40/40 => 20/20 : Très bien

WOZNIAK.cpp


```
/**********************
Capteur.h
**********************/
```
5/5

```
#pragma once
#include <iostream>
#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();
};



/**********************
CapteurTemperatureExterieure.h
**********************/
```
2/2

```
#pragma once
#include "CapteurTemperature.h"
class CapteurTemperatureExterieure :
    public CapteurTemperature
{
public:
    CapteurTemperatureExterieure(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;
        nombreDeCapteurs++;
}
int Capteur::getNombreCapteurs() {
        return Capteur::nombreDeCapteurs;
}
Capteur::~Capteur() {
        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() {
	return (((rand() / RAND_MAX) * (this->temperatureMaxAutorisee - this->temperatureMinAutorisee)) + this->
temperatureMinAutorisee);
}

/**********************
CapteurTemperatureExterieure.cpp 3/3
**********************/
#include "CapteurTemperatureExterieure.h"
CapteurTemperatureExterieure::CapteurTemperatureExterieure(string derniersChiffresNumeroSerie, double min, doubl
e max) : CapteurTemperature("33"+derniersChiffresNumeroSerie, min, max) {

}

bool CapteurTemperatureExterieure::verifieConformite() {
	if (this->temperatureMinAutorisee < -50 && this->temperatureMaxAutorisee>60)
		return true;
	else
		return false;
}

/**********************
TestCapteurs_A_Completer.cpp 17/17
**********************/
/*********************************************
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 <time.h>
//on inclut la classe CapteurTemperatureExterieure
#include "CapteurTemperatureExterieure.h"

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


/*=============================================
Fonction principale
=============================================*/
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 nbCapteurs = Capteur::getNombreCapteurs();
	srand(time(NULL));
	string numero;
	// 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 temperature  ont des numéros de séries allant de "111101" à "111120"
	// de temperature mini -20.5 et temperature maxi +80.5
	CapteurTemperature *leCapteurDeBase[20]; //20 capteurs
	for (int i = 0; i < 10; i++) {
		numero = to_string(i+1);
		leCapteurDeBase[i] = new CapteurTemperature("11110"+numero, -20.5, 80.5);
	}
	for (int i = 10; i < 20; i++) {
		numero = to_string(i+1);
		leCapteurDeBase[i] = new CapteurTemperature("1111" + numero, -20.5, 80.5);
	}
	// 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 (int n = 0; n < 20; n++) {
		cout << "Capteur de numéro de série : " << leCapteurDeBase[n]->getNoSerie() << endl;
		for (int i = 0; i < 8; i++) {
			cout << leCapteurDeBase[i]->getTemperature() << " " ;
		}
		cout <<endl << endl;
	}

	// Supprimer de la mémoire 10 capteurs
	for (int j = 10; j < 20; j++) {
```

```
                delete leCapteurDeBase[j];
        }


        // 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 temperature mini −85.5 et temperature maxi +300.5
        // le dernier capteur a une température mini de −45.5 et maxi de 60.6
        CapteurTemperatureExterieure* leCapteurExterieur[12];
        for (int i = 0; i < 10; i++) {
                numero = to_string(i+1);
                leCapteurExterieur[i] = new CapteurTemperatureExterieure(+"550" + numero, −85.5, 300.5);
        }
        leCapteurExterieur[10] = new CapteurTemperatureExterieure(+"5511",−85.5,300.5);
        leCapteurExterieur[11] = new CapteurTemperatureExterieure(+"5512", −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 (int g = 0; g < 12; g++) {
                cout << "Capteur de Numero de Serie : " << leCapteurExterieur[g]−>getNoSerie();
                if (leCapteurExterieur[g]−>verifieConformite())
                        cout << "\t" << " CONFORME" << endl;
                else
                        cout << "\t" << " NON CONFORME" << endl;
                for (int i = 0; i < 8; i++) {
                        cout << leCapteurExterieur[i]−>getTemperature() << " ";
                }
                cout << endl << endl;
        }


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

        // Supprimer tous les capteurs
        for (int y = 0; y < 10; y++) {
                delete leCapteurDeBase[y];
        }

        for (int i = 0; i < 12; i++) {
                delete leCapteurExterieur[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
}
```