

Série 3.2: Surcharge des opérateurs

Exercice 1: Classe Vecteur. Surcharge de << = et []

Implémentez une classe de vecteurs dynamique, dans le sens où les constructeurs doivent allouer dynamiquement la taille nécessaire. Les éléments sont de type `double`.

1. Implémentez les constructeurs

1. par défaut (pas d'allocation),
2. standard (le premier argument définit la taille et le second le contenu, par défaut le contenu est nul),
3. par recopie.

2. Surchargez les opérateurs suivants:

1. insertion de flux `operator << ()`,
2. affectation `operator = ()`. Il faut faire une copie en profondeur,
3. addition `operator + ()`. A déclarer comme fonction amie
4. accès `operator [] ()`. Il doit permettre:

- de lire un élément du vecteur `x=v[4];`
- de modifier un élément. `v[4]=4;`

Si l'indice est hors du vecteur, l'opérateur utilise la première case du vecteur `v[0]`.

```
int main()
{
    Vector v1;
    Vector v2(3);
    Vector v3= Vector(3,10);
    cout << "v1: " << v1 << " v2: " << v2 << " v3: " << v3 << endl << endl;

    cout << "v1: " << v1 << " v3: " << v3 << endl;
    cout << "v1=v3;" << endl;
    v1=v3;
    cout << "v1: " << v1 << " v3: " << v3 << endl << endl;

    cout << "v1: " << v1 << endl;
    cout << "v1[2]=100;" << endl;
    v1[2]=100;
    cout << "v1: " << v1 << endl << endl;

    cout << "v1[2] " << v1[2] << endl;
    cout << "v3[2] " << v3[2] << endl << endl;

    cout << "v1: " << v1 << endl;
    cout << "v1[1000]=200; " << endl;
    v1[1000]=200;
    cout << "v1: " << v1 << endl << endl;

    Vector v4(5,5);
    cout << "v1: " << v1 << " v3: " << v3 << " v4: " << v4 << endl;
    cout << "v1=v3+v4; " << endl;
    v1=v3+v4;
    cout << "v1: " << v1 << " v3: " << v3 << " v4: " << v4 << endl << endl;

    cout << "v1: " << v1 << " v2: " << v2 << " v3: " << v3 << endl;
    cout << "v1=v3+v3; " << endl;
    v1=v3+v3;
    cout << "v1: " << v1 << " v2: " << v2 << " v3: " << v3 << endl << endl;

    return 0;
}
```