationController и управляет кнопкой «Назад»
- Хранит стек ViewController'ов в него помещенных
- Чтобы перейти на новый – pushViewController:animated:
- Чтобы перейти к предыдущему – popViewController:animated:

Создание (в applicationDidFinishLaunchingWithOptions:):

```
UIViewController* ctrl = [ CustomViewController viewController ];
```

```
UINavigationController* naviCtrl = [ [ UINavigationController  
alloc ] initWithRootViewController: ctrl ] autorelease ];
```

```
self.window.rootViewController = naviCtrl;
```

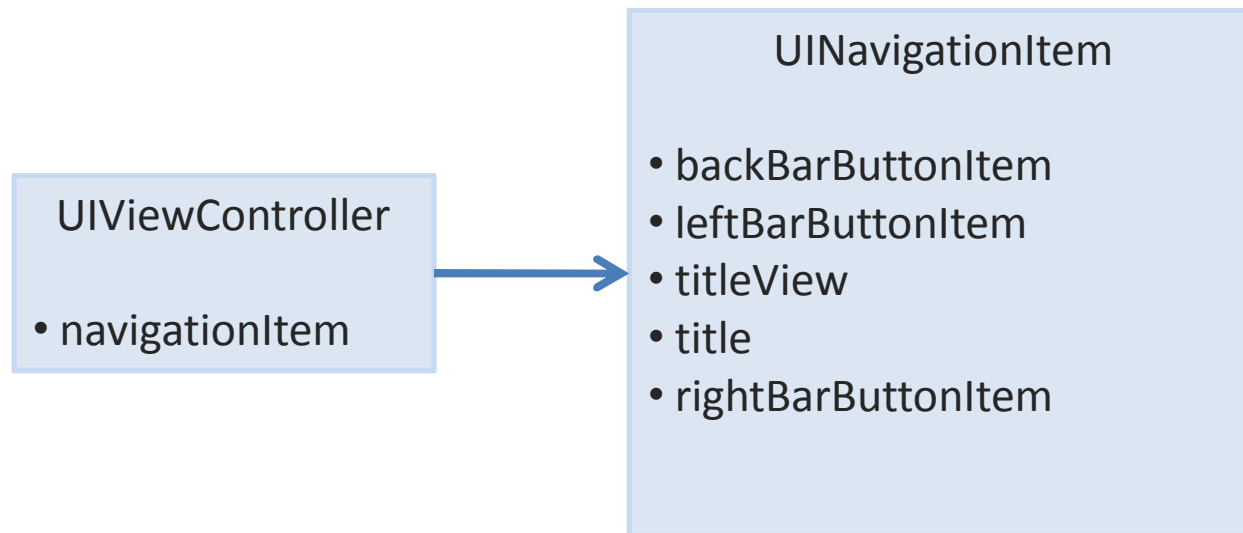
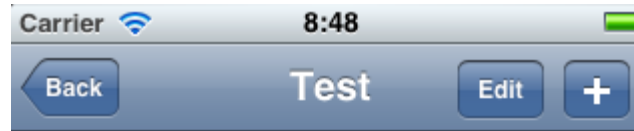
UINavigationController

Переход на новый контроллер:

