Java学习day08

1.多态：多态：事物存在的多种形态；

人：男人 女人

动物：猫 狗

猫 x = new 猫();

动物 x = new 猫();

动物共性的方法是吃

动物可以分成猫、猪、狗

1).多态的体现： 父类的引用指向了子类的对象

2).多态的前提： 必须类与类之间存在关系，要么继承，要么实现； 还有一个前提：存在覆盖

3).多态的好处： 大大提高了程序的扩展性

4).多态的弊端： 只能访问父类中定义好的成员，子类中特殊的就没法访问

5).多态的应用：

1. Animal c = new Cat(); 新的理解，这是一个向上转换； 可以理解为强转换， 比如：byte a ; int c = a; byte类型就转换成了int类型。 子类化成父类

始终对父类操作的强转换会报错， 父类指向的引用需要始终对子类操作， 而new Animal() 是对父类的操作，因此不能够强转换

Animal a = new Animal();

Cat c2 = (Cat) a;

这里在现实举例的一个理解:

胡老爷 x = new 胡晨熙(); 胡晨熙化妆成了胡老爷。其中胡老爷是引用型变量， 胡晨熙是创建的对象所属的类名。

胡晨熙.看电视

胡晨熙 y = (胡晨熙) x;

y.玩游戏（）

同理还有向下转型， 父类想用子类的中特有的内容

1. 多态中成员的特点：

一个小结：

看编译是否成功，看引用型变量中的类是否有方法。也就是看创建对象时左边的类中的方法

看运行是否成功，看创建对象所属的类中的方法是否存在，也就是看创建对象时右边的类的方法。

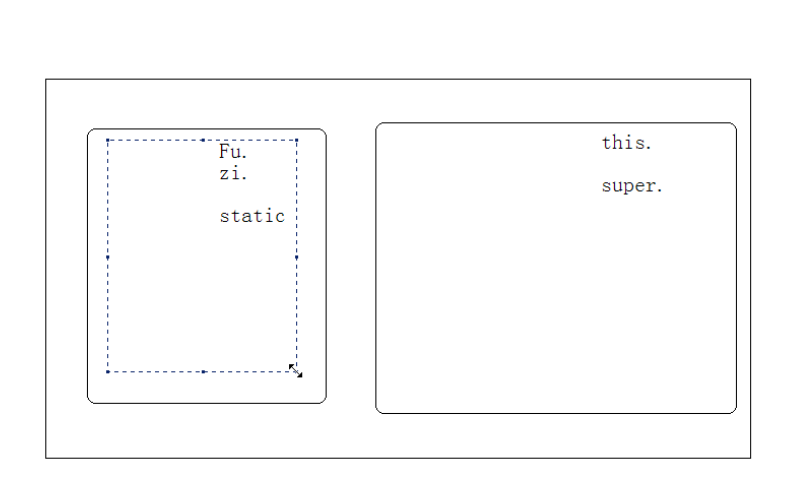
总结：成员函数在多态调用的时候，编译看左边，运行看右边。(非静态)

如下图所示分别为静态区和非静态区。

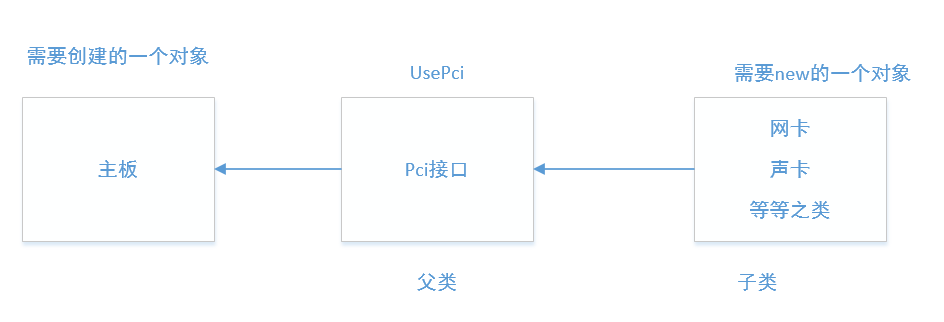
静态区一般直接用类名. 来进行调用； 而非静态区用super与this来调用。

一个小的总结，无论是编译还是运行，在访问成员变量时，始终是调用 创建对象时左边类中的成员变量。

静态成员函数在多态调用的时候都是看左边。



1. 多态的主板示例



多态 这个例子有必要再写一遍

电脑主板实例

有一个pci-e的接口

从接口可以接入 网卡、声卡； 接口可以提高程序扩展性 多态

1. 多态的扩展示例:

注意这是一个思想，不能具体运行。

需求：数据库的操作。

数据是：用户信息

1.连接数据库。 JDBC Hibernate

2. 操作数据库。

增删改添 c create r read u update d delete

1. 关闭数据库连接
2. Objects类

object:所有类直接或者间接父类，就是上帝

所有的子类都是Object的直接或者间接子类

equals方法与toString方法的重写 以及 多态的使用 105讲与106讲的例子。