

## Program szkolenia:

# Projektowanie, tworzenie i testowanie aplikacji mobilnych na platformie iOS

## Informacje:

<b>Nazwa:</b>	<b>Projektowanie, tworzenie i testowanie aplikacji mobilnych na platformie iOS</b>
<b>Kod:</b>	<b>ios-core</b>
<b>Kategoria:</b>	iOS
<b>Grupa docelowa:</b>	developerzy
<b>Czas trwania:</b>	4 dni
<b>Forma:</b>	40% wykłady / 60% warsztaty

Szkolenie zostało przygotowane z myślą o programistach pragnących rozpocząć pracę na platformie iOS oraz rozwinąć kompetencja do poziomu biegłego.

Szkolenie zawiera szereg praktycznych rozwiązań opracowanych na podstawie doświadczeń w dojrzałych projektach.

Program szkolenia został rozszerzony o aspekty architektury, wzorców i testowania automatycznego.

## Wymagania

Umiejętność programowania w dowolnym języku obiektowym oraz komputery Mac z zainstalowanym środowiskiem Xcode 5.x

## Zalety szkolenia:

- Aspekty wydajności
- Architektura i wzorce projektowe
- Typowe pułapki oraz najlepsze praktyki

## Szczegółowy program:

### 1. Język Objective-C.

- 1.1. Cel: Stworzenie projektu z szablonu, uruchomienie na symulatorze
- 1.2. Metody - a właściwie komunikaty (dynamiczna natura języka)
- 1.3. Klasy, protokoły, kategorie (czyli: klasy, interfejsy i mixiny)
- 1.4. Properties (czyli pola i niejawne gettery i settery)
- 1.5. Typy kolekcji (tablice, słowniki, zbiory)
- 1.6. Typu mutable i immutable
- 1.7. Wzorce projektowe: MVC, delegat, obserwator
- 1.8. Zarządzanie pamięcią: automatyczne (ARC) i manualne. Pule auto-release.
- 1.9. Bloki kodu (closures)
- 1.10. Architektura aplikacji iOS (AppDelegate, Info.plist, Pętla zdarzeń)
- 1.11. Cykl życia aplikacji (Przechodzenie do tła, powrót)

### 2. Narzędzia developerskie - przegląd

- 2.1. Xcode.
- 2.2. Workspace i projekty.
- 2.3. Schematy.
- 2.4. Statyczny analizator kodu.
- 2.5. XCTest - framework do testów,
  - 2.5.1. pisanie testów i uruchamianie

### 3. UIKit - biblioteka do programowania interfejsu

- 3.1. Cel: budowanie prostej aplikacji mieszczącej się na jednym ekranie
- 3.2. ViewController - kontroler widoku.
- 3.3. Wykorzystanie Interface Buildera do projektowania interfejsu.

3.4. Komponenty: Pola tekstowe, przyciski, suwaki, jak z nich korzystać (wzorzec delegate)

3.5. Łączenie komponentów z kodem: IBOutlet, Action.

3.6. Auto-layout: system dostosowania układu elementów na ekranie na podstawie zadanych ograniczeń

3.7. Budowanie aplikacji z widokiem listy (UITableViewController)

3.7.1. Budowanie własnych komórek listy (UITableViewCell)

3.8. Budowanie aplikacji z wieloma ekranami

3.8.1. Przełączanie między częściami aplikacji za pomocą Tab Bar.

3.8.2. UINavigationController - stos widoków - przechodzenie do kolejnych ekranów i powrót.

3.8.3. Przejścia między ekranami z wykorzystaniem Storyboard

## 4. Zagadnienia dodatkowe

4.1. Testy jednostkowe XCTest

4.2. Menedżer zależności CocoaPods (gdy zostanie użyta jakaś zewnętrzna biblioteka)

4.3. Parser JSON i XML.

4.4. Pobieranie danych z web-services.

4.5. Zapisywanie/odczytywanie preferencji aplikacji.