

La classe vector de la STL C++

Sommaire :

I - Présentation de la classe vector.....	1
II - Classes Capteurs et ListeCapteurs.....	2

I - Présentation de la classe vector

La classe **vector** fait partie de la **bibliothèque STL (Standard Template Library)**. Les **vectors** sont des **tableaux dynamiques** avec une gestion de la mémoire et des méthodes de gestion intégrées.

Pour créer une application utilisant la classe **vector** de la **STL**, il faut inclure le fichier **vector** :
#include <vector>

Exemple de programme :

```
#include <iostream>
#include <vector>
//using namespace std;

int main (void)
{
    std::vector<int> listeCapteur;    // Déclaration d'un vecteur contenant des int

    listeCapteur.push_back(10);    // Ajout d'un int de valeur 10 à la fin du vector
    listeCapteur.push_back(20);    // Ajout d'un int de valeur 20 à la fin du vector
    listeCapteur.push_back(30);    // Ajout d'un int de valeur 30 à la fin du vector

    // Parcours du vector
    for(std::size_t i=0; i<listeCapteur.size(); i++)
    {
        std::cout << "Indice= " << i << ' ';
        std::cout << "Valeur= " << listeCapteur[i] << ' ';
        std::cout << std::endl;
    }

    //Recherche d'un entier dans le vector avec sa valeur
    int val = 20;
    std::cout << "Valeur = " << val << ' ';
    for(std::size_t i=0; i<listeCapteur.size(); i++)
    {
        if (listeCapteur[i] == val)
        {
            std::cout << "Indice= " << i << ' ';
            std::cout << std::endl;
        }
    }
    return 0;
}
```

II - Classes Capteurs et ListeCapteurs

Nous voulons utiliser la classe **vector** de la librairie standard **STL C++** afin de gérer un **tableau dynamique** contenant une **liste de capteurs**.

Nous allons donc commencer par créer la classe **Capteur**. La déclaration de la classe **Capteur** est la suivante (fichier **capteur.h**) :

```
class Capteur
{
private:
    int id;        // identifiant du capteur
    float temp;   // grandeur de température du capteur
public:
    Capteur(int); // Constructeur créant un capteur identifié par un int
    int getId (void); // Récupération de l'identifiant d'un capteur
    void setTemp(float); // Affectation de la température du capteur
    float getTemp (void); // Récupération de la température du capteur
};
```

On désire maintenant créer une classe **ListeCapteurs** gérant des objets de type **Capteur** dans un **vector**. Nous allons donc commencer par créer cette classe. La déclaration de la classe **ListeCapteurs** est la suivante (fichier **listecapteurs.h**) :

```
#include <vector>
#include "capteur.h"
class ListeCapteurs
{
private:
    Capteur* capteur;
    std::vector<Capteur*> listeCapteurs;
public:
    int getNbCapteurs(void); // Récupération du nombre de capteurs
    int EcrireInfosCapteur(int, float); // Affectation d'un capteur
    Capteur* LireInfosCapteur(int); // Récupération d'un capteur
};
```

Remarque 1 : La méthode **int getNbCapteurs(void)** permet de retourner le nombre d'objets de type **Capteur** contenu dans dans le **vector listeCapteurs**.

Remarque 2 : La méthode **int EcrireInfosCapteur(int id, float temp)** permet, s'il n'existe pas, de **créer** un objet de type **Capteur** identifié par **id**, de l'**ajouter** dans le **vector listeCapteurs** et enfin de lui **affecter** une valeur de température **temp**.

Remarque 3 : La méthode **Capteur* LireInfosCapteur(int id)** permet de retourner un objet de type **Capteur** identifié par **id**.