



# Programmazione iPhone

*Giovanni Perbellini, Damiano Federti*  
Università di Verona – EDALab S.r.l.

## Agenda

- Introduzione iPhone
  - Overview
  - SDK
  - Objective-C
- Utilizzo di Xcode
- Esempi:
  - HelloWorld
  - Slider
  - TwoButtonOneLabel
- Esercizi

## Introduzione a iPhone OS (I)

- Sviluppato da Apple per dispositivi iPhone o iPod touch
- Si basa sul kernel di Mac OS X
  - Mack Kernel
- Xcode (iPhone SDK)
  - E' l'ambiente di sviluppo che contiene il codice e tutti gli strumenti necessari per programmare, eseguire test, emulare, compilare, etc.
- Linguaggio di programmazione: Objective-C

3

## Introduzione a iPhone OS (II)

- Include quattro layer di servizi
  - Core OS, Core Services
    - Interfaccia verso iPhone OS (C-based)
      - File system, Socket, Network, SQLite (Embedded DB), Security, Power Management, POSIX thread, etc.
  - Media
    - Audio, video, grafica 2D e 3D
    - Interfaccia C-based e Objective-C
  - Cocoa Touch
    - Foundation framework: support object-oriented per networks, file-management, etc.
    - UIKit framework: infrastruttura per la realizzazione dell'interfaccia grafica
      - Classi per windows, view, controls e i controllori per gestire questi oggetti
    - Altri framework per accedere a dispositivi hardware (e.g., accelerometer)

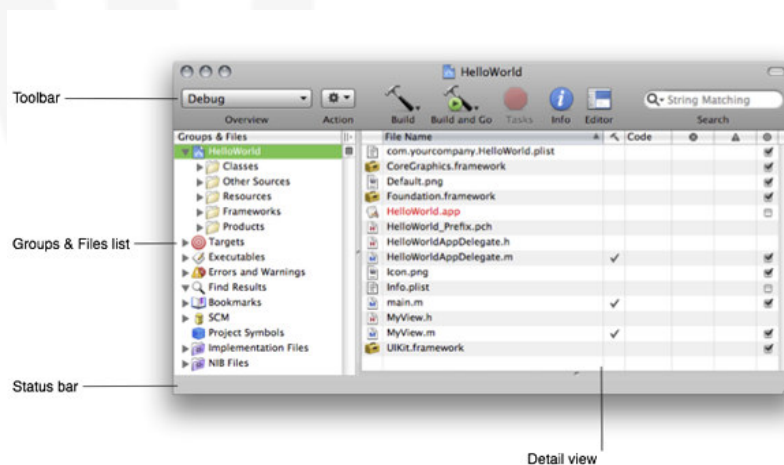
4

## SDK – Xcode

- E' l'ambiente di sviluppo per applicazioni che eseguono su iPhone OS
- Fornisce tutti i tool per creare il progetto iPhone, manipolare i file sorgenti, compilare l'applicazione, eseguire e debuggare il codice sul simulatore o sul dispositivo.

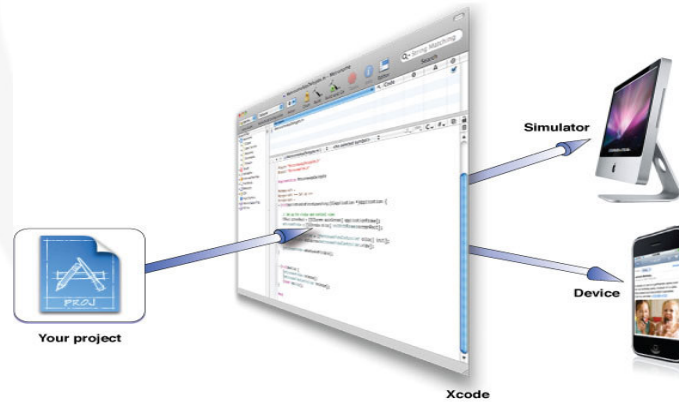
5

## Xcode – Project window



6

## Xcode – Running project



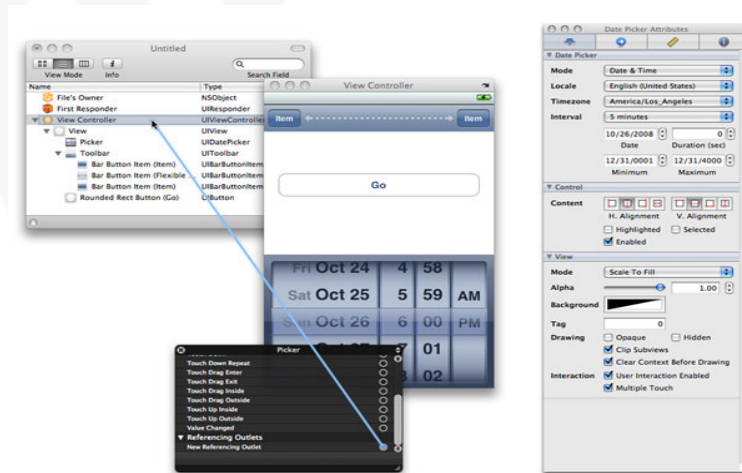
7

## Xcode – Interface Builder (I)

- L'ambiente per creare l'interfaccia grafica
- Drag and drop di widget grafici
- Switches, text fields, buttons
- Configurare gli attributi degli oggetti grafici
- Stabilire le relazioni tra gli oggetti e il codice
- Il formato del file ottenuto ha estensione .nib

