

Partiel TO2

19 mars 2011 - Duree 2h

Exercice 1 (Questions de cours)

1. Récursivité

- (i) Rappelez les règles à appliquer lors de la conception d'un programme récursif.
- (ii) Dans le cas de la fonction de Morris, on a le code suivant :

```
static int morris(int m, int n) {  
    if (m == 0) return 1;  
    else      return morris(m - 1, morris(m, n));  
}
```

Les règles mentionnées à la question (i) sont-elles bien appliquées ? Justifiez clairement votre réponse.

2. Tris de tableaux

On dispose d'un tableau d'entiers que l'on voudrait trier.

- (i) Décrivez un algorithme de tri de votre choix et présentez son code Java.
- (ii) Quelle est la complexité (dans le pire et dans le meilleur des cas) de votre algorithme ?

Exercice 2 On considère la méthode suivante :

```
public static int toto(int x, int y) {  
    int a = 0, b = x;  
    while(b >= y) {  
        b -= y;  
        a++;  
    }  
    return a;  
}
```

- 1. Donnez une précondition **Pre** et une postcondition **Post** de la méthode toto.
- 2. Donnez un invariant **I** pour la boucle.
- 3. Prouvez que **I** est un invariant de la boucle (faire le lien avec **Pre** et **Post**).
- 4. Que fait la méthode toto ?

Exercice 3 On considère une classe Recipient avec :

- (i) un attribut privé volume de type entier (le volume maximal de liquide que peut contenir le recipient) et un attribut privé contenu de type entier (le volume contenu actuellement dans le recipient),
- (ii) un constructeur Recipient(int volume) qui construit un recipient de capacité volume, complètement vide,
- (iii) et des méthodes publiques :
 - { void vider() qui vide le contenu du recipient courant,
 - { void remplir() qui remplit complètement le recipient courant,
 - { boolean vide() qui teste si le recipient courant est vide,
 - { boolean plein() qui teste si le recipient courant est plein,
 - { void transvaser(Recipient R) qui transvase le contenu du recipient courant dans le recipient R (attention à ne pas transvaser plus de liquide que la place qu'il reste dans R).

A. Programmation des objets

1. Ecrivez la classe `Recipient`.
2. Ecrivez une classe `TestRecipient` avec une méthode `main` qui teste les méthodes de la classe `Recipient`.

B. Comparaison des volumes

Dans cette partie, on veut comparer les volumes de deux recipients `R1` et `R2`, en utilisant uniquement les méthodes de la classe `Recipient` (et pas l'attribut `volume` qui est privé!). Pour cela, on s'autorise à changer les contenus de `R1` et `R2`.

1. Comment peut-on comparer le volume des deux recipients `R1` et `R2` en utilisant uniquement les méthodes de la classe `Recipient`? Dans la classe `TestRecipient`, écrivez deux méthodes :

```
{ public static boolean plusVolumeux(Recipient R1, Recipient R2) qui teste  
  si le recipient R1 a un volume supérieur ou égal à celui de R2,  
{ public static boolean memeVolume(Recipient R1, Recipient R2) qui teste si les  
  deux recipients ont le même volume.
```
2. Donnez un exemple d'utilisation des méthodes de la question précédente dans la méthode `main`. Pourquoi a-t-on déclaré ces deux méthodes en `static`?
3. Ecrivez une méthode `public static int quotient(Recipient R1, Recipient R2)` qui renvoie le quotient du volume du recipient `R1` par celui du recipient `R2`. Souvenez-vous que pour calculer le quotient de a par b , on calcule "combien de fois il y a b dans a ".

C. Minimum et maximum simultanés dans un tableau

1. Ecrivez une méthode `minEtMax` qui prend en argument un tableau `t` de recipients, trouve les indices des recipients de volume minimum et maximum de ce tableau, et renvoie le quotient du volume maximum par le volume minimum des recipients de ce tableau. Vous pouvez utiliser les méthodes `plusVolumeux` de la question B.1 et `quotient` de la question B.3 même si vous ne les avez pas programmées.
2. Combien de comparaisons utilise votre méthode `minEtMax`? Combien de liquide gaspillez-vous si votre tableau contient n recipients de volume v , tous vides au départ?
3. Lorsque l'on appelle `plusVolumeux(R1, R2)`, on 900(a4ipcJ-333(les)-334(ca4ip9mparaiso8-359(deux