

czech

iOS Development

Marcel Baláš

2. září 2014

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator
- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator
- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

Objective C

- Kompletní a "čisté" C
- Plně objektová nadstavba
- Dynamický jazyk (dynamické typy + dynamický binding)
- Komunikace mezi objekty: zasílání zpráv

Objective C

Zprávy mezi objekty:

```
[Obj doSomething];
```

Lze řadit do sekvence:

```
UIImage * img = [[UIImage alloc] initWithData:data];
```

Objective C

- Properties:

```
@property (readonly, weak, nonatomic) something;
```

- ARC (Automatic Reference Counting)

- Blocks:

```
[self beginTaskWithName:@"MyTask" completion:^(  
    NSLog(@"The task is complete");  
)];
```

- přesměrování zpráv, proxy objekty, isaswizzling, trampolíny, higher order messages, ...

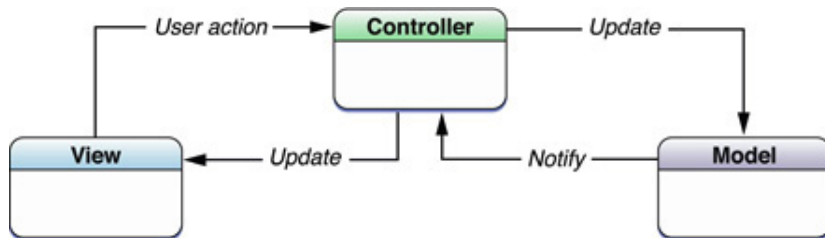
Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch**
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator
- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

Cocoa Touch

- Cocoa pro iOS = Cocoa Touch
- Foundation Kit stejný jako pro Mac OS X
- Nově navržený AppKit = UIKit
- Základní Design Pattern: MVC

MVC



Cocoa Touch

- **Views:** UITableView, UITextView, UIWebView, UITabBarView, UIButton, UILabel, ...
 - Views lze skládat do hierarchických sekvencí
 - Views jsou animovatelné přes Core Animation
- **Controllers:** UITableViewController, UINavigationController, UITabBarController, ...
- **Model:** Vlastní třídy, využívající tříd a služeb Foundation frameworku

Cocoa Touch

- **Cocoa** je ucelené, kompaktní, poměrně jednoduché API.
- Namísto vytváření milionu podtříd je v Cocoa zvykem použít alternativy:
 - Class clusters
 - Delegation
 - Target/Action
 - Kategorie

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 **XCode**
 - Interface Builder
 - Storyboard
- 4 iOS Simulator
- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

XCode

- Interface Builder
- Version editor (v pozadí git)
- iOS Simulator
- Instruments: analýza aplikace

XCode

- LLVM kompilátor
- Průběžná, inkrementální kompilace
- CLANG: Statická analýza

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator
- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

InterfaceBuilder

- Storyboards: kompletní UI aplikace včetně přechodů mezi Views
- Díky MVC odděleno od modelu a kontrolerů
- Auto Layout: novinka, zobrazení prvků dle *constraints* vůči ostatním prvkům, okrajům obrazovky, rodičovskému View, atd.

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 **XCode**
 - Interface Builder
 - **Storyboard**
- 4 iOS Simulator
- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

Storyboard

Storyboard: kompletní UI aplikace

- Scény: jednotlivé View / ViewController
- Segues: přechody mezi scénami

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator
- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

iOS Simulator

- Simuluje jakékoliv iOS zařízení (iPhone 3.5" + iPhone 4.0", iPad, 32 + 64 bit, Retina + nonRetina)
- Simulace všech možných gest
- Až na pár HW výjimek (proximity sensor, gyroscope, camera, ...) obsahuje kompletní funkcionalitu reálného zařízení
- V mém případě: Core Data, lokace (mapy), lokalizace, web

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator

- 5 **FlexiTicket**
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator

- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

App Definition Statement

Application Definition Statement

Application for selling transport tickets

Target Platform

iOS7+, iPhone

Vlastnosti

- Nabídnout dle zadaných kritérií seznam dostupných jízdenek
- Umožnit jízdenku "koupit"
- Shromažďovat historii nakoupených jízdenek

Hlavní cíle z pohledu UX

- Intuitivní
- Jednoduché na používání
- Decentní grafika (standardní UI objekty, bílé iOS7 schéma, nepřelácané, málo barev)
- Jednoduchý layout pouze se základní funkcionalitou

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator

- 5 FlexiTicket
 - Základní popis
 - **Technologie**
 - Core Data
 - Lokalizace
- 6 Shrnutí

