

Feuille d'exercices 23

listes (suite)

Exercice 23.1 travail sur une liste

Cet exercice reprend la classe `Liste`.

Nous allons travailler sur la liste obtenue au moyen du code Java suivant :

```
ListeIter lal = new ListeIter();  
lal.ajouterAuDebut(4);  
lal.ajouterAuDebut(3);  
lal.ajouterAuDebut(2);  
lal.ajouterAuDebut(1);
```

1. **dessinez** la liste `lal`.
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 `ajouterALaFin`.
3. par quel code Java utilisant la liste `lal` peut-on désigner l'élément 1 de la liste ?
4. par quel code Java utilisant la liste `lal` peut-on désigner l'élément 2 de la liste ?
5. par quel code Java utilisant la liste `lal` peut-on désigner l'élément 3 de la liste ?
6. par quel code Java utilisant la liste `lal` désigner la portion de liste qui ne contient que deux éléments ?
7. par quel code Java peut-on remplacer l'entier 3 par 5, dans la liste ?

Exercice 23.2 - 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, nous humains, à comprendre ce qui se passe en machine.

Cela peut nous aider à raisonner pour faire des programmes corrects.

Question 23.2.1

Notre but 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. Procédons par étape.

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

Voyons d'abord un 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 ou un appel de méthode du genre :

```
- l1 = l2;  
- l=l.getSuivant();  
- l1= new ElementListe(x,l2);  
  - l1.setSuivant(l2);  
-
```

Si vous avez besoin de plus d'une instruction pour décrire le passage d'un état au suivant, c'est que vous avez oublié un état intermédiaire entre les deux.

S'il n'est pas possible de trouver une instruction qui décrive le passage d'un état à un autre, c'est que vous vous êtes permis de faire des choses impossibles à réaliser en Java.

Question 23.2.2

Arrivé ici, vous devez avoir votre séquence de dessins et en parallèle la séquence d'instructions correspondante. Si les choses ne sont pas encore bien claires, refaites la même chose pour d'autres exemples de listes, d'entier à ajouter et de rang.

Ecrivez une procédure qui fonctionne dans tous les cas simples (comme l'exemple étudié).

Dans cette première version, on traitera pas les cas particuliers ni les problèmes qui peuvent arriver. Testez votre procédure.

En gros, on doit retrouver les différentes affectations de l'exemple, organisées avec des structures de contrôle (`while`, `if`). Eventuellement, il faudra de nouvelles variables pour décrire les conditions de ces structures de contrôle.

Question 23.2.3

Nous allons maintenant parachever le travail en traitant les erreurs et les cas particuliers éventuels.

Cas particuliers :

- la procédure marche-t-elle si on veut ajouter l'élément en première position ?
 - la procédure marche-t-elle si on veut ajouter l'élément en dernière position ?
- Modifiez la procédure pour qu'elle marche dans tous ces cas.

Le problème qui peut arriver est que la liste ne comporte pas assez d'éléments pour que le rang demandé existe. Par exemple, il n'est pas possible d'insérer au dixième rang d'une liste de trois entiers.

A quel endroit de la procédure peut-on déceler le problème, et comment ?