

Contrôle de Programmation

Jeudi 17 novembre 2011

Durée : 1h

Ce sujet comporte 2 pages.

Documents autorisés : notes de cours, TD et TP, spécifications des types abstraits.

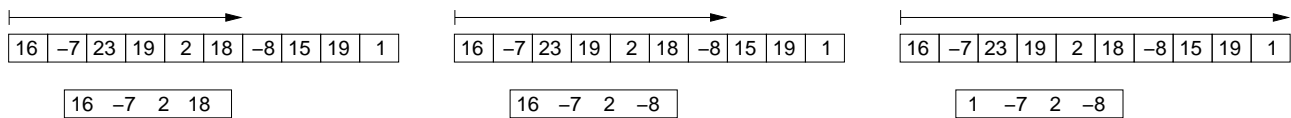
Partie 1.

```
1 public class C<T>
2 {
3     public static <T> void f()
4     {
5         File<T> fileA = new FileAnneau<T>();
6         FileAnneau<T> fileB = new FileAnneau<T>();
7         FileAnneau<T> fileC = new File<T>();
8         g(fileA);
9         g(fileB);
10        g(fileC);
11        h(fileA);
12        h(fileB);
13        h(fileC);
14    }
15    public static <T> void g(File<T> fa) { /* ... */ }
16    public static <T> void h(FileAnneau<T> fb) { /* ... */ }
17 }
```

Question 1 : On rappelle que `File<T>` désigne la spécification et `FileAnneau<T>` l'implémentation à l'aide d'un tableau géré en anneau du type abstrait `File` étudié en cours ; pour chacune des lignes 5 à 13, indiquez si l'instruction est correcte ou non en justifiant à chaque fois votre réponse.

Partie 2.

Le but de cet exercice est de chercher le k ème plus petit élément d'un tableau non trié. L'idée générale est de parcourir le tableau en mémorisant à chaque instant les k plus petits éléments, comme dans l'exemple ci-dessous avec $k = 4$.



Le k ème plus petit (avec $k = 4$) est 2.

Pour réaliser ceci, on va programmer une structure de données `MinSet` pouvant contenir au plus k éléments (entiers) et dont les opérations sont décrites dans la spécification suivante :

```
public interface MinSet {
    // initialiser un MinSet vide de capacité k
    // @pre : k > 0
    // public MinSet(int k)

    // @pre MinSet non vide
    public int getMin(); // renvoie l'élément minimum

    // @pre MinSet non vide
    public int getMax(); // renvoie l'élément maximum

    // ajouter un élément en garantissant qu'on conserve les plus petits éléments
    public void ajouter(int x);
}
```

Soit nb le nombre d'éléments mémorisés à un instant donné; la méthode `ajouter` doit garantir les propriétés suivantes :

- nb est au plus égal à k ($0 \leq nb \leq k$);
- les nb éléments mémorisés sont les nb plus petits éléments qu'on a tenté d'ajouter dans le `MinSet`.

Caractéristiques de l'implémentation :

- les éléments du `MinSet` sont placés dans un tableau *non trié* dont la capacité est définie par le constructeur;
- on doit connaître à tout instant la position de l'élément minimum et celle de l'élément maximum;
- l'ajout (réussi) d'un élément doit se faire sans décalage.

Question 2 : Programmez l'implémentation de l'interface `MinSet`.

Précision : *avant* de programmer la fonction `ajouter`, indiquez le traitement à effectuer dans les cas suivants :

- le tableau n'est pas plein;
- le tableau est plein : citez les différents cas possibles et le traitement correspondant.

Question 3 : Programmez la fonction `mink` qui prend en paramètre un tableau d'entiers et un entier k et qui rend le k ème plus petit élément du tableau.