**反射**

什么是反射机制

简单的来说，反射机制指的是程序在运行时能够获取自身的信息。在java中，只要给定类的名字，那么就可以通过反射机制来获得类的所有信息。

反射机制的优点与缺点

为什么要用反射机制？直接创建对象不就可以了吗，这就涉及到了动态与静态的概念， 静态编译：在编译时确定类型，绑定对象,即通过。动态编译：运行时确定类型，绑定对象。动态编译最大限度发挥了java的灵活性，体现了多态的应用，有以降低类之间的藕合性。一句话，反射机制的优点就是可以实现动态创建对象和编译，体现出很大的灵活性，特别是在J2EE的开发中它的灵活性就表现的十分明显。比如，一个大型的软件，不可能一次就把把它设计的很完美，当这个程序编译后，发布了，当发现需要更新某些功能时，我们不可能要用户把以前的卸载，再重新安装新的版本，假如这样的话，这个软件肯定是没有多少人用的。采用静态的话，需要把整个程序重新编译一次才可以实现功能的更新，而采用反射机制的话，它就可以不用卸载，只需要在运行时才动态的创建和编译，就可以实现该功能。

它的缺点是对性能有影响。使用反射基本上是一种解释操作，我们可以告诉JVM，我们希望做什么并且它满足我们的要求。这类操作总是慢于只直接执行相同的操作。

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一个方法和属性；这种动态获取的信息以及动态调用对象的方法的功能称为java语言的反射机制。

要想解剖一个类,必须先要获取到该类的字节码文件对象。而解剖使用的就是Class类中的方法.所以先要获取到每一个字节码文件对应的Class类型的对象.

反射：就是通过class文件对象，去使用该文件中的成员变量，构造方法，成员方法。

要想这样使用，首先你必须得到class文件对象，其实也就是得到Class类的对象。

Class类：

成员变量 Field

构造方法 Constructor

成员方法 Method

获取class文件对象的方式：

A:Object类的getClass()方法

B:数据类型的静态属性class

C:Class类中的静态方法

public static Class forName(String className)

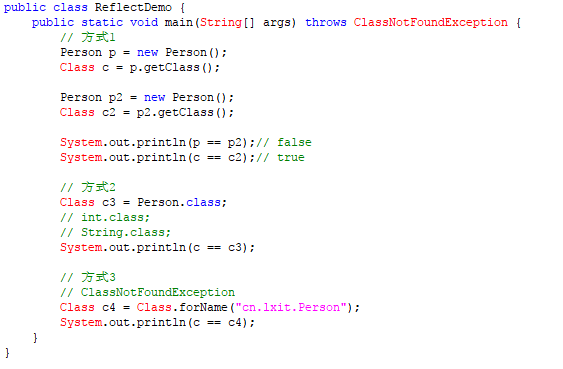
一般我们到底使用谁呢?

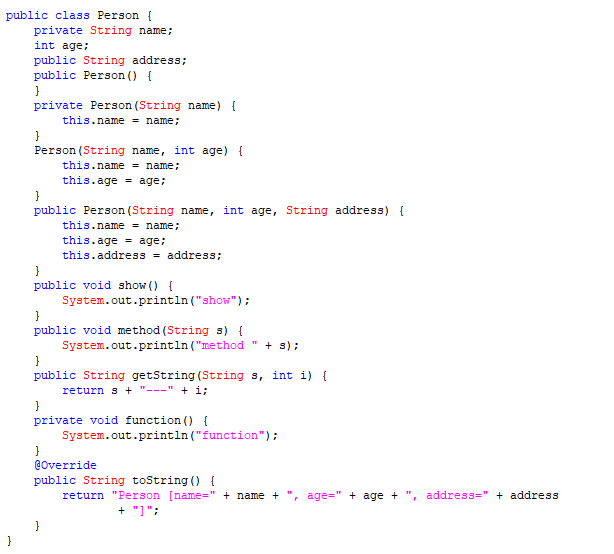
A:自己任选一种，第二种比较方便

B:开发 第三种

为什么呢?因为第三种是一个字符串，而不是一个具体的类名。这样我们就可以把这样的字符串配置到配置文件中。

获取Class对象方法如下：





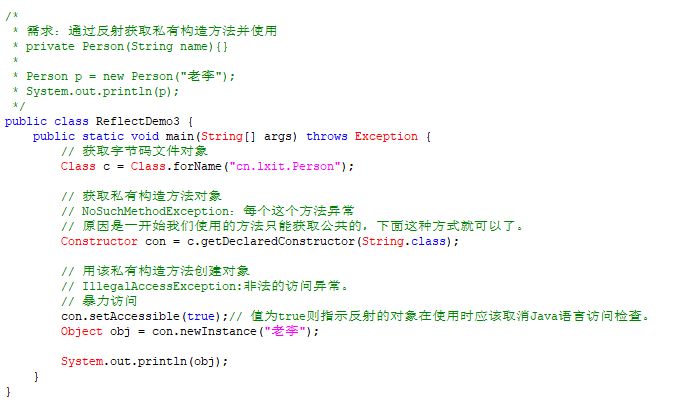
通过反射获取构造方法：



通过反射获取指定构造方法并使用：

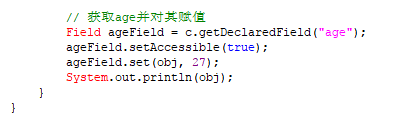


通过反射获取私有构造方法并使用：



获取指定字段





获取方法



反射运行的配置文件

反射越过泛型检查