

Getting Started

448460-1
Fall 2011
09/08/2011
Kyoung Shin Park
Multimedia Engineering
Dankook University

Overview

- Getting Started
- iOS Overview
- OOP

2

Why are we here?

- To build iPhone/iPad applications using Cocoa Touch



Requirements

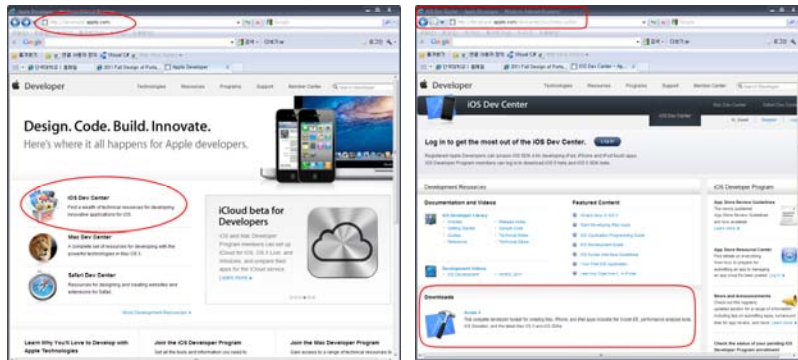


- Hardware & Software
 - Intel-based Macintosh, running Mac OS X 10.6 Snow Leopard
 - Hardware required for final project (iPad) and iPad loaners available
 - iPhone SDK iOS4.3 & Xcode 4.0.2
- Apple Developer Account
 - <http://developer.apple.com/programs/start/register/create.php>
 - Sign up for Apple's iOS Developer Program
 - You must have an Intel-based Mac running on Mac OS X Snow Leopard or later to develop Mac OS X and iOS apps for the App Store.

4

iPhone SDK

- Go to iOS Dev Center
 - <http://developer.apple.com/devcenter/ios/index.action>
- Download Xcode4.0.2
 - Included iOS SDK : iOS 4.3
 - Included Max SDK : Mac OS X 10.6



iPhone SDK

- Uninstall a previous version of Xcode using this command:
 - `sudo /Developer/Library/uninstall-devtools --mode=all`
- Install Xcode 4.0.2
 - Double-clicking on "xcode_4.0.2_and_ios_sdk_4.3.dmg"
 - Double-clicking on "Xcode and iOS SDK.mpkg" (Required Mac OS X 10.6.6)
- Launch Xcode 4.0.2
 - Open "Macintosh HD" (finder) and then go to >Developer>Applications>Xcode



iOS Overview

Mac OS X



iPhone OS



Core OS

OS X Kernel	Power Mgmt
Mach 3.0	Keychain
BSD	Certificates
Sockets	File System
Security	Bonjour

iPhone OS



Core Services

Collections	Core Location
Address Book	Net Services
Networking	Threading
File Access	Preferences
SQLite	URL utilities

iPhone OS



Media

Core Audio	JPG, PNG, TIFF
OpenAL	PDF
Audio Mixing	Quartz (2D)
Audio Recording	Core Animation
Video Playback	OpenGL ES

iPhone OS



Cocoa Touch

Multi-Touch Events	Alerts
Multi-Touch Controls	Web View
Accelerometer	People Picker
View Hierarchy	Image Picker
Localization	Camera

Platform Components

Tools



Xcode



Interface Builder

Frameworks



Foundation



UIKit

Language
(& Runtime)

```
[textView setValue:@"Hello"];
```

Objective-C

Cocoa Touch Architecture

Cocoa Touch

UIKit

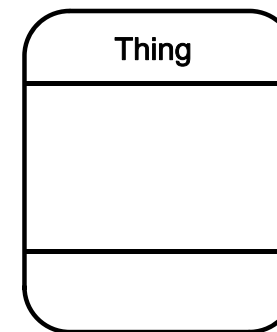
User interface elements
Application runtime
Event handling
Hardware APIs

Foundation

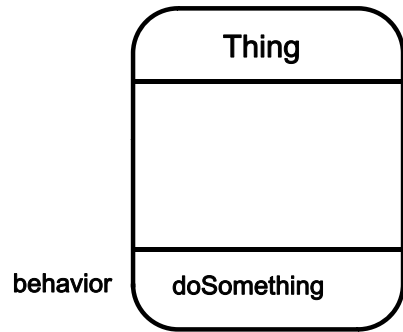
Utility classes
Collection classes
Object wrappers for system services
Subset of Foundation in Cocoa

