TD3

Exercice 1

Surcharger l'opérateur « < » dans la hiérarchie de classes.

```
bool ElementGraphique::operator<(const ElementGraphique& opD) const
{
    if(this->position().x() < opD.position().x())
    {
        return true;
    }

    if(this->position().x() > opD.position().x())
    {
        return false;
    }

    return this->position().y() < opD.position().y();
}
```

Ajoutez dans la classe « ZoneDessin » une méthode « trier() » qui trie tous les éléments graphiques grâce à la fonction « sort » et vérifier son fonctionnement.

Exercice 2

Modifiez la classe « ZoneDessin » en remplaçant l'utilisation du patron « std::vector » par le patron « std::set », toujours avec le type « PEG ».

Renommez « ajouterFin() » en « ajouter() ». Expliquez pourquoi.

Remplacez « enlever(int) » par « enlever(const ElementGraphique &) ». Expliquez pourquoi.

Remplacez « int indice(const ElementGraphique &) const » par « ElementGraphique * recherche(const ElementGraphique &) const ». Expliquez pourquoi.

Supprimez « element() » et « trier() ». Expliquez pourquoi.

Ajouter « ElementGraphique * getFirst() ».

Faîtes en sorte que le nouveau programme principal fonctionne correctement.

Exercice 3

Nous souhaitons maintenant que l'ordre soit du plus grand au plus petit. Pour cela, vous devrez ajoutez dans la hiérarchie de classe la surcharge de l'opérateur « operator> ». Cet opérateur permettra de dire, étant donnés deux éléments graphiques lequel est le plus grand. Il faudra utiliser la relation d'ordre totale sur les éléments graphiques :

- On regarde les points de base : d'abord les abscisses puis en cas d'égalité les ordonnées.

Exercice 4

Modifiez la classe « ZoneDessin » de fait qu'elle devienne une classe générique et qu'elle ait comme donnée membre « std::map<T1, T2> », « T1 » et « T2 » étant des types génériques (voir page 5 du cours, diapo « Patrons de classes » pour un exemple).

Ecrire un programme principal qui :

- Crée une instance de « ZoneDessin <PEG, string> ».
- Insère un carré de coordonnées (1, 2) et de côté 1, et avec la chaîne de caractères "Carré 1".
- Insère un carré de coordonnées (11, 12) et de côté 2, et avec la chaîne de caractères "Carré 2".
- Affiche le contenu.
- Supprime le carré de coordonnées (11, 12) et de côté 2.
- Affiche le contenu.

Exercice 5

Modifiez la classe « ZoneDessin » de fait qu'elle ait comme donnée membre « std::multiset<T> », « T » étant un type générique.

Ajouter les méthodes suivantes :

- "EraseFirst" qui supprime la première instance trouvée d'un élément du multiset correspondant au paramètre de type T.
- "EraseAll" qui supprime toutes les instances d'un élément du multiset correspondant au paramètre de type T.

Ecrire un programme principal qui :

- Crée une instance de « ZoneDessin <PEG> ».
- Insère un carré de coordonnées (1, 2) et de côté 1.
- Insère un carré de coordonnées (11, 12) et de côté 2, et avec la chaîne de caractères "Carré 2".
- Insère un carré de coordonnées (1, 2) et de côté 1.
- Affiche le contenu.
- Supprime via "EraseAll()" tous les carrés de coordonnées (1, 2) et de côté 1.
- Affiche le contenu.