```
package com.sxt.test;

import java.lang.reflect.Method;
import java.lang.reflect.Modifier;
import java.util.Arrays;

import javassist.ClassPool;
import javassist.CtClass;
import javassist.CtConstructor;
import javassist.CtField;
import javassist.CtMethod;
import javassist.CtNewMethod;
import javassist.NotFoundException;

/**
 * 测试javassist的API
 * @author 江
 *
 */
@SuppressWarnings("all")
public class Demo02 {
        /**
         * 处理类的基本用法
         */
        public static void test01() throws Exception{
                 ClassPool pool=ClassPool.getDefault();
         CtClass cc=pool.get("com.sxt.test.Emp");

         byte[] bytes=cc.toBytecode();
         System.err.println(Arrays.toString(bytes));

         System.err.println(cc.getName());   //获取类名
         System.err.println(cc.getSimpleName());   //获取简要类名
         System.err.println(cc.getSuperclass());   //获得父类
         System.err.println(cc.getInterfaces());   //获得接口
        }

        /**
         * 测试产生新的方法
         */
        public static void test02() throws Exception {
                ClassPool pool=ClassPool.getDefault();
                CtClass cc=pool.get("com.sxt.test.Emp");

                //CtMethod m=CtMethod.make("public int add(int a.int b){return a+b;}", cc);

                CtMethod m=new CtMethod(CtClass.intType,"add",new CtClass[]{CtClass.intType,CtClass.intType},cc);
                m.setModifiers(Modifier.PUBLIC);
                m.setBody("{System.out.println(\"www.sxt.cn\");return $1+$2;}");
                cc.addMethod(m);

                //通过反射调用新生成的方法
                Class clazz=cc.getClass();
                Object obj=clazz.getDeclaredConstructor().newInstance();   //通过调用Emp无参构造器，创建新的Emp对象
                Method method=clazz.getDeclaredMethod("add", int.class,int.class);
                Object result=method.invoke(obj, 200,300);
                System.err.println(result);


        }


        /**
         * 修改已有的方法信息
         */
        public static void test03() throws Exception {
                ClassPool pool=ClassPool.getDefault();
                CtClass cc=pool.get("com.sxt.test.Emp");

                CtMethod cm=cc.getDeclaredMethod("sayHello",new CtClass[]
{CtClass.intType});

cm.insertBefore("System.out.pritnln($1);System.out.println(\"start!!!\");");

cm.insertAt(9,"System.out.pritnln($1);System.out.println(\"b=\"+b);");

cm.insertAfter("System.out.pritnln($1);System.out.println(\"end!!!\");");

                //通过反射调用方法
                Class clazz=cc.getClass();
                Emp emp=(Emp)clazz.getDeclaredConstructor().newInstance();
                Method method=clazz.getDeclaredMethod("sayHello", int.class);
                method.invoke(emp,888);

        }

        /**
         * 创建属性
         */
        public void test04() throws Exception {
                ClassPool pool=ClassPool.getDefault();
                CtClass cc=pool.get("com.sxt.test.Emp");

                //CtFielt f1=CtField.make("private int empno", cc);
                CtField f1=new CtField(CtClass.intType,"salary",cc);
                f1.setModifiers(Modifier.PRIVATE);
                cc.addField(f1);

                //增加相应的set和get方法
                cc.addMethod(CtNewMethod.getter("getSalary", f1));
                cc.addMethod(CtNewMethod.setter("setSalary", f1));

        }

        /**
         * 构造器的操作
         */
        public static void test05() throws Exception {
                ClassPool pool=ClassPool.getDefault();
                CtClass cc=pool.get("com.sxt.test.Emp");

                CtConstructor[] cs=cc.getDeclaredConstructors();
                for(CtConstructor c:cs) {
                        System.err.println(c.getLongName());
                        c.insertBefore("int a=0");
                }
        }

        public static void test06() throws NotFoundException, 
ClassNotFoundException {
                ClassPool pool=ClassPool.getDefault();
                CtClass cc=pool.get("com.sxt.test.Emp");

                Object[] all=cc.getAnnotations();
                Author a=(Author)all[0];
                String name=a.name();
                int year=a.year();
                System.err.println();

        }
     public static void main(String[] args) throws Exception {
         //test01();
         //test02();
          test03();
        }
```

```
}
```