

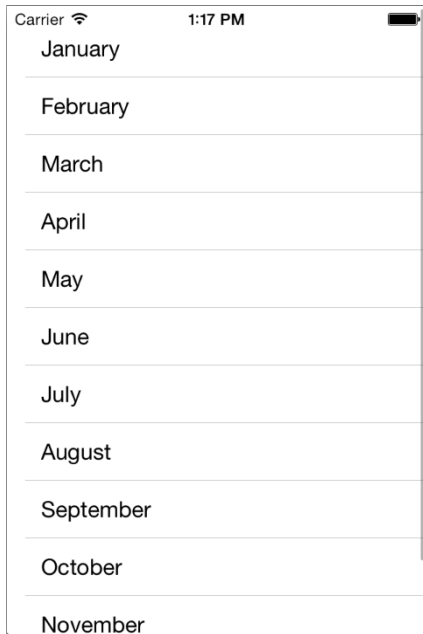
# Разработка iOS приложений

Лекция 5.

Стандартные ViewController'ы

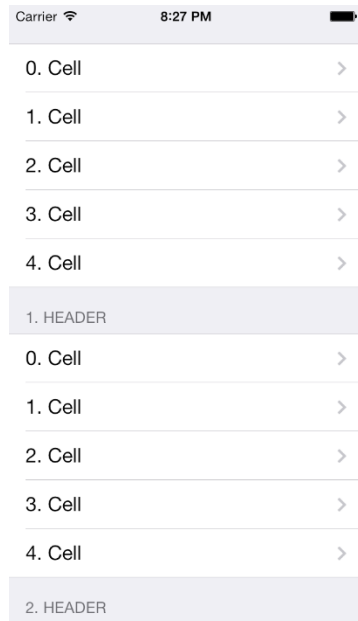
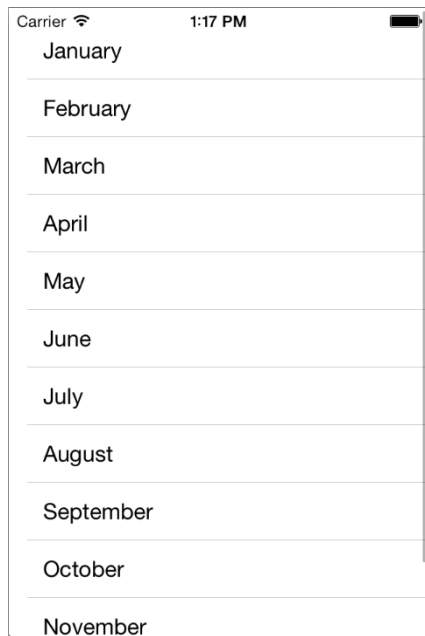
# UITableViewController

---



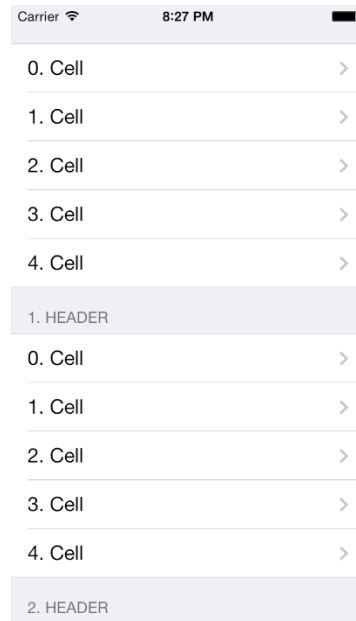
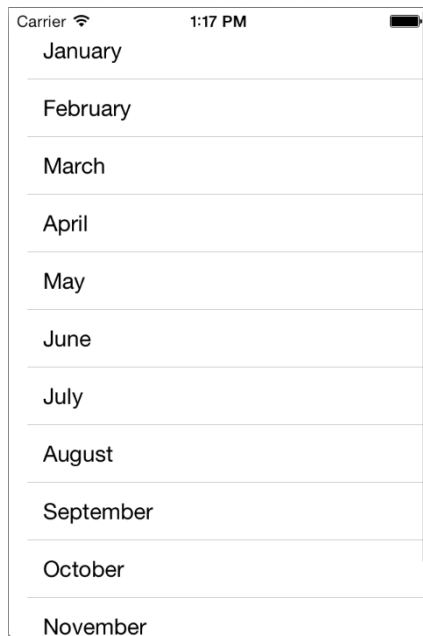
# UITableViewController

---

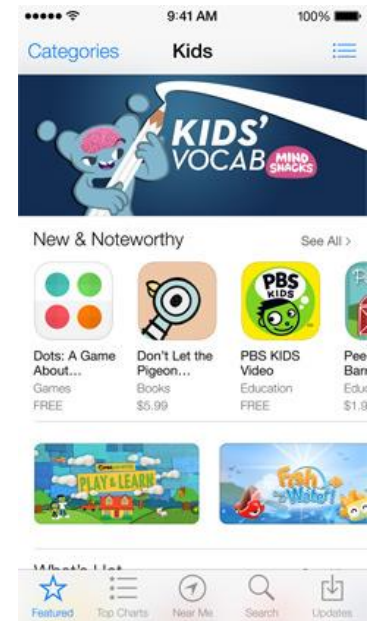
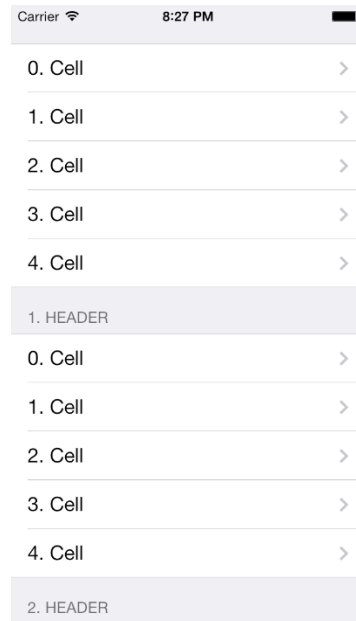
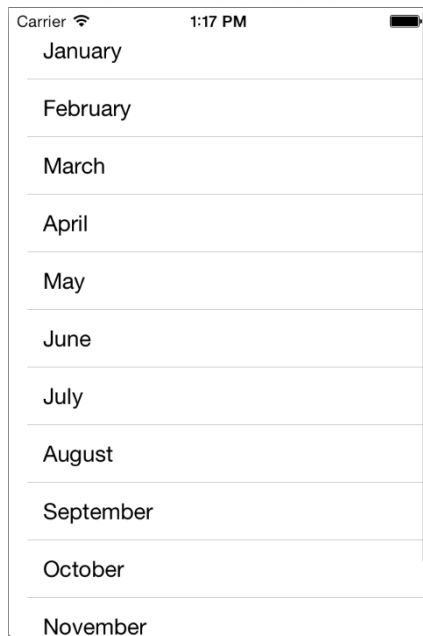


