**public** **class** ValuePassing {

**public** **static** **void** main(String[] args) {

Point2D point1=**new** Point2D(100,200);

Point2D point2=**new** Point2D(200,300);

Point2D point3=**new** Point2D(400,500);

System.*out*.println("before chaning:");

System.*out*.println("str1是:"+point1);

System.*out*.println("str2是:"+point2);

System.*out*.println("str3是:"+point3);

*change*(point1,point2,point3);

System.*out*.println("after chaning:");

System.*out*.println("str1是:"+point1);

System.*out*.println("str2是:"+point2);

System.*out*.println("str3是:"+point3);

}

**public** **static** **void** change(Point2D p1,Point2D p2,Point2D p3) {

int x = 3333;

p1.setX(x);

p1= **new** Point2D(7777,8888);

}

}

Java所有形参与实参的结合方式都为**值传递**：

例如：

形参：p1, p2, p3

实参：point1，point2，point3

形参：newX

实参：x

值传递图示：

赋值

p1

point1

赋值

newX

3333

通过值传递后，对于基本数据类型的变量而言，函数内部所有改变形参值的行为与实参毫无瓜葛。

通过值传递后，对于对象变量而言:

1. 函数内部所有改变形参值的行为与实参毫无瓜葛；例如：p1本身存的是指向对象的地址，p1的赋值本身发生任何改变，都不影响实参point1。
2. 函数内部所有改变形参所指对象的行为就是改变实参所指对象的行为；例如：通过p1改变p1指向对象的行为，就会影响实参的对象本身,p1.setX(x)是改变了对象变量所指向的对象的属性值。