### 

### 8PRO128 Programmation orientée objet

###### Automne 2019

# Travail pratique #2 (10% de la session)

##### Date de remise

**À remettre d’ici le 24 octobre 2019 (jeudi)**

(Peut se faire seul ou en équipe de 2)

**Objectif général**

Ce travail a pour objectif d’introduire l’étudiant à la notion héritage en programmation orientée objet et sa traduction en langage C++.

**Notions du langage C++ à acquérir**

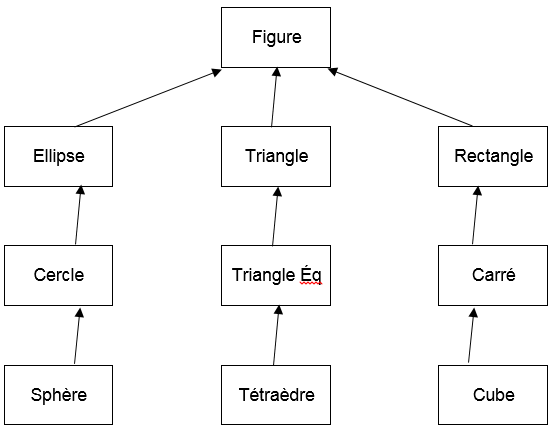
Déclaration et définition d’héritage, droits d’accès aux membres, constructeurs, destructeurs, compilation en fichiers séparés, …

**INSTRUCTIONS:**

* Votre travail pratique sera remis sous forme de dossier compressé. Ce dossier pourrait ne contenir qu’un document Word (le rapport décrit plus bas) et un dossier de projets (ou une solution contenant des projets) Visual Studio.
* Pour chacune des classes, vous aurez à créer un fichier de déclaration (.h) et un fichier de définition (.cpp). Le nom de chacun de ces fichiers doit être significatif de son contenu. Exemple : « Point.cpp », « Point.h » ou autre. Mettre des commentaires dans votre programme (fichier, but, auteur, date ou autre). Décrire les variables au besoin. Vous remettrez également une copie d’écran montrant l’exécution de votre programme.
* Remettre un diagramme de classes*.*

### Partie 1

À partir des classes que vous avez déjà créées, écrivez une hiérarchie d’héritage pour obtenir les classes du diagramme suivant. Écrivez un programme pilote qui instancie et affiche les objets de chacune de ces classes. Déclarez vos figures avec des pointeurs.



Voici un tableau de formules qui pourraient vous être utiles :

|  |  |  |  |
| --- | --- | --- | --- |
| Figure | Attributs | Comportement | formules |
| Carré | côté | aire()  périmètre()  diagonale()  afficher() | côté2  côté \* 4 |
| Cercle | rayon | aire()  périmètre()  diamètre()  afficher() | πr2  2πr  2r |
| TriangleEq | côté | aire()  périmètre()  hauteur()  afficher() | côté2 \* / 4  côté \* 3  côté \* / 2 |
| Rectangle | largeur  longueur | aire()  périmètre()  diagonale()  afficher() | largeur \* longueur  (largeur + longueur) \* 2 |
| Triangle | côté1  côté2  côté3 | aire()  périmètre()  hauteur(côtéi)  afficher() | côté1 + côté2 + côté3  2 \* aire / côtéi |

où p = demi périmètre (voir formule de Héron)

Voici d’autres formules :

|  |  |  |  |
| --- | --- | --- | --- |
| Figure | Attributs | Comportement | formules |
| Tétraèdre | base  (triangle) | aire()  face()  volume()  hauteur()  afficher() | \* côté2  côté2 \* / 4 (aire du triangle)  côté3 \* / 12  / 3 \* côté |
| Cube | base  (carré) | aire()  face()  volume()  afficher() | 6 \* aire de la base  aire de la base  côté3 |
| Sphère | grand cercle (cercle) | diamètre()  aire()  volume()  afficher() | 2 \* rayon du cercle  4 \* aire du cercle  4 \* π \* rayon3 / 3 |

### Partie 2

Modifiez votre classe de test pour que les données permettant d’instancier un objet soient obtenues de la lecture d’un fichier. Exemple de fichiers :

carre.txt rectangle.txt

4

0 0

-1 1

2 4

3 3

|  |
| --- |
| 3  1  2  5 |

La première ligne représente le nombre d’objets à instancier. Chaque ligne suivante représente les valeurs des attributs (***côté*** pour le carré; ***longueur*** et ***largeur*** pour le rectangle).

Testez plusieurs figures.

Vos résultats obtenus seront écrits dans un fichier. Exemple d’un fichier :

testCarre.txt

|  |
| --- |
| no côté aire diagonale périmètre  1 1 1 1.4142 4  2 2 4 2.8284 8  3 5 25 7.071 20  Surface occupée : 30.0 |

**Bon travail!**