

```

/*****
Capteur.h
*****/ 4/5
#pragma once
#include <string>

using namespace std;

class Capteur
{
protected:
    int static nombreDeCapteurs;
    string type, noSerie;

public:
    Capteur(string type, string noSerie);
    string getType() { return this->type; }
    string getNoSerie() { return this->noSerie; }
    int static getNombreCapteurs();
    ~Capteur();
};

/*****
CapteurTemperature.h
*****/ 2/2
#pragma once
#include "Capteur.h"
class CapteurTemperature : public Capteur
{
protected:
    double temperatureMinAutorisee;
    double temperatureMaxAutorisee;

public:
    CapteurTemperature(string numeroSerie, double min, double max);
    double getTemperature();
};

/*****
CapteurTemperatureExterieur.h
*****/ 2/2
#pragma once
#include "CapteurTemperature.h"
class CapteurTemperatureExterieur : public CapteurTemperature
{
public:
    CapteurTemperatureExterieur(string derniersChiffresNumeroSerie, double min, double max);
    bool verifieConformite();
};

/*****
Capteur.cpp
*****/ 4/4
#include "Capteur.h"

int Capteur::nombreDeCapteurs = 0;

Capteur::Capteur(string type, string noSerie)
{
    this->type = type;
    this->noSerie = noSerie;
    this->nombreDeCapteurs++;
}

Capteur::~Capteur()
{
    this->nombreDeCapteurs--;
}

int Capteur::getNombreCapteurs()
{
    return nombreDeCapteurs;
}

/*****
CapteurTemperature.cpp
*****/ 7/7
#include "CapteurTemperature.h"

```

```

CapteurTemperature::CapteurTemperature(string numeroSerie, double min, double max) : Capteur("TEMP", numeroSerie)
{
    this->temperatureMinAutorisee = min;
    this->temperatureMaxAutorisee = max;
}

double CapteurTemperature::getTemperature()
{
    double temp = 0;
    temp = rand();
    temp = temp / 32767;
    temp = temp * (temperatureMaxAutorisee - temperatureMinAutorisee);
    temp = temp + temperatureMinAutorisee;
    return temp;
}

/*****
CapteurTemperatureExterieur.cpp
*****/
#include "CapteurTemperatureExterieur.h"

CapteurTemperatureExterieur::CapteurTemperatureExterieur(string derniersChiffresNumeroSerie, double min, double max) : CapteurTemperature("33"+derniersChiffresNumeroSerie,min,max)
{
}

bool CapteurTemperatureExterieur::verifieConformite()
{
    if (temperatureMinAutorisee < -50 && temperatureMaxAutorisee > 60)
        return true;

    else
        return false;
}

/*****
TestCapteurs_A_Completer.cpp
*****/
/*****
Programme principal à compléter
Au final, l'exécution du programme doit
produire un résultat similaire à l'exécutable fourni
*****/
#define _CRT_SECURE_NO_WARNINGS

#include <iostream> // bibliothèque de gestion des E/S
#include <conio.h> // gestion de la console (ici _getch())
#include <windows.h>
#include <vector>
#include "Capteur.h"
#include "CapteurTemperature.h"
#include "CapteurTemperatureExterieur.h"

using namespace std; // utilisation de l'espace de nommage standard

/*=====
Fonction principale 13/17
=====*/
int main()
{
    // Prise en compte des accents
    // Il faudra choisir la police de caractères Consolas ou Lucida
    SetConsoleOutputCP(1252);

    //==== Déclaration d'éventuelles variables locales ====
    //...

    // Affichage du nombre de capteurs
    cout << "ETAPE1: NOMBRE DE CAPTEURS = " << Capteur::getNombreCapteurs() << " (attendu 0)" << endl << endl;

    // Créer un tableau de 20 capteurs de température (modèles de base)
    // Ces capteurs de température ont des numéros de séries allant de "111101" à "111120"
    // de température mini -20.5 et température maxi +80.5
    CapteurTemperature* capteur[20];

    for (unsigned i = 0; i < 20; i++)
    {
        capteur[i] = new CapteurTemperature("1111", -20.5, 80.5);
    }
}

```

Tous vos capteurs ont le même numéro de série

```

}

// Affichage du nombre de capteurs
cout << "ETAPE2: NOMBRE DE CAPTEURS = " << Capteur::getNombreCapteurs() << " (attendu 20)" << endl << endl;

// Afficher 8 mesures par capteurs (20 lignes de 8 valeurs)
cout << "CAPTEURS DE TEMPERATURE (MODELE DE BASE)" << endl;
cout.precision(4); // 4 digits de précision
for (unsigned i = 0; i < 20; i++)
{
    cout << "Capteur de numero de serie: " << capteur[i]->getNoSerie() << i + 1 << endl;
    for (unsigned j = 0; j < 8; j++)
    {
        cout << capteur[i]->getTemperature() << "\t";
    }
    cout << endl;
}

// Supprimer de la mémoire 10 capteurs
for (unsigned i = 0; i < 10; i++)
{
    delete capteur[i];
}

// Affichage du nombre de capteurs
cout << "ETAPE3: NOMBRE DE CAPTEURS = " << Capteur::getNombreCapteurs() << " (attendu 10)" << endl << endl;

// Créer 12 capteurs de température extérieure dont les numéros de séries finissent par "5501" à "5512"
// Les 11 premiers capteurs ont une température mini -85.5 et température maxi +300.5
// le dernier capteur a une température mini de -45.5 et maxi de 60.6
CapteurTemperatureExterieur* capteurExt[12];

for (unsigned i = 0; i < 11; i++)
{
    capteurExt[i] = new CapteurTemperatureExterieur("55", -85.5, 300.5);
}
capteurExt[11] = new CapteurTemperatureExterieur("55", -45.5, 60.6);

// Afficher 8 mesures par capteurs (12 lignes de 8 valeurs) et afficher pour chaque capteur s'il est con
forme ou non
cout << "\n\nCAPTEURS DE TEMPERATURE (MODELE EXTERIEUR)" << endl;
for (unsigned i = 0; i < 12; i++)
{
    cout << endl;
    cout << "Capteur de numero de serie: " << capteurExt[i]->getNoSerie() << i + 1;
    if (CapteurTemperatureExterieur::verifieConformite)
        cout << "\t Conforme" << endl;
    else
        cout << "\t Non Conforme" << endl;
    for (unsigned j = 0; j < 8; j++)
    {
        cout << capteurExt[i]->getTemperature() << "\t";
    }
    cout << endl;
}

// Affichage du nombre de capteurs
cout << "ETAPE4: NOMBRE DE CAPTEURS = " << Capteur::getNombreCapteurs() << " (attendu 22)" << endl << endl;

// Supprimer tous les capteurs
for (unsigned i=10; i < 20; i++)
{
    delete capteur[i];
}
for (unsigned i = 0; i < 12; i++)
{
    delete capteurExt[i];
}

// Affichage du nombre de capteurs
cout << "ETAPE5: NOMBRE DE CAPTEURS = " << Capteur::getNombreCapteurs() << " (attendu 0)" << endl << endl;

_getch(); // attente d'appui sur une touche
return 0; // sortie du programme
}

```

Tous vos capteurs ont le même numéro de série. Vous avez fait une bidouille en ajoutant ((i+1)) à l'affichage pour que l'on ait l'impression que les numéros sont différents :-)

Tous vos capteurs ont le même numéro de série

Tous vos capteurs ont le même numéro de série. Vous avez fait une bidouille en ajoutant ((i+1)) à l'affichage pour que l'on ait l'impression que les numéros sont différents :-)