

Examen de Programmation Objet

Durée : 0h30

Aucun document autorisé

Mobiles interdits

Nom :

Prénom :

1 Question 1 : récursivité

```
/**
 * Antécédent : l.length >= n
 * Rôle : renvoie la somme des n premiers doubles de la liste l
 * écriture récursive
 */
public static double somme_recursive(List<Double> l, int n) {
    return n == 0 ? l.get(n) : l.get(n) + somme_recursive(l, n-1);
}

/**
 * Antécédent : l.length >= n
 * Rôle : renvoie la somme des n premiers doubles de la liste l
 * écriture itérative
 */
public static double somme_itérative(List<Double> l, int n) {
    double res = 0.0;
    for(int i = 0; i < n; i++) res += l.get(i);
    return res;
}
```

2 Question 2 : pile

```
/**
 * Rôle : insère l'entier myItem dans la pile courant à sa place
 */
public void push(Integer myItem) {
    LinkedList<Integer> tmpList = new LinkedList<Integer>();
    //mettre tous les entiers plus petit que we myItem dans tmpList
    while (!this.empty() && (this.peek() < myItem)) {
        tmpList.addFirst(this.peek());
        this.pop();
    }
    //empiler myItem dans MyLinkedListInteger
    MyLinkedListInteger.addFirst(myItem);
    //empiler tous les entiers de tmpList dans MyLinkedListInteger
    while (!tmpList.isEmpty())
        MyLinkedListInteger.addFirst(tmpList.removeFirst());
}
```

3 Question 3 : liste

```
/**
 * Rôle : concatène la liste l à la liste courante
 */
void concaténer(MaListe<T> l) {
    for (T x : l) this.add(x);
}
```

Le sujet suggérait de ne pas utiliser la méthode `get`. Dans ce cas, l'écriture de la méthode `inverser` nécessite l'utilisation d'une liste auxiliaire.

```
/**
 * Rôle : inverse la liste l à la liste courante
 */
void inverser() {
    //copier la liste courante dans la liste auxiliaire
    //en ordre inverse
    MaListe<T> lAux = new MaListe<T>();
    for (T x : this) lAux.add(0,x);
    //affecter les éléments de lAux à ceux de this
    int i=0;
    for (T x : lAux) this.set(i++,x);
}
```

Bien sûr, ceux qui ont utilisé la méthode `get` n'ont pas été pénalisés.

```
/*
 * Rôle : échange les éléments de rangs i et j
 * de la liste courante
 */
void échanger(int i, int j) {
    T aux = this.get(i);
    this.set(i, this.get(j));
    this.set(j, aux);
}

/**
 * Rôle : inverse la liste l à la liste courante
 */
void inverser() {
    int i=0, j=this.size()-1;
    while (i<j)
        //échanger ième(i) et ième(j)
        échanger(i++, j--);
}
```

.....