**关于抽象类实现接口的探究**

抽象类 M

接口 I

子类 A

如下：

**public** **interface** I {

**int** ***Final\_Count*** = 3;

**public** **void** clickable();

**public** **void** showable();

}

**public** **abstract** **class** M **implements** I {

**public** **static** **final** **int *Final\_Count*** = 2;

**public** **abstract** **void** run();

**public** **final** **int** Final\_Count=3;

@Override

**public** **void** showable() {

System.***out***.println("abstract:showable");

}

}

**public** **class** A **extends** M {

@Override

**public** **void** run() {

}

@Override

**public** **void** clickable() {

**super**.showable();

}

@Override

**public** **void** showable() {

System.***out***.println("A:showable");

}

}

总结：

子类里只能使用super调用父类里面已经实现的方法。父类里能 调用所有实现和未实现的方法（包括实现的接口里面的方法）

首先关于抽象类M实现接口I：

抽象类M不用重写接口I的所有方法，可以重写接口I的部分方 法，但是子类A必须重写抽象类M未重写的接口I其余方法，也 可以重写接口I全部方法(包括已经被父类重写的方法)。

**public** **static** **void** main(String[] args) {

M m = **new** A();

I i = **new** A();

A a=**new** A();

}

变量m，i，a分别能调用相应类里面的所有方法。

注意：

在java中多继承中（实际是一个父类加上多个实现接口），假如父类里面变量与接口中的变量（默认是由**public** **static** **final**修饰的）出现同名（可以是不同的修饰符），那么接口中的变量只能有以下两种方式调用：

1. 直接使用接口的名称（例：I.***Final\_Count***）来调用。
2. 使用接口的引用（i.***Final\_Count***）；这里变量i可以是 个空值，即：

I i = **null**;

**int** c=i.***Final\_Count***;

引申：任何类里面由**public** **static** 修饰的变量均可由如下调用方式：

M m = **null**;

System.***out***.println(m.*Final\_Count*);

但是m.*Final\_Count*会出现警告：静态的变量最好使用静态的方式去调用。即：M.*Final\_Count。*