

19/40 => 10/20

LHEUREUX.cpp

Page 1/3

```

/*****
Capteur.h
*****/
#include <string>

#pragma once

using namespace std;

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

public:

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

};

```

```

/*****
CapteurTemperature.h
*****/
#pragma once
#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
*****/
#pragma once

#include <string>
#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;
}

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

```

```

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

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

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

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

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

double CapteurTemperature::getTemperature()
{
    return 0;
}

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

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

bool CapteurTemperatureExterieur::verifieConformite()
{
    if (temperatureMaxAutorisee < -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 6/17
*****/
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;

// Affichage du nombre de capteurs
cout << "ETAPE1:NOMBRE DE CAPTEURS=" << Capteur::getNombreCapteurs() 1 (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* capteur1[20]; 1

for (unsigned i = 0; i < 19; i++)
{
    capteur1[i] = new CapteurTemperature("", -20.5, 80.5); 1
}

// 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 < Capteur::getNombreCapteurs(); i++)
{
}

// Supprimer de la mémoire 10 capteurs
for (unsigned i = 0; i < 9; i++) 0,5
    delete capteur1[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* capteur2[12]; 1

for (unsigned i = 0; i < 11; i++)
{
    capteur2[i] = new CapteurTemperatureExterieur("", -85.5, 300.5); 1
}
for (unsigned i = 11; i < 12; i++)
    capteur2[i] = new CapteurTemperatureExterieur("", -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;
//...

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

// Supprimer tous les capteurs
for (unsigned i = 0; i < 12; i++) 0,5
    delete capteur2[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
}

```