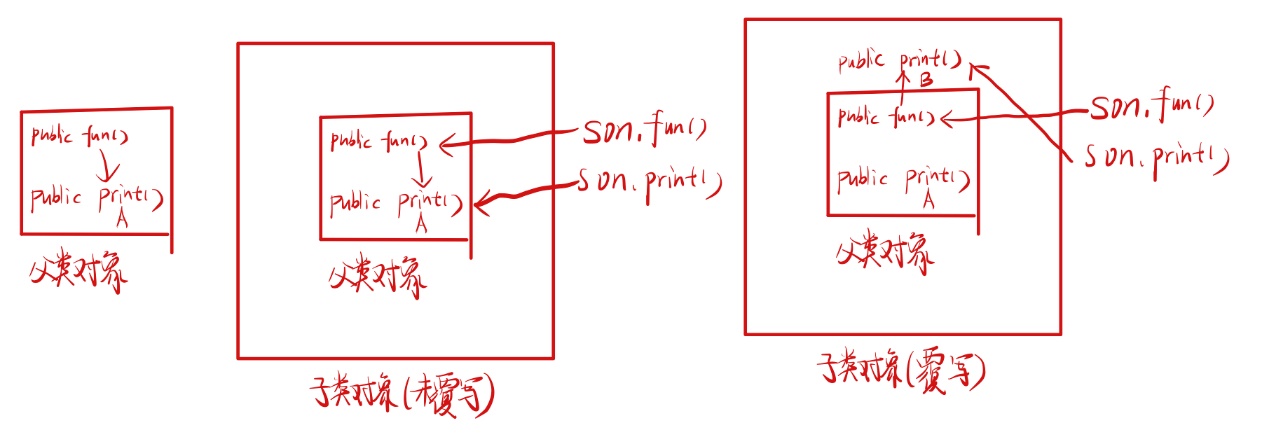
1. **方法的覆写**
2. **非private的方法——可以被覆写**

覆写之后，父类的方法就被隐藏了，任何对该方法的调用，都会调用子类覆写后的方法。

但是子类中方法，可以通过super调用父类中被覆写的方法



覆写规则：两同两小一大

* 两同: 方法名相同 / 形参列表相同
* 两小: 子类方法返回值类型应比父类方法返回值类型小或相等. / 子类方法声明抛出的异常类应比父类方法声明抛出的异常类更小或相等.
* 一大: 子类方法的访问权限应比父类方法的访问权限大或相等.

1. **private方法——无法被覆写**

父类的private方法对子类不可见，所以子类无法对其进行覆写

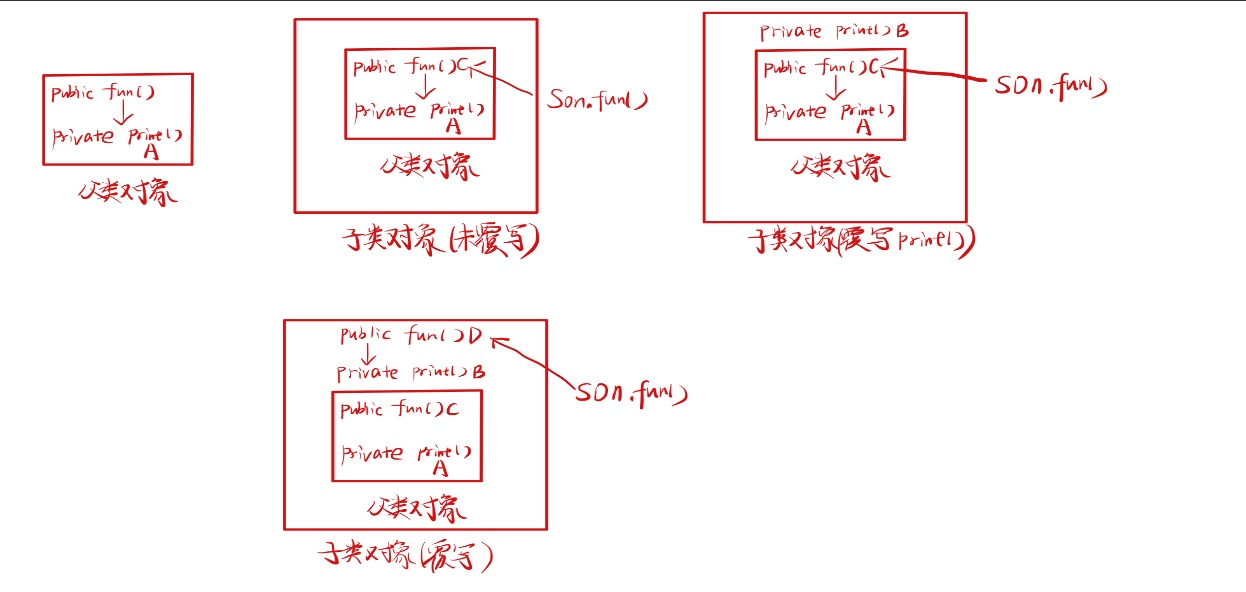
如果子类定义了一个与父类private方法 相同方法名，相同参数表，相同返回类型——这不是重写，只是在子类中定义了一个新的方法