# UITableViewController

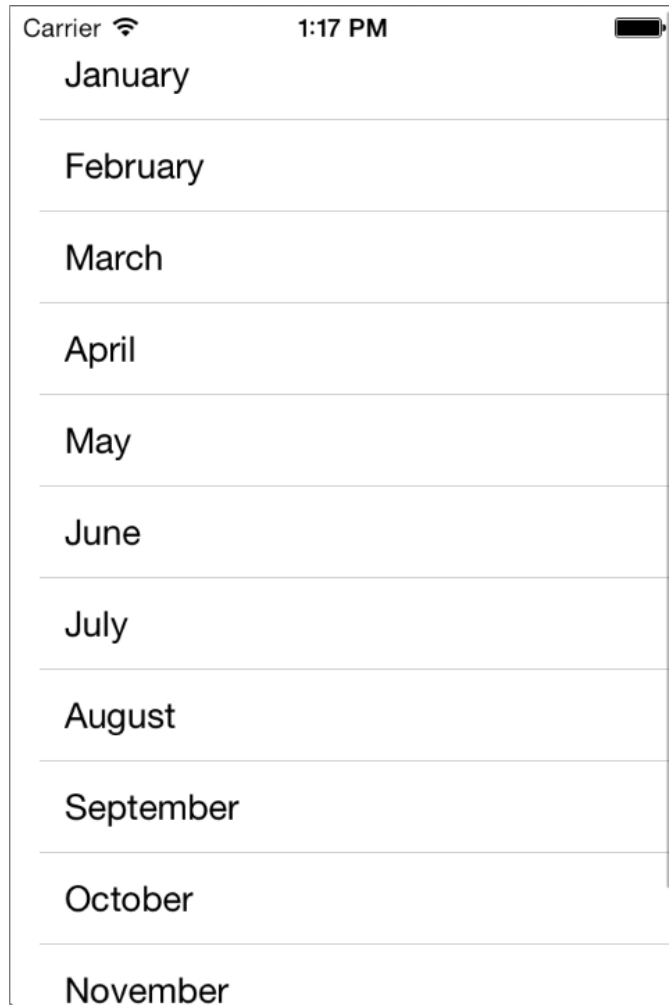
---



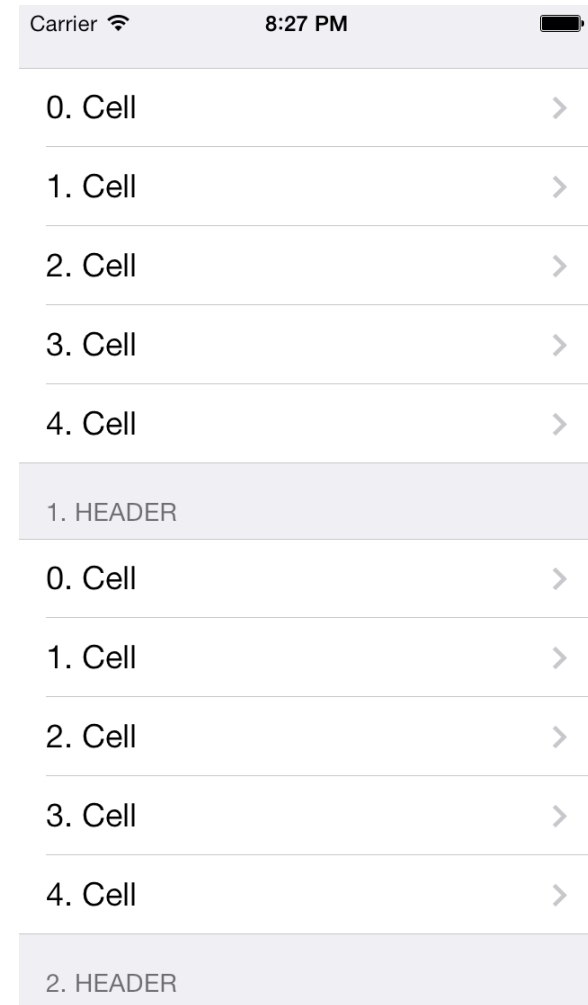
# UITableViewController



# UITableViewStyle









Plain









Grouped







# UITableViewCellStyle

Default Cell Style	
	Text Label
	Dahlia
	Daisies
	Dandelion
	Echinacea
	Lavender

Default

Subtitle Cell Style	
	Text Label Detail text label
	Dahlia This is a dahlia
	Daisies These are daisies
	Dandelion This is a dandelion
	Echinacea This is echinacea
	Lavender

Subtitle

Value 1 Cell Style	
	Text Label      Detail text label
	Dahlia      This is a dahlia
	Daisies      These are daisies
	Dandelion      This is a dandelion
	Echinacea      This is echinacea
	Lavender      This is a field of lav...

