

Laboratoire no. 1

Objectifs

Pratiquer :

- les concepts de classe et d'objet
- la surcharge des opérateurs
- la compilation séparée

Donnée

- 1) Implémenter une classe *Temps* permettant de modéliser des temps.

Un temps se caractérise par 3 données membres (de type *unsigned int*) : les heures (0 – 23), les minutes (0 – 59) et les secondes (0 – 59).

Quelle que soit l'opération réalisée, un temps doit toujours rester compris dans l'intervalle [00:00:00; 23:59:59].

La classe doit mettre à disposition :

- Un constructeur par défaut permettant d'initialiser les heures, minutes et secondes à zéro
- Un constructeur à 1 paramètre (de type *time_t*) permettant, par exemple, de construire un temps correspondant à l'heure courante du système
- Un constructeur à trois paramètres (*heures*, *minutes* et *secondes* dans cet ordre). Les paramètres *heures* et *minutes* doivent obligatoirement être renseignés à l'appel du constructeur; le paramètre *secondes* lui est facultatif. S'il n'est pas renseigné à l'appel du constructeur, cela signifie 0 secondes.
- Des accesseurs pour les trois données membres
- Tous les opérateurs de comparaison possibles (<, <=, >, >=, == et !=)
- Un opérateur ++ (en version pré-incrémentation et post-incrémentation)¹
- Un opérateur -- (en version pré-décrémentation et post-décrémentation)¹
- Un opérateur + permettant d'additionner deux temps et retournant un temps
- Un opérateur +=, permettant de faire, par ex : t1 += t2
- Un opérateur - permettant de soustraire deux temps et retournant un temps
- Un opérateur -=, permettant de faire, par ex : t1 -= t2
- Un opérateur permettant d'afficher un temps *t* au format "hh:mm:ss" (par ex 23:05:55) en écrivant cout << t
- Un opérateur de conversion permettant de convertir un objet de type Temps en temps décimal exprimé en heures. Exemple : la conversion de 01:30:00 doit donner 1.5 [h].

¹ Incrémenter, respectivement décrémenter, un temps signifie ajouter, respectivement soustraire, une seconde.

- 2) Ecrire un programme de test (*main*) permettant de vérifier le bon fonctionnement de la classe *Temps*.

Important

- Résoudre le problème ci-dessus en exploitant la compilation séparée.
- Implémenter la classe *Temps* en évitant au maximum toute redondance du code et en appliquant les formes canoniques proposées en cours.
- Il n'est pas demandé :
 - de vérifier la validité des paramètres des constructeurs ou des fonctions
 - de commenter systématiquement et dans le style *Doxygen* les divers éléments de la classe *Temps*. Quelques commentaires jugés utiles seront toutefois les bienvenus.
- A l'exception de *ctime*, il n'est pas autorisé d'utiliser des notions non vues en cours.

A réaliser

- ☐ Seul
☒ Par groupe de 3

Travail à rendre le 09.03.2020, au début de la séance de laboratoire

- ☒ Fiche de laboratoire (tirage papier)
☒ Listings des 3 fichiers sources C++
(imprimés portrait, recto/verso et noir/blanc via Notepad++ ou équivalent)
☒ Fichiers sources (.h et .cpp) UNIQUEMENT dans :
\\eistore1\cours\tic\RRH\INF2\Rendus\<votre répertoire>\Labo_1
où <votre répertoire> = répertoire du membre du groupe venant en premier dans l'ordre alphab.