

Java深入:认识classClass及其应用(2) PDF转换可能丢失图片或格式, 建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Java_E6_B7_B1_E5_85_A5_c104_144847.htm

2. 用Class对象进行类型比较

2.1. 直接比较两个Class对象不论是通过equals()函数还是直接用==运算符进行比较, 比较的都是类型是否相同。2.2. 通过Class.isInstance(object)函数进行比较, 如class Cat{} class

```
class Dog{} class Duck{} public class Test{ public static void  
main(String[] args){ Class c2, c3. Duck d = new Duck(). c2 =  
Dog.class. c3 = d.getClass(). System.out.println("c2.isIntance(d) : "  
(c2.isInstance(d))). System.out.println("c3.isIntance(d) : "  
(c3.isInstance(d))). } }
```

运行结果为: c2.isIntance(d) : false c3.isIntance(d) : true

2.3. 通过instanceof关键字进行比较。但是两个处于不同的继承体系中的类对象和Class对象进行比较, 会产生编译错误。class Cat{} class Dog{} class Duck{} public class Test{ public static void main(String[] args){ Class c3. //Object d = new Duck(). (1) Duck d = new Duck(). // (2) c3 = d.getClass(). System.out.println("d instanceof Dog : " (d instanceof Dog)). // (3) } }

由于Duck和Dog处于两个不同的继承体系中, 所以代码(3)会发生编译错误。如果把代码(2)注释掉, 并去掉代码(1)的注释, 编译将通过。这是因为所有class都是继承自Object的, 所以Object与Dog处于同一个继承体系中, 可以进行比较。

2.4. 综合实例class Base{} class Derived{} public class Test{ public static void test(Object x){ System.out.println("Testing x of type " x.getClass()). System.out.println("x instanceof Base " (x instanceof Base)).

```
System.out.println("x instanceof Derived " (x instanceof Derived)).
System.out.println("Base.isInstance(x) " Base.class.isInstance(x)).
System.out.println("Derived.isInstance(x) "
Derived.class.isInstance(x)). System.out.println("x.getClass() ==
Base.class " (x.getClass() == Base.class)).
System.out.println("x.getClass() == Derived.class " (x.getClass() ==
Derived.class)). System.out.println("x.getClass().equals(Base.class) "
(x.getClass().equals(Base.class))).
System.out.println("x.getClass().equals(Derived.class) "
(x.getClass().equals(Derived.class))). } public static void
main(String[] args){ test(new Base()). test(new Derived()). } } 运行
结果为 : Testing x of type class Basex instanceof Base true x
instanceof Derived false Base.isInstance(x) true Derived.isInstance(x)
false x.getClass() == Base.class true x.getClass() == Derived.class
false x.getClass().equals(Base.class)
true x.getClass().equals(Derived.class) false Testing x of type class
Derived x instanceof Base false x instanceof Derived
true Base.isInstance(x) false Derived.isInstance(x) true x.getClass() ==
Base.class false x.getClass() == Derived.class
true x.getClass().equals(Base.class)
false x.getClass().equals(Derived.class) true 100Test 下载频道开通
, 各类考试题目直接下载。 详细请访问 www.100test.com
```