

接口常用来实现抽象,多继承，松耦合。

## 接口

- 接口用interface来修饰
- 接口没有构造方法
- 接口不能用于实例化对象
- 接口不是被类继承了，而是要被类实现
- 接口可以继承接口，支持多继承
- 接口中方法默认为public abstract，jdk1.8以后可以使用静态方法及方法

## Test.java

```
1 interface A {
2     void printA(); //接口的方法默认指定为public abstract（抽象方法），变量默认指定为public static final
3
4     static int add(int value) { //jdk1.8以后有静态方法
5         return value + value;
6     }
7 }
8
9 class TestA implements A {
10     public void printA() { //
11         System.out.println("test A implements A");
12     }
13 }
14 /* test b */
15 interface B {
16     void printB();
17 }
18
19 class TestB implements A,B { //接口不能被类继承，而是被类实现
20     public void printA() { //方法体必须加public修饰，不然编译报错
21         System.out.println("test B implements A");
22     }
23     public void printB() {
24         System.out.println("test B implements B");
25     }
26 }
27 /* test c */
28 interface C extends A,B { //接口可以继承接口，支持多继承
29     void printA();
30     void printB();
31     void printC();
32 }
33
34 class TestC implements C {
35     public void printA() {
36         System.out.println("test C implements C extends A");
37     }
38     public void printB() {
```

```

39     System.out.println("test C implements C extends B");
40 }
41 public void printC() {
42     System.out.println("test C implements C");
43 }
44 }
45
46 public class Test {
47     public static void main(String []args) {
48         System.out.println("----A----");
49         //A testa = new A(); //接口不能实例化对象，没有构造方法
50         A testa = new TestA();
51         testa.printA();
52         //System.out.println(testa.add(10)); //不能被类实现
53         System.out.println(A.add(10)); //只能通过接口实现
54         System.out.println("----B----");
55         TestB testb = new TestB();
56         testb.printA();
57         testb.printB();
58         System.out.println("----C----");
59         TestC testc = new TestC();
60         testc.printA();
61         testc.printB();
62         testc.printC();
63     }
64 }

```

## 运行结果

```

jzzh@jizizihe:~/workspace/java/interface$ java Test
----A----
test A implements A
20
----B----
test B implements A
test B implements B
----C----
test C implements C extends A
test C implements C extends B
test C implements C

```