

17/40 =&gt; 8,5 /20

LAIGLE.cpp

Page 1/3

```

/*****
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() {
        return this->type;
    };
    string getNoserie() { return this->noSerie; };
    static int 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
*****/ 3/4
#include "Capteur.h"

using namespace std;

static int nombreDeCapteurs = 0;

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

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

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

```

```

/*****
CapteurTemperature.cpp 1/7
*****/
#include "CapteurTemperature.h"

using namespace std;

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

double CapteurTemperature::getTemperature()
{
    double valTemp;
    valTemp = temperatureMaxAutorisee - temperatureMinAutorisee;
    valTemp = rand();

    return rand();
}

/*****
CapteurTemperatureExterieur.cpp 2/3
*****/
#include "CapteurTemperatureExterieur.h"

using namespace std;

CapteurTemperatureExterieur::CapteurTemperatureExterieur(string derniersChiffresNumeroSerie, double min, double max) : CapteurTemperature("33", min, max)
{
    this->noSerie = derniersChiffresNumeroSerie;
}

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 "CapteurTemperatureExterieur.h"
//...

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

/*=====
Fonction principale 2/17
=====*/
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 z=0;
    int j = 0;

```

```

// Affichage du nombre de capteurs
cout << "ETAPE1: NOMBRE DE CAPTEURS = " << z << " (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* C[20]; 1

for (unsigned i = 0; i < CapteurTemperature::getNombreCapteurs(); i++)
{
    C[i] = new CapteurTemperature ("111101", -20.5, 80.5); 1
    j++;
    cout << C[i]->getNserie() << " ";
}

// Affichage du nombre de capteurs
cout << "ETAPE2: NOMBRE DE CAPTEURS = " << " (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
//...

// Supprimer de la mémoire 10 capteurs
//...
delete C[10];

// Affichage du nombre de capteurs
cout << "ETAPE3: NOMBRE DE CAPTEURS = " << " (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
}

```