

## 1、课程名称：instanceof 关键字



## 2、本课程预计讲解的知识点

MLDN

本章目标

- 掌握instanceof关键字的作用及使用时机
- 可以使用instanceof对对象的转型进行安全验证

E-MAIL: [mldnqa@163.com](mailto:mldnqa@163.com) [www.MLDNJAVA.cn](http://www.MLDNJAVA.cn)

## 3、具体内容

在 JAVA 中可以使用 instanceof 关键字判断一个对象到底是那个类的实例

MLDN

instanceof 关键字

- 在java中可以使用instanceof关键字判断一个对象到底是那个类的实例
  - 对象 instanceof 类 → 返回boolean类型

```
E-MAIL: mldnqa@163.com www.MLDNJAVA.cn

class A{
    // 定义类 A
    public void fun1(){
        // 定义 fun1()方法
        System.out.println("A --> public void fun1()");
    }
    public void fun2(){
        this.fun1();
        // 调用 fun1()方法
    }
};
class B extends A{
    // 此方法被子类覆写了
    public void fun1(){
        System.out.println("B --> public void fun1()");
    }
    public void fun3(){
        System.out.println("B --> public void fun3()");
    }
};
```

```
};
public class InstanceofDemo01{
    public static void main(String asrgs[]){
        A a1 = new B(); // 通过向上转型实例化对象
        System.out.println("A a1 = new B() : " + (a1 instanceof A));
        System.out.println("A a1 = new B() : " + (a1 instanceof B));
        A a2 = new A(); // 通过向上转型实例化对象
        System.out.println("A a2 = new B() : " + (a2 instanceof A));
        System.out.println("A a2 = new B() : " + (a2 instanceof B));
    }
};
```

使用 instanceof 关键字可以做什么呢？

从之前在对象多态性中讲解的代码可以发现一个问题。

如果现在传入的实例是 B 类的实例，则要调用 fun3()方法，如果是 C 类，则要调用 fun5()方法。

```
class A{
    // 定义类 A
    public void fun1(){
        // 定义 fun1()方法
        System.out.println("A --> public void fun1()");
    }
    public void fun2(){
        this.fun1();
        // 调用 fun1()方法
    }
};
class B extends A{
    // 此方法被子类覆写了
    public void fun1(){
        System.out.println("B --> public void fun1()");
    }
    public void fun3(){
        System.out.println("B --> public void fun3()");
    }
};
class C extends A{
    // 此方法被子类覆写了
    public void fun1(){
        System.out.println("C --> public void fun1()");
    }
    public void fun5(){
        System.out.println("C --> public void fun5()");
    }
};
```

```
public class InstanceofDemo02{
    public static void main(String asrgs[]){
        fun(new B());
        fun(new C());
    }
    public static void fun(A a){
        a.fun1();
        if(a instanceof B){
            B b = (B) a;
            b.fun3();
        }
        if(a instanceof C){
            C c = (C) a;
            c.fun5();
        }
    }
};
```

在开发中一定要注意，对于向下转型操作最好增加验证，以保证转型时不会发生 ClassCastException。

如果现在要增加新的子类，则肯定要修改 fun()方法，这样一来程序就失去了灵活性，所以在程序的开发中重点的设计应该放在父类上，只要父类设计的足够合理，则开发肯定会非常的方便。

而且在开发中一定要记住以下一点：

一个类永远不要去继承一个已经实现好的类。而只能继承抽象类或实现接口。

## 4、总结

- instanceof 用于判断一个对象是否是某个类的实例
- 在对象向下转型之前最好使用 instanceof 关键字进行验证

## 5、预习任务

MLDN

下一章内容

E-MAIL: [mldnqa@163.com](mailto:mldnqa@163.com) [www.MLDNJAVA.cn](http://www.MLDNJAVA.cn)