Value1

Value 2 Cell Style	
<a href="#">Text Label</a>	Detail text label
<a href="#">Dahlia</a>	This is a dahlia
<a href="#">Daisies</a>	These are daisies
<a href="#">Dandelion</a>	This is a dandelion
<a href="#">Echinacea</a>	This is echinacea
<a href="#">Lavender</a>	This is a field of lavender

Value2

# AccessoryView

---

September 11, 2013

(0:03)

Type: recording 1

(0:03)

Type: Recording 2



Accessory



# AccessoryView

---

September 11, 2013

(0:03)

Type: recording 1



(0:03)

Type: Recording 2



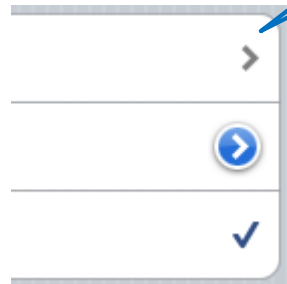
Accessory

# AccessoryView

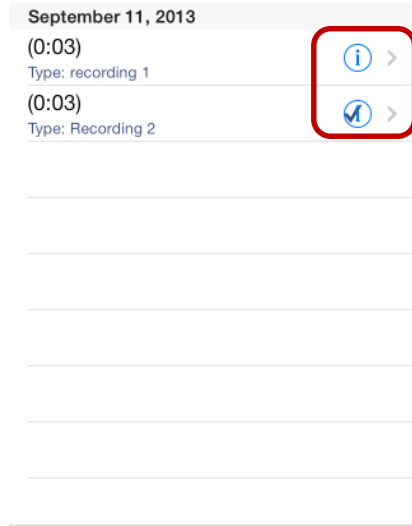
September 11, 2013	
(0:03)	 >
Type: recording 1	
(0:03)	 >
Type: Recording 2	

Accessory

При нажатии на ячейку  
будет переход

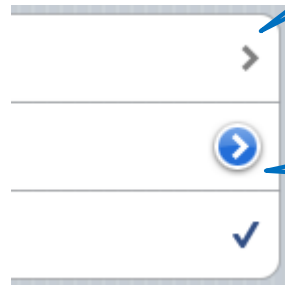


# AccessoryView



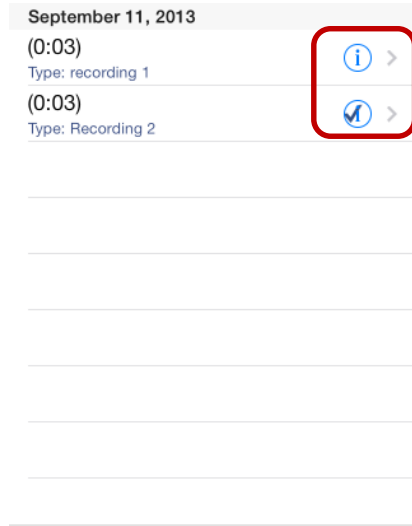
Accessory

При нажатии на ячейку  
будет переход



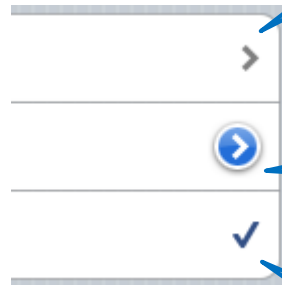
При нажатии на СТРЕЛОЧКУ  
будет отображена  
дополнительная информация

# AccessoryView



Accessory

При нажатии на ячейку  
будет переход



При нажатии на СТРЕЛОЧКУ  
будет отображена  
дополнительная информация

Просто галочка

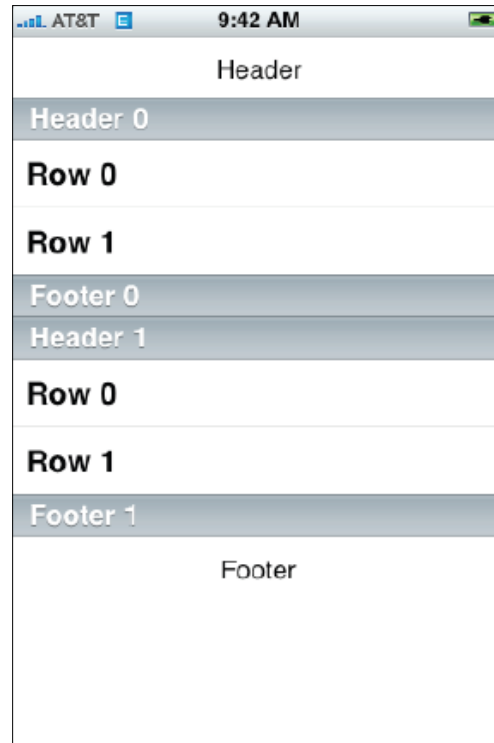
# Элементы TableView (Plain)

---

Заголовок таблицы

Секция

«Подвал» таблицы



The screenshot shows an iPhone screen with a status bar at the top displaying 'AT&T', signal strength, and the time '9:42 AM'. Below the status bar is a UITableView. The table contains the following elements from top to bottom: a 'Header' row, a section header 'Header 0', two data rows 'Row 0' and 'Row 1', a section footer 'Footer 0', another section header 'Header 1', two more data rows 'Row 0' and 'Row 1', a section footer 'Footer 1', and finally a table footer 'Footer'.