8

## Xcode – Interface Builder (II)



9

## Objective C

- E' l'estensione ad oggetti del linguaggio C
- .h header file
  - Contiene classi, tipi, funzioni, etc.
- .m source file
  - Implementazione della classe
- .mm source file
  - Contiene anche codice C++

10

## Objective C - Classe

```
#import
@interface Photo : NSObject {
    int count;
    id name;
    NSString* name;
}
- (id)initWithString: (NSString *) aName
+ (MyClass *)createMyClassWithString: (NSString *) aName
@end
```

Garantisce che uno stesso file non venga importato più volte

Indica che si sta dichiarando una classe

Oggetto di qualsiasi tipo (è un puntatore)

Tipo di metodo: *instance* oppure *class method*

Nome metodo

Parametro di output

Parametro di input

11

## Objective C – Implementazione (I)

- Esempio:

```
#import
@interface Photo : NSObject {
    NSString* caption;
    NSString* photographer;
}
- (NSString*) caption;
- (NSString*) photographer;
- (void) setCaption: (NSString*)input;
- (void) setPhotographer: (NSString*)input;
@end
```

12

## Objective C – Implementazione (II)

```

#import "Photo.h"
@implementation Photo

- (NSString*) caption {
    return caption;
}

- (NSString*) photographer {
    return photographer;
}

- (void) setCaption: (NSString*)input {
    [caption autorelease];
    caption = [input retain];
}

- (void) setPhotographer: (NSString*)input {
    [photographer autorelease];
    photographer = [input retain];
}

```

← Importo la dichiarazione  
 ← Indica che si sta implementando una classe  
 ← autorelease: rilascio automatico della memoria al termine della funzione.  
 ← release: rilascio immediato della memoria.  
 ← retain: copia di tipo shallow; duplica semplicemente il valore del puntatore (condivisione dei dati).  
 ← copy: copia di tipo deep; duplica i dati dell'oggetto originale e li assegna al nuovo oggetto.

13

## Objective C - Metodi

- Invocare un metodo di un oggetto
  - Instance method
    - [object method];
    - [object methodWithInput:input];
    - Output = [object methodWithOutput];
    - Output = [object methodWithInputAndOutput:input];
  - Class method
    - Id myObject = [NSString string];
    - NSString\* myString = [NSString string];
  - Invocazione di metodi con più parametri (e assegnazione di nomi ai metodi)
    - - (int) colorWithRed:(int) r green:(int) g blue:(int) b
    - [myColor colorWithRed:5 green:2 blue:6];

14

## Objective C – Creare oggetti (I)

```
NSString* myString = [[NSString alloc] init];
```

Allocazione (`alloc` è un class method)

Costruttore di default: ogni Oggetto ha un costruttore di default definito implicitamente (`init` è un instance method).

Override del metodo `init`:

```
-(id) init {
    if ( self = [super init] ) {
        variabile = 1;
    }
    return self;
}
```

Inizializza la superclasse e verifica che sia Avvenuta con successo

15

## Objective C – Creare oggetti (II)

C++
Foo* foo = new Foo;
Objective C
Foo* foo1 = [Foo alloc]; [foo1 init]; //bad : the returned value should be used
Foo* foo3 = [[Foo alloc] init]; //ok, usual way to do

16



## Objective C – Creare oggetti (III)

C++
<pre>class Point2D { public:     Point2D(int x, int y); private:     int x;     int y; }; Point2D::Point2D(int anX, int anY) {x = anX; y = anY;} ... Point2D p1(3,4); Point2D* p2 = new Point2D(5, 6);</pre>
Objective C
<pre>@interface Point2D : NSObject { int x; int y; } //Note : id is somewhat similar to void* in Objective-C.(id) is the general type for an object -(id) initWithX:(int)anX andY:(int)anY; @end @implementation Point2D -(id) initWithX:(int)anX andY:(int)anY { //an initializer of the superclass must be called if (!(self = [super init])) return nil; self-&gt;x = anX; self-&gt;y = anY; return self; //and return the object itself } @end ... Point2D* p1 = [[Point2D alloc] initWithX:3 andY:4];</pre>

17

## Objective C - Property

- La direttiva property permette di generare automaticamente i metodi per l'accesso alle variabili (metodi getter e setter)

```
#import                                     #import "Photo.h"
@interface Photo : NSObject {              @implementation Photo
    NSString* caption;
    NSString* photographer;
}
@property (retain) NSString* caption;
@property (retain) NSString* photographer;
@end
                                             @synthesize caption;
                                             @synthesize photographer;

- (void) dealloc {
    [caption release];
    [photographer release];
    [super dealloc];
}
@end
```

Il retain racchiuso nelle parentesi invece indica che i setter devono restituire il valore in input.

