Interfaces des Programmes d'Applications, TD1 : Héritage et Délégation

19 septembre 2005

1 Eclipse

Pour l'ensemble des TP, Eclipse est l'environnement de développement préconisé. Ainsi, nous vous invitons dès à présent à vous référer à la documentation sur le site officiel : www.eclipse.org mais également aux nombreux tutoriaux disponibles sur Internet pour prendre en main ce nouvel outil.

- **Initialisation du système** :
  Pour configurer correctement votre environnement pour ce premier TP, vous devez :
  - Lancer *eclipse*,
  - Indiquer comme *workspace* le répertoire dans lequel vous travaillerez,
  - Cliquer sur workbench,
  - Changer de version courante de jdk dans windows/preferences/Java/Installed JREs –> /opt/local/stow/jdk-1.5.0/jdk1-5
  - Créer un projet *distance*
  - puis un package *dist*

2 Premier programme Java

Pour votre première approche du langage Java, nous vous donnons la classe *Point2D* qui caractérise les coordonnées d'un point dans un plan en deux dimensions.

- Télécharger le fichier *Point2D.java* dans le répertoire *src* de votre nouveau projet *distance*.
- Après l'avoir parcouru, implémenter un jeu de test *TestPoint2D.java* qui, à partir d'un tableau de 4 points, affiche la distance entre chaque couple de points.

À présent, on souhaite exprimer la position d'un point non plus dans un plan 2D mais dans un plan 3D. Pour la caractériser, il faut ajouter un attribut concernant la profondeur du point aux coordonnées planaires.

3 Héritage de classe

Un premier moyen est d'étendre la classe *Point2D* à une classe *Point3D* qui introduit la troisième coordonnée. On procède par héritage de la classe *Point2D*.

3.1 Première version

Une première version vous est donnée dans le fichier *Point3D.java*.

- Télécharger le fichier.
- Dans un programme de test, commencer par identifier toutes les déclarations et initialisations de points 2D et de points 3D possibles (ie toutes les déclarations d'instance de *Point2D* et de *Point3D*).
- Pour chaque combinaison de ces instances, appliquer la méthode déterminant leur éloignement. Qu'observez-vous ? Ajouter si nécessaire des traces dans les méthodes *distance* pour identifier laquelle est employée.
- Certaines combinaisons ne donnent pas un résultat correct. Lesquelles et pourquoi ?

3.2 Seconde version

Pour lever l'ambiguité quant au choix de la méthode employée au sein de la classe *Point3D*, nous décidons de ne pas accepter de surcharge de la méthode *distance*.

- À partir de l'API Java (http ://java.sun.com/j2se/1.5.0/docs/api/), déterminer le rôle et le comportement de l'opérateur *instanceof*.
- Modifier votre classe *Point3D*.
- Rejouer le même jeu de test que pour la version précédente. Que se passe-t-il à présent ?

4 Délégation

On peut aussi utiliser la délégation.

- Télécharger le fichier *Point3DDeleg.java*.
- Implémenter toutes les méthodes données.
- Ecrire un programme de test.
- Quelles sont les différences avec la version par héritage ? Avantages, inconvénients ?