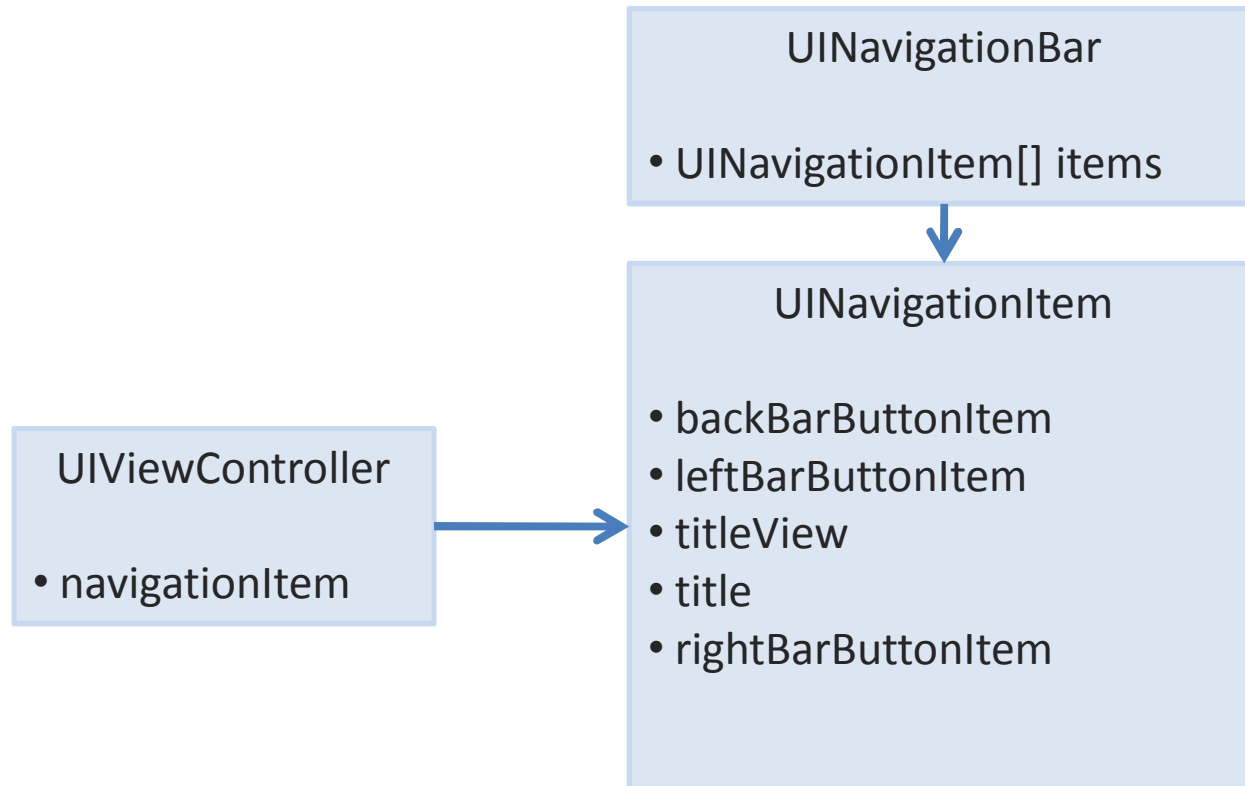
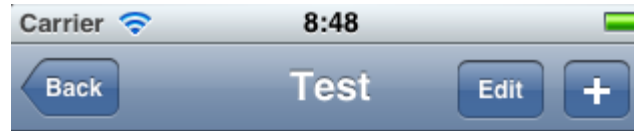
```
NewViewController* ctrl = [ NewViewController viewController ];  
ctrl.Something = anything;  
ctrl.delegate = self;
```

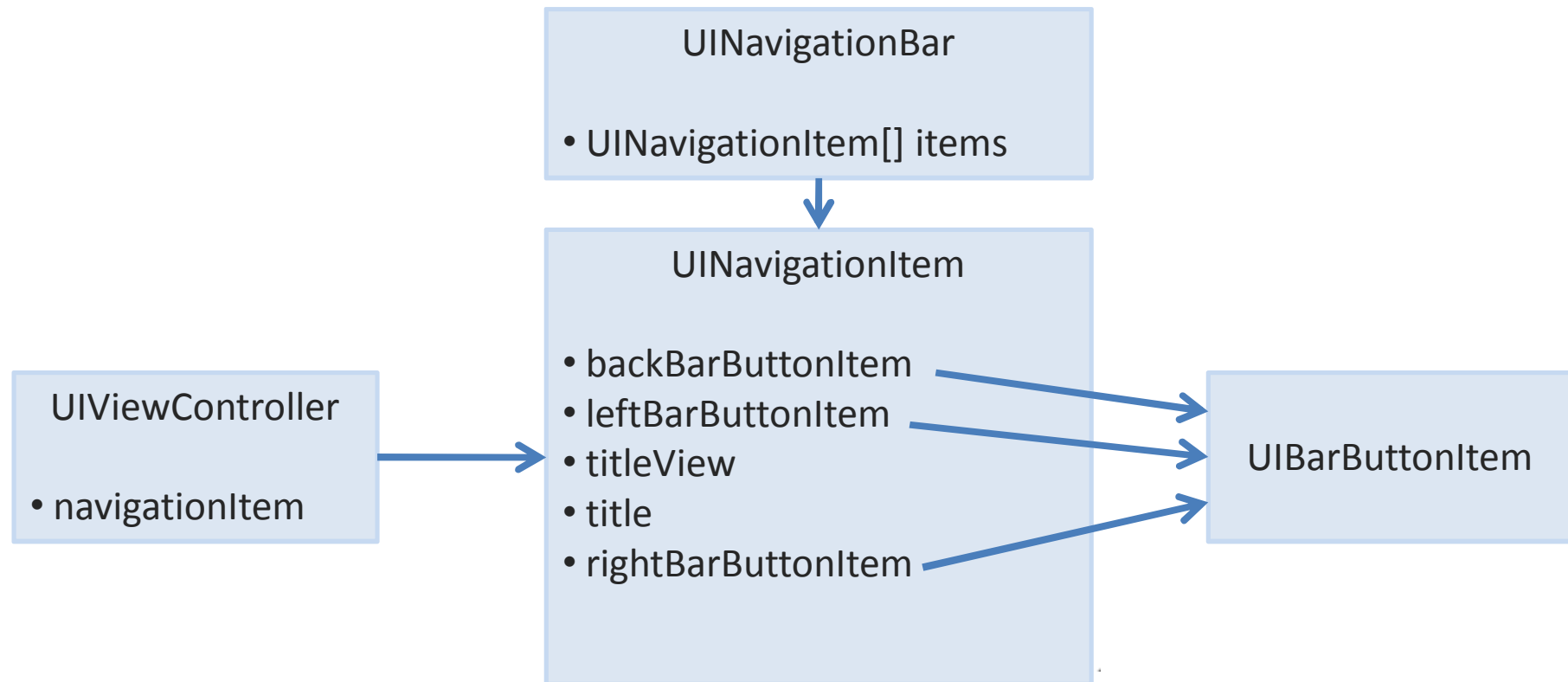
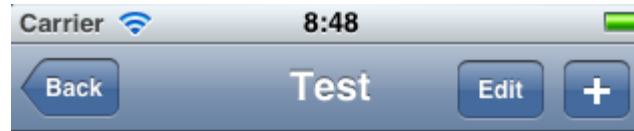
```
[self.navigationController pushViewController: ctrl animated: YES];
```

