# **Android通过javassist修改类**

[](https://www.jianshu.com/u/0cdf27019891)

[yaozhukuang](https://www.jianshu.com/u/0cdf27019891" \t "/Users/wujiawen/Documents\\x/_blank)关注

2018.08.26 16:23:34字数 1,572阅读 4,351

###一、原理介绍

1、App构建是将代码编译为.class文件，然后打包成dex文件之后输出apk

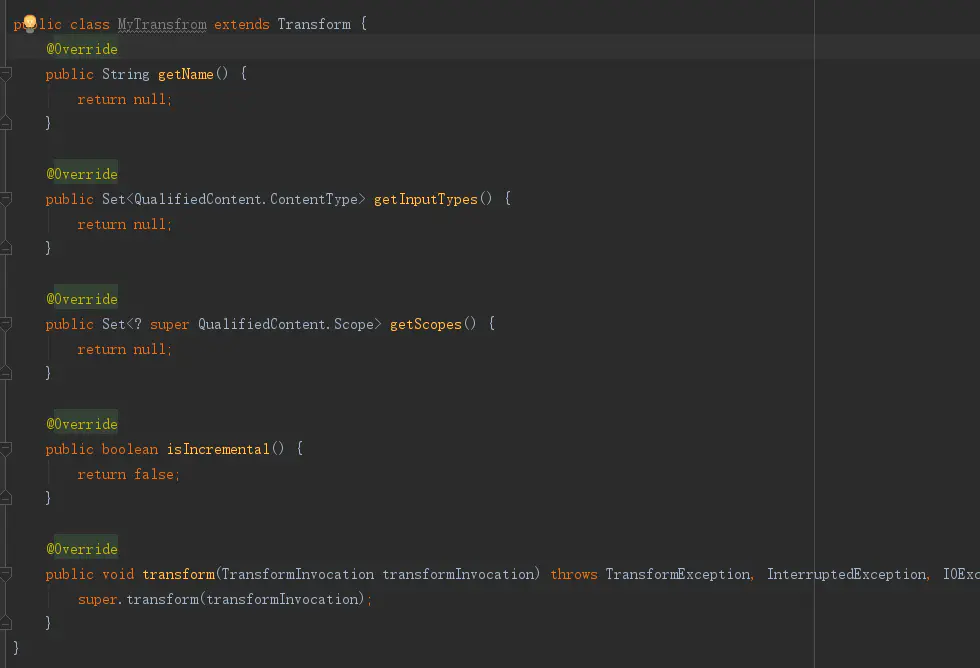
2、Gradle构建App由一个个Task组成，每个Task作用实际上是接收一个输入(编译App所需的资源)然后进行处理然后有一个输出

3、Gradle1.5以后提供了transform-api可以在代码转化为.class文件之后再打包成dex文件之前对它进行处理

4、Javassist可以处理.class文件

所以我们可以通过自定义gradle插件的方式利用javassist在打包的过程中修改.class文件，这样编译出来的apk文件中就会是我们修改过的class

###二、Transfrom-API介绍



**getName**：用于指明本Transform的名字，这个 name 并不是最终的名字，在TransformManager 中会对名字再处理

**getInputTypes**：用于指明Transform的输入类型，可以作为输入过滤的手段

    –CLASSES表示要处理编译后的字节码，可能是 jar 包也可能是目录

     –RESOURCES表示处理标准的 java 资源

**getScopes**：用于指明Transform的作用域

    –PROJECT                         只处理当前项目

    –SUB\_PROJECTS  只处理子项目

    –PROJECT\_LOCAL\_DEPS  只处理当前项目的本地依赖,例如jar, aar

    –EXTERNAL\_LIBRARIES  只处理外部的依赖库

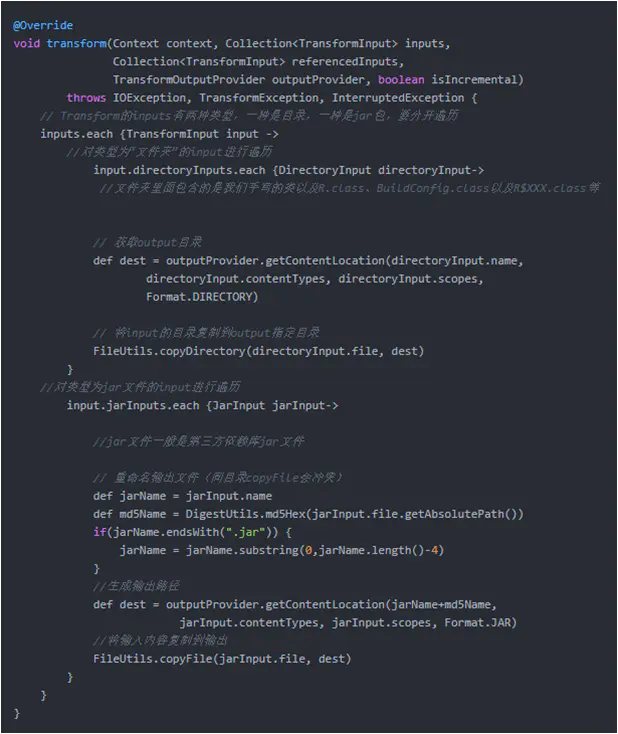
    –PROVIDED\_ONLY  只处理本地或远程以provided形式引入的依赖库

