

内部类：一个类的定义放在另一个类的定义内部。

## 内部类

- 普通内部类：普通内部类对象需要依赖外部类对象
- 静态内部类：静态内部类对象不需要依赖外部类对象
- 匿名内部类：主要是针对抽象类和接口，重写其方法。
- 局部内部类：声明在方法体内，不在方法体内无法使用其类。

### 普通内部类 TestOne.java

```
1 public class TestOne {
2     public int test = 1;
3     private int test2 = 2;
4
5     public TestOne() {
6         InnerOne innerObj = new InnerOne();
7         //外部类对象也可以通过内部类的对象引用来访问内部类中定义的所有访问权限的字段
8         System.out.println("inner="+innerObj.inner);
9         System.out.println("inner2="+innerObj.inner2);
10    }
11
12    public class InnerOne { //普通内部类：依赖内部类TestOne
13
14        public int inner = 3;
15        private int inner2 = 4;
16        //内部类对象可以访问外部类对象中所有访问权限的字段.
17        public InnerOne() {
18            System.out.println("test="+test);
19            System.out.println("test2="+test2);
20        }
21    }
22    public static void main(String []args) {
23        TestOne testObj = new TestOne();
24        /* static是独立于类的任何一个对象。所以不在外部类TestOne时，
25        指名外部类对象testObj，再使用内部类InnerOne */
26        /*
27        InnerOne innerObj = testObj.new InnerOne();
28        System.out.println("inner="+innerObj.inner);
29        System.out.println("inner2="+innerObj.inner2);
30        */
31    }
32 }
```

### 运行结果

```
jzzh@jizizihe:~/workspace/java/innerclass$ javac TestOne.java
jzzh@jizizihe:~/workspace/java/innerclass$ java TestOne
test=1
test2=2
inner=3
inner2=4
```

### 静态内部类 TestTwo.java

```

1 public class TestTwo {
2     public static int test = 1;
3     private static int test2 = 2; //必须是静态变量：静态变量无法访问外部类变量
4
5     public TestTwo() {
6         /* 静态内部类innerObj对象不需要依赖外部类对象testObj */
7         /* InnerTwo innerObj = new InnerTwo(); */
8         //System.out.println("inner="+innerObj.inner);
9         /*System.out.println("inner2="+innerObj.inner2);*/
10    }
11
12    public static class InnerTwo { //静态内部类
13        public int inner = 3;
14        private static int inner2 = 4; //可使用静态变量
15
16        public InnerTwo() {
17            System.out.println("test="+test);
18            System.out.println("test2="+test2);
19        }
20    }
21    public static void main(String []args) {
22        TestTwo testObj = new TestTwo();
23        InnerTwo innerObj = new InnerTwo();
24        System.out.println("inner="+innerObj.inner);
25        System.out.println("inner2="+innerObj.inner2);
26    }
27 }

```

## 运行结果

```

jzzh@jizizihe:~/workspace/java/innerclass$ javac TestTwo.java
jzzh@jizizihe:~/workspace/java/innerclass$ java TestTwo
test=1
test2=2
inner=3
inner2=4

```

## 匿名内部类 TestThree.java

```

1 public class TestThree {
2     public int test = 1;
3     private int test2 = 2;
4
5     public void printTest() {
6         InnerThree inner = new InnerThree() { //new创建匿名内部类实现接口的方法,然后赋值给Inner对象
7             int innerValue = 3; //只能在匿名内部类使用
8             public void printThree() {
9                 System.out.println("---override printThree method---");
10                System.out.println("test="+test);
11                System.out.println("test2="+test2);
12            }
13        };
14        inner.printThree(); //执行重写的方法
15        //System.out.println("inner="+inner.innerValue); //无法使用匿名内部类的变量,因为外部类无法获取当前匿名内部类的类名。

```

```

16     }
17
18     interface InnerThree { //匿名内部类：主要是针对抽象类和接口
19
20         void printThree();
21     }
22
23     public static void main(String []args) {
24         TestThree testObj = new TestThree();
25         testObj.printTest();
26     }

```

运行结果

```

jzzh@jizizihe:~/workspace/java/innerclass$ javac TestThree.java
jzzh@jizizihe:~/workspace/java/innerclass$ java TestThree
---override printThree method---
test=1
test2=2

```

局部内部类 TestFour.java

```

1 public class TestFour {
2     public int test = 1;
3     private int test2 = 2;
4
5     public void printTest() {
6         class InnerFour { //局部内部类：声明在一个方法体内部。只能在方法体内部使用。
7
8             public int innerValue = 3;
9             private int innerValue2 = 4;
10
11             public InnerFour() {
12                 System.out.println("test="+test);
13                 System.out.println("test2="+test2);
14             }
15         }
16
17         InnerFour inner = new InnerFour();
18         System.out.println("innerValue="+inner.innerValue);
19         System.out.println("innerValue2="+inner.innerValue2);
20     }
21
22     public static void main(String []args) {
23         TestFour testObj = new TestFour();
24         testObj.printTest();
25     }

```

运行结果

```
jzzh@jizizihe:~/workspace/java/innerclass$ javac TestFour.java
jzzh@jizizihe:~/workspace/java/innerclass$ java TestFour
test=1
test2=2
innerValue=3
innerValue2=4
```