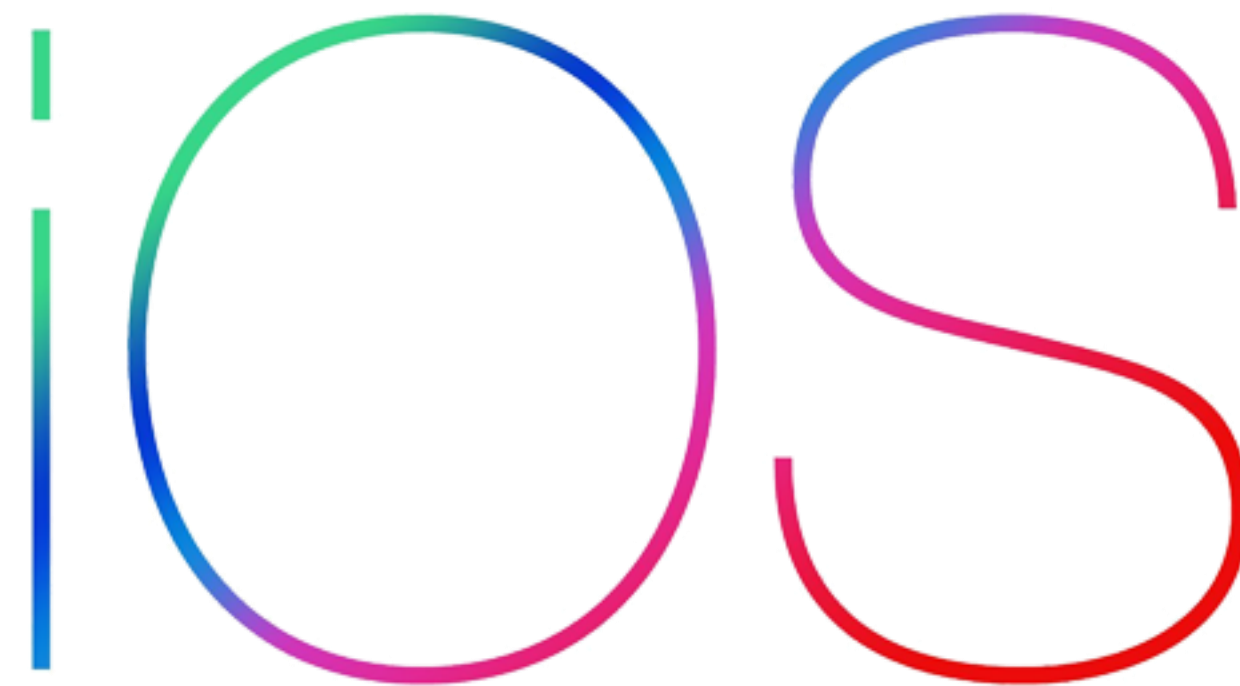


Operativni sistem



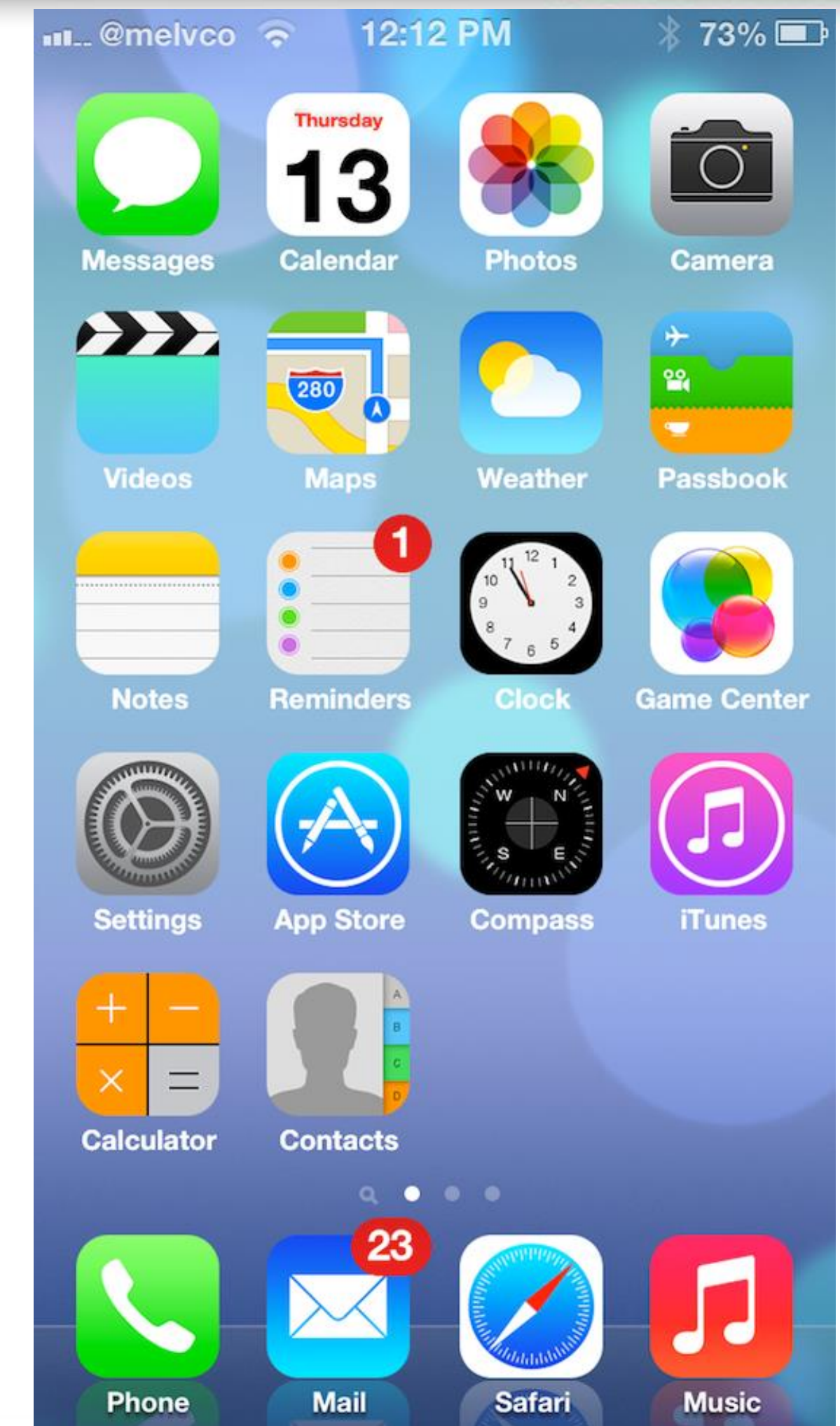
Uvod, istorija, razvoj i arhitektura

Ivan Nedeljković

ivan.nedeljkovic@outlook.com

Uvod

- iOS je namenjen isključivo uređajima kompanije Apple
- Korisnički interfejs baziran je na direktnoj manipulaciji
- Multi-touch gestures:
 - tap
 - swipe
 - pinch
 - reverse pinch
- Korišćenje internih uređaja za komunikaciju (akcelerometar)
 - shake - brise uneti tekst
 - okretanje uređaja menja prikaz (portrait/landscape)



Uvod

- Trenutno iOS postoji na sledećim uređajima:
 - iPhone
 - iPad
 - iPod
 - Apple TV

Istorija i razvoj

- Prvi put predstavljen je javnosti 2007. kao operativni sistem prve verzije iPhone-a
- 2008. Apple predstavlja App Store kada i objavljuje SDK za razvoj third-party aplikacija
- Sâm operativni sistem biva nazvan “iPhone OS”, a svoj današnji naziv “iOS” dobija 2010. godine
- U maju 2010. broj aplikacija u App Store-u dostiže 500.000

Istorija i razvoj

- Od početka razvoja do septembra 2013. izgled korisničkog interfejsa se nije suštinski menjao
- Septembra 2013. Apple objavljuje potpuno redizajnirani iOS 7
