

iOS Developing

Κεφάλαια Αικατερίνη
Μουρίκης Ευάγγελος

<http://arch.icte.uowm.gr>

Επιβλέπων: Μηνάς Δασυγένης

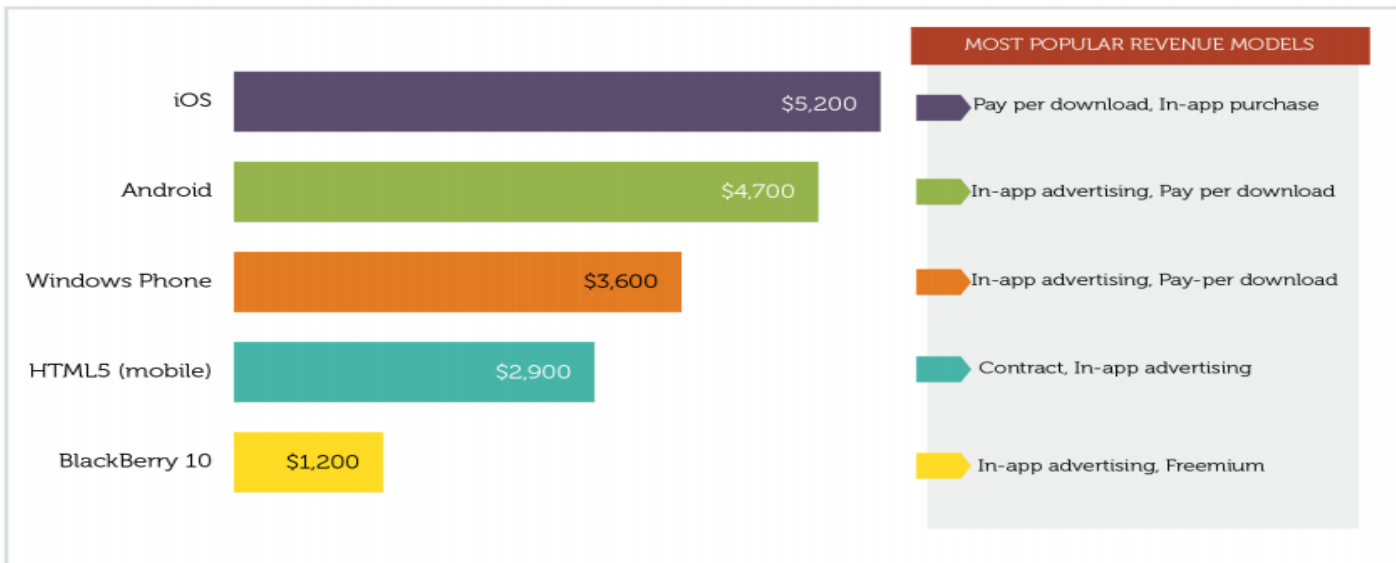
iOS Συσκευές



Λόγος επιλογής προγραμματισμού σε iOS

iOS leads the revenue chart

Average monthly developer revenue for each platform (n = 2,085)

