

TD Concepts objet et Java

2016-17

Exercice 5

On souhaite mettre en place un éditeur simple de dessins à base de formes géométriques, sous forme d'un package `et3.java.geometrie` regroupant les classes permettant de manipuler des formes géométriques bidimensionnelles simples, ici : des cercles, des rectangles, et des carrés. On aura les contraintes suivantes :

- Chaque forme possède un centre de gravité (instance de `java.awt.Point`) ainsi qu'une couleur (instance de `java.awt.Color`). Les attributs correspondants ne devront pas être directement partagés avec l'extérieur de la classe. De plus :
 - un cercle possède un rayon ;
 - un rectangle possède une largeur et une hauteur.
- Toute forme géométrique doit pouvoir avoir les comportements suivants :
 - *translation*, prenant en paramètres deux nombres représentant un déplacement horizontal et vertical
 - *homothétie*, prenant en paramètre un nombre représentant le ratio de l'homothétie
 - toute forme est capable de donner sa représentation sous la forme d'une chaîne de caractères donnant le nom de la forme (classe) et la description textuelle de chacun de ses attributs. Par exemple, la chaîne de caractères produite pour un cercle pourrait être :

```
[Cercle
 [centre de gravité : x=10 , y=4]
 [rayon : 20]
 [couleur : r=82 , g=255 , b=0]
]
```

- Des constructeurs utiles devront être proposés.
1. Implémentez la spécification demandée.
 2. On souhaite ajouter une forme de carré comme une spécialisation de la classe pour un rectangle. On veillera à ce que cette classe ne puisse plus être dérivée.
 3. On souhaite à présent pouvoir construire des listes de formes géométriques pour opérer des regroupements d'objets.
 - (a) Commencez par créer une classe de test définissant un programme créant une collection sous forme de liste utilisant la classe `java.util.ArrayList<E>`. Ajoutez à cette collection un objet de chaque type de forme définie, et affichez le contenu de la liste à l'aide d'un *itérateur*. Quelles sont les méthodes `toString` qui sont appelées pour chaque objet ? Que se passe-t-il si l'on enlève un élément de la collection à un indice qui n'existe pas ?

- (b) Créez à présent une classe pour représenter une simple collection pour des formes géométriques uniquement par réutilisation de la classe `java.util.ArrayList<E>`.
- (c) Ajoutez à la classe précédente des méthodes `translation` et `homothetie` qui déplacent ou modifient la taille de l'ensemble des objets d'une telle collection de formes géométriques, et redéfinissez la méthode `toString` de manière à ce qu'elle produise un affichage plus approprié pour la collection.
- (d) Testez l'ensemble des méthodes définies.
- (e) On souhaite à présent définir une collection qui permet d'imposer que ses membres soient des formes géométriques de même sous-type. Définissez une nouvelle classe sur le modèle de la précédente, avec notamment la définition de méthodes pour l'homothétie et la translation, ainsi que la redéfinition de la méthode `toString`. Confirmez à l'aide de tests qu'il n'est pas possible de mélanger des formes géométriques de types différents et que cela est bien détecté à la compilation.