    –TESTED\_CODE                         只处理测试代码

**isIncremental**：用于指明是否是增量构建。

**transform**：核心方法，用于自定义处理,在这个方法中我们可以拿到要处理的.class文件路径、jar包路径、输出文件路径等，拿到文件之后就可以对他们进行操作

利用Transform-api处理.class文件有个标准流程，拿到输入路径->取出要处理的文件->处理文件->移动文件到输出路径



Transform-api处理流程

上图展示的代码中没有包含处理过程，我们只需要在FileUtils.copy函数之前对拿到的文件进行处理即可

###三、javassist介绍

**介绍**：Javassist是一个动态类库，可以用来检查、”动态”修改以及创建 Java类。其功能与jdk自带的反射功能类似，但比反射功能更强大。

**常用类**：

    ClassPool：javassist的类池，使用ClassPool类可以跟踪和控制所操作的类,它的工作方式与 JVM类装载器非常相似，

    CtClass： CtClass提供了检查类数据（如字段和方法）以及在类中添加新字段、方法和构造函数、以及改变类、父类和接口的方法。不

        过，    Javassist 并未提供删除类中字段、方法或者构造函数的任何方法。

    CtField：用来访问域

    CtMethod ：用来访问方法

   CtConstructor：用来访问构造器

**基本用法**

         1、添加类搜索路径

          ClassPool pool =ClassPool.getDefault();

           pool.insertClassPath("/usr/local/javalib");

        2、添加方法

         CtClass point =ClassPool.getDefault().get("Point");

         CtMethod m =CtNewMethod.make( "public int xmove(int dx) { x += dx; }", point);point.addMethod(m);

        3、修改方法

         CtClass point =ClassPool.getDefault().get("Point");

         CtMethod m= point.getDeclaredMethod(“show", null)

         m.insertAfter(“System.out.prinln(“x:” + x + “,y:) + y”))

        4、添加字段

         CtClass point =ClassPool.getDefault().get("Point");

          CtField f = newCtField(CtClass.intType, "z", point);

          point.addField(f);

###四、自定义Gradle插件

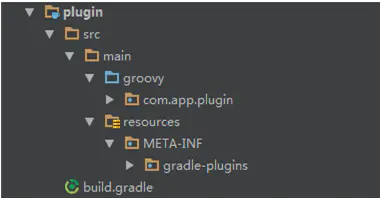
    自定义插件可以直接新建一个名为buildSrc的module不过这样的插件只能自己工程用，还有另外一种方式可以发布到jcenter提供别别人用也可以发布到本地自己使用。这里是第二种方式的步骤：[这里是原文](https://links.jianshu.com/go?to=https://deemons.cn/2017/10/16/%E8%87%AA%E5%AE%9A%E4%B9%89 Gradle %E6%8F%92%E4%BB%B6/" \t "/Users/wujiawen/Documents\\x/_blank)

    1、新建一个 Module，此Module 用于开发插件，类型选什么都无所谓，后面会大改。

    2、在 Project 目录视图模式下，清空build.gradle 文件的内容，删除其余的所有文件。

    3、然后在 module 中新建多个文件夹 src/main/groovy ，再新建包名文件夹。 在 main 目录下再新建resources 目录，在resources 目录下再新

    建 META-INF 文件夹，再新建文件夹gradle-plugins，这样就完成了 gradle插件的目录结构搭建



Module文件结构

    4、打开 build.gralde 文件，替换全部内容



Build.gradle内容

    5、编写插件内容。在刚刚新建的包名下 再次新建一个文件 MyPlugin.groovy ，注意文件类型，一定是 groovy 类型文件

    6、在resources/META-INF/gradle-plugins 目录下新建一个 .properties 文件，注意该文件的命名就是你使用此插件时的名称，这里命名

    为 com.app.myplugin.properties ，一定要注意后缀名称，那么使用时的名称就是com.app.myplugin，文件里面的内容填写如下：     implementation-class=com.app.plugin.MyPlugin，这里是 key = value 的形式，值就是刚刚自定义的 插件（ groovy 文件 ）的全名，也就是径加上类名称。

    7、发布插件到本地，修改build.gradle文件，这个时候右侧的 gradle Toolbar 就会在module下多出一个task :upload-uploadArchives。 clean工程然后运行upload-uploadArchives



发布插件

    8、引用插件



引用插件

      a、修改工程根目录下的build.gradle

       b、在module的目录下的build.gradle中添加

      apply plugin:'com.app.plugin.myplugin' //这里就填写.properties 文件的名称

###五、实操

    我们可以按照上面的步骤进行一次demo编写，这里要做的功能是在代码中所有的View.OnClickListener类的onClick方法中插入一个Toast，在每次编译之后我们可以到文件夹D:\as\_workspace\sample\javassist\app\build\intermediates\transforms\ModifyTransform\debug下查看.class文件是否修改成功，D:\as\_workspace\sample\javassist\这个是代码工程目录ModifyTransform是自定义Gradle插件的名称。修改class文件在实际中还是挺有用处的，比如我们可以用来做无埋点或者修改第三方jar包中的bug。

代码：[https://github.com/yaozhukuang/javassist](https://links.jianshu.com/go?to=https://github.com/yaozhukuang/javassist" \t "/Users/wujiawen/Documents\\x/_blank)