UINavigationController

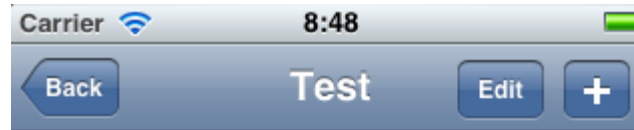


UINavigationController





UINavigationController



previous

UINavigationController

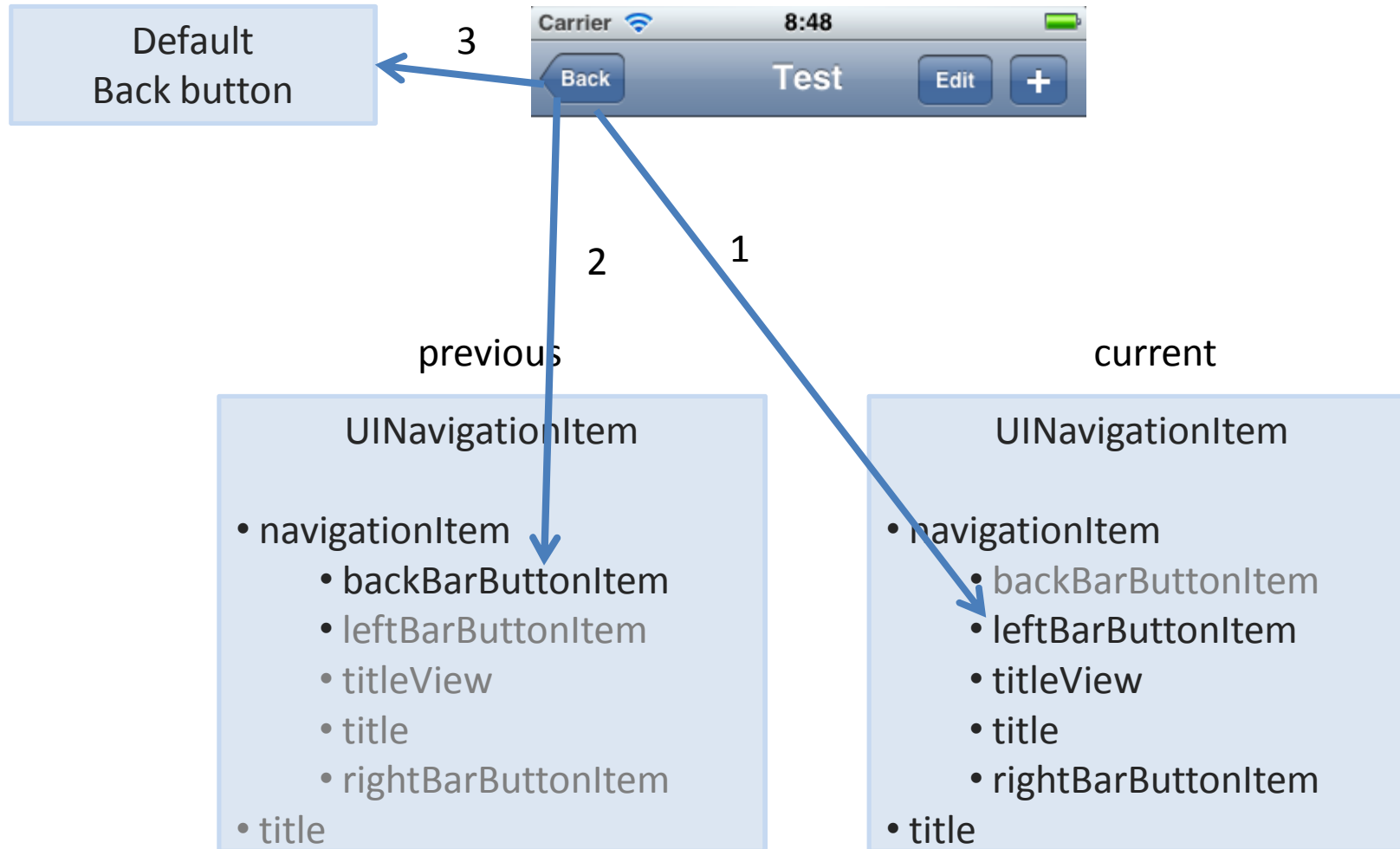
- navigationItem
 - backBarButtonItem
 - leftBarButtonItem
 - titleView
 - title
 - rightBarButtonItem
- title

