

TP 1 - Découverte d'Objective C et d'Xcode

Enseignant : Adrien Humiliere

11/03/2015

A la fin de ce TP :

- Faire une archive contenant les projets Xcode des exercices.
- Envoyer l'archive à adrien.humiliere@djit.fr.
- Le rendu ne sera pas noté mais fera l'objet d'une correction.
- Si le TP est fait à plusieurs, préciser les noms et adresses mail de chacun.

Exercice 1 : Liste chaînée

L'objectif de l'exercice est d'implémenter simplement une liste chaînée, uniquement avec les types de base d'Objective-C.

- Lancer Xcode et créer un nouveau projet de type Command Line Tool (OSX > Application > Command Line Tool).
- Créer une classe `Node`, avec deux attributs :
 - `key`, un entier non modifiable
 - `next`, une référence vers un objet de type `Node`Implémenter à minima la méthode `-description` qui, à la façon de la méthode `toString()` de Java, permet d'afficher l'état courant d'un objet dans le terminal.
- Créer une classe `List`, qui implémente une liste simplement chaînée. Cette classe devra implémenter la méthode `-description` et répondre aux méthodes suivantes :

```
- (BOOL)isEmpty;  
- (void)addFirst:(Node *)node;  
- (Node *)removeFirst;  
- (void)addLast:(Node *)node;  
- (Node *)removeLast;  
- (Node *)nodeForKey:(int)key;
```

- Tester le programme en utilisant dans la fonction `main()` chacune des méthodes implémentées pour la classe `List`, avec au moins 10 instances de `Node` dans la liste chaînée.

Exercice 2 : Première UI

L'objectif de cet exercice est d'avoir un premier contact avec les classes de Cocoa Touch et avec les outils de création d'UI fournis par Xcode.

- Créer un nouveau projet d'application iOS (iOS > Application > Single View Application).

- (b) Prendre le temps de se familiariser avec les différents fichiers créés par Xcode pour un projet de ce type (AppDelegate, ViewController, Storyboard, xib, ...).
Utiliser `main.storyboard` pour ajouter un bouton et deux labels dans l'application.
- (c) Récupérer l'action (`IBAction`) du bouton pour l'évènement `touchUpInside`, ainsi qu'une référence (`IBOutlet`) vers chaque label.
Au clic sur le bouton, changer le texte affiché dans les labels.
- (d) Au clic sur le bouton, afficher la date au moment du clic dans le premier label. Afficher le temps depuis le dernier clic dans le second label.
Utiliser la classe `NSDate`.
- (e) En utilisant la classe `NSDateFormatter`, afficher les information de date et heure dans un format facilement lisible par l'utilisateur.