

Cours de C++, 1ère année, HE-Arc

Série 7.2

Exercice 1 : Modèle de class

Écrire une classe générique *Vector* qui permet de manipuler des données de n'importe quel type. On utilisera un tableau par valeur.

```
Vector<float> myFloatVector;
```

L'instruction ci-dessus va créer une variable *myFloatVector* de 3 composantes de type float.

Pour créer un vector d'une taille différente de 3, la création se fera comme suit :

```
Vector<int, 4> myIntVector;
```

Indications

Il faudra prévoir un constructeur par copie, un opérateur = pour attribuer un vecteur à partir d'un autre de même type. Exemple :

```
myFloatVector2 = myFloatVector;
```

Une autre surcharge de l'opérateur = permettra de d'initialiser un vecteur avec une même valeur. Dans cet exemple, tous les éléments de *myFloatVector* auront la valeur 5.5.

```
myFloatVector = 5.5;
```

Il est également demandé d'implémenter l'opérateur [] pour protéger l'exécution contre les débordements de tableau.

L'opérateur de flux << sera également implémenté.

Reprenez la classe *Point*. Que faut-il modifier ?

Le programme ci-dessous montre un exemple d'utilisation :

```
int main()
{
    Vector<int, 4> vi = {0};
    vi[3] = 5;
    vi[2] = 2;
    cout << vi;

    Vector<double> vd; // 3 elements by default
    vd[0] = 0.0;
    vd[1] = 0.1;
    vd[2] = 0.2;
    cout << vd;
    cout << "\nvd[-8]-->out of range: " << vd[-8] << endl; //vd[0] changed
    cout << "\nvd[12]-->out of range: " << vd[12] << endl; //vd[2] changed
    cout << "vd[12] = 99.99" << endl;
    vd[12] = 99.99 ;// vd[12] doesn't exist, --> last element is modified
    cout << "vd[2]: " << vd[2] << endl;
    Vector<double> vd3;
    cout << vd;
    vd3 = vd;
    cout << vd3;

    Vector<Point,2> vpt = {Point (0.0, 0.0, "default")};
    Vector<Point,2> vpt2 = {Point (1.0, 2.0, "other")};
    cout << vpt;
    vpt = vpt2;
    cout << vpt;
    Vector<Point,2> vpt3 (vpt2);
    cout << vpt3;

    cout << "\nPlease hit ENTER to continue... ";
    cin.get();
    return 0;
}
```