

TD1 : MODELISATION DE FORMES GEOMETRIQUES SIMPLES

A. BUT DU TD

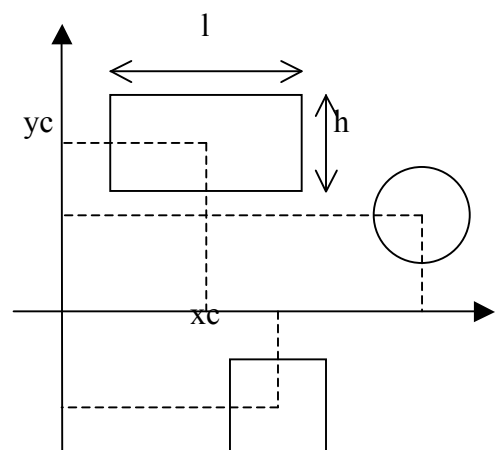
Nous abordons dans ce TD deux concepts de base de la Programmation Objet, l'encapsulation et l'héritage, qui ont été introduits lors du premier cours.

B. INTRODUCTION

Lors de ce TD, nous cherchons à représenter des formes géométriques simples et à montrer l'intérêt de l'héritage pour factoriser les propriétés communes.

Nous nous limitons à un espace à 2 dimensions avec un repère orthonormé. Dans cet espace, nous choisissons de représenter les formes géométriques simples suivantes :

- Les cercles représentés par leur centre $C(x_c, y_c)$, leur couleur (c), et leur diamètre (d)
- Les rectangles représentés par leur centre $C(x_c, y_c)$, leur couleur (c), leur largeur selon l'axe des x (l) et leur hauteur selon l'axe des y (h)
- Les carrés représentés par leur centre $C(x_c, y_c)$, leur couleur (c) et leur largeur (l)



C. MODELISATION AVEC UML

Représenter les 3 classes avec les diagrammes de classes UML. Dans un premier temps, on ne considère que les attributs que l'on déclarera privés. Il est recommandé de commencer les noms des classes par une majuscule et les attributs par une minuscule.

D. HERITAGE

Introduire l'héritage comme une opération de généralisation permettant de regrouper les 3 attributs x_c , y_c et c communs aux 3 objets dans une nouvelle classe « Forme ». De la même manière montrer que Carre est une spécialisation de Rectangle (où $h=l$). Faire le diagramme UML correspondant (avec les attributs seulement).

E. METHODES

Déterminer les méthodes. Prévoir :

- Des méthodes d'initialisation des attributs, permettant de définir les paramètres des formes
- Des méthodes de lecture des données
- Des méthodes de calcul comme « perimetre » et « surface »

Compléter le diagramme UML du §D avec les méthodes.

F. Python

Ecrire en Python les classes abordées précédemment. Prévoir 2 fichiers :

- Un fichier « formes.py » comportant la définition des classes
- Un fichier « test_formes.py » contenant le programme principal dont le rôle sera de créer 1 rectangle, 1 cercle et 1 carré (attention : pas d'entrée interactive !).

Pour chaque objet, faire appel aux différentes méthodes de modification des attributs et afficher :

- Ses paramètres (centre, couleur, dimension)
- Son périmètre
- Sa surface

G. Pour aller plus loin... (partie optionnelle)

On considère dans cette partie une classe "Dessin" comme une agrégation d'objets de la classe Forme :

- Un objet de la classe Dessin « contient » 0, 1 ou plusieurs objets de la classe Forme.
- Un objet de la classe Forme n'est inclus que dans 1 seul objet de la classe Dessin.

Cette classe Dessin devra comporter des méthodes permettant d'ajouter des formes, d'en supprimer et d'afficher les propriétés des objets qu'elle contient.

Ecrire le diagramme de classe correspondant, puis écrire le code Python associé.