18

## Esempio Helloworld (I)

- Xcode
  - Application → View-based application → Helloworld
    - Main.m
    - Delegate – Gestione eventi
    - UIcontroller – User Interface
- Interface Builder
  - HelloWorldViewController.xib

19

## Esempio Helloworld (II)

- Interface Builder -> Libray -> widget grafici
  - UILabel, UIButton, etc.

- Xcode: HelloWorldViewController.h

```
@interface HelloWorldViewController : UIViewController {
    IBOutlet UIButton *Btn;
    IBOutlet UILabel *Lbl;
}
```

```
- (IBAction)onClick:(id)sender;
```

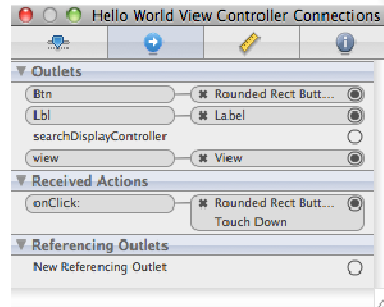
```
@property (nonatomic,retain) IBOutlet UIButton *Btn;
@property (nonatomic,retain) IBOutlet UILabel *Lbl;
```

```
@end
```

20

## Esempio Helloworld (III)

- Interface Builder: creazione link



- Xcode: HelloWorldViewController.m

```
- (IBAction)onClick:(id)sender{
    Lbl.text = @"Hello World !!!";
}
```

21

## Esempio Helloworld (IV)

- Build and Run



22

## Esempio Switch\_Slider (I)

- Xcode
  - New Project → View-based application → Switch\_Slider
    - main.m
    - AppDelegate
    - ViewController
- Interface Builder
  - Switch\_SliderViewController.xib

23

## Esempio Switch\_Slider (II)

- Interface Builder → Libray → widget grafici
  - UIImageView, UISwitch, UISlider, etc.
- Xcode: Switch\_SliderViewController.h

```

@interface Switch_SliderViewController : UIViewController {
    IBOutlet UISlider *Slider;
    IBOutlet UISwitch *Switch;
    IBOutlet UIImageView *ImageView;
    CGPoint p;
}
- (IBAction)enableMove:(id)sender;
- (IBAction)Move:(id)sender;
@property (nonatomic,retain) IBOutlet UISlider *Slider;
@property (nonatomic,retain) IBOutlet UISwitch *Switch;
@property (nonatomic,retain) IBOutlet UIImageView *ImageView;
@end

```

24

## Esempio Switch\_Slider (III)

- Xcode: Switch\_SliderViewController.m

```
- (IBAction)enableMove:(id)sender {
    if (Switch.on == TRUE)
        [Slider setEnabled:TRUE];
    else
        [Slider setEnabled:FALSE];
}

- (IBAction)Move:(id)sender {
    p.y = 150;
    p.x = 150 + Slider.value;

    [ImageView setCenter:p];
}
```

25

## Esempio Switch\_Slider (IV)

- Build and Run



26

## Esempio TwoButtonOneLabel (I)

- Xcode
  - Application → View-based application → TwoButtonOneLabel
    - Main.m
    - Delegate – Gestione eventi
    - UIcontroller – User Interface

27

## Esempio TwoButtonOneLabel (II)

- TwoButtonOneLabelViewController.h

```

@interface TwoButtonOneLabelViewController : UIViewController {
    IBOutlet UILabel *statusText;
}
-(IBAction)buttonPressed:(id)sender;

@property (nonatomic,retain) IBOutlet UILabel *statusText;

@end
  
```

28

## Esempio TwoButtonOneLabel (III)

- TwoButtonOneLabelViewController.m

```

-(IBAction)buttonPressed:(id)sender{
    NSString *title = [sender titleForState:UIControlStateNormal];
    NSString * newText;
    newText = [[NSString alloc] initWithFormat:@"Button select: %@", title];
    statusText.text = newText;
    [newText release];
}

```

- Rilascio automatico della memoria
  - NSString\* str1 = [NSString string];
- Rilascio manuale della memoria
  - NSString\* str2 = [[NSString alloc] init];
  - [str2 release];

29

## Esempio TwoButtonOneLabel (IV)

- TwoButtonOneLabelViewCo  
ntroller.xib

- Aggiunta dei componenti grafici utilizzando InterfaceBuilder
- Aggiunta Label
- Aggiunta PulsanteOne
- Aggiunta PulsanteTwo



30

## Esempio TwoButtonOneLabel (V)

- Mettere in relazione statusText con la Label
  - File's Owner -> Label
- Mettere in relazione gli oggetti grafici Button (PulsanteOne e PulsanteTwo) con i metodi dichiarati precedentemente
  - Selezionare proprietà del Button
  - Selezionare Touch Up Inside
  - File's Owner -> buttonPressed

31

## Esercizi

- Esercizi:
  - Inserire RESET
  - Inserire UITextField da cui prelevare la stringa da impostare nella label alla pressione del pulsante.

32