Header
Header 0
Row 0
Row 1
Footer 0
Header 1
Row 0
Row 1
Footer 1
Footer

Заголовок секции

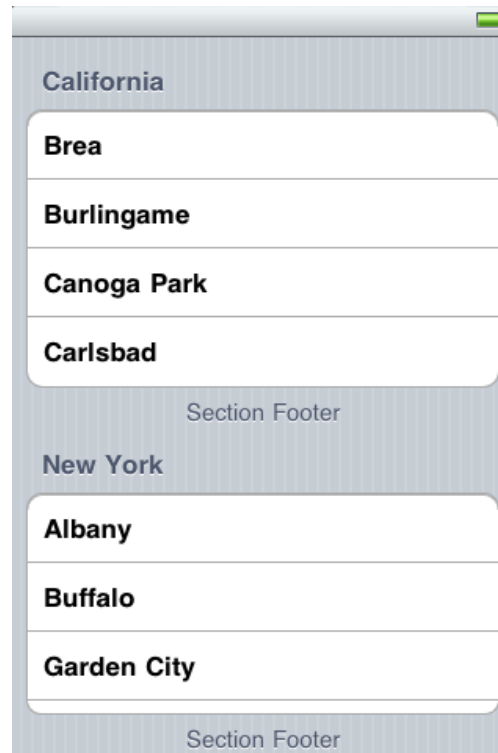
Ячейка таблицы

«Подвал» секции

# Элементы TableView (Grouped)

---

Заголовок таблицы {



} Заголовок секции

} Ячейка таблицы

} «Подвал» секции

Секция {

{ «Подвал» таблицы

# 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 ];
    }

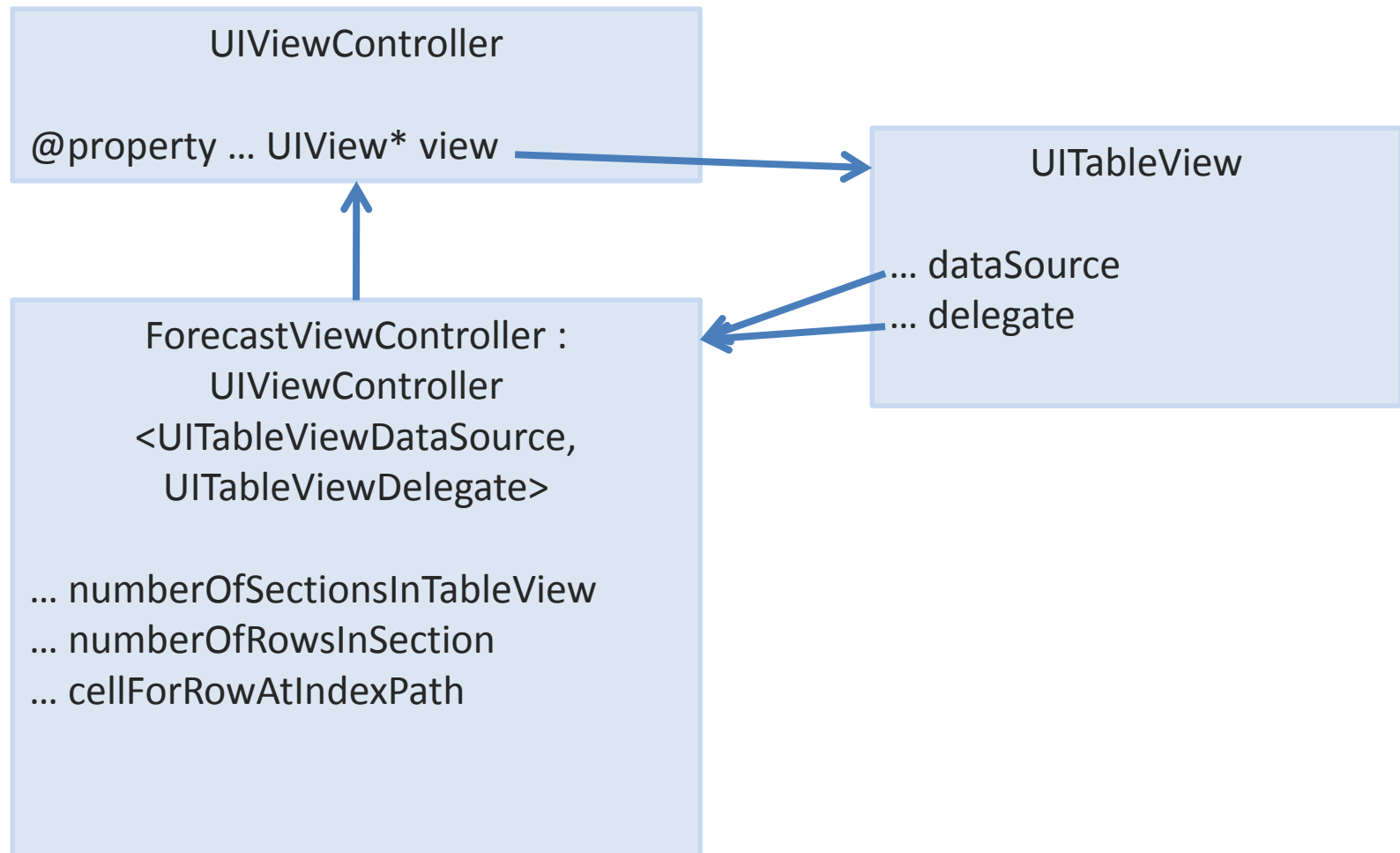
    cell.textLabel.text = @"something";