父类有一个private的print()，A

子类重写了一个private的print() ,B

如果父类中的方法调用了print()——用A

如果子类中的方法调用了print()——用B



1. **静态方法的覆写**

由于静态方法必须是public，所以静态方法一定可以进行覆写，覆写后，调用的就只是覆写后的静态方法

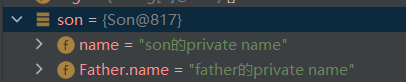
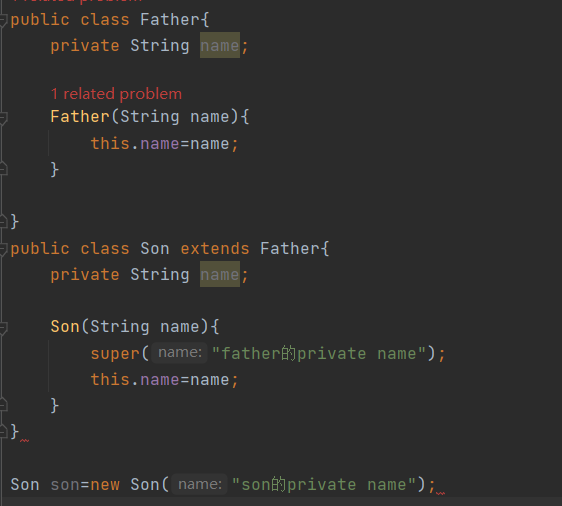
1. **构造代码块的覆写**

