

# Interfaces des Programmes d'Applications, TD1 : Classes, Interfaces et Classes Abstraites

15 septembre 2006

## 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](http://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 *FormesGeom*
- puis un package *formes*

## 2 Héritage simple

On souhaite écrire un programme pour manipuler des formes géométriques. On doit disposer de méthodes donnant leur périmètre, leur surface, leur centre

de gravité. On doit pouvoir leur appliquer différentes transformations géométriques :

- une translation de vecteur donné (par ses deux coordonnées),
- une homothétie de rapport donné (un réel  $k$ ) et de centre le point  $C$ , centre de gravité de la forme (on rappelle qu'alors pour tout point  $P$  de la forme, son image  $P'$  par l'homothétie vérifie  $\overrightarrow{CP'} = k\overrightarrow{CP}$ ),
- une homothétie de rapport et de centre un point quelconque donné.

On doit disposer également d'une méthode *placer* positionnant la forme (c'est-à-dire son centre de gravité) en un point donné.

1. Chercher dans l'API (<http://java.sun.com/j2se/1.5.0/docs/api/>), les classes relatives aux points. Comprendre et justifier l'arborescence de ces classes.
2. Ecrire une interface *Forme* correspondant aux spécifications ci-dessus.
3. On considèrera en premier lieu deux types de formes géométriques : les cercles et les carrés. Elles seront caractérisées par un point et un réel, représentant le rayon ou le côté.  
Ecrire les classes *Cercle* et *Carre* implémentant toutes les fonctionnalités des formes géométriques indiquées dans l'interface *Forme*.
4. Implémenter un jeu de test *TestFormes.java* qui crée des cercles et des carrés et teste leurs fonctionnalités.
5. Rappeler le concept de classe abstraite. Comparer le code de vos classes *Cercle* et *Carre*. Quel défaut présente-t-il ? Proposer une solution basée sur l'héritage simple, pour y remédier.
6. L'interface est-elle toujours nécessaire ? Justifier votre réponse.
7. Ecrire une classe *Afficheur*, qui contient une méthode d'affichage pour les cercles et pour les carrés. Critiquer en terme d'extensibilité. Proposer une solution.

### 3 Délégation

Rappeler le principe de la délégation.

- Créer un nouveau projet *FormeDeleg* et écrire une autre version du programme précédent utilisant la délégation.
- Quelles sont les différences avec la version par héritage ? Avantages, inconvénients ?