Classe NombreDeHits

NombreDeHits.h

```
/**************************************************************************
NombreDeHits  -  fichier d'en-tête de la structure <NombreDeHits>
-------------------
début                : 23/11/2015 15:48:12
copyright            : (C) 2015 par Quentin SCHROTER, Nicolas GRIPONT
e-mail               : quentin.schroter@insa-lyon.fr , nicolas.gripont@insa-lyon.fr
**************************************************************************/

//---------- Interface de la structure <NombreDeHits> (fichier NombreDeHits.h) ------
#if ! defined ( NombreDeHits_H )
#define NombreDeHits_H

//------------------------------------------------- Interfaces utilisées

//------------------------------------------------------------ Constantes

//------------------------------------------------------------------ Types

//-------------------------------------------------------------------------
// Rôle de la structure <NombreDeHits>
// Class permettant de stocker un nombre de hits par heure et par statut
// (reussi  ou echec)
//
//-------------------------------------------------------------------------

class NombreDeHits
{
//----------------------------------------------------------------- PUBLIC
public:
        //-------------------------------------------------- Méthodes publiques
        int NombreDeHitsPourUneHeure ( int uneHeure, bool UniquementReussi = true );
        // Mode d'emploi :
   // Methode permettant de recuperer le nombre de hits pour une heure donnee
   // (UniquementReussi = true , uniquement les hits reussis sinon tous)
        //
   // Contrat : aucun.
        //

   int NombreDeHitsTotal( bool UniquementReussi = true ) const;
   // Mode d'emploi :
   // Methode permettant de recuperer le nombre de hits
   // (UniquementReussi = true , uniquement les hits reussis sinon tous)
   //
   // Contrat : aucun.
   //

   void MAJHits ( bool statut, int heure );
   // Mode d'emploi :
   // Méthode permettant d'incrémenter le nombre de hits en fonction d'un
   // statut (reussi : true, echec : false) et d'une heure.
   // Ne fait rien si heure < 0 et heure >= 24
```

```
    //
    // Contrat : aucun.
    //


        //------------------------------------------------- Surcharge d'opérateurs
        NombreDeHits & operator = (const NombreDeHits & unNombreDeHits);
        // Mode d'emploi :
    // Redefinition de l'operateur =
    //
    // Contrat : aucun.
        //

        //-------------------------------------------- Constructeurs - destructeur
        NombreDeHits(const NombreDeHits & unNombreDeHits);
    // Mode d'emploi : constructeur de copie
        //
        // Contrat :
        //

        NombreDeHits();
    // Mode d'emploi : constructeur
        //
    // Contrat : aucun.
        //

        virtual ~NombreDeHits();
    // Mode d'emploi : destructeur.
        //
    // Contrat : aucun.
        //

    //------------------------------------------------------------------ PRIVE

protected:
    //----------------------------------------------------- Méthodes protégées


    //------------------------------------------------------- Attributs protégés

    static const int NB_HEURE_PAR_JOUR = 24;           //nombre d'heure par jour

    int nombreDeHitsReussisParHeure[NB_HEURE_PAR_JOUR]; // tableau contenant le
                                                        // nombre de hits reussis
                                                        // par heure

    int nombreDeHitsEchouesParHeure[NB_HEURE_PAR_JOUR]; // tableau contenant le
                                                        // nombre de hits echoues
                                                        // par heure

};

//--------------------------- Autres définitions dépendantes de <NombreDeHits>

#endif // NombreDeHits_H
```

NombreDeHits.cpp

```
/*************************************************************************
NombreDeHits  -  fichier de réalisation de la structure <NombreDeHits>