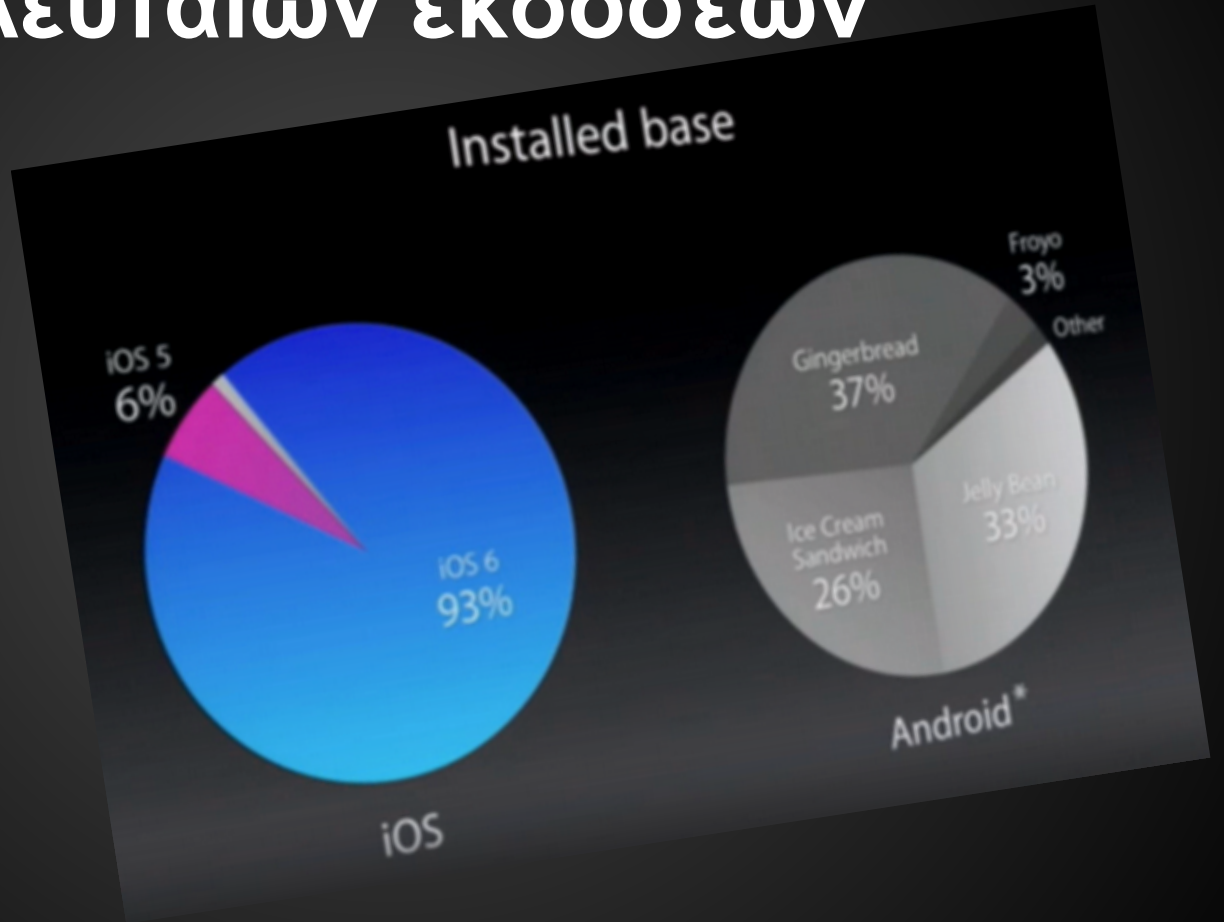


Source: Developer Economics Q3 2013 - State of the Developer Nation

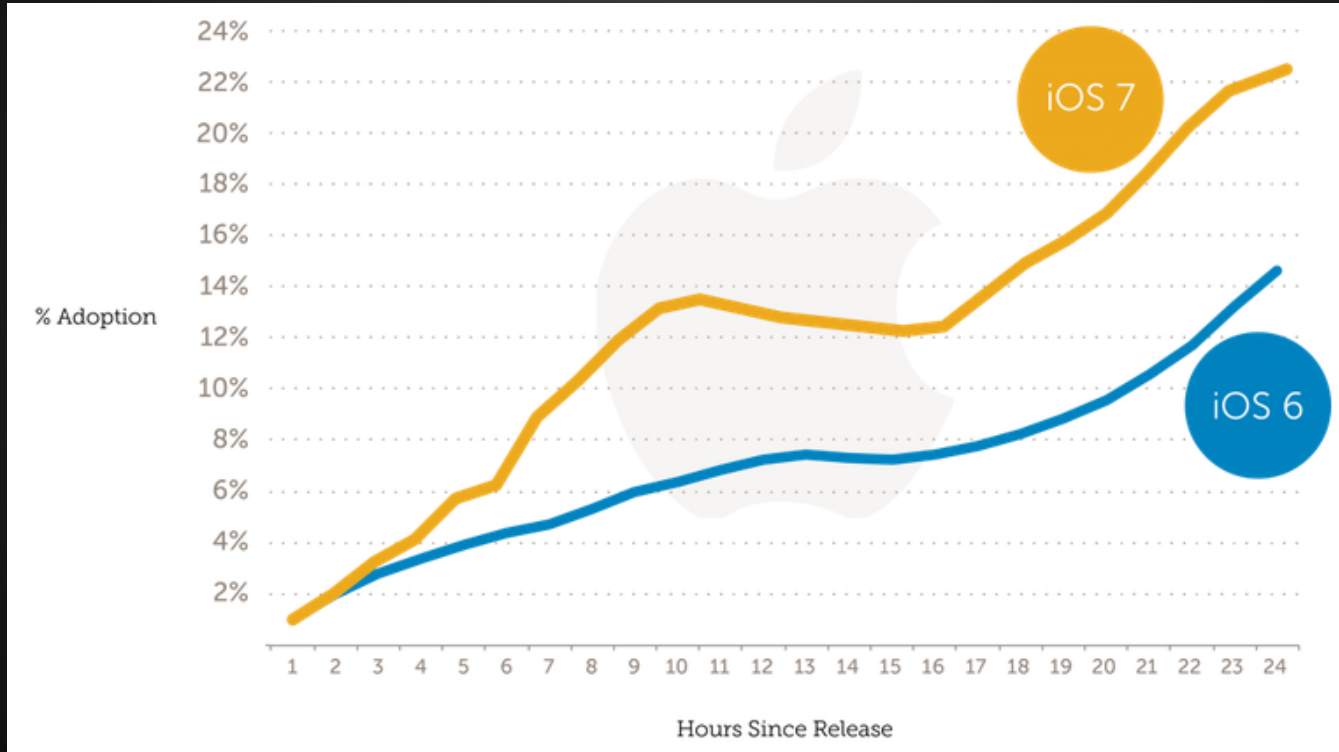
www.DeveloperEconomics.com/go

Licensed under Creative Commons Attribution 3.0 License

Δείγμα τελευταίων εκδόσεων



Υιοθέτηση των IOS 6 και IOS 7 (1/2)



Υιοθέτηση των IOS 6 και IOS 7 (2/2)

Κύριοι λόγοι

1. ασφάλεια
2. σύστημα αναβάθμισης

Τεχνολογίες σχετικά με το iOS 7

Hardware

Software

Πρότυπο

Αναπτυξιακό
Περιβάλλον



Cocoa



Cocoa Touch

Προαπαιτούμενα για προγραμματισμό σε iOS

- υπολογιστής με λειτουργικό OSX
- Xcode

iOS Apple Developer Program

Για να ανέβει η εφαρμογή στο App Store χρειάζεται εγγραφή με κόστος 99\$/χρόνο.

Για προγραμματισμό εφαρμογών δεν απαιτείται καθόλου κόστος παρά μόνο τα δύο προαπαιτούμενα που αναφέρθηκαν.

Εγγραφή για Apple Developer Program (1/5)

<https://developer.apple.com/programs/ios/>

iOS Developer Program

The fastest path from code to customer.

Enroll Now

\$99/year

Εγγραφή για Apple Developer Program (2/5)

Enrolling in Apple Developer Programs

Get everything you need to develop and distribute apps for iOS and OS X.

It's easy to get started.

✓ Choose an enrollment type.

Individual: choose this option if you are an individual or sole proprietor/single person business.

Company/Organization: choose this option if you are a company, non-profit organization, joint venture, partnership, or government organization.

✓ Submit your information.

Provide basic personal information, including your legal name and address. If you're enrolling as a company/organization, we'll need a few more things, like your legal entity name and D-U-N-S® Number, as part of our verification process.

✓ Purchase and activate your program.

Once we verify your information, you can purchase your program on the Apple Online Store. After you have completed your purchase, we'll send you an email within 24 hours on how to activate your membership.



Continue

Εγγραφή για Apple Developer Program (3/5)

Sign in or create an Apple ID.

You can enroll in the iOS Developer Program or Mac Developer Program with the same Apple ID you use for other services like iCloud and the Apple Online Store. However, if you have an iTunes Connect account for distributing another media type (music, TV, movies, or books) or are enrolled in the iOS Developer Enterprise Program, you need to use a different Apple ID for your enrollment.

Existing Apple ID

Enroll in an Apple Developer Program with the same Apple ID you use for other services like iCloud and the Apple Online Store.

