15/40 => 7,5/20

DARDARI.cpp


```
/**********************
Capteur.h
**********************/
#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 type; }
        string getNoSerie() { return noSerie;}
        static int getNombresCapteurs();
        ~Capteur();
};



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



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

int Capteur:: nombreDeCapteurs = 0;


Capteur::Capteur(string type, string noSerie)
{
        this->type = type;
        this->noSerie = noSerie;
}
int  Capteur::getNombresCapteurs()
{
        return nombreDeCapteurs;
}
Capteur::~Capteur()
{
        nombreDeCapteurs--;
}


/**********************
CapteurTemperature.cpp
```

5/5

2/2

2/2

3/4

```
**********************/
#include "CapteurTemperature.h"
```

1/7

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

double CapteurTemperature::getTemperature()
{
        double ValeurTemperature;
        ValeurTemperature = temperatureMaxAutorisee - temperatureMinAutorisee;
        ValeurTemperature = rand();
        return rand();
}

/**********************
CapteurTemperatureExterieur.cpp
**********************/
```

0/3

```
#include "CapteurTemperatureExterieur.h"
#include "CapteurTemperature.h"

CapteurTemperatureExterieur::CapteurTemperatureExterieur(string derniersChiffresNumeroSerie, double min, double max) :CapteurTemperature("33", max, min)
{
        noSerie = derniersChiffresNumeroSerie;
        temperatureMinAutorisee = min;
        temperatureMaxAutorisee = max ;
}

bool CapteurTemperatureExterieur::verifieConformite()
{
        if (temperatureMaxAutorisee > 60)
        {
                return true;
        }
        if (temperatureMinAutorisee < 50)
        {
                return true;
        }
        else
        {
                return false;
        }
}
```

Faux ! il faut que les deux conditions soient vraies pour retourner true

```
/**********************
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>

//...

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


/*============================================
Fonction principale
============================================*/
```

2/17

```
#include "Capteur.h"
#include "CapteurTemperature.h"
#include "CapteurTemperatureExterieur.h"
int main()
{
        // Prise en compte des accents
        // Il faudra choisir la police de caractères Consolas ou Lucida
        SetConsoleOutputCP(1252);

        //==== Déclaration d'éventuelles variables locales ====
        //...
        CapteurTemperature* capt;
```

```
        // Affichage du nombre de capteurs
        cout << "ETAPE1 : NOMBRE DE CAPTEURS = " << capt->getNombresCapteurs() << " (attendu 0)" << endl << endl;

        // Créer un tableau de 20 capteurs de température (modèles de base)
        // Ces  capteurs de temperature  ont des numéros de séries allant de "111101" à "111120"
        // de temperature mini −20.5 et temperature maxi +80.5
        // ...
        CapteurTemperature* capt2;

        int i,valeur,y;

        for (i = 0; i < 20; i++)
        {
                string strDebut = "1111";
                string strFin = to_string(valeur);
                capt2[i] = new CapteurTemperature(strDebut += strFin, y);
                y = y + 1;
                valeur = valeur + 1;
        }
        // Affichage du nombre de capteurs
        cout << "ETAPE2 : NOMBRE DE CAPTEURS = " <<capt2->getNombresCapteurs()<<" (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
        //...
        CapteurTemperature* capt3;

        for (i = 0; i < 10; i++)
        {
                delete capt3[i];
        }

        // Affichage du nombre de capteurs
        cout << "ETAPE3 : NOMBRE DE CAPTEURS = " <<capt3->getNombresCapteurs() << " (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 temperature mini −85.5 et temperature 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

        delete capt;

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

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

1 pt

1 pt