

# Programmation Orientée Objet (en C++)

## Objectifs

- Comprendre la notion de classe et objet
- Instancier des objets
- Définir et appeler des fonctions membres
- Surcharger des fonctions membres
- Définir et appeler des constructeurs
- Associer entre les classes

## Exercice 1 : Somme et produit de complexes

L'objectif de cet exercice est de définir les opérateurs arithmétiques en C++ d'une classe **Complexe** en utilisant les fonctions membres.

Soit la classe **complexe** pour gérer les nombres complexes :

### Attributs :

- **Re** la partie réelle de type **double**
- **Img** la partie imaginaire de type **double**

### Méthodes :

- **Constructeur** avec arguments (valeurs d'initialisation par default)
- Fonction d'**affichage**
- Fonction qui retourne le **module** sachant que :  $|z| = \sqrt{a^2 + b^2}$ .
- Fonction qui retourne le **conjugué** sachant que :  $\bar{z} = a - bi$
- Surcharge des opérateurs suivants :
  - **"+"** : (**complexe + complexe**) sachant que :
 
$$(a + bi) + (c + di) = (a + c) + (b + d)i ;$$
  - **"+"** pour (**complexe + double**) et **"+"** pour (**double + complexe**)
  - De même pour **"\*"**, **"-"** sachant que :

$$(a + bi)(c + di) = (ac - bd) + (ad + bc)i.$$

2. Créer un programme de test **main ()**.

### Exercice 2 : Point et cercle.

Ecrire une classe en C++ permettant de représenter des cercles qui se caractérisent par un rayon et un centre de type point (Classe Point).

Les opérations possibles sur un cercle sont :

- L'affichage des caractéristiques du cercle (méthode affiche ()).
- Le changement de son rayon,
- La translation de son centre,
- Le calcul de sa surface,
- Le calcul de son périmètre,
- Le test de l'égalité du cercle avec un autre cercle,
- La vérification de l'appartenance d'un point au cercle,
- Supposons que les constructeurs de la classe point sont privés. Donnez une solution pour créer des instances de cette classe.

### Exercice 3 :

1. Créer un projet Banque sous votre répertoire de travail.
2. Déclarer une classe Compte dans le fichier compte.h ayant les attributs suivants :

- **numCompte** : entier
- **nomProprietaire** : chaîne de caractères
- **solde** : réel

Et les opérations suivantes

- **retirerArgent** : retire du solde une valeur passée en paramètre et retourne vrai. Si le solde est insuffisant, la méthode retourne faux.
- **deposerArgent** : additionne une somme donnée au solde.
- **consulterSolde** : affiche le solde actuel.
- **transférerArgent** : transfère une somme donnée vers un compte donné en paramètre

Définir les opérations en dehors de la classe compte dans un fichier compte.cpp

3. Créer un programme principal main.cpp permettant de :

Définir un Compte A et lui attribuer les valeurs suivantes :

50003

User1 user1

80050.175

Ajouter au compte 50003 un montant de 12000MAD et afficher le nouveau solde.