[Sign In](#)

or

New Apple ID

Create a new Apple ID if you have an existing iTunes Connect account, participate in the Volume Purchase Program, are enrolled in the iOS Developer Enterprise Program, or prefer to have an Apple ID dedicated to your business transactions.

[Create Apple ID](#)

Εγγραφή για Apple Developer Program (4/5)

Are you enrolling as an individual or organization?

Individual

Select this option if you are an individual or sole proprietor/single person company.



Seller Name

Your personal legal name will be listed as the seller of your apps on the App Store.

Example:

Seller: John Smith



Individual Development Only

You are the only one allowed access to program resources.



You will need:

- A valid credit card for purchase.
We may also require additional personal documentation to verify your identity.

Individual

Company/Organization

Select this option if you are a company, non-profit organization, joint venture, partnership, or government organization.



Seller Name

Your organization's legal entity name will be listed as the seller of your apps on the App Store.

Example:

Seller: ABC Company, Inc.



Development Team

You can add additional developers to your team who can access program resources. Companies who have hired a contractor to create apps for distribution on the App Store should enroll with their company name and add the contractors to their team.



You will need:

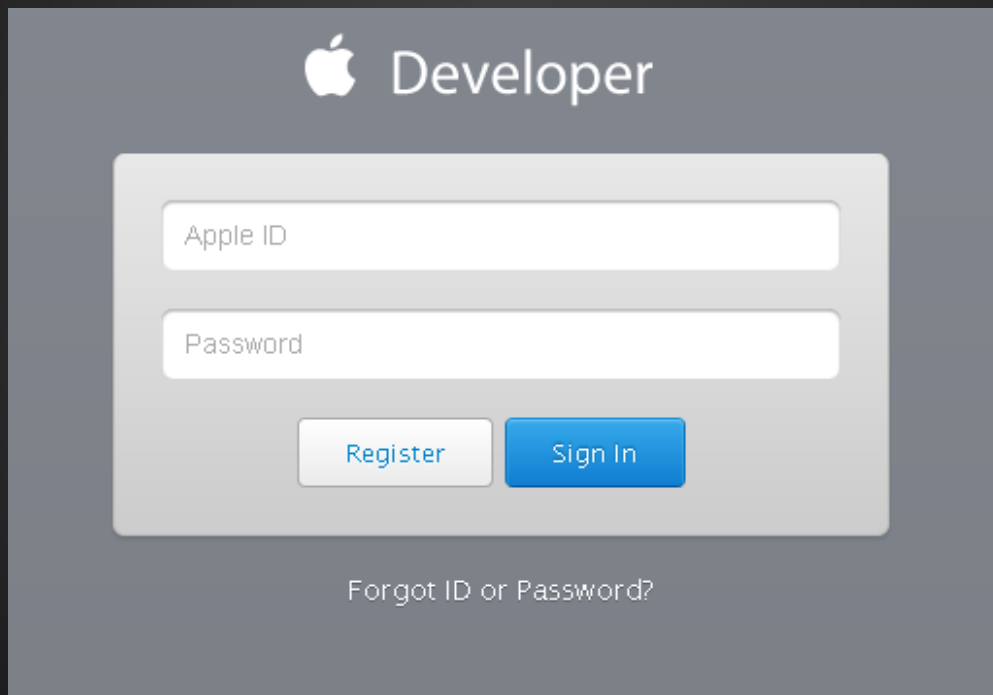
- The legal authority to bind your company/organization to Apple Developer Program legal agreements.
- An address for the company's principal place of business or corporate headquarters.
- A D-U-N-S® Number assigned to a legal entity.
D-U-N-S Numbers, available from D&B for free in most jurisdictions, are unique nine-digit numbers widely used as standard business identifiers. To learn more, read our FAQs. Before enrolling, check to see if D&B has assigned you a D-U-N-S Number. If not, please request one.

Note: We do not accept DBAs, Fictitious Business, Trade names, or branches at this time.

- A valid credit card for purchase.

Company

Εγγραφή για Apple Developer Program (5/5)



The image shows a screenshot of the Apple Developer Program login and registration interface. At the top, the Apple logo is followed by the word "Developer". Below this, there is a light gray rounded rectangle containing two input fields: "Apple ID" and "Password". Underneath the input fields are two buttons: "Register" (light gray) and "Sign In" (blue). At the bottom of the form, there is a link that says "Forgot ID or Password?".

Apple Developer

Apple ID

Password

Register Sign In

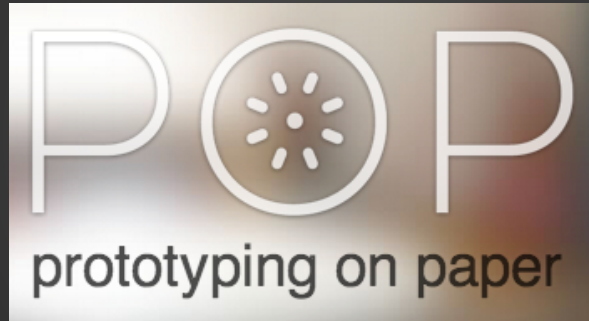
[Forgot ID or Password?](#)

Xcode 5.0



<https://itunes.apple.com/us/app/xcode/id497799835?mt=12>

Εφαρμογή Prototyping (1/2)



<https://itunes.apple.com/app/id555647796>

Εφαρμογή Prototyping (2/2)

1. Σχέδιο στο χαρτί

2. Φωτογραφία του σχεδίου

3. Προσομοίωση του σχεδίου

Εισαγωγή σε Xcode και Objective-C

Σχεδιαστικό πρότυπο που χρησιμοποιείται στο iOS

Στον iOS προγραμματισμό κυριαρχεί ένα είδος
σχεδιαστικού προτύπου, το MVC.

Ορισμός MVC

Σχεδιαστικό πρότυπο

Ορίζει το ρόλο που παίζουν τα αντικείμενα στην εφαρμογή, αλλά και τον τρόπο με τον οποίο επικοινωνούν.

Ακρωνύμιο MVC

Model

View

Controller

MVC - Μοντέλο

Αντικείμενα μοντέλου ενσωματώνουν συγκεκριμένα στοιχεία σε μια εφαρμογή και καθορίζουν τη λογική και τον υπολογισμό για την επεξεργασία των δεδομένων.