Technologie

- Asynchronní komunikace se serverem (seznam dostupných jízdenek)
- XML (parsování seznamu jízdenek)
- Core Data pro lokální úložiště
- Lokační služby

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator

- 5 FlexiTicket
 - Základní popis
 - Technologie
 - **Core Data**
 - Lokalizace
- 6 Shrnutí

Core Data základy

- Model layer framework
- Object relational mapping
- Persistentní technologie (serializace do: XML / binárky / SQLite)
- Poskytuje: change tracking, undo management, migrace schémat, "faulting", ...
- Snadno integrovatelné do UITableViewController (a odtud do UI)

Core Data užití

- V IB navrhne datový model
- XCode automaticky vygeneruje kód pro modelové třídy
- Doplníme kód pro ukládání nových instancí, vytahování instancí z persistentního modelu, atd.

CoreData request 1

```
// Let's have following class (NSManagedObject as a
    parent):

@interface Employee : NSManagedObject

@property (nonatomic) NSString *firstName, *lastName;
@property (nonatomic) Department *department;
@property (nonatomic) Employee *manager;
@property (nonatomic) NSDecimal *salary;
@end
```

Core Data request 2

```
// We are searching for Employee entities.
NSEntityDescription * ed = [NSEntityDescription
    entityForName:@"Employee"
    inManagedObjectContext:moc];
NSFetchRequest * request = [[NSFetchRequest alloc] init];
[request setEntity:ed];

// We want all employees whose name begins with 'name'
// and whose manager's salary is greater than
// 'minSalary'.
NSPredicate * pred = [NSPredicate predicateWithFormat:
    @"(lastName BEGINSWITH[c] %@) AND (manager.salary >
    %@)", name, minSalary];
[request setPredicate:pred];
```


Core Data request 3

```
// And we want it in an ascending order based on
    employee's first name.

NSSortDescriptor * sortDescriptor = [[NSSortDescriptor
    alloc] initWithKey:@"firstName" ascending:YES];
[request setSortDescriptors:@[sortDescriptor]];

// Execute the fetch request.

NSError * error;
NSArray * array = [moc executeFetchRequest:request
    error:&error];
```

Core Data NSPredicate

- Základní porovnání `=`, `!=`, `<`, `>`, `<=`, `>=`, `BETWEEN`
- Složené predikáty: `AND`, `OR`, and `NOT`
- Stringové predikáty: `BEGINSWITH`, `LIKE`, `CONTAINS`, `ENDSWITH`, `MATCHES`
- Agregátní predikáty: `ANY`, `SOME`, `ALL`, `NONE`, `IN`, `array[index]`, `array[FIRST]`, `array[LAST]`, `array[SIZE]`

Core Data propojení

Core Data lze relativně snadno propojit s UI přes `NSFetchResultsController`:

- UI zobrazující seznam položek v `UITableView`
- Obsah (data source) je dodáván přes `NSFetchResultsController`
- Jakákoliv změna v modelu je přes notifikace delegátovi propagována až do UI

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator

- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

Lokalizace

Lokalizace na několika úrovních:

- Storyboard: XCode sám oskenuje Storyboard a pro každý jazyk vytvoří .string soubor obsahující všechny UI texty
- Obrázků: Pro každý jazyk je možné mít speciální sadu obrázků
- Dynamické texty: Manuálně vytvořený .string soubor pro každý jazyk
- Date/Measurement formatters, např.:

```
NSDateFormatter * dateFormatter = [[NSDateFormatter  
    alloc] init];  
[dateFormatter setLocale:[NSLocale currentLocale]];
```

Lokalizace textů

- Soubor je ve formátu:

```
/* Optional comment used for context. */  
"Key" = "Value"
```

- V kódu pak namísto statických stringů používáme:

```
// NSString * msg = @"Value";  
NSString * msg = NSLocalizedString(@"Key", nil);
```

Outline

-
- 1 ObjectiveC
- 2 Cocoa Touch
- 3 XCode
 - Interface Builder
 - Storyboard
- 4 iOS Simulator
- 5 FlexiTicket
 - Základní popis
 - Technologie
 - Core Data
 - Lokalizace
- 6 Shrnutí

Shrnutí

Líbí

- Objective C
- XCode
- CocoaTouch
- Bezkonkurenční dokumentace (<https://developer.apple.com/>)
- Spousta sample kódu

Nelíbí

- Nikdo znalý iOS v okolí
- Přepínání projektů
- Málo času

S čím se smířit:

- Každý rok nové API, nové přístupy.