# 反射

## 概述

Java反射机制 :

是在运行状态中,对于任意一个类,都能够知道这个类的所有属性和方法;

对于任意一个对象,都能够调用它的任意属性和方法;

**说白了,利用反射创建类的对象,调用类的属性和方法时,可以无视修饰符**

这种动态获取信息以及动态调用对象方法的功能称为Java语言的反射机制

**先获取配置文件中的信息,动态获取信息并创建对象和调用方法**

**配置文件读到什么就创建/调用什么.**

## 获取class对象

1. Class.forName(“全类名”);

全类名获取方式,目录栏类位置右键Copy Reference

2. 类名.class;

3. 对象.getClass();

特别的:

Class对象是一个类的整体,类的三大部分:构造方法,成员变量成员方法对应了Class对象的三大组成部分:Constructor, Field, Method.

## 获取Constructor对象

**返回所有公共构造方法对象的数组: 公共构造表示修饰符是public**

Constuctor<?>[] getConstructors()

也就是返回public修饰的所有构造方法的数组

**返回所有构造方法对象的数组:**

Constuctor<?>[] getDeclaredConstructors()

无视修饰符,能返回公共 + 私有的,也就是所有的偶早方法的数组

**返回指定公共构造方法对象**

Constuctor<?> getConstructor(class<?>... parameterTypes)

注意,小括号的形参要跟你需要得到的构造方法的形参保持一致.比如说你要空参构造那就没有参数,如果你需要满参构造,那形参和满参构造的参数保持一致,顺序也不能错,否则报错

**返回单个构造方法对象**

Constuctor<?> getDeclaredConstructor(class<?>... parameterTypes)

和上面获得单个一样,但是这个可以获得私有,上面只能获得公共.

## 利用Constructor创建对象

**T newInstance(Object... initargs) : 根据指定的构造方法创建对象**

在获取到Constructor对象后调用newInstance方法就可以创建类的对象,根据不同的Constructor获取到对应构造方法(参数)创建的对象.

**特别的,**在使用反射获得了私有化构造方法来创建对象时,直接创建对象是不行的,需要构造方法对象调用setAccessible()方法并且传参true,表示需要临时取消访问检查后,再创建私有构造的对象并使用.

## 小结

**1.获取class对象**

**三种方式---常用Class.forName(“全类名”)**

**2.获取里面的构造方法对象**

**getConstructor(class<?>... parameterTypes)**

**getDeclaredConstructor(class<?>... parameterTypes)**

**3.如果是public的,直接创建对象**

**newInstance(Object... initargs)**

**4.如果是非public的,需要临时取消检查,然后再创建对象**

**setAccessible(boolean) 暴力反射**

**protected修饰的和public一样不需要暴力反射**

## 获取Field对象

**Field[] getFields() :** 返回所有公共成员变量对象的数组;

**Field[] getDeclaredFields() :** 返回所有成员变量对象的数组;

**Field getField(String name) :** 返回单个公共成员变量对象;

成员变量必须存在且非私有

**Field getDeclaredField(String name) :** 返回单个成员变量对象;

可以获取私有成员变量,也不需要暴力反射.

## 利用Field赋值和获取值

void set(Object obj, Object value) : 给指定对现象的成员变量赋值;

Object get(Object obj) : 返回指定对象的Field的值

赋值和获取之前,需要创建一个对象,在获取私有成员的时候是暴力反射,需要通过成员变量对象调用**setAccessible(true)**, 其他没什么东西了~

## 获取Method对象

**Method[] getMethods() :** 返回所有公共成员方法对象的数组,包括继承的;

**Method[] getDeclaredMethods() :** 返回所有成员方法对象数组,不包含继承的;

**Method getMethods(String name, Class<?>... parameterTypes) :**

参数String name : 类中的方法名

Class<?>... parameterTypes : 该方法的形参

返回单个公共成员方法对象

**Method getDeclaredMethods(String name, Class<?>... parameterTypes) :**

返回单个成员方法对现象

## 利用Method对象运行方法

**Object invoke(Object obj, Object... args) : 运行方法**

参数一 : 用obj对象调用该方法

就是传入类的对象,表明出处.

参数二 : 调用方法的传递的参数(如果没有就不写)

方法形参

返回值 : 方法的返回值(如果没有就不写)