

Examen 3 de Programmation en Java

Christian Rinderknecht

4 février 2015

Durée : deux heures. Les documents et les calculatrices ne sont pas autorisés.

1 Tri à bulles

Le but de cet exercice de vous faire programmer un algorithme de tri simple, dit *tri à bulles*. Pour comprendre le principe, supposons que nous voulions trier par ordre croissant une suite d'entiers.

Le principe est de comparer deux à deux les éléments du tableaux, de la gauche vers la droite. Si les deux éléments sont ordonnés l'un par rapport à l'autre, on passe à la paire suivante, sinon on les échange avant de passer à la paire suivante. Au bout de cette étape, la dernière case du tableau (la plus à droite) contient donc le plus grand élément. Par exemple 3 5 7 2 devient 3 5 2 7. Il faut alors recommencer avec le sous-tableau allant de la gauche jusqu'avant le plus grand élément précédemment trouvé. Dans notre exemple, il faut recommencer avec le sous-tableau 3 5 2, puisque 7 est bien placé. Cela donne 3 2 5 7. Maintenant 5 et 7 sont bien placés et l'on recommence avec le sous-tableau 3 2, ce qui donne 2 3 5 7. On s'arrête là car le sous-tableau restant est réduit à 2, qui est forcément trié.

Écrivez d'abord une classe publique `Bubble` qui contient une méthode publique et statique `print` qui prend un tableau d'entier et l'affiche de gauche à droite à l'écran. Dans la fonction `main`, lisez les arguments passés en ligne de commande par l'utilisateur, stockez-les dans un tableau d'entiers et affichez-les tels qu'ils ont été lus. Par exemple, si l'utilisateur tape

```
$ java Bubble 3 5 7 2
```

alors l'affichage est

```
3 5 7 2
```

Ensuite, ajoutez à la classe `Bubble` une méthode publique et statique `sort` qui prend en paramètre un tableau d'entiers et ne renvoie rien (c'est-à-dire le type `void`).

2 Dictionnaire d'occurrences

Écrivez une classe `Dico` qui traitera un tableau `t` d'entiers et créera deux tableaux d'entiers `occ` et `elt` qui vérifient les propriétés suivantes :

- chaque élément de `t` doit apparaître une fois et une seule dans le tableau `elt`;
- pour chaque `i`, `occ[i]` contiendra le nombre d'occurrences de `elt[i]` dans le tableau `t`.

En d'autres termes, `elt` fournira un dictionnaire des valeurs apparaissant dans le tableau et `occ` indiquera combien de fois elles apparaissent. Par exemple :

```
$ java Dico 1 3 0 3 3 1 7 7
elt = 1 3 0 7
occ = 2 3 1 2
```

3 Matrices

Écrivez une classe `Poly` contenant

- une méthode `somme` qui réalise le calcul de la somme de deux vecteurs réels;
- une méthode `prod_scal` qui réalise le calcul du produit scalaire de deux vecteurs;
- une méthode `prod` qui réalise le calcul du produit d'une matrice par une autre matrice.

Pour chacune des méthodes, quelles conditions devez-vous vérifier avant de procéder au calcul ?