- Aktuelna verzija iOS 8 donela je mnoge novosti:
 - Health
 - Apple Pay
 - HomeKit

Trenutne statistike

- Broj aplikacija na App Store-u: **1.200.000**
 - PlayStore: **1.300.000**
- Novac isplaćen programerima: **5.000.000.000\$**
 - Google: **900.000.000**
- Prosečna zarada po download-u: **0.1\$**
 - Google: **0.018\$**
- Prosečna zarada po developer-u: **21.276\$**
 - Google: **6.000\$**

Učešće na tržištu (3. kvartal 2014)

Zemlja/Platforma	iOS	Android	Trend iOS
Nemačka	20.9	71	+3.6
USA	47.7	47.6	+3.8
Velika Britanija	42	50.5	+11.1
Kina	21.5	77	+2.5
Japan	60.2	38.3	-8.5
Australija	45.1	43.7	10.3
Italija	18.3	67.3	5.5

Bitni koncepti

- **In-app purchases** - kupovina dodatnih sadržaja u okviru aplikacije
- **Game Center** - dozvoljava korisnicima da igraju i dele igre sa svojim prijateljima. Prati progres i prikazuje tabele.
- **Touch ID** - aplikacije mogu koristiti čitač otiska prsta za autentifikaciju
- **Apple Pay** - upotreba Touch ID-a za lako i sigurno plaćanje dobara i usluga
- **HomeKit** - novi framework za komunikaciju i kontrolu povezanih uređaja u korisnikovom domu

