

TD 3 – Classe Complexe

Le but de ce TD est de faire une classe **Complexe** efficace en termes de :

- mémoire, (utiliser le moins possible de mémoire)
- temps d'exécution (le plus court possible)
- utilisabilité (ergonomie, intuitif, confort d'interaction avec l'utilisateur, temps minimal requis pour atteindre un résultat, grande souplesse d'utilisation, ...)

Préparation : UML et rappels

1-a En s'appuyant sur l'illustration du cours, donner une modélisation complète de la classe **Complexe**. Les fonctionnalités des objets seront :

- les 4 opérations arithmétiques avec des complexes
- les 4 opérations arithmétiques avec des « double »
- calculs (par retour de valeur) des : conjugué, module, argument
- initialisation à partir de 0, 1 ou 2 *double*, et à partir d'un autre complexe
- affichage
- un test d'égalité

Pour favoriser l'utilisation, les fonctions seront le plus possible des « operator ». Par exemple : operator+, operator==, ...

1-b Comment calculer l'argument d'un **Complexe** ? Donner le code C++ de la fonction.

1-c Combien de fonctions « operator+ » existent dans cette classe ? Sur quoi le compilateur s'appuiera-t-il pour exécuter la bonne fonction ? Comment se nomme ce principe ?

1-d Quels sont les intérêts d'utiliser des passages des paramètres par référence pour les objets de type **Complexe** ? Vis-à-vis des objectifs de la classe, lequel de ces intérêts est le plus pertinent ? Pourrait-on faire autrement ?

1-e A quoi sert le mot clé *const* ? Pour une fonction membre, comment peut-il être utilisé ? En quoi l'ajout des mots clés *const* est-il pertinent pour l'utilisateur ?

Exercice 2 : Réalisation

2-a Réaliser sous *QtCreator* votre classe. Petit à petit, c'est plus sûr... Vous pourrez utiliser le squelette *Complexe.cpp* et *Complexe.h* après les avoir téléchargés (TD3.zip).

2-b Valider votre classe **Complexe** avec quelques exemples simples, puis déterminer :

- Impédance Z à 50 Hz avec R et C en parallèle, R=50 Ohms et C=10nF
- Rotation de 45 degrés de $w=1 + 0j$

2-c Bon et les objectifs ? Vitesse et empreinte mémoire... et facilité d'utilisation ? Ajouter le fichier *TD3_main_Bench_CPU.cpp* et adapter le code (normalement que le nom du fichier .h inclus). Comparer les résultats entre votre classe et celle de la std. Essayer d'améliorer les performances de votre classe. Quels sont les compromis ?

2-d Combien de constructeurs sont appelés par : $c = c*(a/b + c)$; Utilisation mémoire ?