1.

编译和与运行是完全不同的事.编译时主要做的是声明对象的类型,分配属性,检查语法错误等

运行时做的是,将对象加载内存(一般用new,反射也常用), 运行代码执行功能等.

.当我们编译Supclass sup=new Subclass()时,sup对象被声明为Supclass类,于是sup对象的属性便是父类对象的属性的值,

　它是编译时被确定好的.(声明4,可以对这段进行解释).这段话可以对测试三进行解释,即为什么此时对象的属性是父类对象的属性.

　　 2.当我们运行Supclass sup=new Subclass()时,此时sup对象的内存空间是子类对象的内存空间(声明4,可以对这段进行解释),

　　 注意此时sup对象的内存空间由两部分组成父类的一些必要信息和子类对象信息(声明1,可以对这段进行解释).

　　 当我们不重写父类的方法时,由于此时内存空间已有父类的一些必要信息,所以继承父类的方法public void print()当然能被调用。

　　 这段话可以对测试一进行解释.

　　3.接着2继续,若我们重写父类的方法时,由于sup对象的内存空间是子类对象的内存空间,子类的方法public void print()已被加载到内存中.

　　 所以我们调用的是子类的方法public void print();如果你需要调用父类的被重写方法,需要用super.

　　 这段话可以对测试二进行解释.

　　 总结:

　　 以下纯是个人观点:

　　 将子类对象赋值给父类对象,所得到对象是这样的一个对象:

　　 它是一个编译是为父类对象,但运行却是一个子类对象,具体特征如下.

　　 1.被声明为父类对象

　　 2.拥有父类属性

　　 3.占用子类的内存空间

　　 4.子类方法覆盖父类的方法时,此时对象调用的是子类的方法;否则,自动调用继承父类的方法.

　　 5.我人认为这个对象既不是父类对象,也不是子类对象.当我们用到它的方法时,

　　 我便把它看成子类对象;若用到它的属性时,我把它看成父类对象.

　　 它是一个占用父类属性,而使用子类方法的对象.至于到底是什么对象,我认为还是得根据声明来,它应算是父类对象,但拥有子类方法.