

目录

1. 探探对 java 多态的理解..... 2
2. Java 中实现多态的机制是什么？（源码）3

1. 探探对 java 多态的理解

面向对象的三大基本特征：**封装、继承、多态**

多态是指：父类引用指向子类对象，在执行期间判断所引用对象的实际类型，根据其实际的类型调用其相应的方法。（**同一消息可以根据发送对象的不同而采用多种不同的行为方式。**）

多态的作用：消除类型之间的耦合关系。

实现多态的技术称为：动态绑定（dynamic binding），是指在执行期间判断所引用对象的实际类型，根据其实际的类型调用其相应的方法。

实现多态的三要素：继承，重写，父类引用指向子类对象（即，声明是父类，实际指向的是子类的一个对象）

```
void doSomething(Shape shape) {  
    shape.draw();  
    .....  
    shape.erase();  
}  
  
Circle circle = new Cricle();  
Traingle traingle = new Traingle();  
Line line = new Line();  
doSonething(circle);
```

```
doSomething(triangle);
```

```
doSomething(line);
```

2. Java 中实现多态的机制是什么？（源码）

多态就是指一个引用变量到底会指向哪个类的实例对象，该引用变量发出的方法调用到底是哪个类中实现的方法，必须在由程序运行期间才能决定。

因为在程序运行时才确定具体的类，这样，不用修改源程序代码，就可以让引用变量绑定到各种不同的类实现上，从而导致该引用调用的具体方法随之改变，即不修改程序代码就可以改变程序运行时所绑定的具体代码，让程序可以选择多个运行状态，这就是多态性。

特点：

1) 指向子类的父类引用由于向上转型了，它只能访问父类中拥有的方法和属性，而对于子类中存在而父类中不存在的方法，该引用是不能使用的，尽管是重载该方法。

2) 若子类重写了父类中的某些方法，在调用该方法的时候，必定是使用子类中定义的这些方法（动态连接、动态调用）。

Java 实现多态有三个必要条件：继承、重写、向上转型。

调用的优先级方法，该优先级为：`this.show(0)`、`super.show(0)`、`this.show((super)0)`、`super.show((super)0)`。