    return cell;
}
```



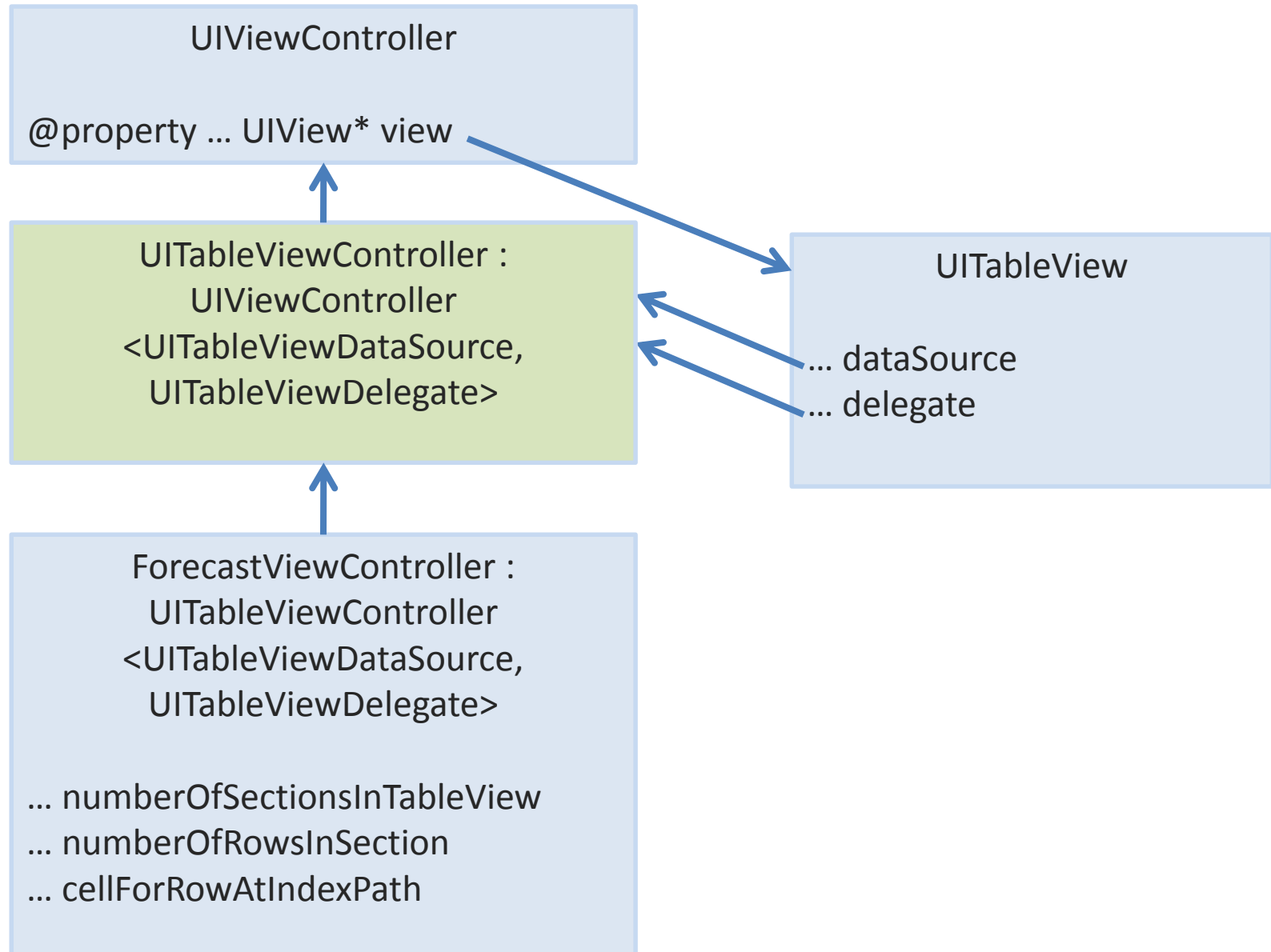
# UITableViewController

---



# UITableViewController

---



# UITableViewController

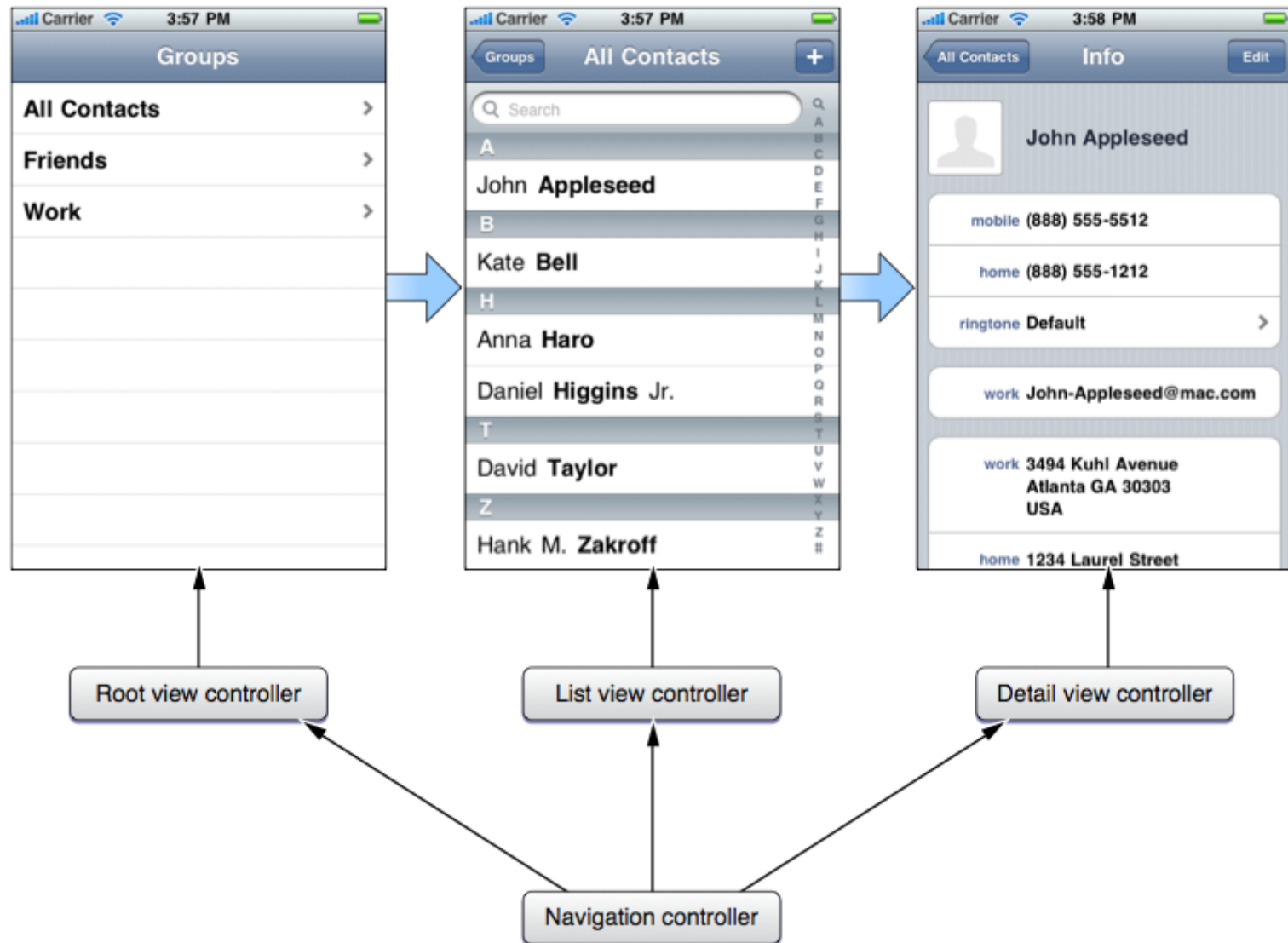
