

## TP-3 POO en C++ (3<sup>ème</sup> Séance)

Ce TP aborde les points suivants :

- Manipulation plus élaborée des classes et des objets
- Membres statiques, Constructeurs de copie, Listes d'initialisations, Surcharge de fonctions, Fonctions Amies

Chapitre 3,4 du Polycopie : Classes et Objets (suite)

### Exercice 1

**Q1.1** Réaliser une classe «**Complexe**» dans laquelle on définira deux membres privés, la partie réelle et la partie imaginaire. On prévoira des constructeurs et destructeur. On prévoira également les méthodes nécessaires au fonctionnement de cette classe : l'affichage d'un nombre complexe, récupérer la partie réelle ou la partie imaginaire, additionner, soustraire et multiplier deux complexes.

**Q1.2** Réaliser une classe polynôme du second degré «**Poly2deg**». Les membres privés (*\_a*, *\_b*, *\_c*) sont les coefficients entiers constantes.

On créera les constructeurs nécessaires et les fonctions membres utiles dont la fonction solution qui devra donner la solution de l'équation du second degré, quelle soit réelle ou complexe. On utilisera bien sur la classe complexe créée en **Q1.1**.

Ecrire un programme d'application mettant en pratique cette nouvelle classe.

### Exercice 2

**Q2.1** Réaliser une classe nommée «**Set\_Of\_Integer**» permettant de manipuler des ensembles de nombres entiers. On devra pouvoir réaliser sur un tel ensemble les opérations classiques suivantes :

- Lui ajouter un élément, connaître son « cardinal » (nombre d'éléments), savoir si un entier donné lui appartient.
- On conservera les différents éléments de l'ensemble dans un tableau alloué dynamiquement par le constructeur.
- Un argument (dont il existera une valeur par défaut) précisera le nombre maximal d'élément de l'ensemble.

**Q2.2** Ecrire un programme utilisant la classe «**Set\_Of\_Integer**» pour déterminer le nombre d'entiers différents contenus dans un tableau lu en donnée.

**Q2.3** Modifier la classe de façon à ce que maintenant l'ensemble d'entiers soit représenté par une «**liste chaînée**».

### Exercice 3

Créer deux classes :

L'une, nommée «**Vecteur3d**» permettant de représenter des vecteurs à 3 composantes de type double ; avec constructeurs, destructeur et une fonction membre d'affichage.

L'autre nommée «**Matrice**» permettant de représenter des matrices carrées de dimension **3x3** ; elle comportera aussi entre autres un constructeur avec argument (adresse d'un tableau **3x3** valeurs) qui initialisera la matrice avec les valeurs correspondantes.

Réaliser deux fonctions amies «**produit**». Une permettant de fournir le vecteur correspondant au produit d'une matrice par un vecteur. L'autre permettant de fournir la matrice correspondant au produit d'un vecteur par une matrice. Ecrire un programme pour tester les méthodes réalisées.