

TD 5 Plus sur l'héritage

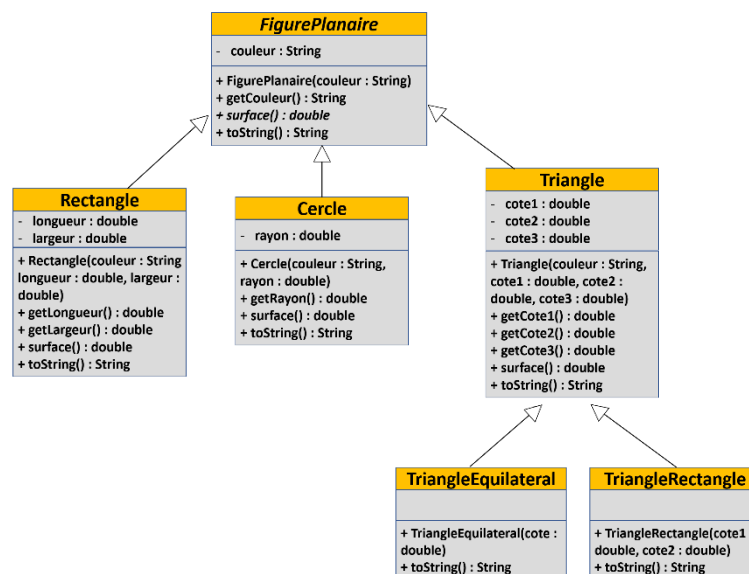
Dans ce TD nous allons approfondir un peu plus l'idée de l'héritage et nous allons utiliser la notion de classe abstraite. J'inclus à la fin de ce TD un rappel des éléments de géométrie nécessaires (les formules qui donnent les surfaces de différentes figures et la formule pour le volume d'un prisme).

1. Nous allons considérer des diverses figures planes (par exemple : un cercle, un rectangle, un carré, un triangle, etc.). Ces figures ont des méthodes communes : par exemple des méthodes `get`. Nous allons vouloir calculer la surface de chaque figure, mais en fonction de figure, la surface est calculée selon une formule différente.

La classe *FigurePlanaire* sera donc une classe abstraite (le texte en italique dénote le caractère abstrait d'une méthode ou d'une classe sur les diagrammes de classe) qui aura un attribut de type `String`, quelques méthodes concrètes (le constructeur, une méthode `get` pour l'attribut et une méthode `toString()` qui affiche `Figure plane, couleur <couleur>`.) et une méthode abstraite (la méthode *surface()*).

A partir de cette classe nous aurons trois classes concrètes représentant trois figures planes différentes : un cercle, un rectangle et un triangle. Ces figures auront des attributs différents et des méthodes implémentant la méthode *surface()*, ainsi qu'un constructeur (pour chaque classe) et une méthode `toString()` différente (qui affiche `<nom de la figure>`, `couleur <couleur>`, `<nom des attributs>` `<autres attributs>` -- par exemple `cercle, couleur <couleur>`, `rayon <rayon>`).

Consultez le diagramme de classe ci-dessous.



- Ecrivez la classe *FigurePlanaire* .

- Pour les sous-classes Rectangle, Cercle, Triangle, écrivez leurs constructeurs, ainsi que les méthodes surface() et toString(). Vous pouvez utiliser la commande Math.PI pour avoir la valeur de π , tandis que pour calculer la racine carrée d'une variable x de type double on utilise Math.sqrt(x)
 - Pour les classes TriangleEquilateral et TriangleRectangle écrivez leurs constructeurs
 - Dans une autre classe MonApplication écrivez une méthode main, dans laquelle vous calculez et affichez les surfaces de :
 - Un cercle rouge de rayon 10
 - Un rectangle bleu 5 par 10
 - Un carré noir de côté 10
 - Un triangle vert avec côtés 3, 5 et 7
 - Un triangle équilatéral marron de côté 3
 - Un triangle rectangle jaune de côtés 5 et 10.
2. Nous allons maintenant étendre notre programme. Nous voulons calculer les modules des prismes avec la base ayant la forme d'une figure plane donnée. La classe Prisme aura deux attributs :

Un attribut base de type *FigurePlanaire*

Un attribut hauteur de type double

Elle aura les méthodes suivantes :

Un constructeur avec la signature Prisme(FigurePlanaire, double)

Des méthodes FigurePlanaire getBase() et double getHauteur()

Une méthode double volume() qui calcule le volume d'une prisme

Une méthode String toString() qui retourne "Prisme, couleur <couleur>, hauteur <hauteur>, base <base>".

- Faites le diagramme de classe de la classe Prisme
- Comment est-elle liée à la classe FigurePlanaire ?
- Ecrivez le constructeur et les méthodes volume() et toString() de la classe Prisme
- Dans la classe MonApplication, affichez pour chacune des figures déjà créées, le volume d'un prisme avec ces figures en tant que base, et une hauteur de 10.

Rappel Math :

Cercle : surface $\pi \cdot r^2$

Rectangle : surface *longueur* · *largeur*

Triangle : calculez $s = \frac{cote1 + cote2 + cote3}{2}$, et la surface $\sqrt{s(s - cote1)(s - cote2)(s - cote3)}$

Volume prisme : *surface_{base}* · *hauteur*