# 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.

# 反射相关API

反射API用来生成JVM中的类、接口或则对象的信息。   
- **Class类**：反射的核心类，可以获取类的属性，方法等信息。   
- **Field类**：Java.lang.reflec包中的类，表示类的成员变量，可以用来获取和设置类之中的属性值。   
- **Method类**： Java.lang.reflec包中的类，表示类的方法，它可以用来获取类中的方法信息或者执行方法。   
- **Constructor类**： Java.lang.reflec包中的类，表示类的构造方法。

# 使用反射步骤

## 获取Class对象

1、调用某个对象的getClass()方法

Class clazz=p.getClass();

2、调用某个类的class属性来获取该类对应的Class对象

Class clazz=Person.class;

3、使用Class类中的forName()静态方法; (最安全/性能最好)

Class clazz=Class.forName("类的全路径");

## 获取方法和属性

1、通过类对象来获取方法和属性集合

Method[] method=clazz.getDeclaredMethods();

Field[] field=clazz.getDeclaredFields();

Constructor[] constructor=clazz.getDeclaredConstructors()

## 创建对象

当我们获取到所需类的Class对象后，可以用它来创建对象，创建对象的方法有两种：

使用Class对象的newInstance()方法来创建该Class对象对应类的实例，但是这种方法要求该Class对象对应的类有默认的空构造器。

先使用Class对象获取指定的Constructor对象，再调用Constructor对象的newInstance()方法来创建 Class对象对应类的实例,通过这种方法可以选定构造方法创建实例。