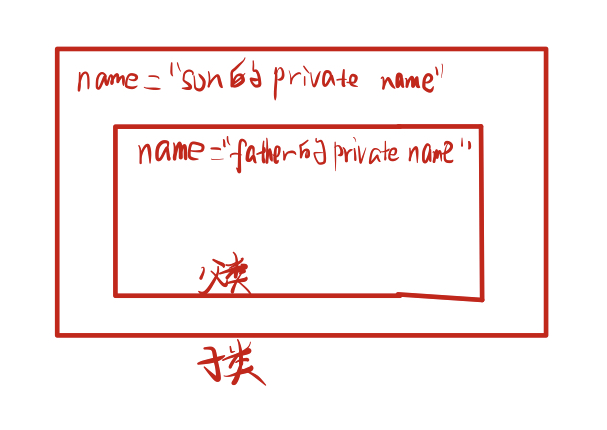
构造代码块无法被覆写

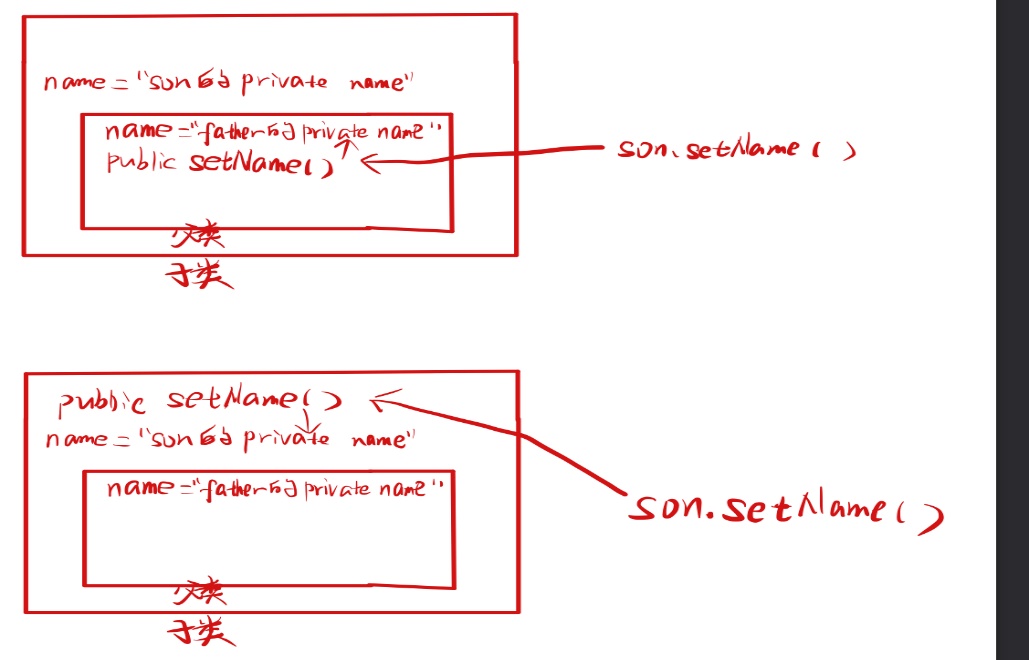
1. **属性的覆写**

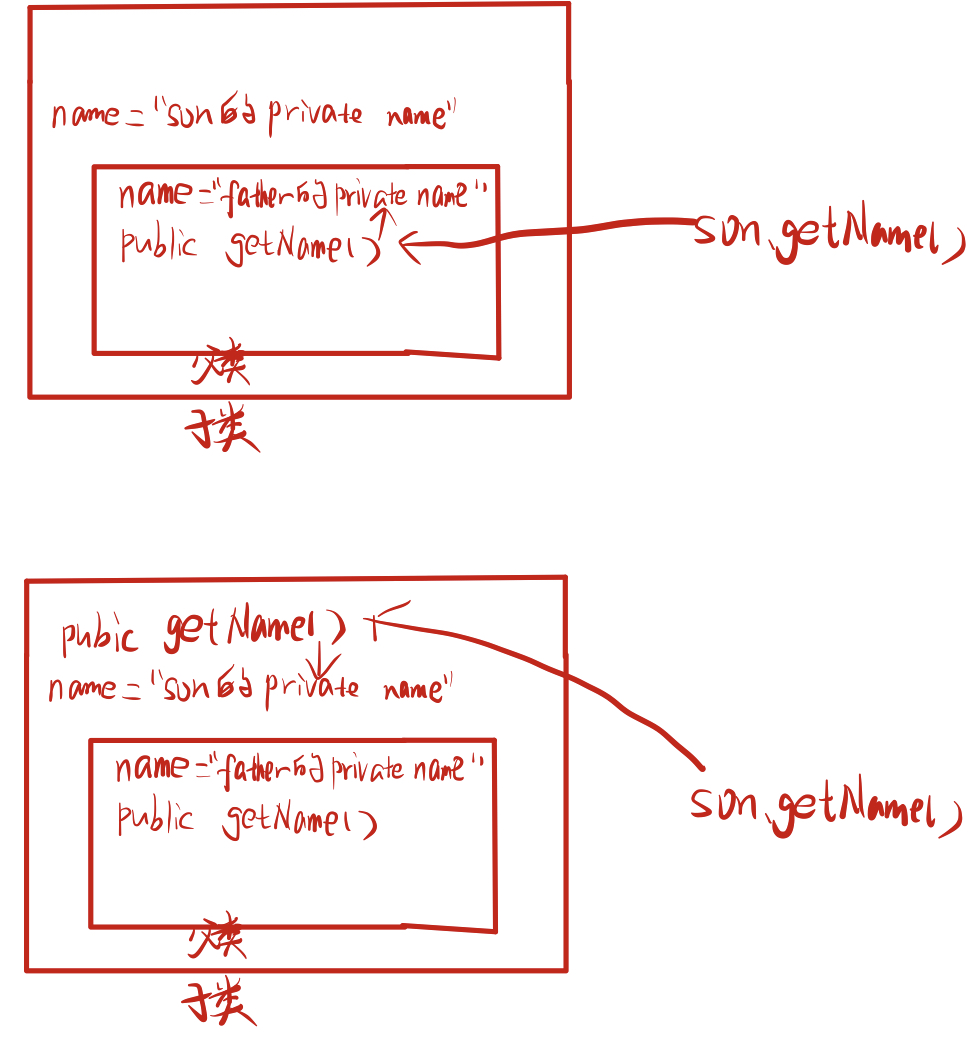
属性无法被覆写，子类跟父类同名的属性只会同时存在

由于封装的缘故，属性一般都是private的，所以接下来只讨论private的属性









因为父类的属性是private，子类的属性也是private

所以父类继承来的方法无法访问子类的private属性，只能访问父类内部的private属性

子类覆写的方法的方法，子类增加的方法无法访问父类的private属性，只能访问子类的private属性

