

© Copyright 2010 tv <thierry.vaira@orange.fr>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License,

Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover.

© You can obtain a copy of the GNU General Public License : write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

SOMMAIRE

Le Patron Vecteur.....	3
Introduction.....	3
Travail.....	3
Liens.....	3
Validation.....	4

LE PATRON VECTEUR

Introduction

Dans ce TP, nous nous proposons de développer à partir de la classe C++ nommée `Vecteur` créée durant le TP précédant, un patron de classe `Vecteur` permettant de traiter des vecteurs dont les éléments sont d'un type quelconque. La dimension reste modifiable en fonction des besoins de la classe, comme au TP précédant.

Travail

Généralisez votre classe `Vecteur` en un patron de classe `Vecteur` recevant en paramètre le type de ses éléments (qui au TP précédant étaient des `double`).

Une fois l'écriture du patron effectuée, vous le testerez en écrivant un programme qui calcule la somme d'une série de vecteur de `double` en dimension 5 se trouvant dans [ce fichier](#) (`test1.txt`) :

```
<5:1,2,3,4,5>  
<5:2,3,4,5,6><5:3,4,5,6,7>
```

Remarque : Attention, vous n'avez pas le droit de modifier le contenu du fichier.

Sur le même principe, vous calculerez le vecteur moyen de la série de vecteur de ce fichier (`test2.txt`) de `Vecteur` de `Ratio` en dimension 4 :

```
<5:1,2,3,4,5>  
<5:2,3,4,5,6><5:3,4,5,6,7>
```

Liens

- Page du cours de programmation C/C++ d'Éric REMY :

<http://www.iut-arles.up.univ-mrs.fr/eremy/Ens/Info1.C++/index.html>

- Planche de TP n°3 d'Éric REMY :

<http://pluton.up.univ-mrs.fr/eremy/Ens/LPSIL.C++/tp3/>

- Pour approfondir vos connaissances en C et C++ :

<http://www.iut-arles.up.univ-mrs.fr/eremy/Ens/Info1.C++/approfondir.html>

Validation

```

1 . testPVecteur.cc
#include <iostream>
#include <exception>
#include <iomanip>
#include <math.h>

using namespace std;

#include "PVecteur.h"

double fonction(double, double);

int main() {
    double t[5] = {1.0, 2.0, 3.0, 4.0, 5.0};
    PVecteur<double> v0, v1(4, 0.0), v2(5, t);

    cout.precision(2);
    //cout << "V0 = " << setprecision(1) << v0 << endl;
    cout << "V0 = " << v0 << endl;
    cout << "V1 = " << v1 << endl;
    cout << "V2 = " << v2 << endl;
    v0 = v2;
    cout << "V0 = " << v0 << endl;
    cout << "Norme de V0 = " << v0.norme() << endl;
    v0.normalise();
    cout << "V0 = " << v0 << endl;
    cout << "Norme de V0 = " << v0.norme() << endl;
    cout << "Entrez un vecteur : ";
    cin >> v0;
    if (!cin) {
        cout << "ERREUR de lecture\n";
        exit (-1);
    }
    cout << "V0 = " << v0 << endl;
    cout << "V0 * 2 = " << v0 * 2.0 << endl;
    cout << "2 * V0 = " << 2.0 * v0 << endl;
    cout << "-V0 = " << -v0 << endl;
    cout << "V0 + V1 = ";
    try {
        cout << (v0 + v1) << endl;
    }
    catch (Erreur & e) {
        cout << e.what() << endl;
    }
}

```

```
v0 = PVecteur<double>(3, t);
v2 = PVecteur<double>(3, t);
cout << "V0 = " << v0 << endl;
cout << "V2 = " << v2 << endl;
cout << "V0 ^ V2 = " << v0 << " ^ " << v2 << " = ";
try {
    cout << (v0 ^ v2) << endl;
}
catch (Erreur & e) {
    cout << e.what() << endl;
}
cout << "Application de sin -> V0" << endl;
cout << v0.appliquer1(sin) << endl;
cout << "Application de sin -> V0 et V2" << endl;
cout << appliquer2(fonction, v0, v2) << endl;

return 0;
}

double fonction(double x, double y) {
    return x / y;
}

2 . testFichiers.cc
#include <iostream>
#include <fstream>
#include <exception>
#include <iomanip>
#include <math.h>

using namespace std;

#include "PVecteur.h"
#include "Ratio.h"

int main() {
    PVecteur<double> v;
    PVecteur<Ratio> vR;
    ifstream *fichier = NULL;

    cout.precision(2);

    cout << "Lecture du premier fichier" << endl;
    /* TODO */

    cout << "Lecture du deuxième fichier" << endl;
    /* TODO */

    return 0;
}
```