

多态为java的三大特性之一，了解学习多态，将更了解java。

多态实现条件

- 继承与重写（方法的多态性）
- 接口（对象的多态性）
- 抽象类与抽象方法（对象的多态性）

向上转型和向下转型

- 向上转型：父类 父类对象 = 子类实例
- 向下转型：子类 子类对象 = （子类）父类对象
- 当父类需要调用子类的扩充方法时才需要向下转型
- 向下转型之前必须要先向上转型

Test.java

```
1  abstract class A{//抽象类的多态
2      abstract void printA();
3  }
4
5  class TestA extends A {
6      void printA() {
7          System.out.println("class TestA extends abstract A");
8      }
9      void printTestA() {
10         System.out.println("printTestA methods");
11     }
12 }
13
14 interface B{//接口的多态
15     void printB();
16 }
17
18 class TestB implements B {
19     public void printB() {
20         System.out.println("class TestB implements B");
21     }
22     void printTestB() {
23         System.out.println("printTestB methods");
24     }
25 }
26
27 public class Test {
28     public static void main(String []args){
29         System.out.println("-----A-----");
30         A a = new TestA();//向上转型：父类 父类对象 = 子类实例
31         a.printA();
32         //testa.printTestA();//会报错，向上转型，因为父类访问不了子类中方法
33         TestA testa = (TestA)a;//向下转型：子类 子类对象 = （子类）父类实例，向下转型和向上转型是一体的
34         testa.printTestA();
35         System.out.println("-----B-----");
36         TestB testb = new TestB();//不想用向上向下转型可直接用子类实现
```

```
37     testb.printB();  
38     testb.printTestB();  
39 }  
40 }
```

运行结果

```
jzzh@jizizihe:~/workspace/java/polymorphic$ java Test  
-----A-----  
class TestA extends abstract A  
printTestA methods  
-----B-----  
class TestB implements B  
printTestB methods
```