# 先获取反射的对象

String path = "com.wiceflow.Reflection.pojo.User";

Class clazz = Class.forName(path);

## 获取类的名字：

### 获取全路径名字

String clazzName = clazz.getName()

### 获取它的类名

String clazzSimpleName = clazz.getSimpleName();

## 获取属性信息

### 获取全部公开的属性

Field[] fields1 = clazz.getFields(); // public

System.out.println(fields1.length); // User 我里面的属性全是私有的,所以获取长度为0

### 获取所有属性（private,public,...）

Field[] fields2 = clazz.getDeclaredFields(); // private,public,protect

System.out.println(fields2.length); // 3

for (Field temp :fields2){

System.out.println(temp);

// private int com.wiceflow.Reflection.pojo.User.id

// private int com.wiceflow.Reflection.pojo.User.age

// private java.lang.String com.wiceflow.Reflection.pojo.User.uname

}   
获取单个属性

Field field = clazz.getDeclaredField("uname");

System.out.println(field);

// private java.lang.String com.wiceflow.Reflection.pojo.User.uname

## 获得方法信息 (和获取属性差不多)

### 获取所有方法

Method[] methods1 = clazz.getDeclaredMethods(); // private,public,protect

### 获取所有公开的方法

Method[] methods2 = clazz.getMethods(); // public

### 获取对应的方法

Method m01 = clazz.getDeclaredMethod("setUname",String.class);

第一个参数为对应参数的名字,第二个参数为 方法中的传参类型,若有多个参数,则可以叠加,如果该方法有参,则必须传上 否则可以传null

## 获取构造器信息

### 获得所有构造器信息 (和获取属性差不多)

Constructor[] c01 = clazz.getConstructors(); // private,public,protect

for (Constructor temp :c01) {

// public com.wiceflow.Reflection.pojo.User(int,int,java.lang.String)

// public com.wiceflow.Reflection.pojo.User()

System.out.println("构造器:" + temp);

}

### 传递不同的参数类型，获得不同的构造器

Constructor c02 = clazz.getConstructor(int.class,int.class,String.class);

# 源代码



