
Xcode

Introduction

Xcode
Objective-C
Swift
Interface Builder
UI Objects
MVC
Outlets and Actions
Xcode Example – outlets and actions



Xcode

- Managing all code

- Running App in the simulator

Objective-C / Swift

- Declaring and implementing objects

- Sending messages between objects

Interface Builder

- Connecting objects to send messages to each other

- Setting up the properties of objects



Xcode

an integrated development environment (IDE) containing a suite of software development tools developed by Apple for developing software for OS X and iOS.

- **Installing Xcode and the iOS SDK**
- **Becoming a registered Apple Developer**
- **Joining the iOS Developer Program**
- **Creating a simple iOS app**

Objective-C was created by Brad Cox and Tom Love in 1983

It was his attempt to add object-oriented programming concepts to the C programming language

Steve Jobs licensed objective-C (then OpenStep) for use in NeXT computers

Most of Apple's present-day Cocoa API is based on OpenStep interface objects, and is the most significant Objective-C environment being used for active development

In 1997, Apple purchased NeXT and transformed NeXTSTEP into MacOS X which was first released in the summer of 2000

Objective-C has been one of the primary ways to develop applications for MacOS for the past 11 years

In 2008, it became the primary way to develop applications for iOS targeting (currently) the iPhone and the iPad and (soon, I'm guessing) the Apple TV



Objective-C makes a small set of extensions to C which turn it into an object-oriented language

Strict superset of C

The Foundation framework contains classes for basic concepts such as strings, arrays and other data structures and provides classes to interact with the underlying operating system

The AppKit contains classes for developing applications and for creating windows, buttons and other widgets



Data types

Variables

Operators

Functions



**Together, Foundation and AppKit are called Cocoa
On iOS, AppKit is replaced by UIKit**

Foundation and UIKit are called Cocoa Touch



Since Objective-C is “C plus objects” any skills you have in the C language directly apply statements, data types, structs, functions, etc.

What the OO additions do, is reduce your need on structs, malloc, dealloc and the like and enable all of the object-oriented concepts we’ve been discussing Objective-C and C code otherwise freely intermix



Introduced in 2014

Swift is a multi-paradigm, compiled programming language created by Apple for iOS and OS X development.

Swift is designed to work with Apple's frameworks and the large body of existing Objective-C code written for Apple products.

Swift contains a lot of modern features that will make it more versatile, more concise, and nicer to read and write



Most of Apple's Apps today are written in Swift

Swift expands on the best of C and Objective-C, without the constraints of C compatibility. ... Swift has a clean slate that is supported by the developed and much adored Cocoa and Cocoa Touch frameworks.

For understanding the basics of programming, Swift is a much easier language to work with. Objective-C is over 30 years old and was written at a time when different considerations had to be made regarding how we interacted with computers. Grasp of both Objective-C and Swift is important

The basics of programming are easier to understand with Swift than with Objective-C. Key word here is basics.

The environment for learning Swift is far more engaging and provides immediate feedback, making concepts easier to learn.

Easier to read and write than Objective-C

[Xcode Playgrounds](#)
[Swift Playgrounds](#)

Objective-C: it's been the de facto language for iOS development for around 10 years.

Third party libraries and frameworks are primarily in Objective-C

The overwhelming majority of learning resources – books, blog posts, conference videos and so on, are all in Objective-C

iOS app development experts have clearly expressed that the significant launch of swift doesn't suggest the demise of Objective-C



With Swift, you're developing apps using the same framework and APIs as you would with Obj-C

Swift:

```
let view = UIView(frame: CGRectMake(0, 0, 200, 200))
view.backgroundColor = UIColor.blueColor()
```

Objective-C:

```
UIView *view = [[UIView alloc] initWithFrame:CGRectMake(0, 0, 200, 200)];
view.backgroundColor = [UIColor blueColor];
```


Interface Builder is a software development application for Apple's Mac OS X operating system.

It is part of Xcode (formerly Project Builder)

Allows developers to create interfaces for applications using a graphical user interface without writing any code.

Simply drag and drop windows, buttons, text fields, and other objects onto the design canvas to create a functioning Mac, iPhone, or iPad user interface.

