

## Zadanie 3 – iOS

### Wstęp:

Zadanie trzecie czas zacząć. Składa się ono z dwóch części: teoretycznej i praktycznej. Tym razem chciałbym także sprawdzić waszą umiejętność dokumentowania kodu. Zadanie powoli wdraża nas do tematyki projektu oraz ma na celu podsumowanie dotychczas zdobytej wiedzy. Zawiera ono często używane sposoby operacji zapisu/odczytu do pliku. A więc, proszę zapoznaj się wnikliwie z poniższymi materiałami.

- przeczytaj tutorial wprowadzający w następujące tematy:

- *Blocks*,

z przewodnika po Objective-C

<http://rypress.com/tutorials/objective-c/introduction.html>.

- zapoznaj się z następującymi rozdziałami:

- *Working with Blocks*,

z podręcznika "*Programming with Objective-C*"

<http://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/ProgrammingWithObjectiveC.pdf> (<http://tiny.pl/q5rf9>).

- zapoznaj się z dokumentacją następujących klas:

- *NSCoding*, *NSKeyedArchiver*, *NSKeyedUnarchiver*, *NSDate*, *NSString* (*Working with Paths*),

[https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/Archiving/Articles/codingobjects.html#apple\\_ref/doc/uid/20000948-BCIHBIDE](https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/Archiving/Articles/codingobjects.html#apple_ref/doc/uid/20000948-BCIHBIDE) (<http://tiny.pl/q5rcn>),

[https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/Archiving/Articles/creating.html#apple\\_ref/doc/uid/20000949-BABGBHCA](https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/Archiving/Articles/creating.html#apple_ref/doc/uid/20000949-BABGBHCA) (<http://tiny.pl/q5rcr>),

[https://developer.apple.com/library/ios/documentation/cocoa/reference/foundation/Classes/NSDate\\_Class/Reference/Reference.html](https://developer.apple.com/library/ios/documentation/cocoa/reference/foundation/Classes/NSDate_Class/Reference/Reference.html) (<http://tiny.pl/q5rc8>),

[https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString\\_Class/Reference/NSString.html#apple\\_ref/doc/uid/TP40003744](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#apple_ref/doc/uid/TP40003744) (<http://tiny.pl/q5rc6>).

- Zapoznaj się z instrukcją Doxygen, na temat dokumentowania w komentarzach:

<http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html#specialblock>.

## Zadanie:

Celem pierwszej części zadania to stworzenie aplikacji, która zapamięta trasę przebytą przez użytkownika (zapisze do pliku). A wymaga to stworzenia klasy **GpsSimulator**. Ma ona za zadanie symulować zmianę położenia użytkownika. To znaczy, że powinna posiadać metodę, która zwróci pomiar jego aktualnej pozycji oraz datę tego pomiaru. Dla uproszczenia przyjmujemy, że ruch użytkownika opisują współrzędne x i y, a datę pomiaru typ **NSDate**. Natomiast dane opisujące ruch użytkownika przechowywane są w klasie **UserData**. Zawiera ona wszystkie zarejestrowane pomiary, średnią prędkość użytkownika, czas trwania podróży i długość przebytej trasy. Klasa ta powinna implementować protokół **NSCoding**, dzięki czemu będzie można ją zapisać i odczytać. Uwaga! Przy zapisie, ścieżka do pliku powinna być generowana za pomocą funkcji z paragrafu „*Working with Paths*” dostępnych w klasie **NSString** (proszę patrzeć w linkach do dokumentacji).

Pokrótkie, program powinien działać w następujący sposób:

- zaczynamy pomiar danych (użytkownik zaczął maszerować),
- po otrzymaniu pojedynczego pomiaru,
- przekazać dane do obiektu typu **UserData**, który zaktualizuje średnią prędkość i długość przebytej trasy,
- zakończenie pomiaru (wymyśl warunek zakończenia pomiaru),
- zapisać obiekt z danymi użytkownika do pliku przy pomocy klasy **NSKeyedArchiver**,
- wyświetlić średnią prędkość i długość przebytej trasy,
- stworzyć nowy obiekt typu **UserData** i wczytać do niego zawartość pliku używając klasy **NSKeyedUnarchiver**,
- sprawdzić czy plik został poprawnie wczytany i wyświetlić informację o stanie poprawności.

Chciałbym, aby symulator GPS-u symulował trasę, która nie jest linią prostą. Powinien on także wykonać minimum 200 pomiarów, jest to nasze założenie dotyczące pojedynczej wycieczki użytkownika.

Rozwiązanie powyższego zadania umieść w pojedynczym pliku archiwum o nazwie **"ImieNazwisko.zip"**

Celem drugiej części zadania jest stworzenie krótkiej notatki opisującej działanie symulatora GPS, co to jest, najważniejsze waszym zdaniem cechy. Ma ona wyglądać w następujący sposób: na początku pliku **.h** powinien pojawić się komentarz opisujący klasę. Zastosujcie tu jedną z konwencji opisaną w poradniku Doxygen. Potraktujcie tę część jak pisanie dokumentacji technicznej.

Następnie po wykonaniu zadania. Zastanów się, jak można by było użyć bloków w sytuacji opisanej w zadaniu. Zaproponuj swoją koncepcję modyfikacji działania aplikacji, możesz też dołączyć jakiś przykład pseudo kodu. Ta część zadania jest również punktowana, a jej cel to sprawdzenie czy zapoznaliście się z materiałem dotyczącym bloków. Chciałbym, aby była ona zamieszczona w pliku **"bloki\_propozycja.txt"**

## Porady:

Pobranie ścieżki do katalogu dokumentów można wykonać w następujący sposób:

```
NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);
```