

```

/*****
                                Cible - description
                                -----
    début                      : 23/11/2015
    copyright                  : (C) 2015 par mfallouh_mvirsolv
*****/

//----- Interface de la classe <Cible> (fichier Cible.h) -----
#if ! defined ( CIBLE_H )
#define CIBLE_H
#include <string>
#include <list>
#include <map>
using namespace std;

#include "Log.h"

//----- Interfaces utilisées

//----- Constantes
extern const char SEP_REQ , SEP , SEP_DATE_FIN, SEP_INT, SEP_PVIRG;
const int NB_HEURES = 24;
const char SEP_DATE_DEBUT = '[', SEP_HEURE = ':', SEP_PT = '.';
const string EXCLUSIE[] = { (string)"jpg", (string)"png", (string)"tga",
(string)"gif", (string)"jpeg", (string)"bmp", (string)"js", (string)"css",
(string)"ico" }; //extensions à exclure si option e spécifiée
const int NB_EXTENSIONS = 9; //nombre d'extensions bannies
//----- Types

//-----
// Rôle de la classe <Cible>
//
// Il s'agit de la classe stockant l'ensemble des logs relatifs à une adresse
// cible.
// Les données sont organisées par heure de Greenwich et type de requête,
// pour accélérer les traitements.
//
//-----

class Cible
{
//----- PUBLIC

public:
//----- Méthodes publiques

    int Ajouter(const string & log);
    // Mode d'emploi :
    // Crée un Log à partir de la string entrée en paramètre.
    // Ajoute ce Log au bon emplacement (bonne heure, bonne requête).
    // Crée au besoin une nouvelle clé (requête) dans la map.
    //
    // Contrat :
    //

    int Compte(const string &requete, const bool e = false, const int t =-1)
const;
    // Mode d'emploi :

```

```

        // Compte le nombre de logs consignés correspondant à la requête entrée en
paramètre.
        // Tient compte des options e et t pour filtrer les résultats.
        //
        // Contrat :
        // 0<=t<24

//----- Surcharge d'opérateurs
Cible & operator = ( const Cible & unCible ); // déclaré mais non défini pour
empêcher son utilisation
// Mode d'emploi :
//
// Contrat :
//

//----- Constructeurs - destructeur

Cible ( const Cible & unCible );
// Mode d'emploi (constructeur de copie) :
//
// Contrat :
//

Cible ( );
// Mode d'emploi :
//
//
// Contrat :
//

virtual ~Cible ( );
// Mode d'emploi :
//
// Contrat :
//

//----- PRIVE
protected:
//----- Méthodes protégées
private:
//----- Méthodes privées

        map <string, list<Log>> lesLogs[24];    // stock des logs. Tableau de 24
cases (1 par heure)
                                                // contenant
des listes de logs triées par type de requête,
                                                // par le
biais d'une map.

protected:
//----- Attributs protégés
private:
//----- Attributs privés

```

```
//----- Classes amies
    friend class Graphe;
//----- Classes privées
//----- Types privés
};

//----- Types dépendants de <Cible>
#endif // CIBLE_H
```