Apple's XCode is used to develop in Objective-C

Behind the scenes, XCode makes use of either gcc or Apple's own LLVM to compile Objective-C programs

Xcode 4 integrated functionality that previously existed in a separate application, known as Interface Builder



XCode is available on the Mac App Store It is free for users of OS



Year

Developed by

1960

ALGOL

International Group

1967

BCPL

Martin Richard

1970

B

Ken Thompson

1973

Traditional C

Dennis Ritchie

1989

ANSI C

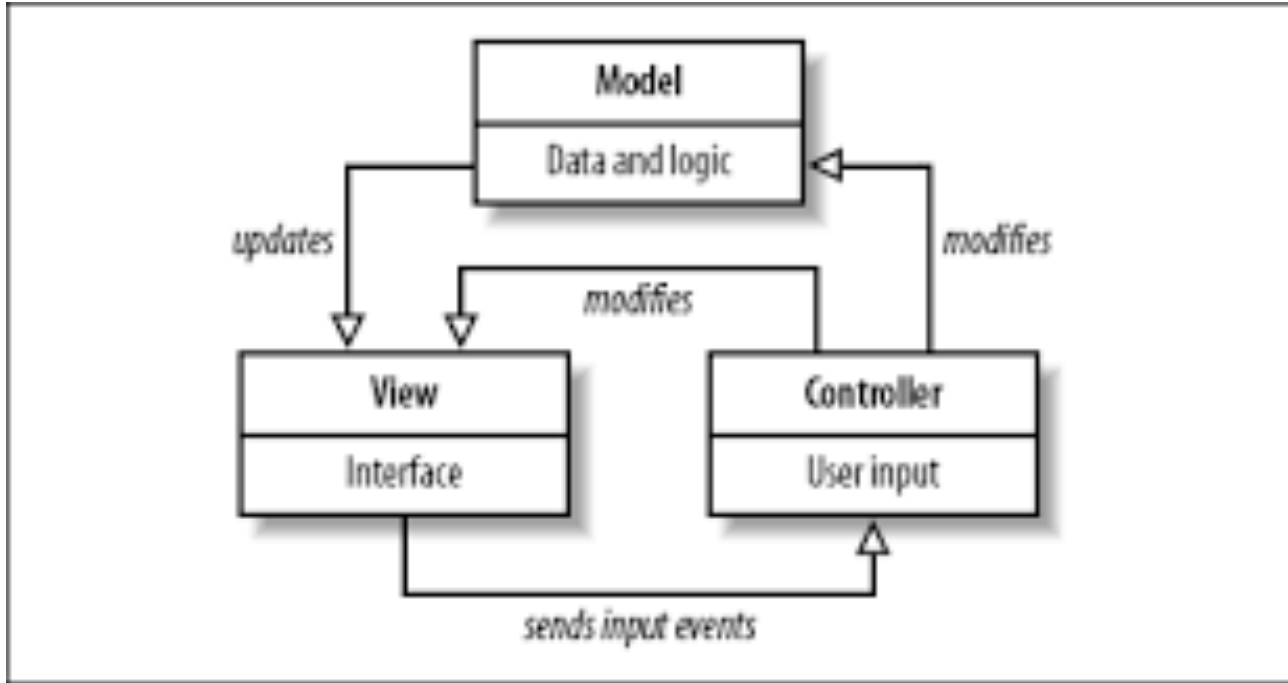
ANSI committee

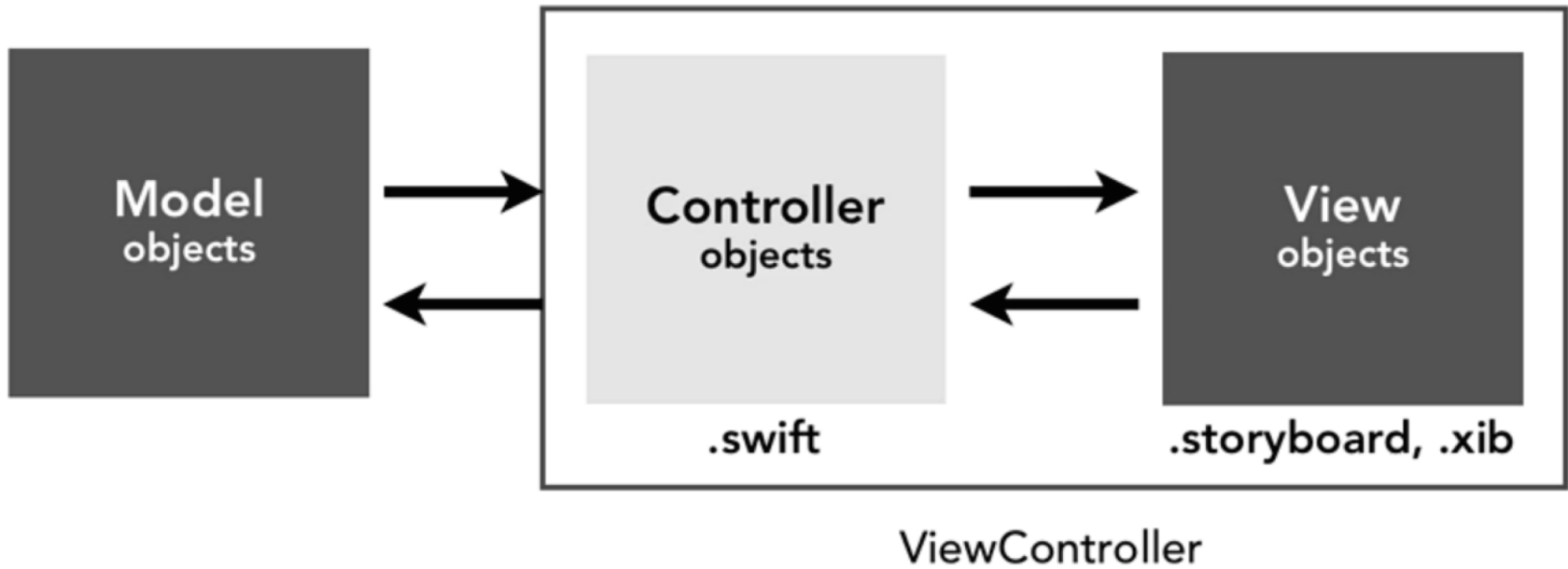
1990

ANSI/ISO C

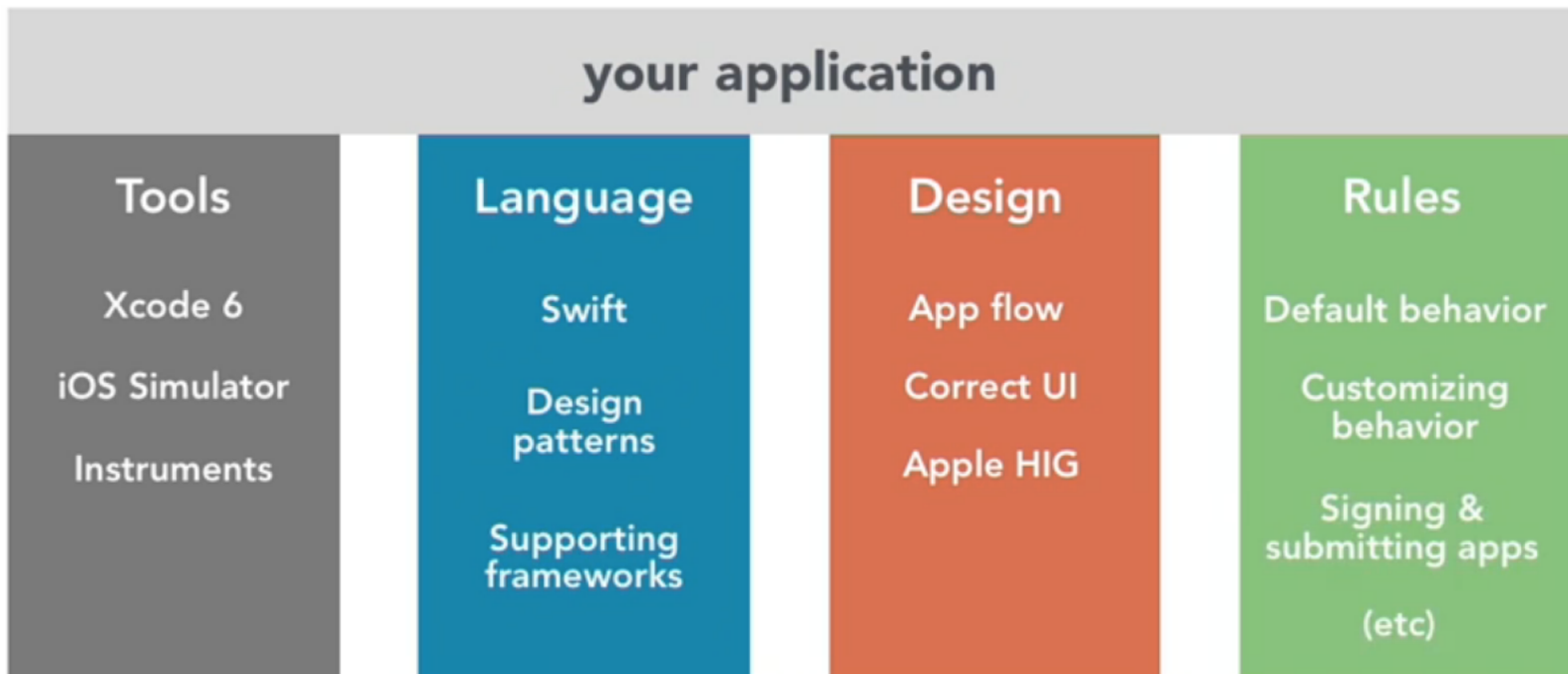
ISO committee

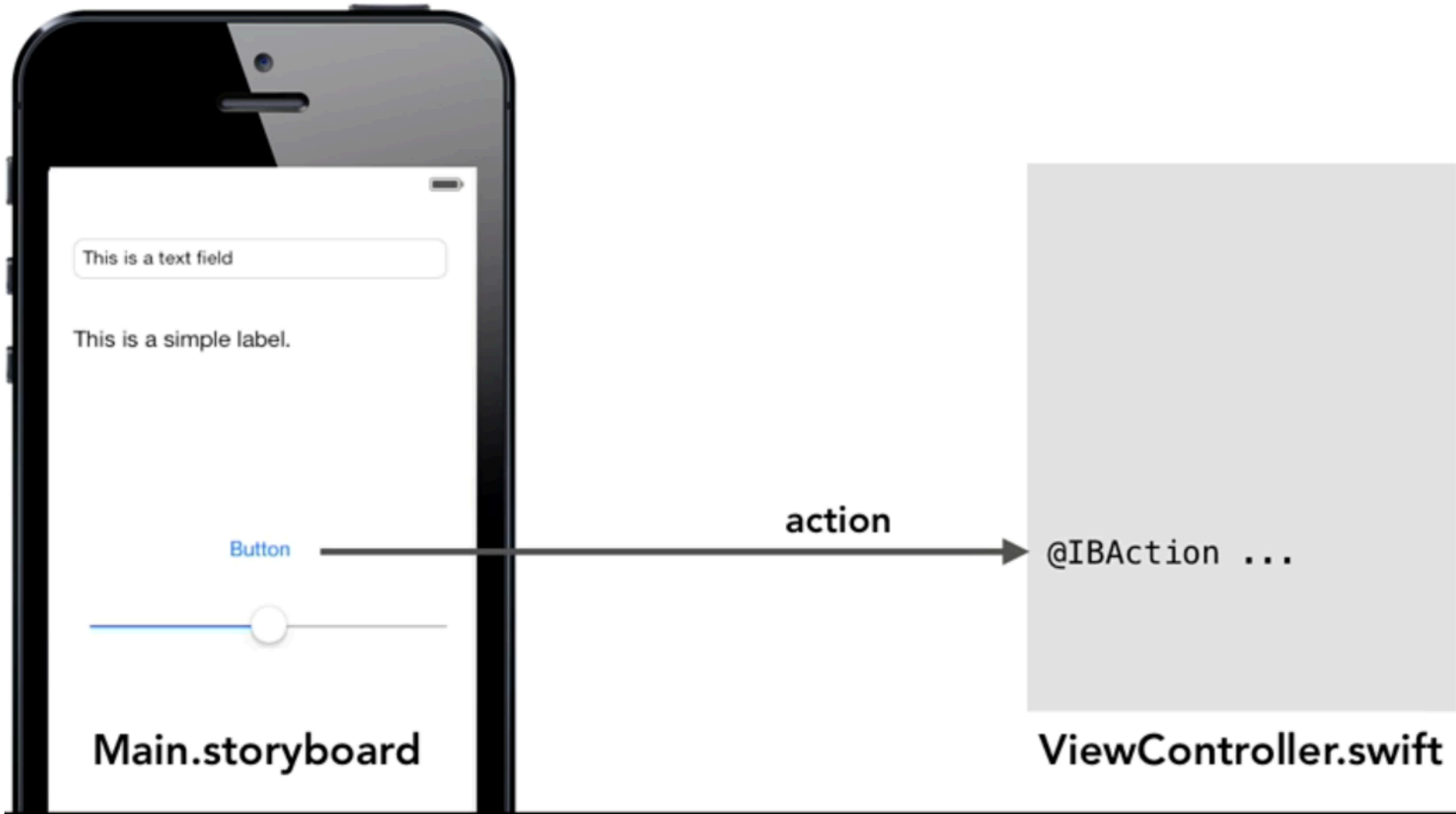


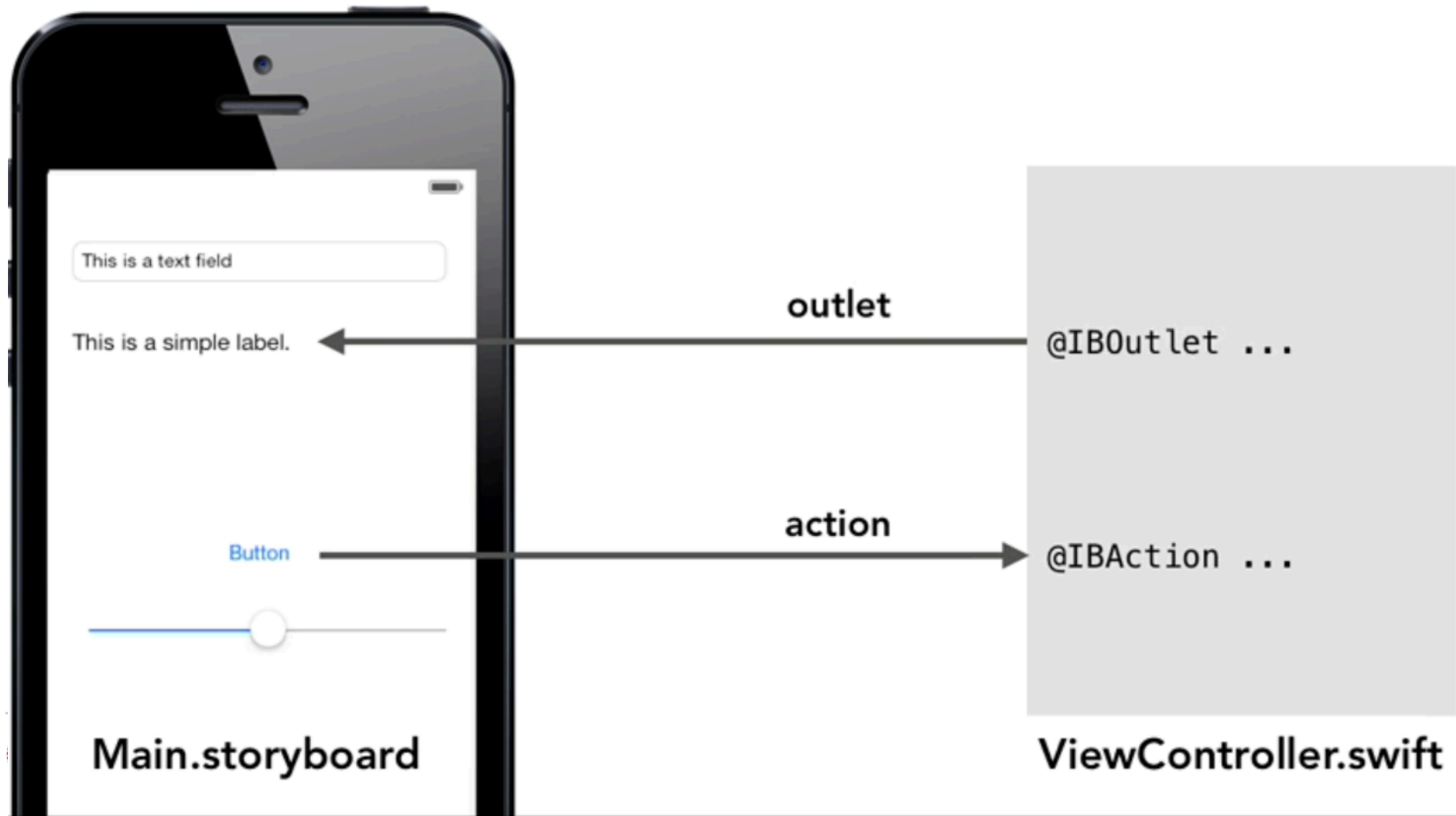


