université PARIS DIDEROT PARIS 7

INITIATION À LA PROGRAMMATION (IP2)

TD5 : Listes simplement chaînées

L'objectif est de manipuler une liste d'entiers de manière récursive. Considérons pour cela les deux classes suivantes. Les attributs étant privés, il faudra penser à implémenter des accesseurs.

```
class Cellule {
        private int valeur;
        private Cellule suivante;
        /* A COMPLETER ... */
}
class ListeDEntiers {
        private Cellule premier;
        /* A COMPLETER ... */
}
```

1. Nous allons tout d'abord ajouter des constructeurs à ces classes.
   (a) Écrire un constructeur pour la classe `Cellule` prenant un argument entier.
   (b) Écrire deux constructeurs pour la classe `ListeDEntiers` :
   – l'un construit la liste vide,
   – l'autre construit la liste à partir d'un tableau d'entiers passé en argument.
2. Nous allons définir des méthodes **récursives** pour manipuler les listes. Pour chaque méthode, il faut indiquer la classe dans laquelle elle est implémentée. Nous avons déjà vu une manière de simplifier la manipulation des listes en déléguant le travail principal à la classe `Cellule` ; il est fortement conseillé de procéder de la même manière ici.
   (a) Écrire une méthode `description` qui renvoie une chaîne de caractères décrivant la liste d'entiers séparés par des espaces.
   (b) Écrire une méthode `ajouter_en_i` qui prend deux arguments entiers `i` et `v` et ajoute `v` en position `i` si elle existe, sinon l'ajoute en fin.
   (c) Écrire une méthode récursive `supprimer_en_i` qui supprime le `i`-ème élément de la liste. Facultatif : Écrire ensuite une méthode `supprimer_k_premieres_occ(int k, int v)` qui supprime les `k` premières occurrences d'un entier `v`.
   (d) Écrire une méthode `egal(ListeDEntiers l)` qui teste l'égalité de deux listes simplement chaînées, à savoir `l` et `this`. Deux listes sont égales si elles contiennent les mêmes valeurs dans le même ordre.
   (e) Écrire une méthode `taille`, sans argument, qui renvoie le nombre d'éléments de la liste.
   (f) Écrire une méthode `somme`, sans argument, qui renvoie la somme des éléments de la liste.
   (g) Écrire une méthode `moyenne`, sans argument, qui renvoie la moyenne des valeurs de la liste sous la forme d'un `double`.
   (h) (Facultatif) Écrire une méthode `obtenir_sous_liste_inf_k (int k)` qui, pour la liste `this`, renvoie sa sous-liste ne contenant que les entiers strictement inférieurs à `k`. Pour cela, on choisit ici de créer une nouvelle liste qui contient les valeurs concernées (on crée de nouvelles cellules), tandis que la liste `this` n'est pas modifiée.
   (i) Donnez des exemples de test de vos constructeurs et méthodes dans un `main`.
3. (Facultatif) Écrire les méthodes précédentes sous forme itérative.