Μπορεί να είναι:

- χαρακτήρας σε παιχνίδι
- επαφή σε αντζέντα

MVC - Αντικείμενα μοντέλου

- Δεν έχουν άμεση σύνδεση με τα αντικείμενα οπτικοποίησης που παρουσιάζουν τα δεδομένα και επιτρέπουν στους χρήστες να τα επεξεργάζονται.
- Δεν ασχολούνται με θέματα διεπαφής χρήστη και παρουσίασης.

MVC - Επικοινωνία Μοντέλου(1/2)

View  Controller  Model

Οι ενέργειες των χρηστών στο view layer που δημιουργούν ή τροποποιούν δεδομένα, διαβιβάζονται μέσω ενός αντικειμένου ελέγχου και οδηγούν στη δημιουργία ή ενημέρωση ενός αντικειμένου μοντέλου.

MVC - Επικοινωνία Μοντέλου(2/2)

View ← Controller ← Model

Όταν ένα αντικείμενο μοντέλου αλλάζει, ειδοποιεί το αντικείμενο ελέγχου, το οποίο αναπροσαρμόζει τα κατάλληλα αντικείμενα οπτικοποίησης.

MVC - Αντικείμενα Οπτικοποίησης

Ένα αντικείμενο οπτικοποίησης

- είναι ένα αντικείμενο σε μια εφαρμογή που μπορούν να δουν οι χρήστες
- γνωρίζει πως να σχεδιαστεί και μπορεί να ανταποκριθεί στις ενέργειες των χρηστών

MVC - Σκοπός αντικειμένων οπτικοποίησης

Ο βασικός σκοπός των αντικειμένων οπτικοποίησης είναι να εμφανίζουν δεδομένα από τα αντικείμενα μοντέλου της εφαρμογής και να καθιστούν δυνατή την επεξεργασία των εν λόγω δεδομένων.

Η σύνδεση με τα αντικείμενα οπτικοποίησης δεν είναι άμεση.

MVC - Επικοινωνία Οπτικοποίησης (1/2)

View → Controller → Model

Τα αντικείμενα οπτικοποίησης πληροφορούνται σχετικά με τις αλλαγές στα δεδομένα μοντέλου μέσω των αντικειμένων ελέγχου της εφαρμογής και ενημερώνουν το χρήστη.

MVC - Επικοινωνία Οπτικοποίησης (2/2)

View ← Controller ← Model

Αντίστοιχα ενημερώνουν τα αντικείμενα μοντέλου μέσω των αντικειμένων ελέγχου για τις δικές τους αλλαγές.

MVC - Αντικείμενα ελεγκτή

Ένα αντικείμενο ελεγκτή

- λειτουργεί ως μεσάζων των αντικειμένων μοντέλου και των αντικειμένων οπτικοποίησης μιας εφαρμογής
- είναι ένας αγωγός μέσω του οποίου τα αντικείμενα οπτικοποίησης μαθαίνουν για τις αλλαγές στα αντικείμενα μοντέλου και το αντίστροφο

MVC - Χρήση αντικειμένων ελεγκτή

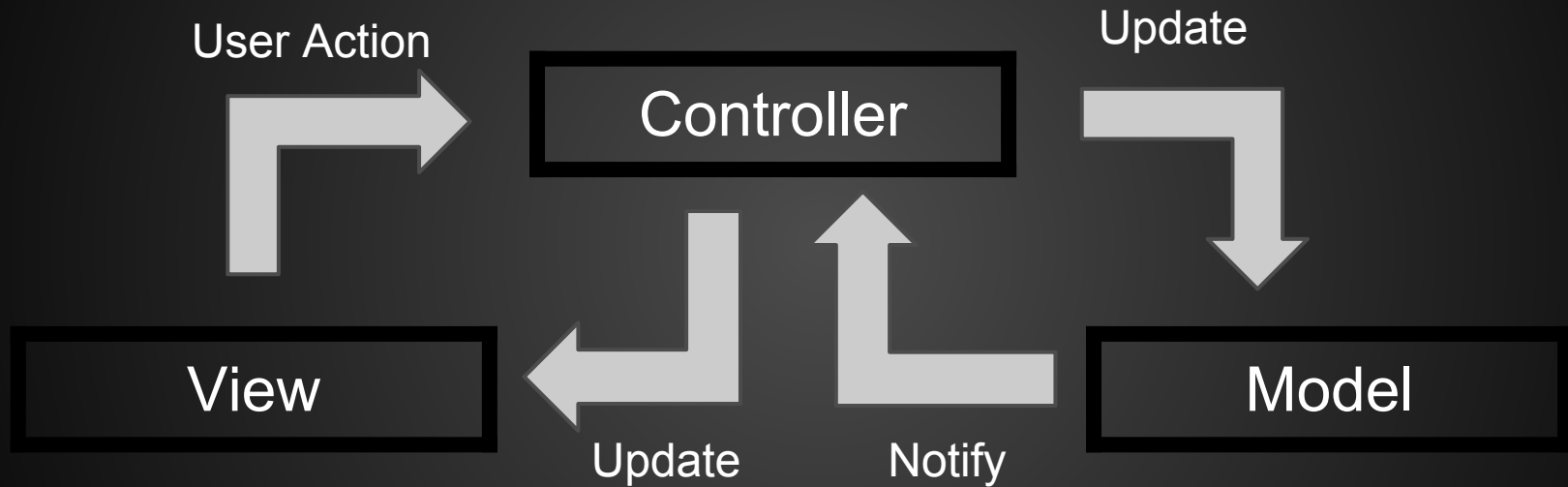
Μπορούν να

- πραγματοποιήσουν εγκαταστάσεις
- συντονίσουν εργασίες για την εφαρμογή
- διαχειριστούν τους κύκλους ζωής των άλλων αντικειμένων

MVC - Επικοινωνία ελεγκτή

- Ένα αντικείμενο ελεγκτή ερμηνεύει τις ενέργειες που πραγματοποιεί ο χρήστης στα αντικείμενα οπτικοποίησης και μεταφέρει τα νέα ή τροποποιημένα δεδομένα στο model layer.
- Όταν τα αντικείμενα μοντέλου αλλάζουν, ένα αντικείμενο ελέγχου μεταφέρει αυτά τα νέα δεδομένα μοντέλου στα αντικείμενα οπτικοποίησης ώστε να τα εμφανίσουν.