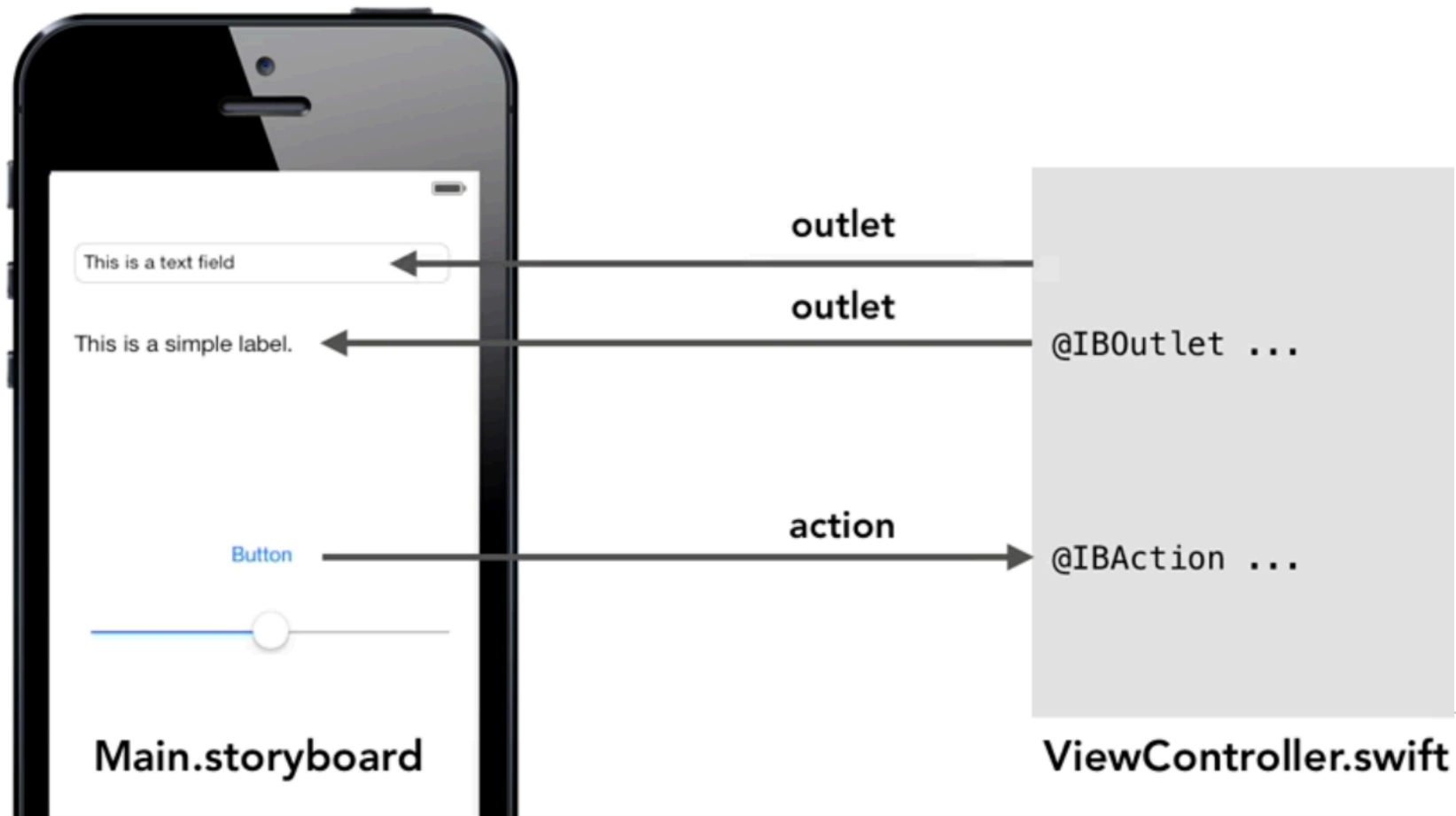


Four pillars of iOS development

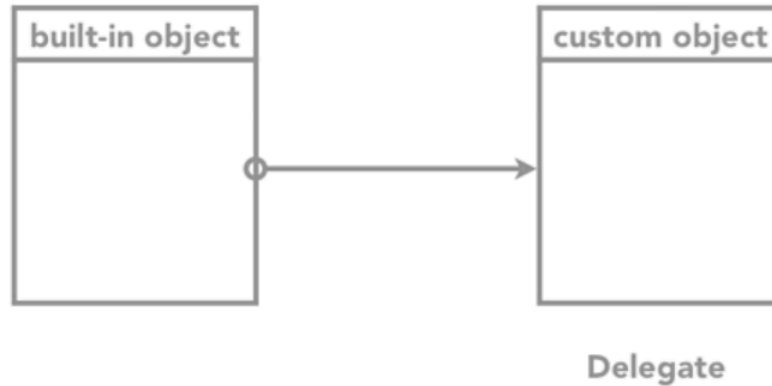








How one **object** can hand off responsibility to another **object**

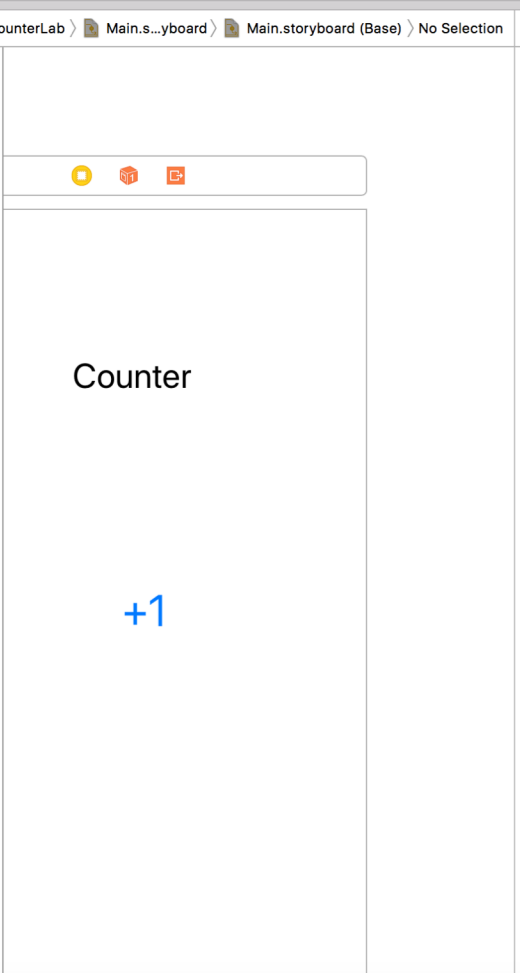


counterLab

- counterLab
 - AppDelegate.swift
 - ViewController.swift M
 - Main.storyboard M
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
- counterLabTests
 - counterLabTests.swift
 - Info.plist
- counterLabUITests
 - counterLabUITests.swift
 - Info.plist
- Products