---

- 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

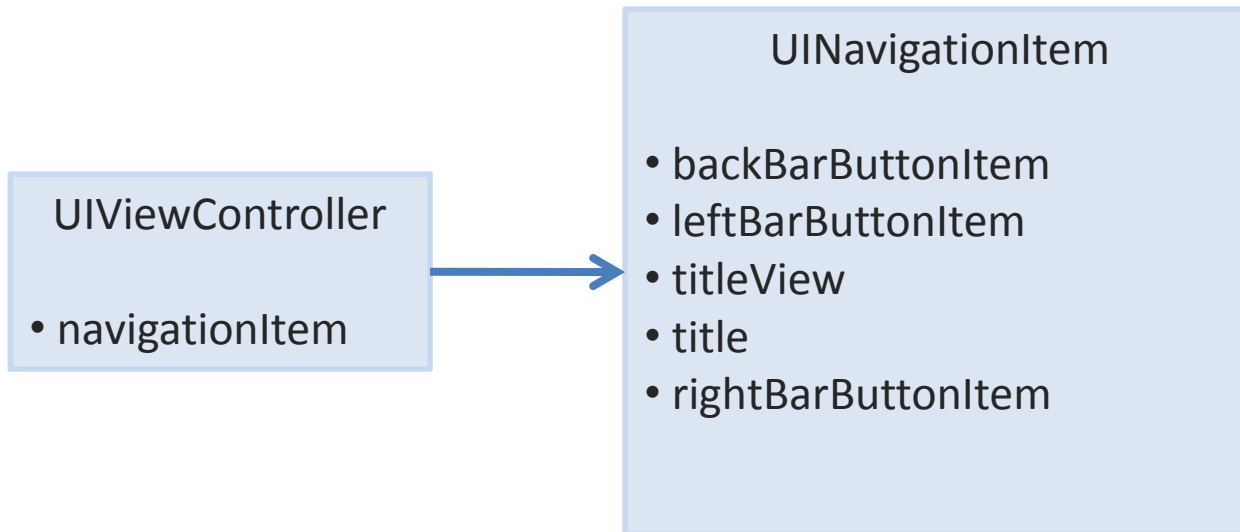
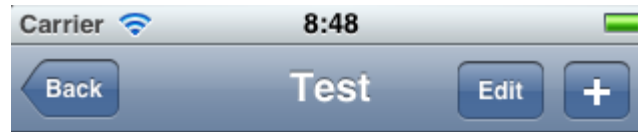
---

Переход на новый контроллер:

