

The background of the slide is a spiral-bound notebook with a light beige, textured cover. The spiral binding is visible on the left side, with the metal wire looping through a series of holes. The text is centered on the page.

Analyse et Programmation Orientées Objets / C++

Classe Vecteur – V 2.0.0

Header de la classe Vecteur (1)

```
# include <stdio.h>
# include <math.h>
# include <malloc.h>
const double EPSILON= 0.0001;
class Vecteur {
private :
double m_x;
double m_y;
public :
```

Header de la classe Vecteur (2)

// Constructeurs

//

Vecteur ();

Vecteur (double, double);

Header de la classe Vecteur (3)

// Accesseurs de consultation

//

inline double **abscisse** () {return m_x;}

inline double **ordonnee** () {return m_y;}

inline double **norme** () {return sqrt (m_x*m_x+m_y*m_y);}

Header de la classe Vecteur (4)

// Operateurs

//

bool operator == (const Vecteur&);

bool operator != (const Vecteur&);

Vecteur* operator + (const Vecteur&);

Vecteur* operator - (const Vecteur&);

Vecteur* operator - ();

Vecteur* operator * (double);

double operator * (const Vecteur&);

Header de la classe Vecteur (5)

// Operateurs (suite)

//

Vecteur& operator = (const Vecteur&);

Vecteur& operator += (const Vecteur&);

Vecteur& operator -= (const Vecteur&);

Vecteur& operator *= (double);

Header de la classe Vecteur (6)

```
// Services
```

```
//
```

```
bool colineaire (const Vecteur&);
```

```
bool orthogonal (const Vecteur&);
```

```
bool memeSens (const Vecteur&);
```

```
char* toString  ();
```

```
};
```

Constructeurs

```
# include "Vecteur.h "
```

```
Vecteur::Vecteur () {  
    m_x= 0;  
    m_y= 0;  
}
```

```
Vecteur::Vecteur (double x, double y) {  
    m_x= x;  
    m_y= y;  
}
```


Exemple d'opérateurs (1)

```
# include "Vecteur.h "
```

```
Vecteur* Vecteur::operator + (const Vecteur& V) {  
    double x, y;  
  
    x= abscisse () + V.abscisse ();  
    y= ordonnee () + V.ordonnee ();  
  
    return new Vecteur (x, y);  
}
```

Exemple d'opérateurs (2)

```
# include "Vecteur.h "
```

```
Vecteur* Vecteur::operator * (double k) {
```

```
double x, y;
```

```
    x= k*abscisse ();
```

```
    y= k*ordonnee ();
```

```
    return new Vecteur (x, y);
```

```
}
```

Exemple d'opérateurs (3)

```
# include "Vecteur.h "
```

```
Vecteur& Vecteur::operator = (const Vecteur& V) {
```

```
    m_x= V.m_x;
```

```
    m_y= V.m_y;
```

```
    return *this;
```

```
}
```

Exemple de services

```
# include "Vecteur.h "
```

```
bool Vecteur::colineaire (const Vecteur& V) {  
    double d1= abscisse()*V.ordonnee();  
    double d2= ordonnee()*V.abscisse();  
        return fabs(d1-d2) < EPSILON;  
}
```