

```
import java.util.Scanner;           // 单类型导入声明
import java.util.*;                 // 按需类型导入声明
import static java.lang.Math.PI;    // 单静态导入声明
import static java.lang.Math.*;     // 按需静态导入声明
```

Chap11

Point2DTester.java

//--- 测试二维坐标类 ---//

```
import point.Point2D;

public class Point2DTester {

    public static void main(String[] args) {
        Point2D p1 = new Point2D();
        Point2D p2 = new Point2D(10, 15);

        System.out.println("p1 = " + p1);
        System.out.println("p2 = " + p2);
    }
}
```

Chap11/Point2DTester.java

运行结果

```
p1 = (0,0)
p2 = (10,15)
```

可以通过简名使用包
point 中的类 Point2D

Point2DTester.class

运行这个类时，会从子目录 point 的
Point2D.class 中读入类 Point2D

point

包 point 的目录

Point2D.java

//--- 二维坐标类 ---//

package point; * 这个源程序中声明的类属于包 point

```
public class Point2D {
    private int x = 0;           // X坐标
    private int y = 0;           // Y坐标

    public Point2D() { }
    public Point2D(int x, int y) { this.x = x; this.y = y; }
    public Point2D(Point2D p) { this(p.x, p.y); }

    public String toString() { return "(" + x + ", " + y + ")"; }
}
```

Chap11/point/Point2D.java

Point2D.class

x

x/P.java

```
package x;
class P { 简名
    void f(X a) {
        OK
        错误
    }
}
```

x/X.java

```
package x;
public class X {
    public void m1() { }
    protected void m2() { }
    void m3() { }
    private void m4() { }

    void f(X a) {
        OK
    }
}
```

y

z/Q.java

```
package z;
import x.X; 完全限定名
class Q {
    void f(X a) {
        OK
        错误
    }
}
```