1. super和this

子类引用指向子类对象

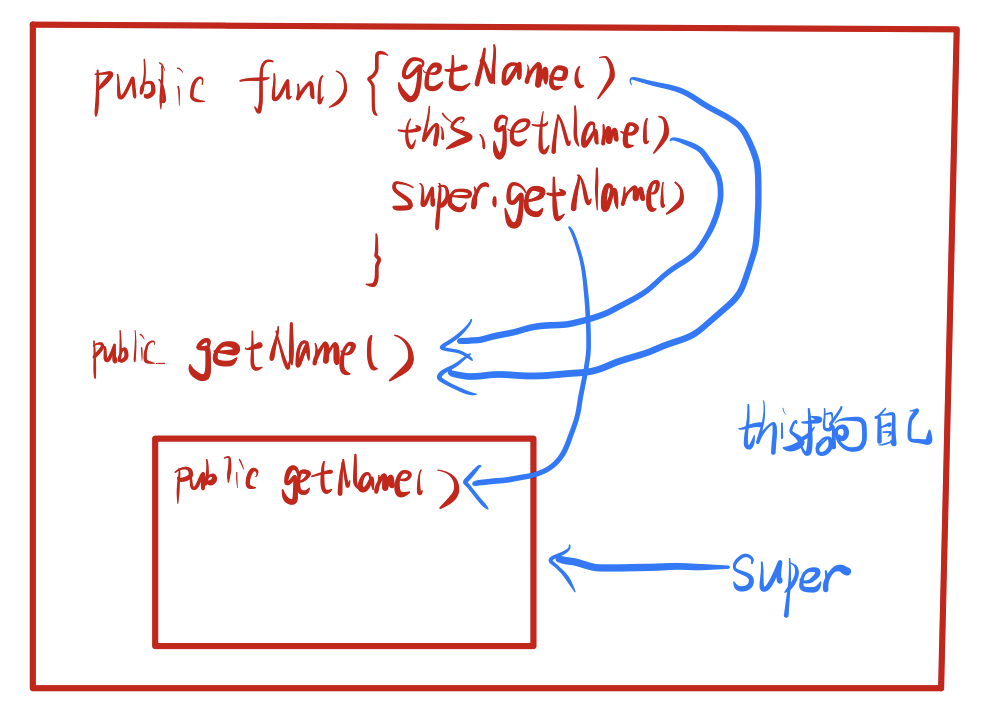
子类引用可以访问子类对象中的非private属性，非private方法（子类对象的private对引用不可见）

子类对象中的非private方法，可以直接调用或者通过this 访问子类对象中的任何属性，任何方法(private也能访问)

子类对象中的非private方法，可以通过super来访问 父类对象中的非private属性，非private方法(也就是被覆盖的方法)

**！！this就是指向自己的引用**

**！！super就是指向内部父类对象的引用**

****