

```
abstract class SuperClass {
    abstract protected String methodFromA(A object);
    abstract protected String methodFromB(B object);
    abstract public String method(SuperClass object);
}

class A extends SuperClass {
    protected String methodFromA(A object)
    {
        return "A.method(A)";
    }

    protected String methodFromB(B object)
    {
        return "B.method(A)";
    }

    public String method(SuperClass object)
    {
        return object.methodFromA(this);
    }
}

class B extends SuperClass {
    protected String methodFromA(A object)
    {
        return "A.method(B)";
    }

    protected String methodFromB(B object)
    {
        return "B.method(B)";
    }

    public String method(SuperClass object)
    {
        return object.methodFromB(this);
    }
}

class Test {
    public static void main(String args[])
    {
        SuperClass a = new A();
        SuperClass b = new B();

        System.out.println(a.method(a));
        System.out.println(a.method(b));
        System.out.println(b.method(a));
        System.out.println(b.method(b));
    }
}
```