Apple Ecosystem

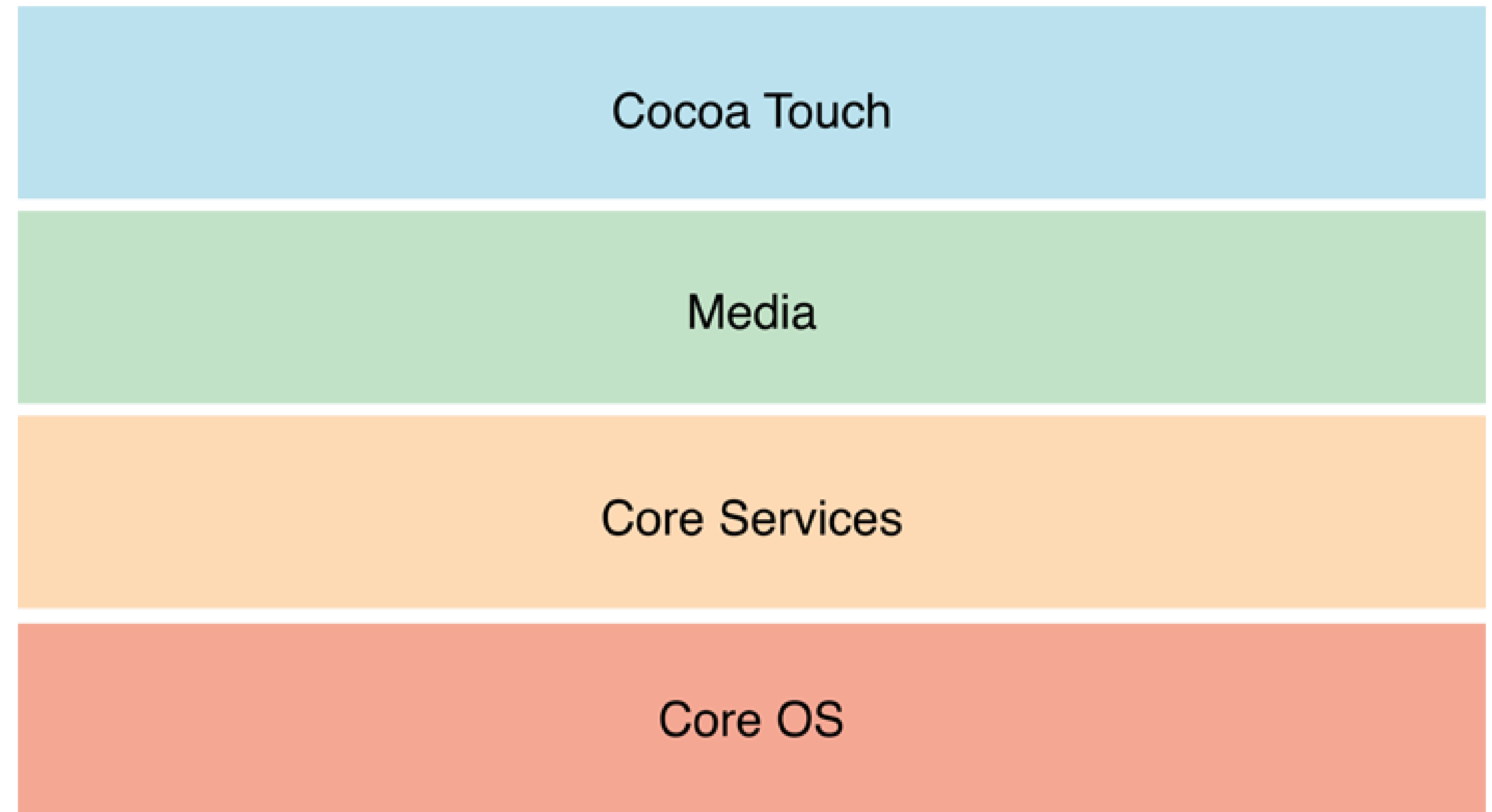


Arhitektura

- Zasnovan na Unix kernelu operativnog sistema Darwin
- Darwin je Apple-ov projekat otvorenog koda
- iOS i MAC OS X dele zajednicki kernel
- Nije moguća portabilnost aplikacija između ova dva sistema

Arhitektura

- Osnovni slojevi
 - Cocoa Touch
 - Media
 - Core Services
 - Core OS



Core OS

- Core Bluetooth Framework
- External Accessory Framework
- System Framework
- Local Authentication Framework

