

TP06

Matière : ATELIER DE PROGRAMMATION OBJET



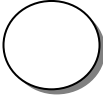
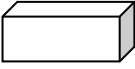

Classes : SEM2

Les objectifs de ce TP sont :

- d'appliquer le concept d'héritage de la programmation objet pour modéliser cinq classes,
- d'utiliser sous eclipse, la génération d'une méthode à redéfinir (Source-Override/Implement Methods...) et de la méthode toString() (Source-Generate toString()...),
- Maîtriser les utilisations du mot clés super,
- Redéfinir les méthodes toString() et equals(...) de la classe Object,
- Utiliser l'opérateur instanceof.

Problème

Le tableau suivant décrit les cinq classes à implémenter :

Classe	Classe mère	Attributs	Méthodes
Solide 	Object	masse (en gramme) matière (Bois, Fer, Acier)	Solide(...) afficher() toString()
Cylindre 	Solide	rayon (en cm) hauteur (en cm)	Cylindre(...) getSurface() getVolume() afficher() toString() equals(...)
Sphère 	Solide	rayon (en cm)	Sphère(...) getSurface() getVolume() afficher() toString() equals(...)
Parallelepipede 	Solide	hauteur (en cm) longueur (en cm) largeur (en cm)	Parallelepipede (...) getSurface() getVolume() afficher() toString() equals(...)
Cube 	Parallelepipede	(hauteur=longueur=largeur)	Cube(...) afficher() toString() equals(...)

La description des méthodes des cinq classes est donnée dans le tableau suivant :

Objet	Description des méthodes
Solide	<p>Solide(...) : prend en paramètre tous les attributs de la classe</p> <p>afficher() : affiche « Je suis un Solide »</p> <p>toString() : retourne « Solide[masse g, matière] »</p> <p>getMasse() getMatiere()</p>
Cylindre	<p>Cylindre(...) : prend en paramètre tous les attributs de la classe</p> <p>$getSurface() = 2 * \pi * rayon * (hauteur + rayon)$</p> <p>$getVolume() = \pi * rayon^2 * hauteur$</p> <p>$getMasseVolumique() = masse / getVolume()$</p> <p>afficher() : affiche « Je suis un Solide » « Je suis un Cylindre »</p> <p>toString() : retourne « Cylindre[masse g, matière, volume cm³, masseVolumique g/cm³] »</p> <p>equals(...) : deux cylindres sont égaux s'ils ont la même matière, le même rayon et la même hauteur</p>
Sphère	<p>Sphere(...) : prend en paramètre tous les attributs de la classe</p> <p>$getSurface() = 4 * \pi * rayon^2$</p> <p>$getVolume() = \pi * rayon^3 * 4/3$</p> <p>$getMasseVolumique() = masse / getVolume()$</p> <p>afficher(): affiche « Je suis un Solide » « Je suis une Sphère »</p> <p>toString() : retourne « Sphère[masse g, matière, volume cm³, masseVolumique g/cm³] »</p> <p>equals(...) : deux sphères sont égaux s'ils ont la même matière et le même rayon</p>
Parallelepiped	<p>Parallelepiped (...) : prend en paramètre tous les attributs de la classe</p> <p>$getSurface() = 2 * ((longueur + largeur) * hauteur + longueur * largeur)$</p> <p>$getVolume() = longueur * largeur * hauteur$</p> <p>$getMasseVolumique() = masse / getVolume()$</p> <p>afficher(): affiche « Je suis un Solide » « Je suis un Parallélépipède »</p> <p>toString() : retourne « Parallélépipède[masse g, matière, volume cm³, masseVolumique g/cm³] »</p> <p>equals(...) : deux parallélépipèdes sont égaux s'ils ont la même matière, la même longueur, la même largeur et la même hauteur</p>
Cube	<p>Cube(...) : prend en paramètre le coté du cube</p> <p>$getSurface() = 6 * longueur^2$</p> <p>$getVolume() = longueur^3$</p> <p>afficher() : affiche « Je suis un Solide » « Je suis un Parallélépipède » « Je suis un Cube »</p> <p>toString() : retourne « Cube[masse g, matière, volume cm³, masseVolumique g/cm³] »</p> <p>equals(...) : deux cubes sont égaux s'ils ont la même matière et la même longueur</p>

- 1- donner l'implémentation des classes **Solide**, **Cylindre**, **Sphere**, **Parallelepiped** et la classe **Cube**.
- 2- donner une classe TestSolide qui permet les différentes méthodes des cinq classes.
- 3- Utiliser l'opérateur instanceof dans la classe TestSolide.
- 4- Ajouter les attributs et les méthodes nécessaires pour compter et afficher le nombre de solides, de cylindres, de sphères, de parallélépipèdes et de cubes instanciés.

