参考信息：

<http://www.sczyh30.com/posts/Java/java-reflection-1/#%E4%B8%80%E3%80%81%E5%9B%9E%E9%A1%BE%EF%BC%9A%E4%BB%80%E4%B9%88%E6%98%AF%E5%8F%8D%E5%B0%84%EF%BC%9F>

在一篇博文中看到这么几句话，感觉可能对深入理解反射有点帮助吧（至于对不对，需要等待以后实践的验证）：一个java类用一个Class类的对象来表示，这个java类中的组成成分：成员变量，方法，构造方法，包等信息也可以用一个个java类来表示。就像汽车是一个类，汽车中的发动机，变速箱等等也是一个个的类。表示java类的Class类提供一系列的方法类获取其中的变量，方法，构造方法，修饰符，包等信息；这些信息就是用相应类的实例对象来表示，他们是Field，Method，Constructor，Package等。

反射（reflection）是Java程序开发语言的特点之一，它允许运行中的Java程序获取自身的信息，并且可以操作类或对象的内部属性。

Java反射框架主要提供以下功能：

1. 在运行时判断任意一个对象所属的类；
2. 在运行时构造任意一个类的对象；
3. 在运行时判断任意一个类所具有的成员变量和方法（通过反射甚至可以调用private方法）；
4. 在运行时调用任意一个对象的方法。

反射最重要的用途是开发各种通用框架。对一般的开发者而言，不深入框架开发则用到的反射就会少一点。

基本应用

Package java.lang.reflect

一、获取Class对象

三种方法：

1. 使用Class类的forName静态方法

Public static Class<?> forName(String classname)

1. 直接获取某个对象的class

Class <?> kclass = int.class;

Class<?> classInt = Integer.TYPE;

1. 调用某个对象的getClass()方法

StringBuilder str = new StringBuilder(“123”);

Class <?> kclass = str.getClass();

1. 判断是否为某个类的实例

一般我们可以用instanceof关键字来判断是否为某个类的实例。同时我们也可以借助反射中Class对象的isInstance()方法来判断是否为某个类的实例，它是一个Native方法。

（注：简单地讲，一个Native Method就是一个java调用非java代码的接口。一个Native Method是这样一个java的方法：该方法的实现由非java语言实现，比如C。这个特征并非java所特有，很多其它的编程语言都有这一机制，比如在C＋＋中，你可以用extern "C"告知C＋＋编译器去调用一个C的函数。 在定义一个native method时，并不提供实现体<这有些像定义一个java interface>，因为其实现体是由非java语言在外面实现的。）

1. 创建实例

1 通过Class对象的newInstance方法来创建Class对象的实例

Class <?> c = String.class();

Object str = c.newInstance();

2 先通过Class对象获取指定的Constructor对象，再调用Constructor对象的newInstance()方法来创建实例。

Class <?> c = String.class;//获取String对应的Class对象

Constructor constructor = c.getConstructor(String.class);//获取String类带一个String参数的构造器

Object obj = constructor.newInstance(“2333”);//根据构造器创建实例

System.out.println(obj)

1. 获取方法

getDeclaredMethod()方法返回类或接口声明的所有方法，包括公共、保护、默认（包）访问和私有方法，但不包括继承的方法。

Public Method[] getDeclaredMethod() throws SecurityException

getMethods()方法返回某个类所有的公共方法，包括其继承类的公用方法。

Public Method[] getMethod() throws SecurityException

getMethod()方法返回一个特定的方法，其中第一个参数为方法名称，后面的参数为方法的参数对应的Class对象。

Public Method getMethod(String name,Class<?>...parameterTypes)

1. 调用方法

当我们从类中获取一个方法后，我们可以用invoke()方法来调用这个方法，invoke方法的原型为：

Public Object invoke (Object obj,Object...args)

Throws IllegalAccessException,IllegalArgumentException,InvocationTargetException

1. 获取成员变量信息

getField：访问公用成员

getDeclaredField：所有已声明的成员变量，但是不能得到其父类的成员变量

getFields：所有共有变量，包括其父类的共有变量

1. 获取构造器信息

获取类的构造器的用法与上述获取方法的用法类似，主要通过Class类的getConstructor方法得到Constructor类的一个实例，而Constructor类有一个newInstance方法可以创建一个对象的实例。