

Cours de programmation orientée-objet

Examen du 9 juin 2010

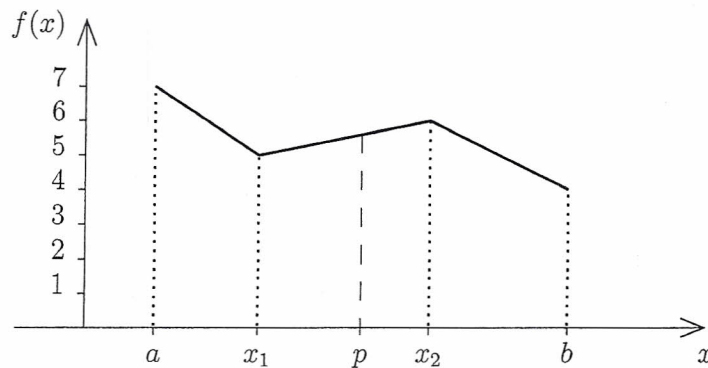
Livres fermés. Durée : 3 heures 1/2.

Veillez répondre à chaque question sur une feuille séparée sur laquelle figurent nom, prénom et section. Soyez bref et concis, mais précis.

1. On désire disposer d'une classe *Fonction* dont chaque instance représente une fonction réelle, continue et linéaire par morceaux définie dans un intervalle fermé borné.

La manipulation de cette classe se fait de la manière suivante :

- il est possible de définir la valeur que doit prendre cette fonction pour certains points du domaine. Ces points sont appelés points caractéristiques.
- l'ensemble des points caractéristiques définis pour la fonction doit toujours comprendre au moins deux éléments et inclut les deux bornes du domaine.
- entre deux points caractéristiques successifs, la fonction prend la valeur de l'interpolation linéaire des deux valeurs définies pour ceux-ci.
- il est possible de calculer la valeur d'une fonction pour n'importe quel point du domaine.



Rappel : La valeur $f(p)$, avec p compris entre les points caractéristiques successifs distincts x_1 et x_2 , est calculée par la formule suivante :

$$f(p) = \frac{(p - x_1)(f(x_2) - f(x_1))}{x_2 - x_1} + f(x_1)$$

Cette classe doit être totalement indépendante de la représentation des nombres utilisés pour représenter les valeurs de la fonction ou les éléments de son domaine. Il faudra donc veiller à mettre en place un mécanisme permettant d'utiliser indifféremment par exemple, des nombres à virgule flottante ou des nombres rationnels représentés par un quotient d'entiers. Bien que vous ne devez pas implémenter ces différentes représentations, il faut que votre solution comprenne les éléments qui permettront de les définir le plus simplement possible.

- (a) Ecrivez le code Java d'une classe ou d'un ensemble de classes répondant aux besoins décrits dans l'énoncé.

Rappel : Vous ne devez pas implémenter les différentes représentations des nombres.

- (b) Expliquez en quoi votre solution permet d'ajouter différentes représentations des nombres et décrivez la marche à suivre pour en ajouter une.

Note : Les situations d'erreur doivent déclencher des exceptions appropriées.

2. (a) En Java, quand utilise-t-on le mot clé `throws` et à quoi sert-il ?
(b) Quelles différences existe-t-il entre la déclaration d'un élément avec l'attribut de visibilité `protected` et la déclaration de ce même élément avec l'attribut `private` ?
(c) Une classe peut-elle accéder à un élément d'une autre instance de la même classe déclaré `private` ?
(d) Dans le contexte de la programmation parallèle, que signifie le terme *opération atomique* ?

3. Voici un extrait d'un programme Java :

```
String d = new String("abcd");  
String s = new String("abcd");  
String c = d;  
String n = c.clone();
```

Donnez, pour chacune des expressions suivantes, la valeur renvoyée par leur évaluation et expliquez soigneusement ces résultats :

- (a) `d == s`
(b) `d == c`
(c) `c == n`
(d) `d.equals(s)`
(e) `s.equals(n)`
(f) `d.equals(null)`

4. Voici un extrait d'un programme Java :

```
public class Stock
{
    private static Type types[];
    private int      size;
    private Element[] storage[];

    public Stock()
    {
        ...
    }

    public void static loadTypes()
    {
        ...
    }

    public Element getElement(int t) throws EmptyStockException
    {
        ...
    }

    public void storeElement(Element e)
    {
        ...
    }
}
```

Cette classe sert à gérer un stock d'éléments distincts. Une instance d'un élément **ne peut pas** être contenue simultanément dans des stocks différents.

- (a) Adaptez cette classe afin qu'elle propose un mécanisme de clonage Java compatible avec les restrictions décrites.
- (b) S'agit-il d'un clonage dit *en profondeur* ? Justifiez.

Note : Vous pouvez considérer que la classe `Element` dispose déjà d'un mécanisme de clonage parfaitement fonctionnel.