
TP 3IF - C++ : CLASSE SIMPLE

1 OBJECTIF

Ce premier TP d'initiation au langage C++ reste relativement simple et doit essentiellement vous permettre de mettre en œuvre les principes élémentaires de conception de logiciel avec la notion d'interface de classe, de réalisation de classe et les syntaxes élémentaires du langage C++. Il doit également vous permettre de mieux prendre en main l'outil de développement Eclipse. Ce TP insiste aussi sur les phases de spécification et de conception d'un logiciel, ainsi que sur la phase de tests.

Observation : L'utilisation d'une quelconque fonction issue d'une quelconque bibliothèque standard du C / C++, hormis la bibliothèque de manipulation des flots d'entrées/sorties **iostream**, est rigoureusement interdite.

2 CAHIER DES CHARGES ...

Le contexte de ce TP est situé dans un espace très éloigné, dans un monde qui n'a pas (ou peu) de liens avec la Terre. Vous faites partie d'une mission d'exploration intergalactique, qui doit chercher des traces de civilisation sur une planète peu connue : Aldebaran. Vous êtes sous la commande de Luke Skywalker, et vous êtes embarqués au bord d'un vaisseau spatial équipé d'un nombre important de capteurs. Votre vaisseau survole la superficie de la planète Aldebaran, et les divers senseurs à votre disposition vous donnent une description de la matière détectée dans chaque point de la planète au moment de son survol. La description de la matière dans un point (DMP) de la superficie d'Aldebaran est donnée sous la forme d'une chaîne de caractères. La cartographie ainsi réalisée sera utilisée pour trouver sur Aldebaran des régions homogènes avec une matière particulière.

Dans le cadre de l'application qui exploitera ces données, vous êtes en charge de la conception et de l'implémentation du module de gestion des mesures de type DMP, et vous devez fournir un API (via l'interface d'une classe : **Cartography**) qui sera utilisée par l'interface utilisateur.

Les méthodes (publiques !) requises pour cette classe sont :

- Un **premier constructeur** sans paramètres pour créer un objet ;
- Une méthode **Afficher/Display** pour afficher, sur la sortie standard (au sens Linux), la **valeur** d'un objet de votre classe ;
- Une méthode **Ajouter/Add** pour ajouter une mesure (DMP) ;
- Une méthode **NombreElements/Count** pour retrouver le nombre de mesures ;
- Une méthode **NombreElements/Count** pour retrouver le nombre de mesures pour une matière spécifiée ;
- Une méthode **Donnée/GetData** permettant l'accès à une mesure ;
- Une méthode **ChercheDMP/FindDMP** permettant de chercher une mesure correspondant à une matière bien définie.

Cette classe comporte également un **destructeur** d'objet.

Le travail supplémentaire, comme l'ajout de méthodes à la liste précédente, ne sera pas considéré : il est donc **interdit d'ajouter** une quelconque **méthode publique** à cette liste. L'accent est mis sur la qualité de la programmation (sa simplicité) et le respect des règles et contraintes de conception et non sur la quantité.

3 RESULTATS ATTENDUS

3.1 UN DOCUMENT DE SPECIFICATION ET DE CONCEPTION

- Les spécifications complètes de votre classe (principales définitions et choix généraux) ;
- Les spécifications complètes, précises et non-ambiguës de méthodes : Add, FindDMP ;
- Les tests unitaires **numérotés**, associés à votre spécification, pour la méthode Count.
- Les tests fonctionnels de la classe, **numérotés**.

Ce document sera rendu en format PDF, avec le nom **B3XXX_spec.pdf** (ou XXX représente votre numéro de binôme).

3.2 UN MANUEL D'UTILISATION

Une documentation détaillée (orientée utilisateur) pour la méthode Count. Dans le manuel d'utilisation, il est interdit *d'expliquer comment la méthode a été réalisée*. L'utilisateur souhaite simplement savoir à quoi sert la méthode, quel est le service rendu, comment l'appeler, mais en aucun cas comment elle rend le service ! Vous pouvez vous inspirer de l'exemple de la fonction *strcpy* et des informations obtenues par la commande Linux man. C'est juste un exemple qui ne doit pas devenir la référence absolue...

Strcpy**Syntaxe**

```
#include <string.h>
char *strcpy ( char *dest, const char *src );
```

Description

Copie une chaîne de caractères dans une autre.

Copie la chaîne de caractères `src` dans la chaîne de caractères `dest`, en s'arrêtant après avoir copié la marque de fin de chaîne (le caractère nul `'\0'`).

Aucune allocation dynamique pour la destination n'est effectuée par `strcpy`.

Valeur de retour

`strcpy` renvoie le pointeur sur la chaîne de caractères `dest`.

Exemple d'utilisation

```
#include <iostream.h>
#include <string.h>
#define TAILLE 10
int main( )
{
    char chaine [ TAILLE + 1 ];
    char *ch1 = "abcdefghij";
    strcpy ( chaine, ch1 ); //...
    cout << chaine << endl; //...
    return 0;
}
```

Ce document sera rendu en format PDF, avec le nom **B3XXX_man.pdf**

3.3 UN DOCUMENT DE REALISATION

Ce document doit contenir :

- La réalisation des tests unitaires pour la méthode `Count`, **en respectant la numérotation** ; -> document **B3XXX_tests.cpp**
- Les listes détaillées et correctement présentées de votre classe (selon le *Guide de Style*). -> documents **B3XXX_class.h**, **B3XXX_class.cpp**
- Tous les autres tests que vous implémentez, avec éventuellement un script qui exécute ces tests pour vérifier la non-régression, seront livrés dans un répertoire **/tests**.

Pour simplifier l'évaluation par l'équipe enseignante de votre travail, le nom de votre classe doit être `Cartography`, et les méthodes suivantes (publiques !) doivent faire partie de la classe :

```
Cartography();
bool Add(float fX, float fY, char* pszDMP);
long Count();
long Count(char* pszDMP);
long FindDMP(char* pszDMP, long lFirstPos, float& fX, float& fY);
```

3.4 UN DOCUMENT D'EVALUATION

Ce document présente les performances et les limites de votre classe. Vous êtes libres pour choisir le contenu. Par exemple, vous pouvez présenter le passage à l'échelle du temps d'exécution de la méthode `FindDMP`, qui est la méthode la plus consommatrice en temps d'exécution. Nom du document : **B3XXX_eval.pdf**

L'ensemble de ces documents sera copié dans le répertoire ... avant la date limite fixée pendant la séance.

NB : Le compte rendu de ce TP ne devra pas excéder 8 pages, hors documents de réalisation que vous présenterez parfaitement selon les conventions d'écriture d'un *Guide de Style C++*. N'oubliez pas qu'un bon logiciel est avant tout un logiciel fiable donc bien testé.