

Exercices dirigés

séance n°6

Exercice 1 : manipulation de liste

Cet exercice reprend la classe `ListeInt` vue en cours. Supposons une liste obtenue au moyen de l'expression java suivante :

```
ListeInt l=new ListeInt(1,new ListeInt(2,new ListeInt(3,new Liste(Int4,null))));
```

Question 1

Dessinez la liste `l`.

Question 2

Définissez une liste contenant les mêmes valeurs dans le même ordre en utilisant une seule fois le constructeur et en insérant les éléments au moyen de la méthode `cons`.

Question 3

Par quel code Java utilisant la liste `l` peut-on désigner l'élément 1 de la liste ?

Question 4

Par quel code Java utilisant la liste `l` peut-on désigner l'élément 2 de la liste ?

Question 5

Par quel code Java utilisant la liste `l` peut-on désigner l'élément 3 de la liste ?

Question 6

Par quel code Java utilisant la liste `l` peut-on désigner la queue de liste qui ne contient que deux éléments ?

Question 7

Par quel code java peut-on remplacer l'entier 3 par 5, dans la liste `l` ?

Exercice 2 : insertion d'élément

Le but de cet exercice est de vous familiariser avec les listes et avec la notation graphique associée. Cette notation n'est qu'une représentation abstraite de ce qui se passe en mémoire. Elle peut nous aider à comprendre ce qui se passe en machine.

Question 1

L'objectif est d'écrire une fonction qui ajoute un élément à un rang donné d'une liste. Nous avons déjà vu l'insertion en début et en fin de liste. Avec l'insertion à un rang donné, nous complétons la panoplie de méthodes permettant d'ajouter un élément à une liste.

L'algorithme consiste à avancer dans la liste jusqu'au bon endroit. Il faut ensuite insérer la nouvelle cellule. Il faut que la cellule précédente pointe sur elle et qu'elle même pointe sur l'élément suivant.

Exemple :

Supposons que la liste contient 1, 2, 3, 4 et qu'on veut insérer 5 en position 3.

- Dessinez la représentation de l'état initial. Faut-il une variable auxiliaire ? Si oui, comment doit-elle être initialisée ? Si oui toujours, ajoutez-la à votre représentation.
- Dessinez ensuite tous les états intermédiaires par lesquels il faut passer pour réaliser l'opération. Faites bien un dessin séparé pour chaque état.
- Essayez de décrire le changement d'état par une instruction Java (ou une portion d'instruction comme nous l'avons fait dans les rappels sur les listes). Cette instruction doit être une affectation du genre :
 - `l1=l2;`
 - `l=l.queue();`
 - `l1= new ListeInt(x, l2);`

Il faut une variable auxiliaire (`aux`) pour parcourir la liste sans perdre le début. De plus, on va utiliser une deuxième variable auxiliaire (`temp`) pour pointer provisoirement sur le nouveau maillon créé. Ceci n'est pas nécessaire, mais rend le cheminement plus simple à comprendre.

Question 2

Ecrivez une procédure qui fonctionne dans tous les cas simples (en vous inspirant de l'exemple). On doit retrouver les différentes affectations de l'exemple, organisées avec des structures de contrôle (boucles, if). On ne traitera pas les cas particuliers ni les problèmes qui peuvent survenir. Testez votre procédure.