```
NewViewController* ctrl = [ NewViewController viewController ];  
ctrl.Something = anything;  
ctrl.delegate = self;  
  
[self.navigationController pushViewController: ctrl animated: YES];
```

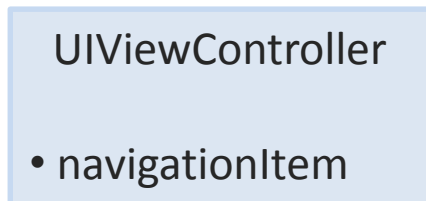
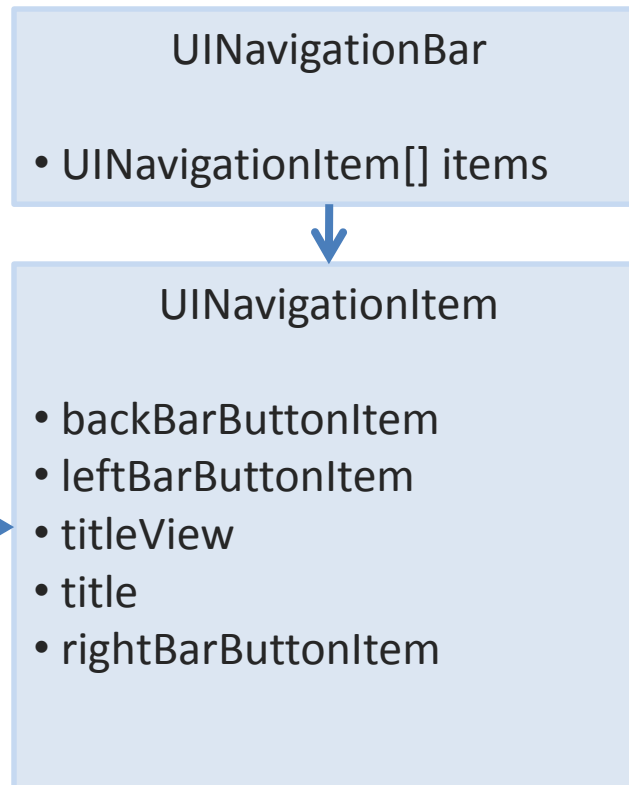
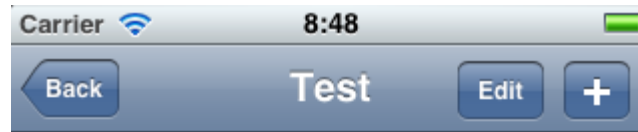
# UINavigationController

---



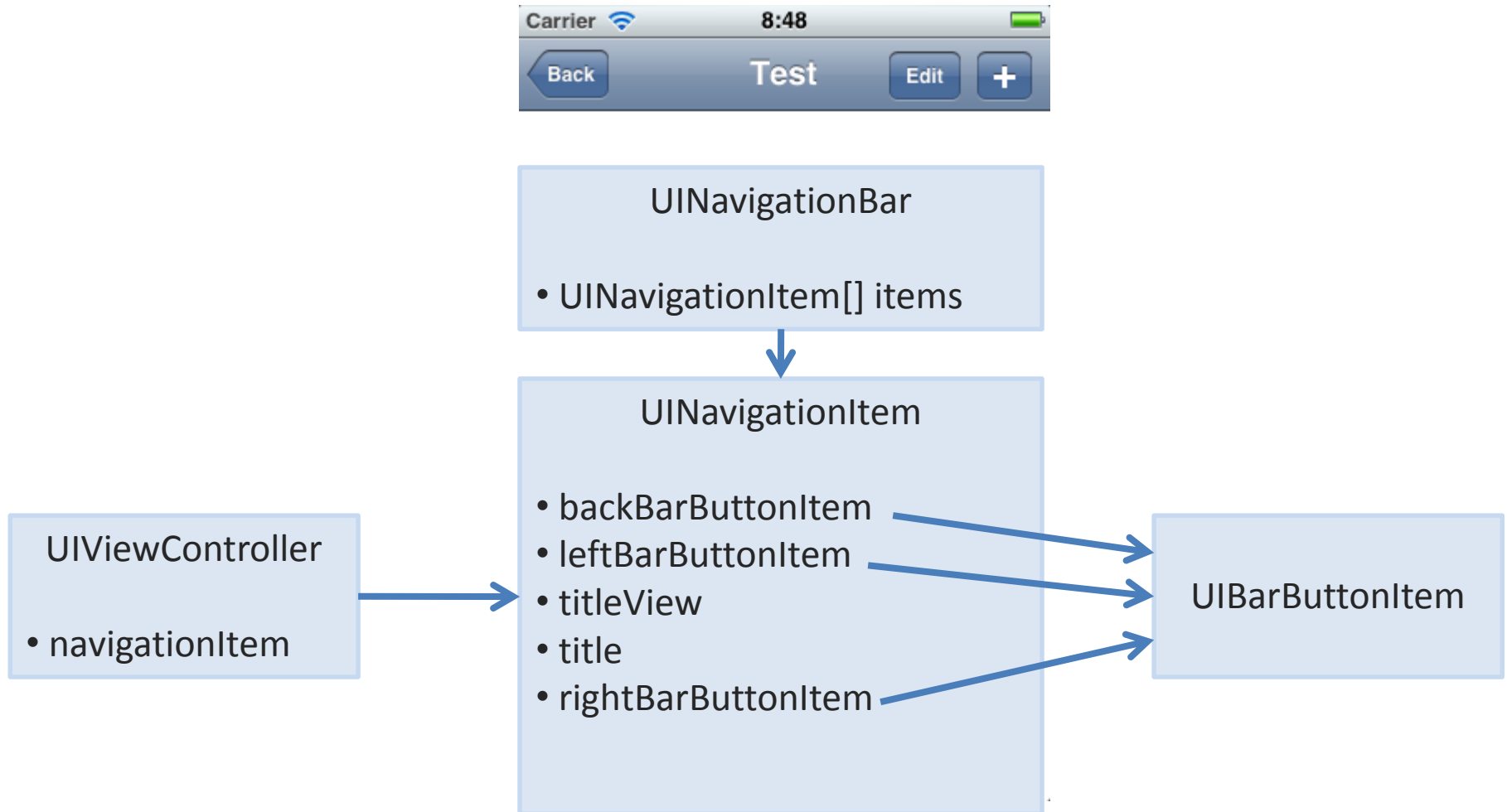
# UINavigationController

---



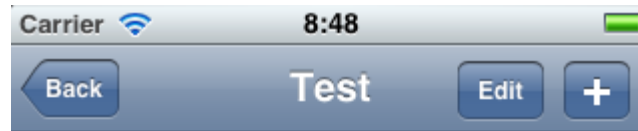


# UINavigationController



# UINavigationController

---



previous

UINavigationControllerItem

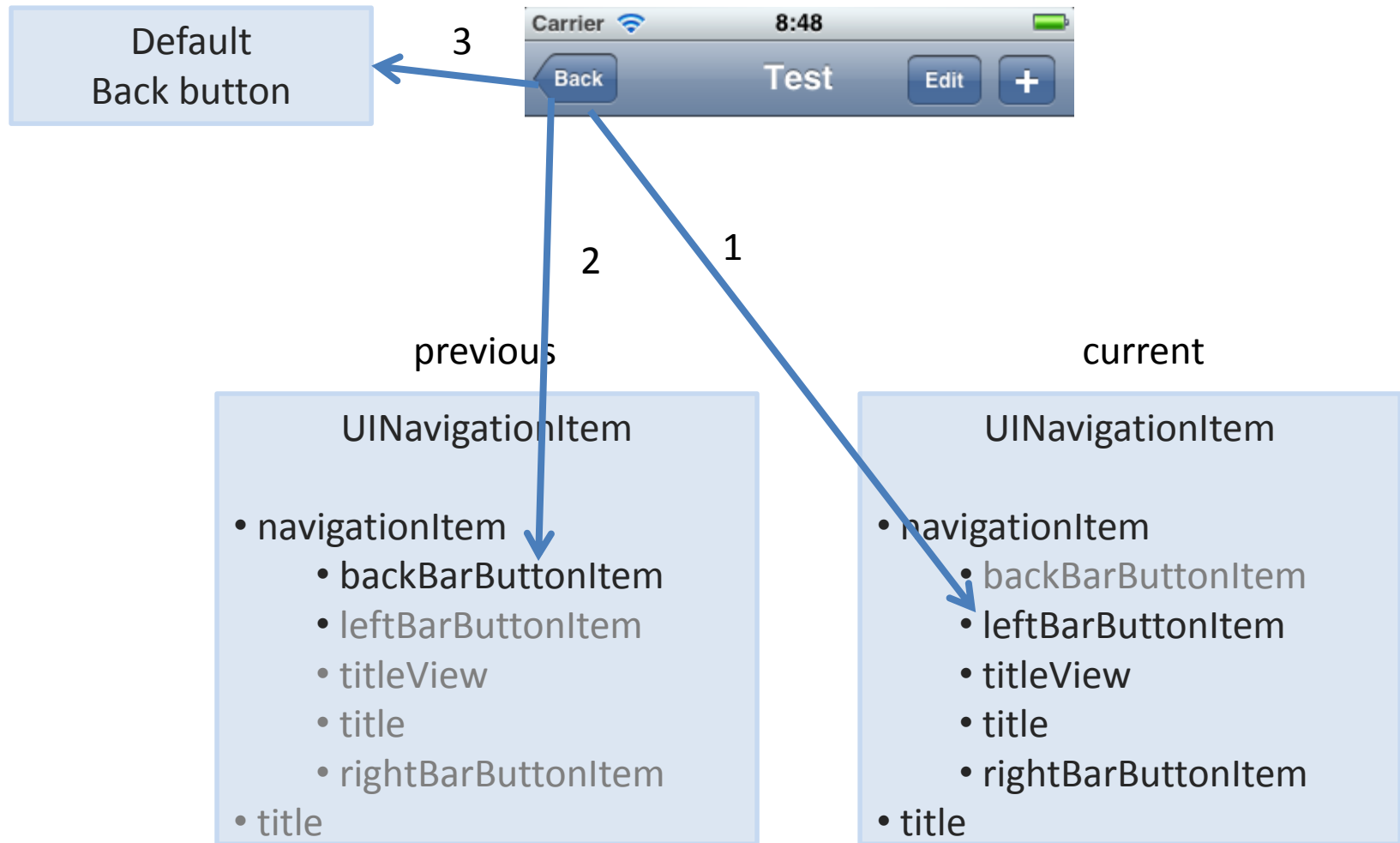
- navigationItem
  - backButtonItem
  - leftBarButtonItem
  - titleView
  - title
  - rightBarButtonItem
- title

current

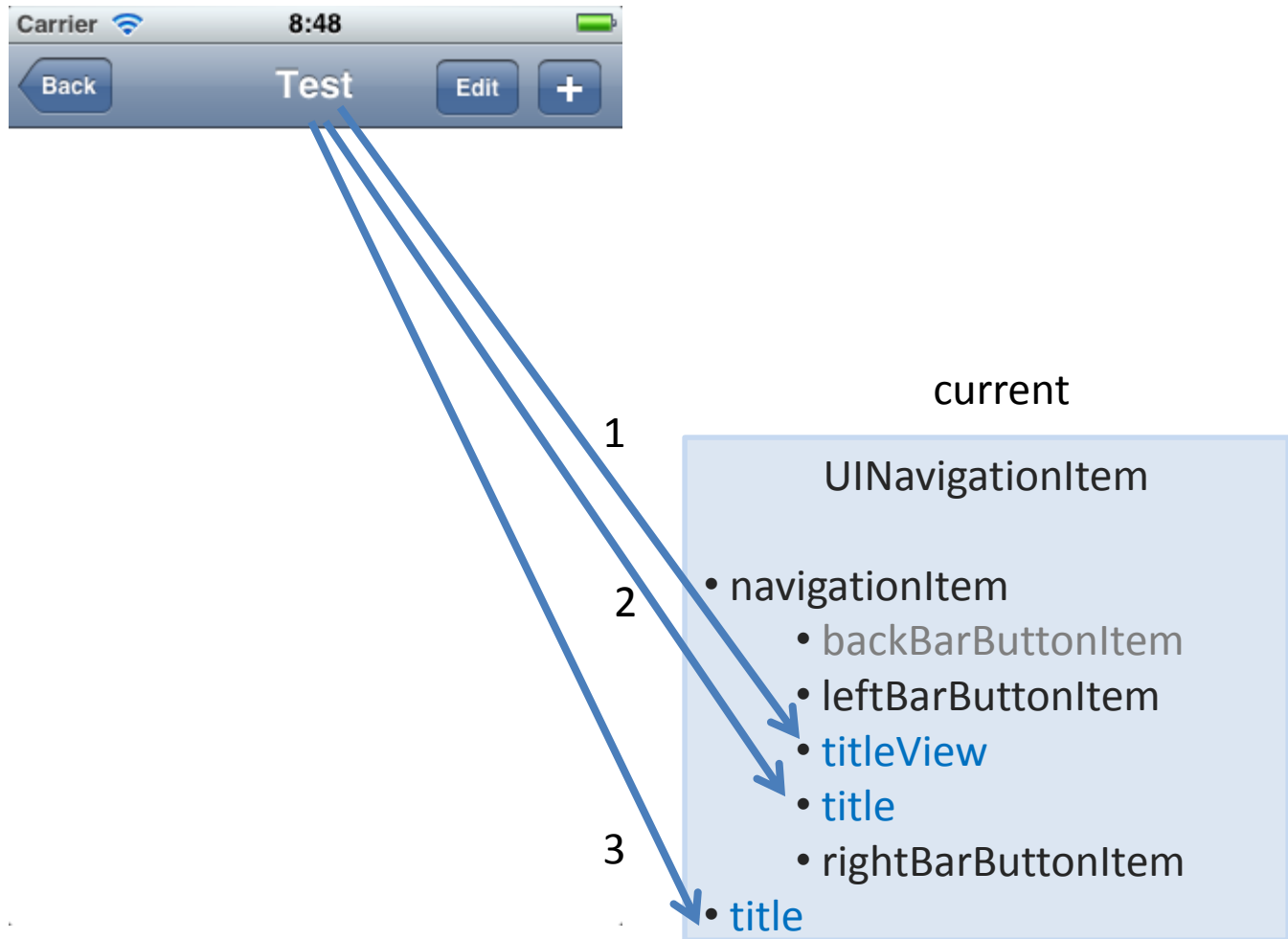
UINavigationControllerItem

- navigationItem
  - backButtonItem
  - leftBarButtonItem
  - titleView
  - title
  - rightBarButtonItem
- title

# UINavigationController

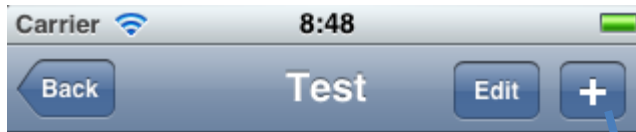


# UINavigationController



# UINavigationController

---



current

UINavigationControllerItem

- UINavigationControllerItem
  - backButtonBarItem
  - leftBarItemBarItem
  - titleView
  - title
  - rightBarItemBarItem
- 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 Forecast в виде таблицы Time'ов
  1. Создать ForecastViewController : UITableViewController
  2. На каждую ячейку UITableViewCell один Time (UITableViewCellStyleSubtitle)
  3. При нажатии на конкретную UITableViewCell сделать переход на уже сделанный нами в предыдущем уроке TimeViewController
  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