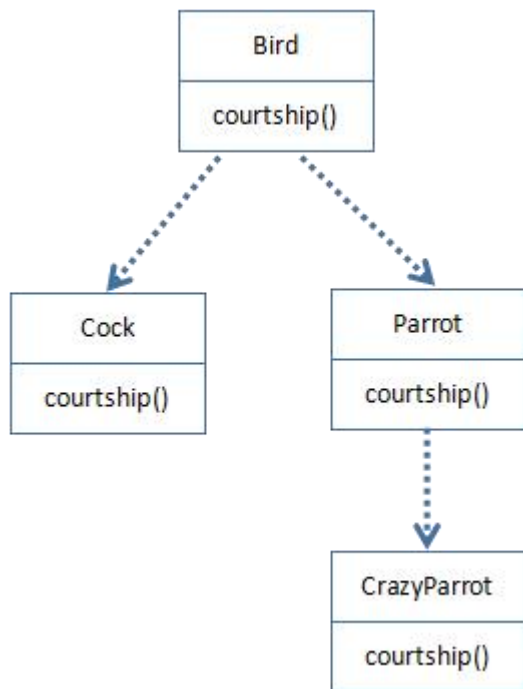
多态的实现原理

Java 里对象方法的调用是依靠类信息里的方法表实现的。

总体而言，当调用对象某个方法时，JVM 查找该对象类的方法表以确定该方法的直接引用地址，有了地址后才真正调用该方法。

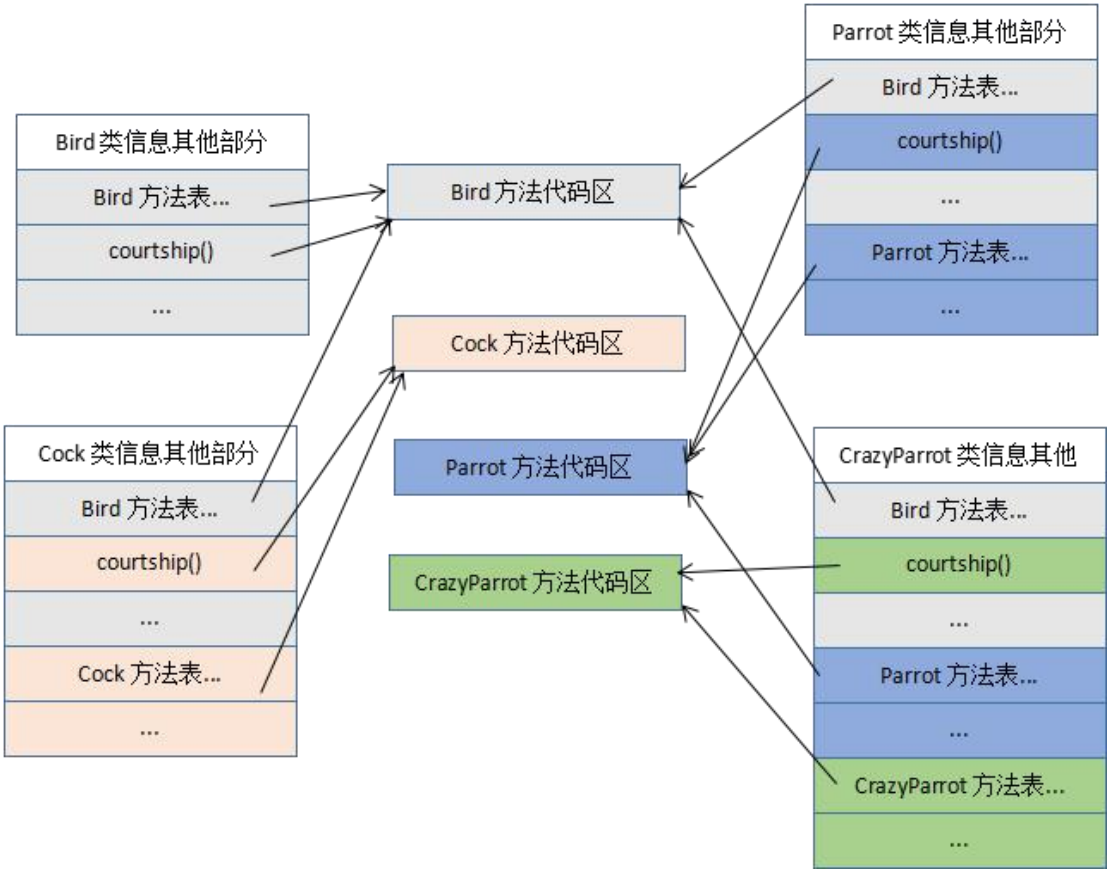
超类继承父类的方法，如果不 Overriding 该方法，那么调用时会指向父类的方法。如果 Overriding 该方法，那么指向该类的代码区。但是超类会存有父类的方法表。

我们知道 java 程序运行时，类的相关信息放在方法区，在这些信息中有个叫方法表的区域，该表包含有该类型所定义的所有方法的信息和指向这些方法实际代码的指针。



当 `Bird`、`Cock`、`Parrot` 和 `CrazyParrot` 这四个类被加载到 Java

虚拟机之方法区后，方法区中就包含了这四个类的信息，下图示例了各个类的方法表。



从图我们可以看到 Cock、Parrot 和 CrazyParrot 的类信息方法表包含了继承自 Bird 的方法。CrazyParrot 的方法表包含了继承自 Parrot 的方法。此外各个类也有自己的方法。

注意看，方法表条目指向的具体方法代码区。对于多态 Overriding 的方法 courtship()，虽然 Cock、Parrot 和 CrazyParrot 的方法表里的 courtship() 条目所在位置是属于继承自 Bird 方法表的部分，但指向不同的方法代码区了。