

# 组合类练习

## 总代码

将求两直线的夹角函数定义在了类体内部，将求两点距离的函数定义在了外部，作为独立的函数

```
#include<iostream>
#include<cmath>
using namespace std;

class Point{
private:
    double X,Y;
public:
    Point(double x, double y){
        X=x;
        Y=y;
    }
    Point(){
        X=0;
        Y=0;
    }
    double getPointX(){
        return X;
    }
    double getPointY(){
        return Y;
    }
};

class Line{
private:
    Point p1,p2;
public:
    Line(Point P1,Point P2){
        p1=P1;
        p2=P2;
    }
    double angle(Line l){
        double ang;
        double tanang,k1,k2;
        k1=(p1.getPointY()-p2.getPointY())*1.0/(p1.getPointX()-p2.getPointX());
        k2=(l.p1.getPointY()-l.p2.getPointY())*1.0/(l.p1.getPointX()-
l.p2.getPointX());
        tanang=(k1-k2)/(1+k1*k2);
        ang=atan(tanang);
        return ang;
    }
};

double dis(Point a,Point b){
    double d;
```

```

        d=sqrt(pow(a.getPointX()-b.getPointX(),2)+pow(a.getPointY()-
b.getPointY(),2));
        return d;
    }
    int main()
    {
        Point a(1,1),b(2,2);
        cout<<dis(a,b)<<endl;
        Line l1(a,b);
        Point c(3,5),d(4,7);
        Line l2(c,d);
        cout<<l1.angle(l2)<<endl;

        return 0;
    }

```

## 定义点类

```

class Point{
private:
    double X,Y;
public:
    Point(double x, double y){
        X=x;
        Y=y;
    }
    Point(){
        X=0;
        Y=0;
    }
    double getPointX(){
        return X;
    }
    double getPointY(){
        return Y;
    }
};

```

## 定义直线类(含求夹角函数)

```

class Line{
private:
    Point p1,p2;
public:
    Line(Point P1,Point P2){
        p1=P1;
        p2=P2;
    }
    double angle(Line l){
        double ang;
        double tanang,k1,k2;
        k1=(p1.getPointY()-p2.getPointY())*1.0/(p1.getPointX()-p2.getPointX());
        k2=(l.p1.getPointY()-l.p2.getPointY())*1.0/(l.p1.getPointX()-
l.p2.getPointX());

```

```
        tanang=(k1-k2)/(1+k1*k2);  
        ang=atan(tanang);  
        return ang;  
    }  
};
```

## 求距离函数

```
double dis(Point a,Point b){  
    double d;  
    d=sqrt(pow(a.getPointX()-b.getPointX(),2)+pow(a.getPointY()-  
b.getPointY(),2));  
    return d;  
}
```