# **Android 混淆从入门到精通**

### 简介

作为Android开发者，如果你不想开源你的应用，那么在应用发布前，就需要对代码进行混淆处理，从而让我们代码即使被反编译，也难以阅读。混淆概念虽然容易，但很多初学者也只是网上搜一些成型的混淆规则粘贴进自己项目，并没有对混淆有个深入的理解。本篇文章的目的就是让一个初学者在看完后，能在不进行任何帮助的情况下，独立写出适合自己代码的混淆规则。

### 说在前面

本文你可以点击喜欢 工作使用时随时翻阅这里有你代码混淆是所需要知道的一切

这里我们直接用Android Studio来说明如何进行混淆，Android Studio自身集成Java语言的ProGuard作为压缩，优化和混淆工具，配合Gradle构建工具使用很简单，只需要在工程应用目录的gradle文件中设置minifyEnabled为true即可。然后我们就可以到proguard-rules.pro文件中加入我们的混淆规则了。

android { ... buildTypes { release { minifyEnabled true proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro' } } }

以上示例代码表示对release版本就行混淆处理。下面我们先来简介下ProGuard的三大作用，并简要说明下它们常用的命令。

### ProGuard作用

压缩（Shrinking）：默认开启，用以减小应用体积，移除未被使用的类和成员，并且会在优化动作执行之后再次执行（因为优化后可能会再次暴露一些未被使用的类和成员）。

-dontshrink 关闭压缩

优化（Optimization）：默认开启，在字节码级别执行优化，让应用运行的更快。

-dontoptimize 关闭优化  
-optimizationpasses n 表示proguard对代码进行迭代优化的次数，Android一般为5

混淆（Obfuscation）：默认开启，增大反编译难度，类和类成员会被随机命名，除非用keep保护。

-dontobfuscate 关闭混淆

混淆后默认会在工程目录app/build/outputs/mapping/release下生成一个mapping.txt文件，这就是混淆规则，我们可以根据这个文件把混淆后的代码反推回源本的代码，所以这个文件很重要，注意保护好。原则上，代码混淆后越乱越无规律越好，但有些地方我们是要避免混淆的，否则程序运行就会出错，所以就有了下面我们要教大家的，如何让自己的部分代码避免混淆从而防止出错。

### 基本规则

先看如下两个比较常用的命令，很多童鞋可能会比较迷惑以下两者的区别。

-keep class cn.hadcn.test.\*  
-keep class cn.hadcn.test.

一颗星表示只是保持该包下的类名，而子包下的类名还是会被混淆；两颗星表示把本包和所含子包下的类名都保持；用以上方法保持类后，你会发现类名虽然未混淆，但里面的具体方法和变量命名还是变了，这时如果既想保持类名，又想保持里面的内容不被混淆，我们就需要以下方法了

-keep class cn.hadcn.test. {;}

在此基础上，我们也可以使用Java的基本规则来保护特定类不被混淆，比如我们可以用extend，implement等这些Java规则。如下例子就避免所有继承Activity的类被混淆

-keep public class \* extends android.app.Activity

如果我们要保留一个类中的内部类不被混淆则需要用$符号，如下例子表示保持ScriptFragment内部类JavaScriptInterface中的所有public内容不被混淆。

-keepclassmembers class cc.ninty.chat.ui.fragment.ScriptFragment$JavaScriptInterface { public \*; }

再者，如果一个类中你不希望保持全部内容不被混淆，而只是希望保护类下的特定内容，就可以使用

<init>; //匹配所有构造器

<fields>; //匹配所有域

<methods>; //匹配所有方法方法

你还可以在<fields>或<methods>前面加上private 、public、native等来进一步指定不被混淆的内容，如

-keep class cn.hadcn.test.One {  
public <methods>;  
}

表示One类下的所有public方法都不会被混淆，当然你还可以加入参数，比如以下表示用JSONObject作为入参的构造函数不会被混淆

-keep class cn.hadcn.test.One {  
public <init>(org.json.JSONObject);  
}

有时候你是不是还想着，我不需要保持类名，我只需要把该类下的特定方法保持不被混淆就好，那你就不能用keep方法了，keep方法会保持类名，而需要用keepclassmembers ，如此类名就不会被保持，为了便于对这些规则进行理解，官网给出了以下表格

| **保留** | **防止被移除或者被重命名** | **防止被重命名** |
| --- | --- | --- |
| 类和类成员 | -keep | -keepnames |
| 仅类成员 | -keepclassmembers | -keepclassmembernames |
| 如果拥有某成员，保留类和类成员 | -keepclasseswithmembers | -keepclasseswithmembernames |

移除是指在压缩(Shrinking)时是否会被删除。以上内容时混淆规则中需要重点掌握的，了解后，基本所有的混淆规则文件你应该都能看懂了。再配合以下几点注意事项，

### 注意事项

1，jni方法不可混淆，因为这个方法需要和native方法保持一致；

-keepclasseswithmembernames class \* { # 保持native方法不被混淆  
native <methods>;  
}

2，反射用到的类不混淆(否则反射可能出现问题)；

3，AndroidMainfest中的类不混淆，所以四大组件和Application的子类和Framework层下所有的类默认不会进行混淆。自定义的View默认也不会被混淆；所以像网上贴的很多排除自定义View，或四大组件被混淆的规则在Android Studio中是无需加入的；

4，与服务端交互时，使用GSON、fastjson等框架解析服务端数据时，所写的JSON对象类不混淆，否则无法将JSON解析成对应的对象；

5，使用第三方开源库或者引用其他第三方的SDK包时，如果有特别要求，也需要在混淆文件中加入对应的混淆规则；

6，有用到WebView的JS调用也需要保证写的接口方法不混淆，原因和第一条一样；

7，Parcelable的子类和Creator静态成员变量不混淆，否则会产生Android.os.BadParcelableException异常；

-keep class implements Android.os.Parcelable { # 保持Parcelable不被混淆  
public static final Android.os.Parcelable$Creator ;  
}

8，使用enum类型时需要注意避免以下两个方法混淆，因为enum类的特殊性，以下两个方法会被反射调用，见第二条规则。

-keepclassmembers enum {  
public static \*[] values();  
public static \*\* valueOf(java.lang.String);  
}

写在最后

发布一款应用除了设minifyEnabled为ture，你也应该设置zipAlignEnabled为true，像Google Play强制要求开发者上传的应用必须是经过zipAlign的，zipAlign可以让安装包中的资源按4字节对齐，这样可以减少应用在运行时的内存消耗。