Object-Oriented Programming

Object

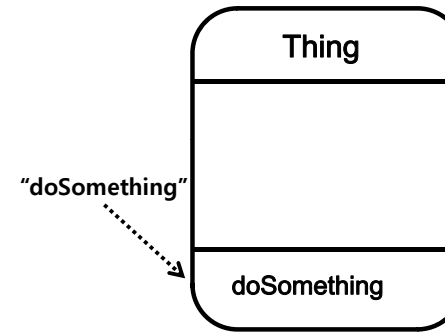


Behavior



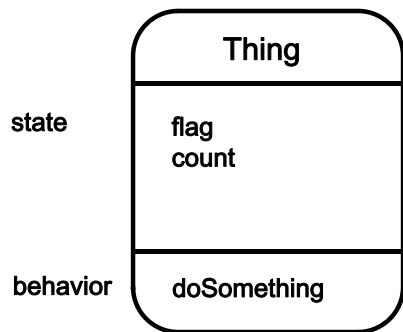
17

Message



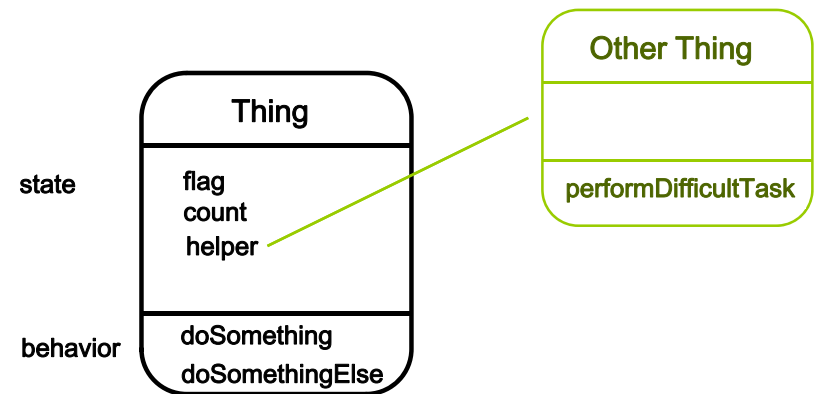
18

State



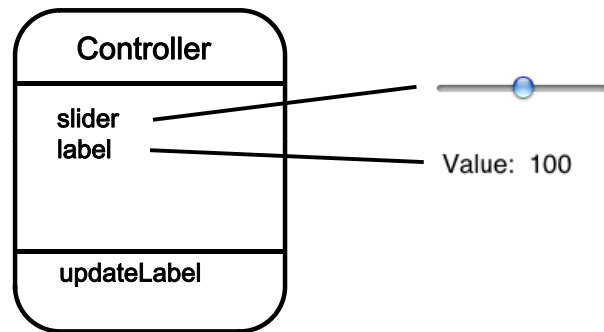
19

Other Objects As State



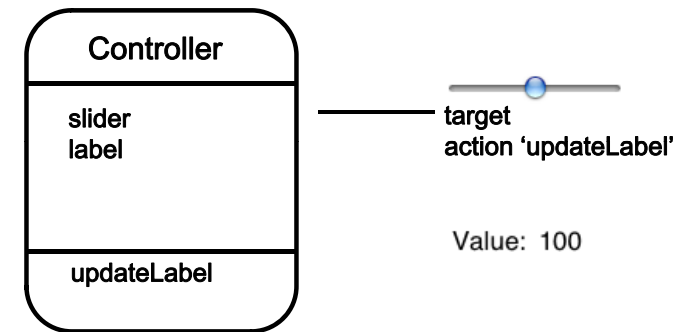
20

Outlets



21

Target/Action



22

Demo



23

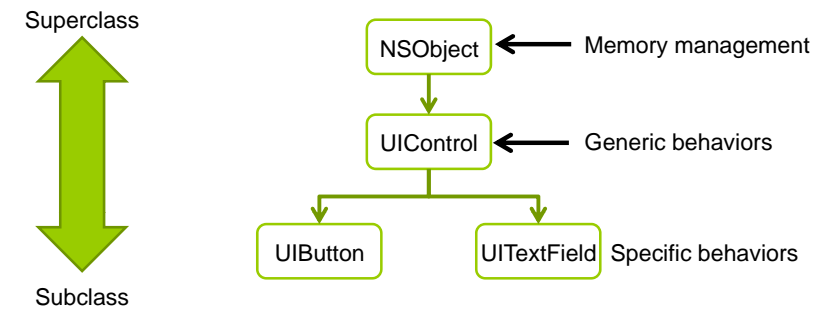
Object-Oriented Programming

- Class
 - Defines the grouping of data and code, the "type" of an object
- Instance
 - A specific allocation of a class
- Message
 - Sent to objects to make them act
- Method
 - A "function" that an object knows how to perform
- Instance variable
 - A specific piece of data belonging to an object

Object-Oriented Programming

- Encapsulation
 - Keep implementation private and separate from interface
- Polymorphism
 - Different objects, same interface
- Inheritance
 - Hierarchical organization, share code, customize or extend behaviors

Inheritance



- Hierarchical relation between classes
- Subclass “inherit” behavior and data from superclass
- Subclasses can use, augment or replace superclass methods

References

- Lecture 1 Slide from iPhone Application Development (Winter 2010) @Stanford University