



Engineering Applications General C++

Exercise

> 3. Class

1. Vector-Class

编写一个向量类:

类名: Vector;

数据成员:	int dim;	//向量的维数
	double * dataPtr;	//向量数据的指针
成员函数:	Vector();	//默认构造函数
	Vector(int N);	//构造函数
	Vector(Vector & v);	//复制构造函数
	~Vector();	//析构函数
	void print(int w = 8);	//输出向量, setw(w)

> 3. Class

1. Vector-Class

编写一个向量类:

类名: Vector;

成员函数:
(续)

```
Vector copy(Vector & v);    //复制
Vector copy(double x);      //所有元素用x赋值
double getData(int count);  //读取元素
int getSize();              //获取向量维数
Vector plus(Vector v);      //向量加法
Vector minue(Vector v);     //向量减法
double dotProduct(Vector v); //向量内积
Vector scale(double s);     //向量标量积(尺度变换)
```

> 3. Class

2. Matrix-Class

编写一个矩阵类：

类名： Matrix;

数据成员：	<code>int _rows;</code>	<code>//矩阵的行数</code>
	<code>int _cols;</code>	<code>//矩阵的列数</code>
	<code>double **dataPtr;</code>	<code>//矩阵数据的指针</code>
成员函数：	<code>Matrix();</code>	<code>//默认构造函数</code>
	<code>Matrix(int row, int col);</code>	<code>//构造函数</code>
	<code>Matrix(Matrix & m);</code>	<code>//复制构造函数</code>
	<code>~Matrix();</code>	<code>//析构函数</code>
	<code>void print(int w = 8);</code>	<code>//输出矩阵, setw(w)</code>

> 3. Class

2. Matrix-Class

编写一个矩阵类：

类名： Matrix;

成员函数：
(续)

```
Matrix copy(Matrix & m);    //复制
Matrix copy(double x);      //所有元素用x赋值
double getData(int m, int n); //读取元素
int getRows();              //获取行数
int getCols();              //获取列数
Matrix plus(Matrix & m);    //矩阵加法
Matrix minus(Matrix & m);   //矩阵减法
Vector product(Vector v);   //矩阵与向量相乘,
Matrix product(Matrix m);   //矩阵与矩阵相乘,
                             注意两种乘法相乘的前提
```

> 3. Class

3. Vector & Matrix Test

在主程序中进行以下测试：

```
Matrix A(3,5);  
A.copy(5.);  
Matrix B (A);  
Vector v1(5);  
  
v1.copy(2.);  
v1.print();  
  
Vector v2 = B.product(v1);  
v2.print();  
  
cout << "Matrix A: " << endl;  
A.print();  
cout << "Matrix B: " << endl;  
B.print();
```