MVC - Ροές επικοινωνίας



Ορισμός Objective-C

- Αντικειμενοστραφής γλώσσα
- Υπερσύνολο της C με επιπλέον χαρακτηριστικά

Βασικά αρχεία προγραμματισμού σε Objective-C

.h

.m

Ρόλος των βασικών αρχείων προγραμματισμού σε Objective-C

.h



interface

.m



implementation

Σημασία βασικών αρχείων προγραμματισμού σε Objective-C

.h  header file

.m  implementation file

Περιεχόμενα του αρχείου .h

- δήλωση κλάσης
- ονομάζουμε τις κλάσεις και τις υπερκλάσεις
- καθορισμός πρωτοκόλλου
- καθορισμός μεταβλητών
- καθορισμός μεθόδων και δήλωση ιδιοτήτων
- εισαγωγή frameworks
- αρχή με @interface
- τέλος με @end

Παράδειγμα αρχείου .h

```
#import <kate.h>
```

```
@interface MyClass : SuperClass {  
    int InstanceVariable;  
}
```

```
+ (void)classMethod;
```

```
- (void)instanceMethod;
```

```
@end
```

Περιεχόμενα του αρχείου .m

- ορισμός κλάσεων
- ορισμός μεθόδων
- εισαγωγή header files
- αρχή με @implementation
- τέλος με @end

Παράδειγμα αρχείου .m (1/2)

```
#import <MyClass.h>
```

```
@implementation MyClass
```

```
+ (void)classMethod{  
    printf("Class Method\n");  
}
```

συνεχίζεται...

Παράδειγμα αρχείου .m (2/2)

```
- (void)instanceMethod{  
    printf("Instance Method\n");  
    printf("InstanceVariable:%d\n",  
InstanceVariable);  
}
```

@end

Εκτός από τα αρχεία .h και .m
χρησιμοποιείται και ένα άλλο
σημαντικό αρχείο, το nib
(γράφεται xib,
αλλά προφέρεται nib)

Ορισμός του αρχείου nib και τρόπος που χρησιμοποιείται (1/2)

Είναι ένας ειδικός τύπος αρχείου πόρων που χρησιμοποιείται για αποθήκευση διεπαφών χρήστη στις εφαρμογές iOS και MAC. Είναι ένα έγγραφο κατασκευής διεπαφών.

Χρησιμοποιείται για να σχεδιάσει τα οπτικά μέρη μιας εφαρμογής και κάποιες φορές για να διαμορφώσει μη οπτικά αντικείμενα (όπως αντικείμενα ελέγχου).

Ορισμός του αρχείου nib και τρόπος που χρησιμοποιείται (2/2)

Μπορεί να περιέχει αντικείμενα κράτησης θέσης που χρησιμοποιούνται για αναφορά σε αντικείμενα έξω από το έγγραφο, αλλά με αναφορές σε αντικείμενα στο έγγραφο. Μια ειδική κράτηση θέσης είναι το αρχείο του ιδιοκτήτη.

Αρχείο ιδιοκτήτη (file's owner) στο αρχείο nib

Κατά το χρόνο εκτέλεσης, το αρχείο nib φορτώνεται χρησιμοποιώντας τη μέθοδο `loadNibNamed:owner:` ή μια παραλλαγή του. Το αρχείο του ιδιοκτήτη είναι μια κράτηση θέσης στο αρχείο nib για το αντικείμενο που θα περάσει ως παράμετρος ιδιοκτήτη της μεθόδου. Όλες οι συνδέσεις από και προς το αρχείο του ιδιοκτήτη, καθορίζονται ξανά όταν φορτώνεται το αρχείο στο χρόνο εκτέλεσης.

Εισαγωγή στο Bundle

Ορισμός δέσμης

Κατάλογος με ομαδοποιημένους εκτελέσιμους κώδικες και στοιχεία που χρησιμοποιούνται σε αυτούς

Χρησιμοποιείται σε

- iOS
- OS X

Λόγος που χρησιμοποιείται και περιεχόμενα της δέσμης

Χρησιμοποιείται για

- εύκολη εγκατάσταση και μεταφορά

Περιλαμβάνει

- εκτελέσιμο κώδικα, εικόνες, ήχους, αρχεία nib, βιβλιοθήκες, plug-ins κλπ

Εισαγωγή στο Framework

Ορισμός framework

Είναι ένα σύνολο (ένας δομημένος κατάλογος) που περιέχει μια δυναμική διαμοιραζόμενη βιβλιοθήκη μαζί με συνδεδεμένους πόρους, όπως nib αρχεία, εικόνες, header αρχεία. Σε μια εφαρμογή το project συνδέεται με ένα ή περισσότερα frameworks.

Λειτουργία του framework

Ο κώδικας έχει πρόσβαση στις δυνατότητες ενός πλαισίου μέσω API που δημοσιεύεται από το πλαίσιο μέσω του header αρχείου του. Πολλές εφαρμογές έχουν πρόσβαση στα frameworks και τους πόρους ταυτόχρονα. Το σύστημα τα φορτώνει στη μνήμη και μοιράζεται το ένα αντίγραφο ενός πόρου με όλες τις εφαρμογές.

Τρόπος που πραγματοποιείται η πρόσβαση στο framework

Η πρόσβαση στα περιεχόμενα μπορεί να γίνει μέσω της κλάσης `NSBundle` ή μέσω `CFBundle` του `Core Foundation`. Δεν επιτρέπονται `third-party frameworks` στο `iOS`. Τα `header` αρχεία ενός `framework` φαίνονται από το `Xcode`.

