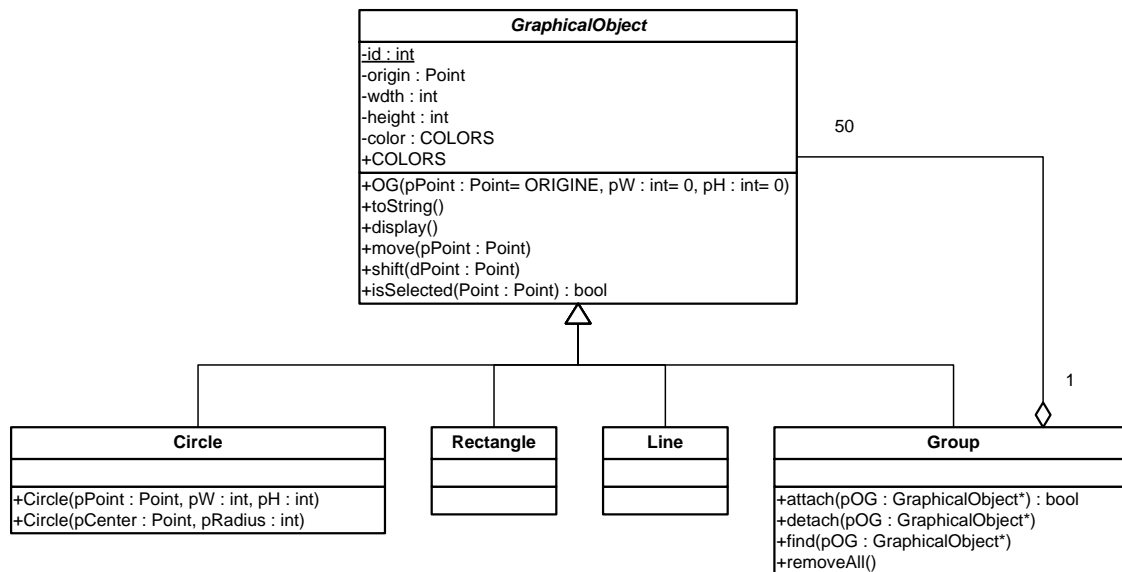


TP n° 3 de C++ - Héritage et polymorphisme

Dans ce TP, vous devez implémenter les classes données par le schéma UML suivant :



L'idée est de coder l'ossature d'un logiciel de dessin vectoriel en C++. Les objets sont affichables à l'écran grâce à la méthode `display()`.

- Le schéma UML est réduit à sa plus simple expression. Il manque notamment les accesseurs aux attributs lorsqu'ils sont nécessaires.
- Vous devrez tester vos classes et les méthodes implémentées.
- Chaque objet aura un identifiant unique géré par la classe `GraphicalObject`. Cet identifiant ne sera pas modifiable mais il sera consultable.
- La méthode `display()` affichera le composant à l'écran sous forme texte.
- La méthode `toString()` permettra de donner les propriétés des objets et pourra être utilisé pour la sauvegarde des objets dans un fichier.
- Le diagramme UML fait appel à une classe `Point` que vous devrez définir. Cette classe comportera les coordonnées x et y du point. Il sera judicieux de définir une fonction permettant de calculer la différence entre deux points.
- `COLORS` est une énumération qui permet de référencer les couleurs disponibles.
- Vous commencerez par écrire la classe `GraphicalObject`, puis `Circle`, `Rectangle` et `Line`.
- La classe `GraphicalObject` devra être abstraite et vous devrez choisir quelles méthodes sont virtuelles (pures ou non)
- La classe conteneur `Group` est plus délicate. Vous pourrez considérer qu'elle peut référencer au maximum 50 instances de `GraphicalObject`. Dans cette classe, le point, la hauteur et la largeur sont calculés comme étant ceux de la boîte englobante et mis à jour lors des opérations d'ajout ou de suppression des objets au groupe.
- Vous pourrez réfléchir aux classes/objets pour mettre en place une scène graphique, un conteneur qui pourrait contenir tous les objets affichables (on pourra limiter le nombre d'objets de la scène par exemple).

- En vous servant du polycopié, vous devrez également écrire des méthodes pour sauvegarder et relire une scène dans un fichier texte.
- Pour les classes qui ont un destructeur complexe, implémenter la forme normale de Coplien : constructeurs par défaut et de copie, destructeur et opérateur d'affectation.

Pour aller plus loin :

Gestion des appels redondants de méthodes virtuelles en cas d'héritage multiple

- Ecrire les classes C++ pour des personnels universitaires. On considère uniquement les enseignants, les chercheurs et les enseignants-chercheur. Réaliser une classification utilisant l'héritage, et proposer les attributs minimaux permettant de caractériser chaque classe (une sorte de fiche d'identité). Fournir les méthodes d'accès / modification pour les attributs de ces classes. Factoriser ce qui est possible dans une superclasse commune.
- Proposer une méthode toHtml qui génère la séquence des codes HTML (en vous documentant) nécessaire à la présentation d'une personne sous une forme visualisable par un navigateur. (Documentez-vous pour trouver les balises HTML élémentaires).
- Ecrire la fonction main dans un fichier séparé. Dans cette fonction, qui peut être considérée comme, vous créez quelques instances de chaque classe. Utiliser un tableau de personnes et appeler la fonction toHtml pour avoir au final une page HTML bien construite.