

16 nov. 21 11:34

gestionEtudiants.cpp

Page 1/2

```

/*****
Etudiant.h
*****/
#pragma once
#include "personne.h"

class Etudiant :      public Personne
{
private:
    string classe;
    double moyenne;
    static int nombreEtudiants ;

public:
    Etudiant(string leNom , string lePrenom , string classe );
    ~Etudiant();
    string getClasse();
    void setMoyenne(double value);
    double getMoyenne();
    static int getNombreEtudiants();
};

/*****
Personne.h
*****/
#pragma once
#include <string>
using namespace std;

class Personne
{
protected :
    string nom , prenom ;
public:
    Personne(string leNom , string lePrenom);
    string getNom() {return nom ; }
    string getPrenom() {return prenom; }
};

/*****
Etudiant.cpp
*****/
#include "Etudiant.h"
int Etudiant::nombreEtudiants=0 ;

Etudiant::Etudiant(string leNom , string lePrenom , string laClasse ):Personne(leNom , lePrenom)
{
    this->classe = laClasse;
    nombreEtudiants++; //idem que => Etudiant::nombreEtudiants++;
}

Etudiant::~Etudiant()
{
    nombreEtudiants--; //idem que => Etudiant::nombreEtudiants--;
}

string Etudiant::getClasse()
{
    return this->classe; //idem que => return classe;
}

void Etudiant::setMoyenne(double value)
{
    this->moyenne=value; //idem que => moyenne = value;
}

double Etudiant::getMoyenne()
{
    return this->moyenne; //idem que => return moyenne;
}

int Etudiant::getNombreEtudiants()
{
    return nombreEtudiants; //idem que => return Etudiant::nombreEtudiants;
}

/*****
Personne.cpp
*****/
#include "Personne.h"
Personne::Personne(string leNom , string lePrenom)
{
    this->nom=leNom;          //idem que => nom = leNom;
    this->prenom=lePrenom;    //idem que => prenom = lePrenom;
}

```

```

/*****
ProgrammePrincipal.cpp
*****/
#include <iostream>      // bibliothèque pour la gestion des E/S
#include <vector>
#include "Etudiant.h"
using namespace std;

int main()
{
    // *** Solution 1 : colonne 5 algorithme sur la création des objets
    Etudiant *snir2[15];
    double moyenneGenerale = 0;

    snir2[0] = new Etudiant("DALTON", "Averell", "SNIR2");
    snir2[0]->setMoyenne(2.26);
    snir2[1] = new Etudiant("DALTON", "Jack", "SNIR2");
    snir2[1]->setMoyenne(7.34);
    snir2[2] = new Etudiant("DALTON", "William", "SNIR2");
    snir2[2]->setMoyenne(11.46);
    snir2[3] = new Etudiant("DALTON", "Joe", "SNIR2");
    snir2[3]->setMoyenne(15.24);

    cout << "===SOLUTION 1=== Nombre etudiants = " << Etudiant::getNombreEtudiants() << endl;

    //===== Affichage des étudiants =====
    for (unsigned i = 0; i < Etudiant::getNombreEtudiants(); i++)
    {
        cout << snir2[i]->getPrenom() << "\t"
              << snir2[i]->getNom() << "\t"
              << snir2[i]->getClasse() << "\t"
              << "Moyenne: " << snir2[i]->getMoyenne() << endl;

        moyenneGenerale += snir2[i]->getMoyenne();
    }

    moyenneGenerale /= Etudiant::getNombreEtudiants();
    cout << "MOYENNE GENERALE=" << moyenneGenerale << endl;

    // Suppression des étudiants
    int memoNombreEtudiants = Etudiant::getNombreEtudiants();
    for (unsigned i = 0; i < memoNombreEtudiants; i++)
        delete snir2[i];

    cout << "===SOLUTION 1=== Nombre etudiants = " << Etudiant::getNombreEtudiants() << endl << endl;

    // *** Solution 2 : Utilisation d'un vecteur
    vector<Etudiant * > lesSnir2;
    lesSnir2.push_back(new Etudiant("DALTON", "Averell", "SNIR2"));
    lesSnir2.push_back(new Etudiant("DALTON", "Jack", "SNIR2"));
    lesSnir2.push_back(new Etudiant("DALTON", "William", "SNIR2"));
    lesSnir2.push_back(new Etudiant("DALTON", "Joe", "SNIR2"));

    cout << "===SOLUTION 2=== Nombre etudiants = " << Etudiant::getNombreEtudiants() << endl;

    // Affectation des moyennes
    lesSnir2[0]->setMoyenne(2.26);
    lesSnir2[1]->setMoyenne(7.34);
    lesSnir2[2]->setMoyenne(11.46);
    lesSnir2[3]->setMoyenne(15.24);

    moyenneGenerale = 0;
    //===== Affichage des étudiants =====
    for (unsigned i = 0; i < Etudiant::getNombreEtudiants(); i++)
    {
        cout << snir2[i]->getPrenom() << "\t"
              << snir2[i]->getNom() << "\t"
              << snir2[i]->getClasse() << "\t"
              << "Moyenne: " << snir2[i]->getMoyenne() << endl;

        moyenneGenerale += snir2[i]->getMoyenne();
    }

    moyenneGenerale /= lesSnir2.size();
    cout << "MOYENNE GENERALE=" << moyenneGenerale << endl;

    // Suppression des étudiants
    for (unsigned i = 0; i < lesSnir2.size(); i++)
        delete lesSnir2[i];

    cout << "===SOLUTION 2=== Nombre etudiants = " << Etudiant::getNombreEtudiants() << endl;

    cin.get(); cin.ignore();
    return 0;
}

```