

父类（**SuperClass**）和 子类（**SonClass**）。

父类的非私有化属性和方法可以默认继承到子类。

```
Class Son extends Father{  
}
```

而如果父类中的私有方法被子类调用的话，则编译报错。

父类的构造方法子类不可以继承，更不存在覆盖的问题。（非构造方法可以）

如果子类访问父类的构造方法，则在编译的时候提示访问不到该方法。

JAVA 中不允许多继承，一个类有且只有一个父类（单继承）。

JAVA 的数据结构为树型结构，而非网状。（**JAVA** 通过接口和内部类实现多继承）

方法的覆盖（**overriding**）

方法的重载并不一定是在一个类中：子类可以从父类继承一个方法，也可以定义一个同名异参的方法，也称为 **overloading**。

当子类从父类继承一个无参方法，而又定义了一个同样的无参方法，则子类新写的方法覆盖父类的方法，称为覆盖。（注意返回值类型也必须相同，否则编译出错。）

如果方法不同，则成重载。

对于方法的修饰词，子类方法要比父类的方法范围更加的宽泛。

父类为 **public**，那么子类为 **private** 则出现错误。

之所以构造方法先运行父类再运行子类是因为构造方法是无法覆盖的。

以下范围依次由严到宽：

private：本类访问；

default：表示默认，不仅本类访问，而且是同包可见。

Protected：同包可见+不同包的子类可见

Public：表示所有的地方均可见。

当构造一个对象的时候，系统先构造父类对象，再构造子类对象。

构造一个对象的顺序：（注意：构造父类对象的时候也是这几步）

- ① 递归地构造父类对象；
- ② 顺序地调用本类成员属性赋初值语句；
- ③ 本类的构造方法。

Super()表示调用父类的构造方法。

Super()也和 **this** 一样必须放在第一行。

This()用于调用本类的构造方法。

如果没有定义构造方法，那么就会调用父类的无参构造方法，即 **super()**。

要养成良好的编程习惯：就是要加上默认的父亲无参的构造方法。

思考：可是如果我们没有定义无参的构造方法，而在程序中构造了有参的构造方法，那么如果方法中没有参数，那么系统还会调用有参的构造方法么？应该不会。

多态：多态指的是编译时类型变化，而运行时类型不变。

多态分两种：

- ① 编译时多态：编译时动态重载；
- ② 运行时多态：指一个对象可以具有多个类型。

对象是客观的，人对对象的认识是主观的。

例：