View Controller Scene

- View Controller
 - View
 - Safe Area
 - Output Label
 - +1
 - First Responder
 - Exit
 - Storyboard Entry Poi...



```

1 //
2 // ViewController.swift
3 // counterLab
4 //
5 // Created by daria tsoupikova
6 // Copyright © 2018 daria
7 // Copyright © 2018 daria
8 // Copyright © 2018 daria
9 // Copyright © 2018 daria
10 // Copyright © 2018 daria
11 // Copyright © 2018 daria
12 // Copyright © 2018 daria
13 // Copyright © 2018 daria
14 // Copyright © 2018 daria
15 // Copyright © 2018 daria
16 // Copyright © 2018 daria
17 // Copyright © 2018 daria
18 // Copyright © 2018 daria
19 // Copyright © 2018 daria
20 // Copyright © 2018 daria
21 // Copyright © 2018 daria
22 // Copyright © 2018 daria
23 // Copyright © 2018 daria
24 // Copyright © 2018 daria
25 // Copyright © 2018 daria
26 // Copyright © 2018 daria
27 // Copyright © 2018 daria
28 // Copyright © 2018 daria
29 // Copyright © 2018 daria

```

Identity and Type

Name: Main.storyboard
 Type: Default - Interface Builder...
 Location: Relative to Group
 Base.lproj/Main.storyboard
 Full Path: /Users/dariatsoupikova/Desktop/MAD/2019_420/week11/lab/counterLab/Base.lproj/Main.storyboard

On Demand Resource Tags

Tags

Interface Builder Document

Opens in: Latest Xcode (9.0)
 Builds for: Deployment Target (11.2)
 Use Auto Layout
 Use Trait Variations
 Use Safe Area Layout Guides
 Use as Launch Screen

