

java初学者实践教程13 - 面向对象之多态 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/270/2021_2022_java_E5_88_9D_E5_AD_A6_c67_270390.htm 上节课我们了解了比较重要的概念面向对象，和java的OOP有封装、继承、多态的特征。但是什么叫做多态，是很多初学者不容易理解的问题。对于继承来说，很容易理解因为你就看字面的意思就知道它是继承着父类的特性。多态字面不容易理解了。下面我们具体讲一下吧！类之间的继承关系使子类具有父类的所有变量和方法，=> 父类所具有的方法也可以在它所有子类中使用，发给父类的消息也可以发送给子类 => 子类的对象也是父类的对象=>子类的对象既可以做本身的类型，也可以做父类的类型。呵呵，上述推导公式好像绕口令似的。我们举个例子理解上述概念。举例：`public class 动物 //动物是父类` `public class 猫 extends 动物 //猫是子类` 动物的所有特性在猫中可以使用，发给动物的信息猫也能收到=>猫的对象`new 猫（）`；既可以作为本身的类型 `猫 a = new 猫（）`；也可以作为父类的类型 `动物 b = new 猫（）`；这样说理解了吗？如有疑问请访问我们的技术论坛。如果大家明白了的话，我们就可以从上述公式推导出结论，所有的子类都可以作为父类的类型（同一种类型）来对待。像刚才那个动物有很多子类啊，可以有很多对象。`动物 a=new 猫（）`；`动物 b=new 狗（）`；`动物 c=new 猪（）`；。这样的将子类型的对象引用转换成父类型的对象引用，叫做上溯造型（upcasting）。我们再来引伸一下，我们在数组那节课里讲了，数组存放的元素是相同类型的数据，但是上溯造型使得java允许创建不同类型对象的数组。例如：

Employee[] staff = new Employee[3] ; staff[0] = new Manager () ; staff[1] = new Secretary () ; staff[2] = new Employee () ; 这是怎么回事啊，数组里面不是相同类型吗？对啊，因为Secretary和Manager是Employee的子类，所以也可以通过上溯造型变成Employee啊。以前我们还学到了所有对象都是从java.lang.Object 继承下来的。如果数组要是 Object型的话 Object[] obj=new Object[]；那就是里面放什么对象都行了。因为什么对象都可以是Object型的。 实践：// java中的多态class Shape { void draw() {} void erase() {} }//圆形class Circle extends Shape { void draw() { System.out.println("Circle.draw()"). } void erase() { System.out.println("Circle.erase()"). }}//正方形class Square extends Shape { void draw() { System.out.println("Square.draw()"). } void erase() { System.out.println("Square.erase()"). }}//三角形class Triangle extends Shape { void draw() { System.out.println("Triangle.draw()"). } void erase() { System.out.println("Triangle.erase()"). }}public class Shapes { public static Shape randShape() { switch((int)(Math.random() * 3)) { default: case 0: return new Circle(). case 1: return new Square(). case 2: return new Triangle(). } } public static void main(String[] args) { Shape[] s = new Shape[9]. // 向数组里添加类型 for(int i = 0. i < s.length. i++) s[i] = randShape(). // 用多态的方法调用 for(int i = 0. i < s.length. i++) s[i].draw(). }}

Java的多态性，有什么意义呢？它的突出优点是使程序具有良好的扩展性。它通过继承，可以派生出任意多个新类型，或向基类增加更多方法时，无须修改原有对基础类进行处理的相关程序。就是扩展性好。我们返回再看面向对象（专

指OOP)的这三个特性封装、继承、多态三者的关系。没有封装就没有继承,没有继承就没有多态。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