Cocoa και Cocoa Touch

Ορισμός Cocoa και Cocoa Touch

Περιβάλλοντα ανάπτυξης εφαρμογών

Περιβάλλον	Cocoa	Cocoa Touch
Περιλαμβάνει	Foundation, Appkit Frameworks	Foundation, UIKit Frameworks
Χρησιμοποιείται	OS X	iOS

Λειτουργία του Foundation Framework

- υλοποιεί την κύρια κλάση, η οποία καθορίζει τις βασικές συμπεριφορές των αντικειμένων
- υλοποιεί κλάσεις με τύπους, όπως αριθμούς και strings και με συλλογές, όπως arrays και dictionaries
- διευκολύνει σε ζητήματα όπως διαχείριση αρχείων, επεξεργασία XML

Appkit και UIKit Framework και περιεχόμενα

- χρησιμοποιούνται για ανάπτυξη διεπαφής χρήστη μιας εφαρμογής
- είναι ισοδύναμα σε σχέση με το σκοπό, αλλά διαφορετικά σε σχέση με την πλατφόρμα
- περιλαμβάνουν κλάσεις για χειρισμό γεγονότων και εικόνων, επεξεργασία κειμένου κλπ
- περιλαμβάνουν στοιχεία διεπαφής χρήστη, όπως πίνακες, κουμπιά, πεδία κειμένου κλπ

Κλάσεις

Ορισμός κλάσης

Σύνολο δηλώσεων που περιγράφει τα χαρακτηριστικά και τη συμπεριφορά ενός συγκεκριμένου συνόλου αντικειμένων.

Ορισμός μεθόδων και οι κατηγορίες τους

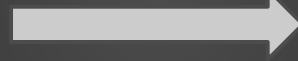
Συμπεριφορές που διαθέτει μία κλάση

1. Μέθοδοι κλάσης (δηλώνονται με +)
2. Μέθοδοι αντικειμένων (δηλώνονται με -)

Δήλωση και ορισμός μιας κλάσης

MyClass

Δήλωση



MyClass.h

Ορισμός



MyClass.m

Ομαδοποίηση κλάσεων

Αρχιτεκτονική:

abstract superclass



private subclass

concrete subclass

Χαρακτηριστικά ομαδοποίησης κλάσεων

- απλοποιημένο περιβάλλον
- καλείται η πιο αποδοτική κλάση ανάλογα με την περίπτωση
- καλύτερη απόδοση

Δήλωση της μεθόδου κλάσης

δήλωση:

```
+ (void)classMethod;
```

αποστολή μηνύματος σε κλάση:

```
[MyClass classMethod];
```


Αναφορά του `self` σε μια μέθοδο κλάσης (1/2)

Αναφέρεται στο ίδιο το αντικείμενο της κλάσης.

```
+ (id)myClass {  
    return [[[self alloc] init] autorelease];  
}
```

Αναφορά του `self` σε μια μέθοδο κλάσης (2/2)

Μπορεί επίσης να αναφέρεται στην κλάση στην οποία στάλθηκε το μήνυμα.

π.χ. μπορεί να υπάρχει μια υποκλάση σε μια κλάση και να στέλνεται μήνυμα σε αυτήν. Τότε το `self` αναφέρεται στην υποκλάση.

Ορισμός κύριας κλάσης

Δεν κληρονομεί από καμία άλλη κλάση και ορίζει διεπαφές και συμπεριφορές κοινές για όλα τα αντικείμενα σε ιεραρχία κάτω από αυτήν. Όλα τα αντικείμενα αυτής της ιεραρχίας κληρονομούν από την κύρια κλάση. Αναφέρεται και ως βασική κλάση.

Κύρια κλάση στην Objective-C και η λειτουργία της

Η κύρια κλάση όλων των κλάσεων στην Objective-C είναι η NSObject, η οποία αποτελεί μέρος του foundation framework. Όλα τα αντικείμενα σε Cocoa και Cocoa Touch κληρονομούν από αυτήν. Δηλώνει τη θεμελιώδη διεπαφή αντικειμένου και εφαρμόζει τη βασική συμπεριφορά του αντικειμένου, όπως διαχείριση μνήμης και κλήση της μεθόδου.

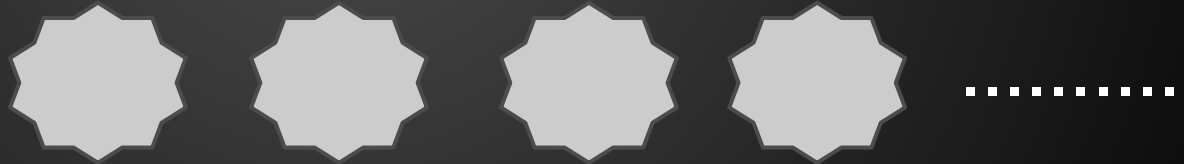
Αντικείμενα

Σχέση κλάσης - αντικειμένου

Κλάση:



Αντικείμενο:



Ορισμός αντικειμένου

- Αποτελεί εκδοχή μιας συγκεκριμένης κλάσης
- Η κλάση παράγει πολλά αντικείμενα
- Κάθε εκδοχή έχει δικές της τιμές για τα συγκεκριμένα δεδομένα

Δήλωση αντικειμένου

- Για τη δημιουργία μιας κλάσης απαιτείται να διατεθεί και να αρχικοποιηθεί ένα μπλοκ μνήμης
- Τα αντικείμενα είναι δείκτες σε αυτό το μπλοκ μνήμης
- * : υποδηλώνει ότι μία μεταβλητή είναι ένα είδος δείκτη

Παράδειγμα δήλωσης αντικειμένου

```
MyObject *someObject = [[MyObject alloc] init];
```

Ορισμός αρχικοποίησης αντικειμένου

Είναι το στάδιο δημιουργίας αντικειμένου που κάνει ένα αντικείμενο να μπορεί να χρησιμοποιηθεί δίνοντάς του αρχικές τιμές. Πρέπει πάντα να γίνεται μετά τη δέσμευση μνήμης. Η αρχικοποίηση μπορεί επίσης να εκτελέσει και άλλες διεργασίες για να θέσει ένα αντικείμενο σε χρήση, όπως προσθήκη πόρων.

Μορφή της δήλωσης αρχικοποίησης

Το όνομα της αρχικοποίησης ξεκινάει πάντα με `init`. Επιστρέφει ένα δυναμικά δηλωμένο αντικείμενο (`id`) ή επιστρέφει `nil` (εάν η αρχικοποίηση δεν πετύχει). Μπορεί να περιλαμβάνει μία ή περισσότερες παραμέτρους που καθορίζουν τις αρχικές τιμές.

