INITIATION À LA PROGRAMMATION(IP2)

TD7 : Listes doublement chaînées

Présentation

On va considérer des listes doublement chaînées d'entiers. Pour cela, on utilise les deux classes suivantes :

```
public class Liste{
    private Cellule debut;

    public Liste(){
        debut = null;
    }
}

public class Cellule{
    private int valeur;
    private Cellule suivante;
    private Cellule precedente;

    public Cellule(int valeur, Cellule suivante, Cellule precedente){
        this.valeur = valeur;
        this.suivante = suivante;
        this.precedente = precedente;
    }
}
```

Exercices

Complétez les deux classes ci-dessous de telle manière que la classe `Liste` possède les méthodes suivantes :

1. la méthode itérative `boolean verifierStructure()` qui vérifiera que les valeurs des attributs `precedente` soient cohérentes avec une structure de liste doublement chaînée.
2. la méthode `ajouterDebut(int val)` qui ajoute une `Cellule` de valeur `val` au début de la liste. La méthode de `Liste` appellera le cas échéant une méthode `Cellule ajouterDebut(int val)` dans `Cellule`.
3. la méthode itérative `afficheAR()` qui affichera les entiers de la liste d'abord dans l'ordre puis dans l'ordre inverse. Par exemple, la liste $[10, 5, 8]$ s'affichera `[10, 5, 8: 8, 5, 10]`.
4. la méthode récursive `ajouterA(int indice, int val)` qui ajoute une cellule de valeur `val` à la position `indice`. (On rappelle que la première cellule est en position 0). Si l'indice ne correspond pas à une position correcte, on ne fait rien.
5. la méthode `supprimerDebut()` qui supprime la première cellule si elle existe.
6. la méthode récursive `supprimerA(int indice)` qui supprime la cellule à la position `indice`. Si l'indice ne correspond pas à une position correcte, on ne fait rien.

7. la méthode itérative `concatener(Liste l)` qui concatène la liste `l` à la liste `this`. Cette méthode ne fait pas de copie de listes, elle modifiera donc `this` et éventuellement `l`. Que se passe-t'il si on concatène une liste ayant au moins un élément avec elle-même ?
8. (facultatif) la méthode récursive `renverser()` qui renverse la liste chaînée.
9. (facultatif) la méthode itérative `eliminer(int val)` qui élimine toutes les cellules de valeur `val`.

Une méthode de tri (facultatif)

Écrivez les méthodes itératives suivantes

1. `boolean estTrie()` qui retourne `true` si les entiers de la liste sont en ordre croissant.
2. `void unePasse()` qui parcourt la liste et échange les couples de cellules consécutives qui ne sont pas dans le bon ordre. Exemple : Si j'applique `unePasse` à la liste $[10, 4, 5, 13, 8]$ on obtient la liste $[4, 5, 10, 8, 13]$. En effet, on a échangé d'abord les cellules de valeurs 10 et 4 à la position 0, puis celles de valeurs 10 et 5 à la position 1, on ne fait rien à la position 3 (valeurs 10 et 13) et on échange les cellules de valeurs 13 et 3 à la position 4.
3. Maintenant faites une méthode qui trie une liste en faisant des passes d'échange jusqu'à ce que la liste soit triée.
4. Comment éviter de faire à chaque fois un parcours pour les échanges et un autre pour les vérifications ?