-------------------
début                : 23/11/2015 15:47:59
copyright            : (C) 2015 par Quentin SCHROTER, Nicolas GRIPONT
e-mail               : quentin.schroter@insa-lyon.fr , nicolas.gripont@insa-lyon.fr
*************************************************************************/

//---------- Réalisation de la structure <NombreDeHits> (fichier NombreDeHits.cpp) --

//------------------------------------------------------------------ INCLUDE

//-------------------------------------------------------- Include système
#include <iostream>
using namespace std;

//------------------------------------------------------ Include personnel
#include "NombreDeHits.h"

//------------------------------------------------------------------ Constantes

//------------------------------------------------------------------ PUBLIC

//----------------------------------------------------- Méthodes publiques
// type NombreDeHits::Méthode ( liste des paramètres )
// Algorithme :
//
//{
//} //----- Fin de Méthode


int NombreDeHits::NombreDeHitsPourUneHeure(int uneHeure, bool UniquementReussi)
{
    int result;
    if ( uneHeure >= NB_HEURE_PAR_JOUR || uneHeure < 0 )
            {
        result =  -1;
            }
    else if (UniquementReussi)
            {
        result = nombreDeHitsReussisParHeure[uneHeure];
            }
            else
            {
        result = nombreDeHitsEchouesParHeure[uneHeure] +
nombreDeHitsReussisParHeure[uneHeure];
            }
    return result;
}
```

```
int NombreDeHits::NombreDeHitsTotal(bool UniquementReussi) const
{
        int s = 0;
   for (int i = 0; i < NB_HEURE_PAR_JOUR; i++)
        {
                s += nombreDeHitsReussisParHeure[i];
        }
   if ( !UniquementReussi )
        {
      for (int i = 0; i < NB_HEURE_PAR_JOUR; i++)
      {
         s += nombreDeHitsEchouesParHeure[i];
      }
         }
         return s;
}

void NombreDeHits::MAJHits (bool statut, int heure )
// Algorithme :
//
{
   if ( heure >= 0 && heure < NB_HEURE_PAR_JOUR )
   {
      if (statut)
      {
         nombreDeHitsReussisParHeure[heure]++;
      }
      else
      {
         nombreDeHitsEchouesParHeure[heure]++;
      }
   }
} //----- Fin de MAJHits

//------------------------------------------------- Surcharge d'opérateurs
NombreDeHits & NombreDeHits::operator = (const NombreDeHits & unNombreDeHits)
// Algorithme :
//
{
   if (this != &unNombreDeHits)
   {
      for ( int i = 0; i < NB_HEURE_PAR_JOUR; i++ )
      {
         nombreDeHitsEchouesParHeure[i] = unNombreDeHits.nombreDeHitsEchouesParHeure[i];
         nombreDeHitsReussisParHeure[i] = unNombreDeHits.nombreDeHitsReussisParHeure[i];
      }
   }
   return *this;
} //----- Fin de operator =


  //-------------------------------------------- Constructeurs - destructeur
NombreDeHits::NombreDeHits(const NombreDeHits & unNombreDeHits)
// Algorithme :
//
{
```

```
#ifdef MAP
        cout << "Appel au constructeur de copie de <NombreDeHits>" << endl;
#endif
    for ( int i = 0; i < NB_HEURE_PAR_JOUR; i++ )
    {
        nombreDeHitsReussisParHeure[i] = unNombreDeHits.nombreDeHitsReussisParHeure[i];
        nombreDeHitsEchouesParHeure[i] = unNombreDeHits.nombreDeHitsEchouesParHeure[i];
    }
} //----- Fin de NombreDeHits (constructeur de copie)


NombreDeHits::NombreDeHits()
// Algorithme :
//
{
#ifdef MAP
        cout << "Appel au constructeur de <NombreDeHits>" << endl;
#endif
    for ( int i = 0; i < NB_HEURE_PAR_JOUR; i++ )
    {
        nombreDeHitsReussisParHeure[i] = 0;
        nombreDeHitsEchouesParHeure[i] = 0;
    }
} //----- Fin de NombreDeHits


NombreDeHits::~NombreDeHits()
// Algorithme :
//
{
#ifdef MAP
        cout << "Appel au destructeur de <NombreDeHits>" << endl;
#endif
} //----- Fin de ~NombreDeHits


  //------------------------------------------------------------------ PRIVE

  //---------------------------------------------------- Méthodes protégées
```