


TP19 – Polymorphisme

Tracé de formes géométriques mieux

 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 du même programme que pour le TP précédent. Cependant, cette fois les formes sont ordonnées comme si elles étaient empilées les unes par-dessus les autres. L'utilisateur a la possibilité de placer la forme la plus haute en bas de la pile par l'appui sur la touche '↓' et inversement de placer la forme la plus basse en haut de la pile par l'appui sur la touche '↑'. Pour que cela soit mieux représenté, les formes sont maintenant remplies par une couleur de remplissage, qui est modifiable par la touche 'C' sur le même principe que les tracés.

0.2 Présentation de la structure du programme

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



Le programme s'articule toujours autour d'un objet de type **CTP** dont les méthodes sont appelées par le programme principal en fonction des actions de l'utilisateur. Cette classe gèrera (une fois que vous l'aurez fait dans la section 0) les formes dans une seule et même liste, de façon à ce que les différentes formes puissent se placer les unes par-dessus les autres quelques soient leurs types.

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





La classe **CForme** possède les mêmes caractéristiques que pour le TP précédent, avec en plus la donnée membre **CForme::m_colRemplissage** qui définit la couleur de remplissage de la forme. La fonction membre **CForme::Afficher()** définit la couleur de remplissage et les attributs du tracé avant affichage, de façon à ce que la forme spécifique (rectangle, ellipse ou arc) n'ait pas à le faire.

Exercice 1 Mise en place du polymorphisme

1.1 La hiérarchie de classes CForme, CRect, CEllipse et CArc.


-  Récupérez les fichiers sources des classes **CRect**, **CEllipse** et **CArc** du TP précédent pour les inclure dans votre projet (par copier-coller !).
-  Rendez la classe **CForme** polymorphe pour permettre l'affichage correct des objets de types dérivés par la manipulation d'un pointeur de type **CForme**.

1.2 La classe CTP

-  Munissez la classe **CTP** d'une liste de pointeurs intelligents vers des **CForme**.
-  Complétez les fonctions de la classe **CTP** pour rendre le code fonctionnel
 -  Référez-vous aux commentaires « **\todo** » des fichiers **TP.h** et **TP.cpp**.
 -  Beaucoup de code peut être copié-collé du TP précédent moyennant quelques modifications.

Exercice 2 Meilleures interactions (facultatif)

Il serait souhaitable de pouvoir modifier les objets qui ont été créés. Pour cela, nous voulons donner la possibilité à l'utilisateur de sélectionner des objets, de les déplacer et de modifier leurs caractéristiques. Nous voulons obtenir le fonctionnement suivant. Lorsque l'utilisateur appuie sur la touche 's', le logiciel passe en mode de sélection. L'utilisateur peut alors sélectionner un objet en cliquant sur la fenêtre. La forme sélectionnée sera alors celle qui est la plus « haute » dont la position de la souris se trouve à l'intérieur de sa boîte englobante. Lorsqu'une forme est sélectionnée, alors sa boîte englobante doit s'afficher en pointillés rouges. Si l'utilisateur fait glisser la forme sélectionnée, alors elle doit être déplacée. Enfin, si l'utilisateur appuie sur la touche 'm', la forme peut être modifiée en fonction de son type (par exemple pour un arc, le prochain clic définit le point P_2 , puis l'angle de départ et enfin l'angle d'ouverture).

-  En analysant le code existant dans le programme principal et la classe **CTP**, implémentez ces comportements.