

今天我来浅谈一下Java多态性，继承与组合，在工作中虽然不是很多见，但是从事Java后台的还是得懂得Java多态性的必要性！

标签：多态性

java引用遍历有两个类型，一个是编译时类型，一个是运行时类型。编译时类型由声明该变量时使用的类型决定，运行时类型由实际赋给该变量的对象决定，如果编译时类型和运行时类型不一致，就可能出现所谓的多态。

```
class BaseClass{
    public int book=99;
    public void base(){
        System.out.println("父类的普通方法");
    }
    public void test(){
        System.out.println("父类的被覆盖方法");
    }
}
public class SubClass extends BaseClass{
    public String book="花花";
    public void test(){
        System.out.println("子类覆盖父类的方法");
    }
    public void sub(){
        System.out.println("子类的普通方法");
    }

    public static void main(String[] args) {
        BaseClass bc =new BaseClass();
        System.out.println(bc.book);
        bc.base();
        bc.test();

        SubClass sc = new SubClass();
        System.out.println(sc.book);
        sc.sub();
        sc.test();
        //编译类型和运行类型不一致，多态发生
        BaseClass bs = new SubClass();
        //输出6，表明访问的是父类对象的实例变量
        System.out.println(bs.book);
        bs.base();
        //子类覆盖父类的方法
        bs.test();
        //因为bs编译时类型是BaseClass，但是这个类没有提供sub()方法，所以下面的代码编译时报错
        //bs.sub();
    }
}
```

运行结果如下：

99

父类的普通方法

父类的被覆盖方法

花花

子类的普通方法

子类覆盖父类的方法

99

父类的普通方法

子类覆盖父类的方法

注意：

1.引用变量在编译阶段只能调用其编译时类型所具有的方法，但运行时则执行它运行时类型的方法，因此，引用变量只能调用声明该变量时所用类里面的方法，如上代码所示，执行bs.sub();时报错。

2.通过引用变量来访问其包含的实例变量时，系统总是试图访问它编译时类型所定义的成员变量，而不是它运行时类型所定义的成员变量。

标签：组合

先看如下代码

```
class Animal{
    private void beat(){
        System.out.println("心脏跳动。。。");
    }
    public void breath(){
        beat();
        System.out.println("呼吸。。。");
    }
}
class Bird {
    private Animal animal;
    public Bird(Animal animal){
        this.animal = animal;
    }
    public void breath(){
        animal.breath();
    }
    public void fly(){
        System.out.println("飞翔。。。");
    }
}
public class Zuhe {
    public static void main(String[] args) {
        Animal a = new Animal();
        Bird b = new Bird(a);
    }
}
```

```
        //心脏跳动。。。
        //呼吸。。。
        b.breath();
    }
}
```

看完代码熟不熟悉，其实我们每天在工作中一直在用，甚至很熟练，可是又有几个人知道这是用到了java中的组合，哈哈哈，反正我每天用，但却不知道是什么原理，懵懵懂的。

问题：到底什么时候用继承？什么时候用组合？

1.继承就是将一个较为抽象的类改造成能适用于某些特定需求的类。而如果两个类之间存在明确的整体，部分关系，则用组合。 2.总之，继承要表达的是(is-a)的关系，而组合表达的是有(has-a)的关系。

