

Réponses aux questions

J. Odersky

C. Vázquez

9 mars 2011

Question P1.1

Un vecteur est représenté par la classe `Vector3D`. Cette classe comprend trois champs privés du type double : `x`, `y` et `z` correspondant aux composantes du vecteur. Ces champs peuvent être accédés respectivement par les méthodes publiques `getX()`, `getY()` et `getZ()`. Les méthodes “opérateurs” sur les vecteurs (par exemple l’addition, la norme, etc...) sont toutes publiques.

Ces méthodes s’appuient entre-elles, par exemple la méthode ‘opposée’ retourne le vecteur multiplié par moins un et la méthode ‘soustraction’ additionne l’opposé. Ces appels consécutifs diminuent la performance d’une façon minimale mais évitent un duplicage de code considérable.

Un vecteur est complètement invariable. C’est-à-dire qu’une fois un vecteur initialisé, on ne peut plus changer ses composantes (pas de méthodes ‘set’). De même, tous les opérations internes du sens mathématique (qui renvoient un vecteur), renvoient une nouvelle instance d’un vecteur. En aucun cas l’instance d’un vecteur n’est modifiée ! Ceci facilite énormément le raisonnement sur toute variable de type vecteur. De plus, l’invariance d’un vecteur paraît naturelle, comme celle d’un nombre réel.

Quelques vecteurs remarquables sont définis comme variables statiques constantes. Parmi ceux-ci notamment le vecteur nulle (`Null`) et les vecteurs unitaires `i`, `j`, `k` selon respectivement les axes `x`, `y` et `z`.

Question P3.1

Nous n’avons pas rajouté un constructeur de copie. Comme la classe `Vector3D` est invariable et ne contient pas de pointeurs ou références sur d’autres objets mutables, elle n’a pas d’état et donc l’utilisation du constructeur de copie par défaut suffit.

Question P3.2

Question P3.3

a)

En ajoutant un constructeur par coordonnées sphériques, les attributs de la classe ne devraient pas forcément être changés. Il serait toute à fait envisageable, de garder les coordonnées carthésiennes comme attributs et d’assigner leurs valeurs équivalentes en coordonnées sphériques par le nouveau constructeur.

b)

La surcharge serait une difficulté majeure pour créer un constructeur par coordonnées sphériques. Etant donné qu’un tel constructeur prendrait comme paramètres deux angles et une longueur, représentés par trois doubles, il serait en conflit avec le constructeur de coordonnées carthésiennes. Il serait alors impossible d’avoir les deux constructeurs dans une classe.

Néanmoins, une solution alternative serait d’implémenter une méthode statique qui prendrait comme paramètres des coordonnées sphériques et qui renverrait un vecteur ayant des coordonnées carthésiennes équivalentes (par exemple `Vector3D Vector3D::fromSpherical(double phi, double theta, double r)`).

Question P3.4

La méthode 'affiche()' d'un vecteur a été implémenté sous forme de l'opérateur '<<' de 'std : ostream'. La méthode 'compare' est équivalent à l'opérateur '='.