Core Services

- Address Book Framework
- Ad Support Framework
- CloudKit Framework
- Core Data Framework
- Core Location Framework
- EventKit Framework

Media

- Core Graphics Framework (Quartz)
- Core Animation
- Core Image
- Photos Library
- Image I/O

Cocoa Touch

- App Extensions (Share, Widgets, Custom keyboard)
- Handoff
- AirDrop
- TextKit
- Multitasking
- Push Notifications
- Local Notifications

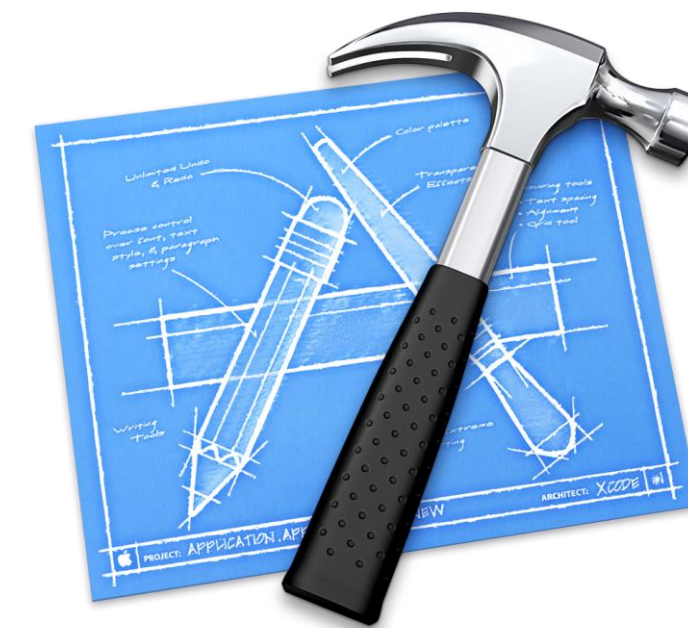
Razvoj i objavljivanje aplikacije

- Potrebni alati:

- Mac računar



- Razvojno okruženje (Xcode)



- Uređaj za testiranje



Razvoj i objavljivanje aplikacije

- Neophodni uslovi
 - Otvoren Apple Developer Account
 - Aplikacija ispunjava **Human Interface Guidelines**
 - Aplikacija ispunjava **App Store Review Guidelines**

Objective C

- Jezik opšte namene
- Nastao sedamdesetih godina 20. veka
- Superset jezika C
- Objektna orijentisanost i sistem poruka preuzeti iz Smalltalk-a
- Koristi se za razvoj iOS i MAC OS X aplikacija

Karakteristike jezika Objective C

- Slanje poruka - message passing
- Dinamičko povezivanje - dynamic binding
- Pointeri
- Svaka klasa se sastoji iz dva fajla (.m i .h fajl)

.h fajl (Header)

- .h fajl predstavlja interfejs klase
- U njemu se definišu varijable, svojstva i metode

```
#import <UIKit/UIKit.h>

@interface classname : superclass
{
    //instance variables
}

@property (attribute1, attribute2) property_primitive_type property_name1;
@property (attribute1, attribute2) property_type *property_name2;

+ classMethod1;
+ (return_type)classMethod2;
+ (return_type)classMethod3:(param1_type)param1_varName;

- (return_type)instanceMethod1With1Parameter:(param1_type)param1_varName;
- (return_type)instanceMethod2With2Parameters:(param1_type)param1_varName
    param2_callName:(param2_type)param2_varName;

@end
```


.m fajl (Messages ili Methods)

- .m fajl sadrži implementaciju klase

```
@implementation classname  
+ (return_type)classMethod  
{  
    // implementation  
}  
- (return_type)instanceMethod  
{  
    // implementation  
}  
@end
```




Center for Career Development

by LINKgroup

```
#import <Foundation/Foundation.h>
```

```
@interface Card : NSObject
```

```
@property (strong, nonatomic) NSString *contents;
```

```
@property (nonatomic, getter=isChosen) BOOL chosen;
```

```
@property (nonatomic, getter=isMatched) BOOL matched;
```

```
- (int)match:(NSArray *)otherCards;
```

```
@end
```

```
#import "Card.h"
```

```
@interface Card()
```

```
@end
```

```
@implementation Card
```

```
- (int)match:(NSArray *)otherCards  
{
```

```
    int score = 0;
```

```
    for (Card *card in otherCards) {
```

```
        if ([card.contents isEqualToString:self.contents]) {  
            score = 1;
```

```
        }
```

```
    }
```

```
    return score;
```

```
}
```

```
@end
```

Stanford CS193p
Fall 2013



Ivan Nedeljković

ivan.nedeljkovic@outlook.com