

```

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

using namespace std;

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

public:
    Capteur(string type, string noSerie);
    string getType();
    string getNoSerie();
    static int getNombreCapteurs();
    ~Capteur();
};

/*****
CapteurTemperature.h
*****/ 2/2
#pragma once
#include <iostream>
#include <math.h>
#include <string>
#include "Capteur.h"

using namespace std;

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 <string>
#include <math.h>
#include "CapteurTemperature.h"

using namespace std;

class CapteurTemperatureExterieur : public CapteurTemperature
{
public:
    CapteurTemperatureExterieur(string derniersChiffresNumeroSerie, double min, double max);
    bool VerifieConformite();
};

/*****
Capteur.cpp
*****/ 4/4
#include <string>
#include "Capteur.h"

using namespace std;

int Capteur::nombreDeCapteurs = 0;

Capteur::Capteur(string type, string noSerie)
{

```

```

        this->type = type;
        this->noSerie = noSerie;

        nombreDeCapteurs++;
};

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

string Capteur::getType()
{
    return this->type;
};

string Capteur::getNoSerie()
{
    return this->noSerie;
};

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

/*****
CapteurTemperature.cpp
*****/
#include <string>
#include <math.h>
#include "CapteurTemperature.h"

using namespace std;
int u;

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

double CapteurTemperature::getTemperature()
{
    u = rand();
    return u;
};

/*****
CapteurTemperatureExterieur.cpp
*****/
#include <string>
#include "CapteurTemperatureExterieur.h"

using namespace std;

CapteurTemperatureExterieur::CapteurTemperatureExterieur(string derniersChiffresNumeroSerie, double min, double max) :
    CapteurTemperature(derniersChiffresNumeroSerie, min, max)
{
};

bool CapteurTemperatureExterieur::VerifieConformite()
{
};

```

17 nov. 21 11:11

KOUADIO.cpp

Page 3/4

```

/*****
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"

//...

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

/*=====
Fonction principale 3/17
=====*/
int main()
{
    // Prise en compte des accents
    // Il faudra choisir la police de caractères Consolas ou Lucida
    SetConsoleOutputCP(1252);

    //==== Déclaration d'éventuelles variables locales ====
    int i;
    int Nombre;
    int Prototype;
    string Convert;
    //...

    // 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)

    for (i = 0; i <= 20; i++)
    {
        CapteurTemperature* LesCapteur[20];
        LesCapteur[i] = new CapteurTemperature(Convert, -20.5, 80.5);

        for (Nombre = stoi(LesCapteur[i]->getNoSerie()); Nombre <= 111120; Nombre++);

        Convert = to_string(Nombre);
        // Afficher 8 mesures par capteurs (20 lignes de 8 valeurs)
        Prototype = LesCapteur[i]->getTemperature();

        for (Prototype = LesCapteur[i]->getTemperature(); Prototype <= 8; Prototype++);
        if (i == 20)
        {
            cout << "CAPTEURS DE TEMPERATURE (MODELE DE BASE) " << LesCapteur[i]->getTemperature() <<
endl;
            cout.precision(4);
        }

    }

    // Ces capteurs de température ont des numéros de séries allant de "111101" à "111120"
    // de température mini -20.5 et température maxi +80.5
    // ...

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

    cout << "CAPTEURS DE TEMPERATURE (MODELE DE BASE) " << endl;
    cout.precision(4); // 4 digits de précision
    //...

    // Supprimer de la mémoire 10 capteurs

```

mercredi 17 novembre 2021

3/4

```
//...

// 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
//...

// 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;
//...

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

// Supprimer tous les capteurs
//...

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

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