

Zadanie 2 – iOS

Wstęp:

Zadanie drugie czas zacząć. Jest ono trudniejsze i będzie surowiej oceniane. Tym razem poprawna kompilacja będzie bardziej punktowana. Zadanie powoli wdraża nas do tematyki projektu, który będziemy realizować. Zawiera ono często używane sposoby składowania danych i komunikację między klasami. Proszę zapoznać się wnikliwie z poniższymi materiałami. W przypadku poważniejszych problemów z kompilatorem GNUStep proszę o kontakt.

- przeczytaj tutorial wprowadzający w następujące tematy:

- *Data Types, Protocols, Categories,*

z przewodnika po Objective-C.

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

- zapoznaj się z następującymi rozdziałami:

- *Values and Collections,*
- *Working with Protocols,*
- *Customizing Existing Classes,*

z podręcznika **"Programming with Objective-C"**.

<http://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/ProgrammingWithObjectiveC.pdf>

- zapoznaj się z dokumentacją kontenera NSArray:

http://developer.apple.com/library/ios/documentation/cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html

Zadanie:

Celem zadania jest stworzenie klasy **ZooTrip** z następującymi własnościami:

- posiada listę predefiniowanych ścieżek do zwiedzania (użyj kontenera **NSArray**),
- powyższa lista powinna być zainicjalizowana następującymi danymi:
(Africa, 1500m,1.5h,6), (South America, 500m,0.4h,4), (Australia, 750m,0.5h,5), (Europe, 2000m,3h,6), (Asia, 1800m,1h,3).
- pojedynczy element listy (klasa **ZooPath**) zawiera pola takie jak:
Nazwę trasy, długość ścieżki, przewidywany czas zwiedzania oraz ocenę zwiedzających.
Dane te powinny być składowane w słowniku (**NSDictionary**) wewnątrz klasy i z niego wyciągane poprzez jej property (getery wyciągające dane ze słownika).
- posiada listę zwiedzanych ścieżek, używając kontenera **NSMutableArray**.

- funkcję **findBestPathByUserPreference**: która zwróci użytkownikowi najdłuższą ścieżkę odległościowo, najkrótszą czasowo, czy najbardziej polecaną przez zwiedzających. Wybór użytkownika będzie argumentem funkcji. Wymagane użycie funkcji **sortedArrayUsingDescriptors** klasy **NSArray**.
- posiada property o nazwie **delegate** udostępniające obsługę protokołu **ZooTripDelegate**
- posiada metodę **start**, która znajdzie trasę bazując na trasie najbardziej polecanej przez zwiedzających. Znalaziona ścieżka powinna zostać dodana do listy zwiedzanych ścieżek. Następnie powinna zostać wywołana pierwsza metoda protokołu, przekazująca znaną ścieżkę przez argument **startPath**.

Zdefiniuj protokół **ZooTripDelegate** z następującymi metodami:

- **-(void)zooTripDidStartOnPath:(ZooPath *)startPath;** (metoda wymagana),
- **-(void)zooTripDidFinish;** (metoda opcjonalna).

Stwórz kategorię nazwaną **Extended** dla klasy **ZooTrip** z następującymi właściwościami:

- posiada metodę **finish**, która usunie aktualną trasę z listy zwiedzanych ścieżek i wywoła drugą metodę protokołu.

Stwórz klasę **RemotePrinter**, która zaimplementuje reakcję na metody protokołu i wydrukuj w konsoli nazwę wywołanej metody oraz jej argument, jeżeli taki posiada.

Np. Reakcja na metodę **-(void)zooTripDidFinish** powinna wyświetlić następujący tekst **"print zooTripDidFinish"**.

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

Porady:

Przykładowy kod testowy

```
int main()
{
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];

    RemotePrinter *rPrinter = [[RemotePrinter new] autorelease];
    ZooTrip *zooTrip = [[ZooTrip new] autorelease];
    zooTrip.delegate = rPrinter;
    id bestPath = [zooTrip findBestPathByUserPreference:FastTrip];
    NSLog(@"Found Fastest path %@", bestPath);
    bestPath = [zooTrip findBestPathByUserPreference:RecommendedTrip];
    NSLog(@"Found Recommended path %@", bestPath);
    bestPath = [zooTrip findBestPathByUserPreference:LongTrip];
    NSLog(@"Found Longest path %@", bestPath);
    [zooTrip start];
    [zooTrip stop];

    [pool drain];
    return 0;
}
```