

```
1. public class A {
2.     public String show(D obj) {
3.         return ("A and D");
4.     }
5.
6.     public String show(A obj) {
7.         return ("A and A");
8.     }
9.
10. }
11.
12. public class B extends A{
13.     public String show(B obj){
14.         return ("B and B");
15.     }
16.
17.     public String show(A obj){
18.         return ("B and A");
19.     }
20. }
21.
22. public class C extends B{
23.
24. }
25.
26. public class D extends B{
27.
28. }
29.
30. public class Test {
31.     public static void main(String[] args) {
32.         A a1 = new A();
33.         A a2 = new B();
34.         B b = new B();
35.         C c = new C();
```

```

36.         D d = new D();
37.
38.         System.out.println("1--" + a1.show(b));
39.         System.out.println("2--" + a1.show(c));
40.         System.out.println("3--" + a1.show(d));
41.         System.out.println("4--" + a2.show(b)); //4--B and A .首先a2是A引/
用, B实例, 调用show ( B b ) 方法, 此方法在父类A中没有定义, 所以B中方法
show(B b)不会调用 ( 多态必须父类中已定义该方法 ), 再按优先级为 :
this.show(O)、super.show(O)、this.show((super)O)、super.show((super)O) ,
即先查this对象的父类, 没有重头再查参数的父类。查找super.show((super)O)时,
B中没有, 再向上, 找到A中show(A a),因此执行。
42.
43.         System.out.println("5--" + a2.show(c)); //同上
44.         System.out.println("6--" + a2.show(d)); //A and D .查找B中没有
show(D d)方法, 再查A中, 有, 执行。
45.         System.out.println("7--" + b.show(b));
46.         System.out.println("8--" + b.show(c)); //B and B .
47.         System.out.println("9--" + b.show(d));
48.     }
49. }

```

运行结果:

1. 1--A and A
2. 2--A and A
3. 3--A and D
4. 4--B and A
5. 5--B and A
6. 6--A and D
7. 7--B and B
8. 8--B and B
9. 9--A and D

