

框架中处处可见反射的运用，你对它了解多少？

题目难度：★★★

知识点标签：反射的运用

学习时长：20分钟

题目描述

框架中处处可见反射的运用，你对反射了解多少？

解题思路

面试官问题可以从几个方面来回答：什么是反射？反射有什么用？反射的工作原理

什么是反射

反射是一种能够在程序运行时动态访问，修改某个类中任意属性和方法的机制（包括private实例和方法）

java反射机制提供了一下几个功能：

- 在运行时判断任意一个对象所属的类
- 在运行时构造任意一个类的对象
- 在运行时判断任意一个类所具有的成员变量和方法
- 在运行时调用任意一个对象的方法

反射涉及到的四个核心类：

- `java.lang.Class.java`：类对象；
- `java.lang.reflect.Constructor.java`：类的构造器对象；
- `java.lang.reflect.Method.java`：类的方法对象；
- `java.lang.reflect.Field.java`：类的属性对象；

反射有什么用

- 操作因访问权限限制的属性和方法；
- 实现自定义注解；
- 动态加载第三方jar包，解决android开发中方法数不能超过65536个的问题；
- 按需加载类，节省编译和初始化APK的时间；

反射工作原理

当我们编写完一个Java项目之后，每个java文件都会被编译成一个.class文件，这些Class对象承载了这个类的所有信息，包括父类、接口、构造函数、方法、属性等，这些class文件在程序运行时会被ClassLoader加载到虚拟机中。

当一个类被加载以后，Java虚拟机就会在内存中自动产生一个Class对象。我们通过new的形式创建对象实际上就是通过这些Class来创建，只是这个过程对于我们是不透明的而已。

反射的工作原理就是借助Class.java、Constructor.java、Method.java、Field.java这四个类在程序运行时动态访问和修改任何类的行为和状态。

反射的特点

优点

- **灵活、自由度高：** 不受类的访问权限限制，想对类做啥就做啥；

缺点

- **性能问题：** 通过反射访问、修改类的属性和方法时会远慢于直接操作，但性能问题的严重程度取决于在程序中是如何使用反射的。如果使用得很少，不是很频繁，性能将不会是什么问题；
- **安全性问题：** 反射可以随意访问和修改类的所有状态和行为，破坏了类的封装性，如果不熟悉被反射类的实现原理，随意修改可能导致潜在的逻辑问题；
- **兼容性问题：** 因为反射会涉及到直接访问类的方法名和实例名，不同版本的API如果有变动，反射时找不到对应的属性和方法时会报异常；

总结

- 通过反射访问方法比实例慢很多；
- 有用到反射的类不能被混淆；
- 反射存在性能问题，但使用不频繁、按需使用时，对程序性能影响并不大；
- 反射存在安全性问题，因为可以随意修改类的所有状态和行为（包括private方法和实例）；
- 使用反射访问Android的API时需要注意因为不同API版本导致的兼容性问题；