


TP18 – Héritage

Tracé de formes géométriques

 Une version interactive de cet énoncé est disponible dans le fichier **VEtudiant.zip**. Décompressez le fichier et naviguez vers le fichier **doc/html/index.html**. Ce fichier est également le point d'entrée vers l'aide en ligne du code déjà réalisé.

Exercice 0 Présentation

0.1 Objectifs du programme

Il s'agit de réaliser un programme simpliste de dessin par ordinateur. Le programme est capable d'afficher des rectangles et des ellipses. Ces formes peuvent être affichées avec différents styles, couleurs et épaisseurs de trait. Elles sont toutes les deux gérées par deux points appelés origine et arrivée. Le point d'origine est le premier point cliqué par l'utilisateur pour créer la forme, le point d'arrivée est la position à laquelle l'utilisateur a relâché la souris. La forme est alors tracée dans la boîte englobante formée par ces deux points.

Pour choisir la forme à créer, l'utilisateur peut appuyer sur la touche 'R' pour un rectangle, sur la touche 'E' pour une ellipse ou sur la touche 'A' pour un arc elliptique. La touche 'T' permet de choisir les attributs de tracé pour la prochaine forme créée.

0.2 Présentation de la structure du programme

0.2.1 La classe CTP (fichiers TP.h et TP.cpp)

Le programme s'articule autour d'un objet de type **CTP** dont les méthodes sont appelées par le programme principal en fonction des actions de l'utilisateur.

0.2.2 La classe CForme (fichiers Forme.h et Forme.cpp)

La classe **CForme** gère toutes les propriétés communes à toutes les formes. Les attributs de tracés sont :







- ➡ **CForme::m_colColor** : La couleur du tracé
- ➡ **CForme::m_fThickness** : l'épaisseur du tracé
- ➡ **CForme::m_eStyle** : le style du tracé
- ➡ **CForme::m_ptP1** : le point d'origine de la forme
- ➡ **CForme::m_ptP2** : le point d'arrivée de la forme
- ➡ **CForme::m_rectBounds** : la boîte englobante de la forme, automatiquement calculée lors de la modification de **CForme::m_ptP1** et **CForme::m_ptP2** par les fonctions **CForme::SetP1** et **CForme::SetP2**.

Enfin, la fonction membre **CForme::Afficher()** définit les attributs du tracé avant affichage, de façon que la forme spécifique (rectangle ou ellipse) n'ait pas à le faire.

Exercice 1 Travail à réaliser




1.1 La classe **CRect**

La classe **CRect** modélise un rectangle. Le rectangle doit être affiché en suivant la boîte englobante de la forme définie par un point d'origine et un point d'arrivée.

-  Vous devez implémenter la classe **CRect** dans les fichiers **Rectangle.h** et **Rectangle.cpp**.
 -  Quel type de relation y a-t-il entre un rectangle et une forme ? Est-ce une relation de composition, d'agrégation ou d'héritage ?
 -  Quel(s) attribut(s) supplémentaire(s) à ceux de la classe **CForme** faut-il ajouter à la classe **CRect** ?
 -  Quel comportement de la forme faut-il modifier ou ajouter à la classe **CRect** ?
-  Vous devez ensuite ajouter à la classe **CTP** une liste de **CRect** et modifier les comportements de la classe **CTP** de façon à rendre opérationnel la création et l'affichage des rectangles.
 -  Référez-vous aux commentaires « **\todo** » des fichiers **TP.h** et **TP.cpp**.





1.2 La classe **CEllipse**

La classe **CEllipse** modélise une ellipse. L'ellipse doit être affichée en suivant la boîte englobante de la forme définie par un point d'origine et un point d'arrivée.

-  Implémentez la classe **CEllipse** sur le même modèle que la classe **CRect**
-  Vous devez ensuite ajouter à la classe **CTP** une liste de **CEllipse** sur le même principe que pour les rectangles et modifier les comportements de la classe **CTP** de façon à rendre opérationnel la création et l'affichage des ellipses.
 -  Référez-vous aux commentaires « **\todo** » des fichiers **TP.h** et **TP.cpp**.

1.3 La classe **CArc**

La classe **CArc** modélise un arc elliptique. L'arc doit être affiché en suivant la boîte englobante de son ellipse. L'arc est défini sur son ellipse génératrice par un angle de départ et un angle de balayage.

-  Implémentez la classe **CArc** sur le même modèle que la classe **CRect**
 -  Quelles données membres faut-il rajouter ?
-  Vous devez ensuite ajouter à la classe **CTP** une liste de **CArc** sur le même principe que pour les rectangles et modifier les comportements de la classe **CTP** de façon à rendre opérationnel la création et l'affichage des arcs.
 -  Référez-vous aux commentaires « **\todo** » des fichiers **TP.h** et **TP.cpp**.