current

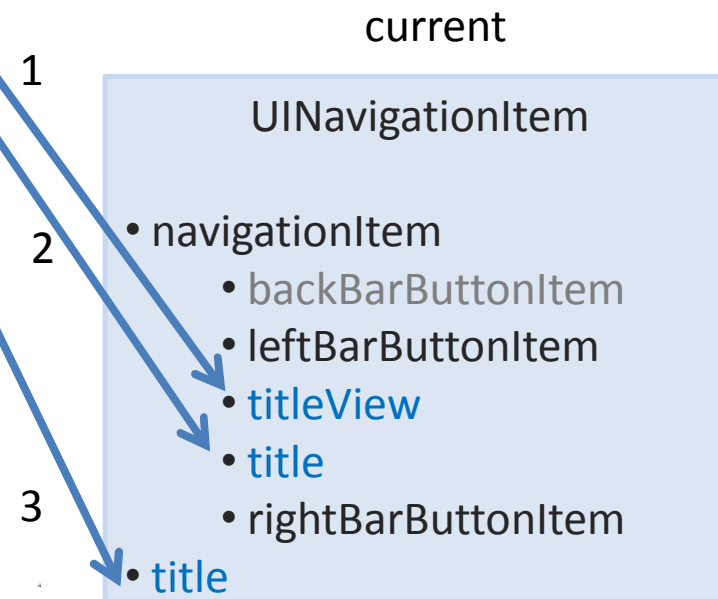
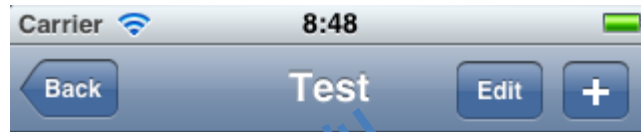
UINavigationController

- navigationItem
 - backBarButtonItem
 - leftBarButtonItem
 - titleView
 - title
 - rightBarButtonItem
- title

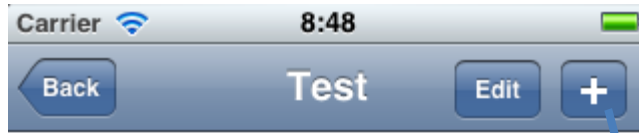
UINavigationController



UINavigationController



UINavigationController



current

UINavigationController

- UINavigationController
 - backButtonItem
 - leftBarButtonItem
 - titleView
 - title
 - rightBarButtonItem
- title

UINavigationController

Создание:

`initWithBarButtonSystemItem:target:action:`

`initWithTitle:target:action:`

`initWithImage:target:action:`

`initWithCustomView:`

Практическое задание

Практикуемся с UITableView и UITableViewController

1. Главная задача – сделать отображение RSSChannel в виде таблице RSSItem'ов
 1. Создать RssChannelViewController : UITableViewController
 2. На каждую ячейку UITableViewCell один RSSItem (UITableViewCellStyleSubtitle)
 3. При нажатии на конкретную UITableViewCell сделать переход на уже сделанный нами RssItemViewController
 4. Для переходов использовать UINavigationController

См. Apple Guides

- Table View Programming Guide for iOS
- Populating the Table View with Data
- View Controller Programming Guide for iOS

Для тех кто быстро справится

- Проверить использование памяти (Analyzer + Profiler)
- В первой ячейке вывести количество item'ов
- * Сгруппировать item'ы по дате