TP6 : Classe abstraite, interface et exception

Enseignants : Madhi Kamel, Majdoub Manel

Groupes : Prepa A02
Objectifs :

- introduire la notion de classe abstraite
- utiliser la notion d'interface
- gérer les exceptions

Exercice 1

On vous demande d'écrire :

1. une interface nommée Homme possédant une seule méthode : Void identite() dont le rôle est d'afficher les informations concernant un homme
2. une classe Personne implémentant cette interface et possédant deux attributs privés nom et prenom de type String et un constructeur paramétré avec le nom et le prénom d'une personne.
3. une classe Client héritant de la classe Personne et implémentant l'interface Homme avec un attribut privé supplémentaire nommé numero de type entier et un constructeur paramétré en conséquence.
4. une classe Peuple contenant deux attributs privés : un tableau d'Hommes nommé pays de capacité 100 et un entier nbHommes représentant le nombre d'hommes dans pays. Cette classe contient également deux méthodes :
   - void naissance (Homme h) : permet d'ajouter un homme dans pays.
   - void affiche () : affiche l'identité de chaque homme dans pays.

   Testez les méthodes naissance () et affiche () après avoir placé dans pays des personnes et des clients.

Exercice2

1. Il s'agit de définir une classe nommée Forme comportant deux méthodes abstraites : float perimetre()et float surface (), et une méthode non abstraite void contenantCarre(float surf) qui affiche le message suivant « Cette forme peut contenir un carré de surface surf » si la forme a une surface au moins égale à celle d'un carré dont la surface est donnée par le paramètre surf et qui affiche sinon « Cette forme ne peut pas contenir un carré de surface surf ».
2. Ecrire les deux classes Rectangle et Cercle héritant de cette classe Forme. La classe Rectangle a deux attributs privés : longueur et largeur. La classe Cercle possède un attribut privé nommé rayon.
3. Tester la méthode contenantCarre avec surf = 20 en l'appliquant pour un rectangle de longueur=2 et de largeur=10, puis pour le cercle de rayon=1.

Exercice3

Cet exercice vise à permettre la création de plusieurs formes tels que les points, les cercles et les cylindres. Pour cela, on considère qu'un cercle étend la notion de point pour caractériser son rayon. De même, un cylindre étend la notion de cercle, pour caractériser sa hauteur.

1. Écrire l'interface Forme, qui comporte les 3 méthodes suivantes :
   — double calcul_aire() : cette méthode permet le calcul de l'aire de la forme
   — double calcul_volume() :cette méthode permet le calcul du volume de la forme
   — void affiche_nom() : cette méthode permet l'affichage du nom de la forme (point, cercle et cylindre).
2. Écrire la classe Point qui implémente l'interface Forme. La classe Point est caractérisée par 2 coordonnées réelles (double) : x,y. La classe Point comporte également une méthode affiche(), permettant l'affichage des coordonnées du point sous forme [x,y].
3. Écrire la classe Cercle qui hérite de la classe Point, elle comporte également une valeur réelle(double) pour son rayon.

   Ecrire 2 constructeurs pour cette classe : le premier va créer un cercle dont le centre est le point (0,0) et de rayon 0 le second comporte des paramètres pour les coordonnées du centre et le rayon De même que pour le point, écrire une méthode affiche(),affichant toutes les informations sur le cercle (coordonnées du centre, valeur du rayon).

   La classe Cercle comporte également une méthode pour le calcul de l'aire du cercle : double calcul_aire().

   On rappelle que l'aire du cercle est égal à 3.14 * rayon * rayon.
4. Écrire la classe Cylindre qui étend la classe Cercle et comporte une valeur réelle (double) hauteur. Implémenter les méthodes calcul_aire()et calcul_volume()pour la classe Cylindre.

   On rappelle que :

   aire (cylindre) = 2*aire (cercle) + 2 * 3.14159 * rayon * hauteur

   volume (cylindre) = aire (cercle) * hauteur.

   Implémenter La méthode affiche()qui permet d'obtenir toutes les informations sur le cylindre : (coordonnées du centre du cercle, rayon du cercle, hauteur).
5. Tester l'implémentation en écrivant une classe FormeTest.
   — Créer les 3 objets suivants :
   un point avec x=5 et y=5
   un cercle avec un rayon de 8 et des coordonnées x=10,y=15 pour centre
   un cylindre avec une hauteur de 20, un rayon de 12 et des coordonnées x=30,y=30.
   — Créer un ArrayList comportant les 3 éléments suivants : un point avec x=5 et y=5
   un cercle avec un rayon de 8 et des coordonnées x=10,y=15 pour centre
   un cylindre avec une hauteur de 20, un rayon de 12 et des coordonnées x=30,y=30.
   — Afficher les coordonnées, les aires, les volumes et le nom de ces 3 formes. On rappelle que la classe ArrayList comporte la méthode get(int index) qui retourne l'élément à la position spécifiée (index) dans la liste.
6. La création d'un cercle doit obéir à ce critère suivant : le rayon du cercle doit être un nombre positif. Pour gérer cette situation, on vous demande de : écrire la classe RayonException.

   Identifier les méthodes concernées par cette exception. Récrire uniquement ces méthodes pour qu'elles puissent propager et lever l'exception RayonException.

   Apporter les modifications nécessaires à la classe FormeTest.