java.lang.reflect.Proxy;

java.lang.reflect.InvocationHandler

Proxy.*newProxyInstance*

动态生成代理类，同时调用方法的时候进行拦截。

Type

Class

Constructor--->getDeclaredConstructors

Method---->getDeclaredMethods

Class<?>[] interfaces ---->getInterfaces

Field--->getDeclaredFields

遍历一个Class和其父类的方法：

Map保存key格式：returntype#方法名称：参数1，参数2

Class<?> currentClass = cls

**while** (currentClass != **null**) {

currentClass.getDeclaredMethods()

//获取直接实现的接口

Class<?>[] interfaces = currentClass.getInterfaces();

anInterface.getMethods()

currentClass = currentClass.getSuperclass();  
}