

# Programmation Java - LI381

## TP 01 – Premiers pas

Julien Sopena

Octobre 2015

Ce premier TP a pour but la mise en œuvre des bases de la programmation Java. Ainsi, un premier exercice permettra d'étudier la chaîne de production d'un exécutable Java. Puis, au travers de la réalisation d'une liste simplement chaînée, il mettra particulièrement l'accent sur les notions d'instance et de référence.

### Exercice 1 : Premiers programmes java

#### Question 1

Écrire un programme java qui affiche le très classique : `Hello world!`.

#### Question 2

Écrire un programme java qui prend en paramètre un nom et affiche : `Hello <nom>!`.

#### Question 3

Écrire un programme java qui prend en paramètre une liste de noms et affiche une série de lignes numérotées : `[<num>] Hello <nom>!`.

### Exercice 2 : Liste chaînée

Le but de cet exercice est de réaliser l'implémentation d'une liste simplement chaînée.

#### Question 1

Écrire une classe `Element` qui contient les attributs suivants :

- `key` : un entier non modifiable.
- `next` : une référence d'un élément, qui pourra être modifiée.

Cette classe sera munie d'une méthode `toString` qui affiche l'état courant d'un élément de clé `6` de la façon suivante :

(6)

## Question 2

On veut maintenant faire une classe **List**, qui implémente une liste simplement chaînée, sans compteur d'élément et sans référence de l'élément final. Quels attributs doit-on déclarer ?

## Question 3

Ajouter à la classe **List** une méthode **isEmpty** qui retourne vrai si la liste est vide et faux dans le cas contraire.

## Question 4

Ajouter à la classe **List** une méthode **toString** qui affiche de la façon suivante l'état courant de la liste chaînée :

-> (3) -> (6) -> (4) ->

## Question 5

Ajouter à la classe **List** une méthode **addFirst** qui ajoute l'élément, passé en argument, au début de la liste chaînée.

## Question 6

Ajouter à la classe **List** une méthode **removeFirst** qui supprime le premier élément de la liste chaînée (s'il existe) et en retourne la référence (ou *null* s'il n'existe pas).

## Question 7

Ajouter à la classe **List** une méthode **addLast** qui ajoute l'élément, passé en argument, à la fin de la liste chaînée.

## Question 8

Ajouter à la classe **List** une méthode **removeLast** qui supprime le dernier élément de la liste chaînée (s'il existe) et en retourne la référence (ou *null* s'il n'existe pas).

## Question 9

Ajouter à la classe **List** une méthode **findKey** qui retourne la référence du premier élément dont la clé correspond à la clé passée en paramètre (ou *null* si un tel élément n'existe pas).

## Question 10

Pour finir, testez votre programme avec la séquence suivante :

1. Ajouter un élément à la fin de la liste ;
2. Ajouter un élément au début de la liste ;
3. Ajouter un élément à la fin de la liste ;
4. Supprimer le premier élément de la liste ;
5. Ajouter à la fin de la liste l'**élément que vous venez de supprimer** ;
6. Enfin, afficher la liste.