- (id)initWithData:(NSData *)data encoding:(NSStringEncoding)encoding

Υλοποίηση της αρχικοποίησης (1/2)

Μια κλάση γενικά υλοποιεί μια αρχικοποίηση για τα αντικείμενά της, αλλά δεν είναι απαραίτητο. Αν μια κλάση δεν υλοποιεί αρχικοποίηση, το C++ καλεί την αρχικοποίηση του πλησιέστερου προγόνου της κλάσης.

Υλοποίηση της αρχικοποίησης (2/2)

Οι υποκλάσεις συνήθως καθορίζουν τις δικές τους αρχικοποιήσεις, αλλά μπορούν να επικαλούνται και από την υπερκλάση. Η κλάση `NSObject` δηλώνει τη μέθοδο `init` ως προεπιλεγμένη.

Βασικά βήματα για τη δήλωση της μεθόδου αρχικοποίησης (1/2)

1. Επίκληση της αρχικοποίησης της υπερκλάσης και έλεγχος για την τιμή που επιστρέφει (χρήση της δεσμευμένης λέξης `super` για την υπερκλάση). Εάν η τιμή δεν είναι `nil`, επιστρέφει ένα έγκυρο αντικείμενο και μπορεί να συνεχίσει η διαδικασία αρχικοποίησης.

Βασικά βήματα για τη δήλωση της μεθόδου αρχικοποίησης (2/2)

2. Ανάθεση τιμών στις instance μεταβλητές του αντικειμένου.
3. Επιστρέφει το αρχικοποιημένο αντικείμενο ή επιστρέφει nil (εάν δεν πέτυχε η αρχικοποίηση).

Σύγκριση αντικειμένων

Σύγκριση ισότητας ενός αντικειμένου με ένα άλλο

Αναφέρεται στην ικανότητα του αντικειμένου να αποφασίσει εάν είναι το ίδιο με ένα άλλο αντικείμενο. Αυτό επιτυγχάνεται στέλνοντας ένα μήνυμα `isEqual`: το ένα αντικείμενο στο άλλο. Η απάντηση μπορεί να είναι YES/NO. Κάθε κλάση αποφασίζει για την ισότητα. Η κύρια κλάση αποφασίζει με σύγκριση δείκτη.

Παράδειγμα σύγκρισης ισότητας αντικειμένου με isEqual:

```
- (BOOL)isEqual:(id)other{
    if (other == self)
        return YES;
    if (![super isEqual:other])
        return NO;
    return ([self name] isEqualToString:[other name]);
}
```

Σύγκριση αντικειμένου με τη μέθοδο `compare`:

Η μέθοδος `compare`: χρησιμοποιείται όταν θέλουμε να συγκρίνουμε δύο αντικείμενα και να μάθουμε εάν το ένα είναι ίσο με το άλλο, μεγαλύτερο ή μικρότερο. Τα αποτελέσματα μπορεί να είναι `NSOrderedSame`, `NSOrderedDescending` ή `NSOrderedAscending` αντίστοιχα.

Παράδειγμα σύγκρισης ισότητας αντικειμένου με compare:

```
result = [myNumber1 compare: myNumber2];  
if (result == NSOrderedSame)  
    NSLog(@"Numbers are equal");  
else if (result == NSOrderedAscending)  
    NSLog(@"Number1 is less than Number2");  
else if (result == NSOrderedDescending)  
    NSLog(@"Number1 is greater than Number2");
```

Δυναμική και Στατική Δήλωση

Ορισμός δυναμικής δήλωσης

Μια μεταβλητή είναι δυναμικά δηλωμένη όταν το είδος του αντικειμένου δεν ελέγχεται κατά τη μεταγλώττιση. Η Objective-C χρησιμοποιεί τον τύπο δεδομένων `id` να αντιπροσωπεύσει μια μεταβλητή που είναι ένα αντικείμενο, χωρίς να προσδιορίζεται το είδος του αντικειμένου.

Σύγκριση δυναμικής και στατικής δήλωσης

- Η δυναμική δήλωση έρχεται σε αντίθεση με τη στατική στην οποία καθορίζεται ρητά η κλάση στην οποία ανήκει ένα αντικείμενο κατά τη μεταγλώττιση.
- Ο έλεγχος στατικού τύπου κατά τη μεταγλώττιση μπορεί να εξασφαλίσει την ακεραιότητα των δεδομένων, αλλά στο δυναμικό τύπο δίνεται μεγαλύτερη ευελιξία στο πρόγραμμα.

Παράδειγμα δυναμικής δήλωσης

```
NSArray *anArray = [NSArray arrayWithObjects:@"A string",  
[NSDecimalNumber zero], [NSDate date], nil];
```

```
NSInteger index;
```

```
for (index = 0; index < 3; index++) {  
    id anObject = [anArray objectAtIndex:index];  
    NSLog(@"Object at index %d is %@", index, [anObject description]);  
}
```

Το αντικείμενο που δείχνεται από τη μεταβλητή κατά το χρόνο εκτέλεσης, πρέπει να είναι σε θέση να ανταποκριθεί σε όλα τα μηνύματα. Η υλοποίηση της μεθόδου καθορίζεται με τη χρήση δυναμικής δέσμευσης.

Δείκτης isa στη δυναμική δήλωση

isa pointer

Κάθε αντικείμενο έχει μια isa μεταβλητή που προσδιορίζει την κλάση του αντικειμένου. Ο χρόνος εκτέλεσης χρησιμοποιεί αυτόν το δείκτη για να καθορίσει την πραγματική κλάση του αντικειμένου όταν αυτό χρειάζεται.

