

TP Supplémentaire 01 – Reprise en main C++



 Ce TP permet de raviver d'anciens souvenirs !

Exercice 0 Un peu de git

Pour ce TP, nous allons utiliser le framework de test LittleTestFramework© de Benjamin ALBOUY-KISSI.

-  Récupérez-le en clonant le dépôt git de LittleTestFramework :
 -  URL : [git@gitlab-lepuy.iut.uca.fr:bealbouy/littletestframework.git](https://gitlab-lepuy.iut.uca.fr:bealbouy/littletestframework.git).



Le fichier utile que vous copierez dans votre solution est **test.h**.



-  Créez une nouvelle solution Visual Studio dans laquelle vous créez votre projet
 -  Assurez-vous de choisir un projet d'application console en C++.




Exercice 1 Une classe de gestion de date

-  Ajouter à votre projet une nouvelle classe **CDate** correspondant aux indications suivantes :







CDate
-m_nYear : int
-m_cMonth : unsigned char
-m_cDay : unsigned char
-check() : void
<<construct>> +CDate()
<<construct>> +CDate(day : unsigned char, month : unsigned char, year : int)
+year() : int
+month() : unsigned char
+day() : unsigned char
+operator==(other : CDate&) : bool
+operator<=>(other : CDate&) : bool

-  Le C++ est formidable : depuis le C++20, il est possible de faire tous les opérateurs <, <=, > et >= en une fois en surchargeant l'opérateur <=>. Plus d'infos ici : https://en.cppreference.com/w/cpp/language/default_comparisons.
-  Assurez-vous d'avoir activé la dernière version supportée du C++ dans votre projet Visual Studio (Propriété du projet / C++ / Langage)








-  Vous ajouterez également un opérateur de sortie vers flux pour afficher la date au format **j/mm/aaaa**.
-  Vous ajouterez également un opérateur d'entrée depuis un flux pour lire une date entrée au format **j/mm/aaaa**.

-  Un autre truc cool, la surcharge de l'opérateur « literal ». Vous pouvez surcharger l'opérateur `""_suffix` pour créer des objets de façon « cool ». Par exemple, il serait possible d'écrire dans le code `CDate date = "8/09/2020"_date;` en surchargeant cet opérateur pour créer une date depuis la chaîne de caractère « 8/09/2020 ».
-  Rendez-vous là : https://en.cppreference.com/w/cpp/language/user_literal
-  Créez un petit programme de test pour vérifier les fonctionnalités de votre classe `CDate`.

Exercice 2 Une classe de gestion d'une personne

-  Créez une classe `CPerson` qui modélisera une personne avec :
 -  Un nom
 -  Un prénom
 -  Une date de naissance
-  Ajoutez à la classe la capacité d'être comparée à une autre instance (selon l'ordre lexicographique usuel)
-  Ajoutez un opérateur de sortie vers flux

Exercice 3 Une classe de gestion d'annuaire

-  Créez une classe de gestion d'annuaire dont les personnes sont classées alphabétiquement.
-  La classe doit pouvoir :
 -  Ajouter une personne
 -  Supprimer une personne
 -  Retrouver toutes les personnes dont les noms et prénoms correspondent à des critères de recherche
 -  Retrouver toutes les personnes dont les dates de naissances sont comprises entre une date min et une date max.
-  Ajoutez un opérateur de sortie vers flux