Global Tint: Default

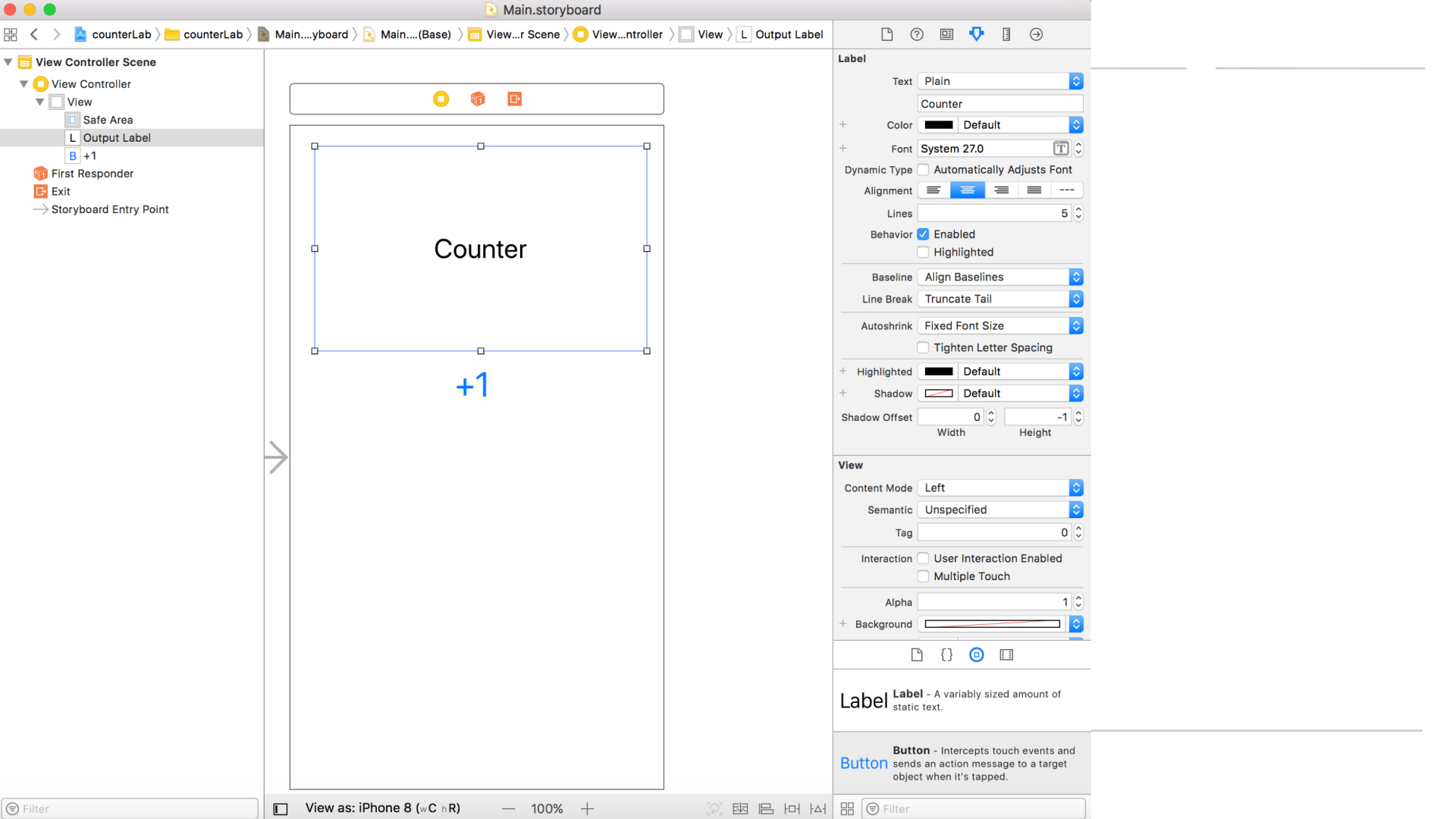
Localization

Base
 English
 Localizable Strings

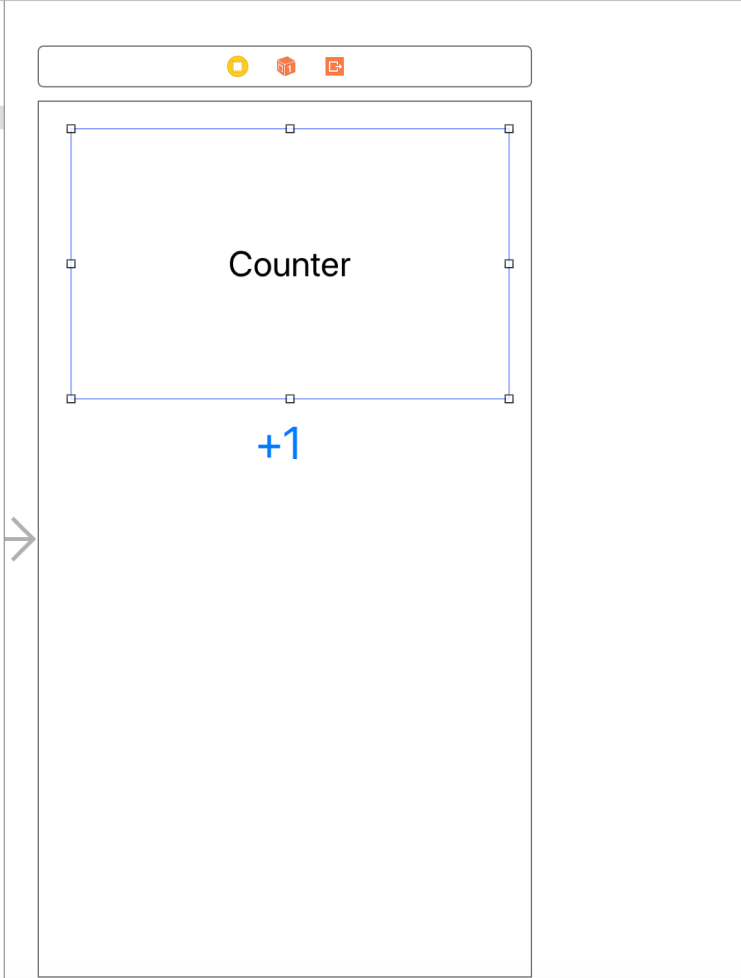
Cocoa Touch Class - A Cocoa Touch class

UI Test Case Class - A class implementing a unit test

Unit Test Case Class - A class implementing a unit test



- View Controller Scene
 - View Controller
 - View
 - Safe Area
 - Output Label
 - +1
 - First Responder
 - Exit
 - Storyboard Entry Point



Label

Text Plain
Counter

Color [Black] Default

Font System 27.0

Dynamic Type Automatically Adjusts Font

Alignment [Left]

Lines 5

Behavior Enabled
 Highlighted

Baseline Align Baselines

Line Break Truncate Tail

Autoshrink Fixed Font Size
 Tighten Letter Spacing

Highlighted [Black] Default

Shadow [Default] Default

Shadow Offset Width 0 Height -1

View

Content Mode Left

Semantic Unspecified

Tag 0

Interaction User Interaction Enabled
 Multiple Touch

Alpha 1

Background [Color]

Label Label - A variably sized amount of static text.

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

```
1 //
2 // ViewController.swift
3 // counterLab
4 //
5 // Created by daria tsoupikova on 11/11/18.
6 // Copyright © 2018 daria tsoupikova. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var outputLabel: UILabel!
14     var currentcount = 0
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18         // Do any additional setup after loading the view, typically from a nib.
19     }
20
21     override func didReceiveMemoryWarning() {
22         super.didReceiveMemoryWarning()
23         // Dispose of any resources that can be recreated.
24     }
25
26
27     @IBAction func addOneNumber(_ sender: UIButton) {
28         currentcount = currentcount + 1
29         outputLabel.text = "The button has been clicked \(currentcount) number of times"
30         outputLabel.textColor = UIColor.brown
31
32     }
33 }
34 }
35
```