Συνθήκη ελέγχου και απαρίθμηση

Σύνταξη συνθήκης ελέγχου if στην Objective-C (1/2)

```
if (condition1) {
```

```
}
```

```
else if (condition2) {
```

```
}
```

```
else {
```

```
}
```

Σύνταξη συνθήκης ελέγχου if στην Objective-C (2/2)

Εάν η συνθήκη είναι μόνο μία, δε χρειάζονται
τα {}

```
if (condition)  
    statement
```

Σύνταξη συνθήκης ελέγχου switch στην Objective-C

```
switch (expression) {  
    case 1:  
        statement  
        break;  
    case 2:  
        statement  
        break;  
    default:  
        statement  
        break;  
}
```

Ορισμός απαρίθμησης

- είναι η διαδικασία διαδοχικής λειτουργίας σε στοιχεία ενός αντικειμένου
- επίσης είναι η διαδικασία επανάληψης
- όταν απαριθμείται ένα αντικείμενο, μπορεί να επιλεγθεί ένα στοιχείο του αντικειμένου κάθε φορά και να εκτελεστεί μια λειτουργία με αυτό

Παράδειγμα απαρίθμησης

```
NSArray *array;  
NSInteger i, count = [array count];  
for (i = 0; i < count; i++) {  
    id element = [array objectAtIndex:i];  
}
```

Η χρήση for loop μπορεί να είναι αναποτελεσματική, καθώς πρέπει να είναι διατεταγμένη η συλλογή στοιχείων. Η απαρίθμηση είναι πιο γενική. Έτσι μέσω Cocoa προσφέρονται δύο επιλογές για απαρίθμηση αντικειμένων:

1. `NSEnumerator class`
2. `fast enumeration`

Παράδειγμα απαρίθμησης με NSEnumerator

```
NSSet *aSet = // a set;
```

```
NSEnumerator *enumerator = [aSet objectEnumerator];
```

```
id element;
```

```
while ((element = [enumerator nextObject])) {
```

```
    /* code */
```

```
}
```


Παράδειγμα απαρίθμησης με Fast Enumeration

```
NSArray *anArray = // an array;
```

```
for (id element in anArray) {  
    /* code */  
}
```

Είναι η πιο αποτελεσματική από όλες τις μορφές απαρίθμησης.

Εξουσιοδότηση αντικειμένου

Ορισμός εξουσιοδότησης

- απλό και ισχυρό μοτίβο με το οποίο ένα αντικείμενο ενεργεί εκ μέρους ή σε συνεργασία με ένα άλλο αντικείμενο
- το αντικείμενο που αναθέτει, έχει επαφή με το άλλο και του στέλνει μήνυμα την κατάλληλη στιγμή
- το μήνυμα ενημερώνει ότι πρόκειται να γίνουν ή έγιναν ενέργειες

Πλεονέκτημα εξουσιοδότησης

Το βασικό πλεονέκτημα είναι ότι επιτρέπει να προσαρμόζουμε εύκολα τη συμπεριφορά διάφορων αντικειμένων με ένα κεντρικό αντικείμενο.

Λόγος για να επιλέξουμε εξουσιοδότηση και χρήση πολλών εκπροσώπων

Βοηθάει στο να

- διαχωριστούν τα μοντέλα από τους ελεγκτές
- μεταφερθούν πληροφορίες μεταξύ των αντικειμένων

Δεν είναι υποχρεωτικό να χρησιμοποιηθεί μόνο ένας εκπρόσωπος ενός αντικειμένου, μπορούν να χρησιμοποιηθούν περισσότεροι.

Τρόπος υλοποίησης της εξουσιοδότησης

Η ανάθεση αντικειμένου γίνεται συνήθως από ένα framework αντικείμενο σε ένα αντικείμενο ελεγκτή.

- π.χ. όταν ο χρήστης θέλει να κλείσει ένα παράθυρο, το ένα αντικείμενο στέλνει μήνυμα ζητώντας την επιβεβαίωση για το κλείσιμο του παραθύρου και το άλλο αντικείμενο επιστρέφει μια τιμή Boolean, ελέγχοντας τη συμπεριφορά του παραθύρου

Συνάρτηση

Σύνταξη συνάρτησης στην Objective-C

```
<return type> <function name> (<arg1 type> <arg1 name>,  
<arg2 type> <arg2 name>, .....)  
{  
    //code  
}
```


Εξαιρέσεις

Ορισμός χειρισμού εξαίρεσης

Είναι η διαδικασία διαχείρισης άτυπων γεγονότων που διακόπτουν την κανονική ροή εκτέλεσης των προγραμμάτων. Χωρίς την κατάλληλη αντιμετώπιση των λαθών, ένα πρόγραμμα που αντιμετωπίζει ένα άτυπο γεγονός πιθανότατα θα τερματίσει δημιουργώντας μια εξαίρεση (throw).

Τύποι εξαίρεσης

Τόσο το hardware όσο και το software μπορούν να προκαλέσουν εξαίρεση.

- αριθμητικά σφάλματα (διαίρεση με το 0)
- υπερχείλιση, υποχείλιση
- απροσδιόριστες οδηγίες (προσπάθεια για μη εφαρμόσιμη μέθοδο)
- πρόσβαση σε στοιχείο συλλογής εκτός ορίων

Χειρισμός εξαιρέσεων μέσω software (1/2)

1. `@try` block περιλαμβάνει κώδικα που μπορεί να δημιουργήσει εξαίρεση
2. `@catch()` block ακολουθεί ένα block `@try` για εξαιρέσεις που έχουν ήδη συμβεί. Μπορούν να υπάρξουν πολλαπλά block `@ catch()` για διάφορα είδη εξαιρέσεων

Χειρισμός εξαιρέσεων μέσω software (2/2)

3. `@finally` block περιέχει κώδικα που πρέπει να εκτελεστεί, είτε εάν είχε υπάρξει εξαίρεση είτε όχι
4. `@throw` δημιουργεί μια εξαίρεση που είναι ουσιαστικά ένα αντικείμενο Objective-C. Συνήθως χρησιμοποιείται ένα `NSException` αντικείμενο αλλά δεν είναι απαραίτητο

Παραδείγματα χειρισμού εξαιρέσεων μέσω software

```
Cup *cup = [[Cup alloc] init];
@try {
    [cup fill];
}
@catch (NSException *exception) {
    NSLog(@"main: Caught %@: %@", [exception name], [exception reason]);
}
@finally {
    [cup release];
}
```

Βιβλιογραφία

Χρησιμοποιήθηκε υλικό από:

www.apple.com

iOS Programming: The big Nerd Ranch Guide

iOS 7 Programming Pushing the Limits

iOS 7 Programming Cookbook