

TP7 : Définition d'une classe (deux séances)

Programmez la classe `tableauReels` qui gère des tableaux de réels triés.

Le fichier `tableauReels.h` est donné en annexe.

Le fichier `tableauReels.h` ainsi que le squelette du fichier `tableauReel.cc` sont stockés dans le répertoire `/net/Bibliotheque/AS/TP_ASD_PROG/TP7/`.

Un exemple basique de création d'un objet de la classe `tableauReels` et de l'appel de ses méthodes.

```
tableauReels t;  
t.ajouter(2.1); t.ajouter(3.2); t.retirer(2.1);  
t.afficher(); t.sauvegarder("toto");
```

Utilisez la compilation séparée telle qu'elle a été vu en cours, une structuration logique du projet, ainsi qu'un fichier « `makefile` ».

La structure logique classique d'un projet C++ est la suivante : un répertoire qui contient uniquement les données propres au projet ; ces données sont réparties en 3 sous répertoires.

- Un répertoire (nommé `src`) contenant le code source (`*.cc`) et le fichier `makefile`.
- Un répertoire (nommé `include`) contenant les définitions des classes, des constantes ... (`*.h`).
- Un répertoire (nommé `bin`) contenant les exécutables et les fichiers objets.

Rappel Système : la commande Unix créant un répertoire est : `mkdir`.

Un exemple structuration logique d'un projet et de son « `makefile` » est présenté dans le répertoire `/net/Bibliotheque/AS/TP_ASD_PROG/TP7/Make`

```
# makefile  
TARGET = essai  
  
#options de compilation  
CC = g++  
CFLAGS = -Wall -g -c  
  
DIR_INCLUDE      = ../include  
DIR_BIN          = ../bin  
INCLDIR          = -I $(DIR_INCLUDE)  
  
$(DIR_BIN)/$(TARGET) : $(DIR_BIN)/main.o $(DIR_BIN)/f1.o $(DIR_BIN)/f2.o  
    @echo "--- Edition des liens ---"  
    $(CC) -o $(DIR_BIN)/$(TARGET) $(DIR_BIN)/main.o $(DIR_BIN)/f1.o $(DIR_BIN)/f2.o  
  
$(DIR_BIN)/main.o : main.cc $(DIR_INCLUDE)/f1.h  
    $(CC) $(CFLAGS) $(INCLDIR) main.cc -o $(DIR_BIN)/main.o  
$(DIR_BIN)/f1.o   : f1.cc $(DIR_INCLUDE)/f1.h $(DIR_INCLUDE)/f2.h  
    $(CC) $(CFLAGS) $(INCLDIR) f1.cc -o $(DIR_BIN)/f1.o  
$(DIR_BIN)/f2.o   : f2.cc $(DIR_INCLUDE)/f2.h  
    $(CC) $(CFLAGS) $(INCLDIR) f2.cc -o $(DIR_BIN)/f2.o  
  
clean:  
    rm $(DIR_BIN)/*.o
```

```

#ifndef TABLEAUREELS_H
#define TABLEAUREELS_H

#include <iostream>
#include <cassert>

using namespace std;

class tableauReels {
private :
    static int const MaxTaille = 1024;
    int m_taille ;
    float m_tab[MaxTaille] ;

public :

    // construit un nouvel objet vide de la classe tableeauRéels;
    tableauReels() ;

    // construit un nouvel objet de la classe tableeauRéels contenant
    // les réels stockés dans « m_tab » ;
    // le deuxième paramètre est la taille de ce tableau
    // le tableau donné en paramètre peut ne pas être trié.
    tableauReels(float[], int) ;

    // OPTIONNELLE (à programmer en dernier)
    // construit un nouvel objet de la classe tableeauRéels contenant les réels
    // stockés dans les deux objets "tableauReels" donnés en argument.
    tableauReels(tableauReels, tableauReels) ;

    // libère l'espace mémoire
    ~tableauReels();

    // retourne la valeur m_taille;
    int getTaille();

    // Si l'argument de getValeur, noté i, vérifie 0 <= i < m_taille
    // alors la méthode retournera la valeur de m_tab[i].
    // Sinon (i < 0 ou i >= m_taille), il s'agit d'une exception - voir TP1.
    float getValeur(int) ;

    // affiche sur la sortie standard le tableau m_tab.
    void afficher() ;

    // Si possible, ajoute le réel donné en argument au tableau m_tab.
    int ajouter(float);

    // Si possible, retire le réel donné en argument au tableau m_tab;
    // l'élément à retirer est trouvé via une recherche dichotomique.
    int retirer(float) ;

    // Si possible, stocke le tableau m_tab dans un fichier.
    // Le nom du fichier est le paramètre.
    int sauvegarder(string) ;

    // Si possible, récupère les données stockés dans un fichier pour
    // les mettre dans m_tab. Le nom du fichier est le paramètre.
    int recuperer(string) ;

};

#endif

```