**x.y related frameworks**

**x.y.1 ProGuard**

**ProGuard** is a Java class file shrinker, optimizer, obfuscator, and preverifier. In the shrinking step it removes the unused classes, methods, fields and attributes. In the optimization step it optimizes the bytecode of the methods. The obfuscation step renames the remaining classes, fields, and methods using meaningless names. All of These steps make the code harder to reverse engineer. The final preverification step adds preverification information to the classes, which is required for Java Micro Edition and for Java 6 and higher. Each of these steps are optional, they may not all be used. Furthermore, ProGuard can be used to list dead code in an application. Fig. 2-z shows the ProGuard process

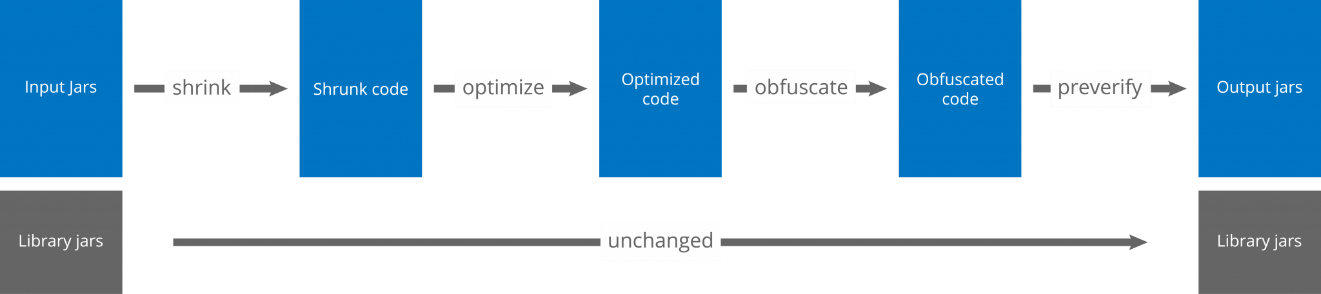


Fig. 2-x ProGuard process

ProGuard first reads the **input jars** (or zips, apks, or directories). It then subsequently shrinks, optimizes, obfuscates, and preverifies them. ProGuard writes the processed results to one or more **output jars** (or zips, apks, or directories). The input may contain resource files, whose names and contents can optionally be updated to reflect the obfuscated class names.

**x.y.2 DexGuard**

DexGuard is a commercial optimizer and obfuscator tool written by Eric Lafortune (who developed ProGuard), it is based on ProGuard and It is used in the place of ProGuard. Rather than targeting Java, DexGuard is specialized for Android resources and Dalvik bytecode.

**x.y.3 ProGuard vs DexGuard**

regarding android, it is more secure to use DexGuard, given that it provides additional security features than ProGuard. Table 2-w shows the differences between both.

Table 2-s Proguard vs DexGuard

|  |  |  |
| --- | --- | --- |
|  | ProGuard | DexGuard |
| Optimization | It is a versatile optimizer for Java bytecode, It enables you to shrink, optimize and obfuscate  desktop applications, embedded applications and mobile applications (Android). | It is specifically designed to protect and optimize Android applications,  It offers functionality that helps you to make optimal use of the Android platform. It comes with a tuned configuration for the Android runtime and for common libraries (Google Play Services, Dagger, Realm, SQLCipher etc.) and automatically splits Dex files that exceed the size limits imposed by the format (MultiDex). |
| Code protection | offers basic protection against static analysis only by harden the source code of the application using a multitude of obfuscation and encryption techniques | shields applications from both static and dynamic analysis by using runtime security mechanisms that check the integrity of the application and of the environment in which it is running and enables the application to react whenever suspicious activity is detected. |
| Obfuscation | offers basic protection in the form of name obfuscation by obfuscate names of classes, fields and methods. | Offers name obfuscation, arithmetic and logical expressions in the code and the control flow of the code inside methods , also it offers encryption of strings and classes and adds reflection to access-sensitive APIs |
| Protection area | ProGuard's action is restricted to the bytecode of Java applications | It provides 360-degree protection. Besides the Dalvik bytecode, it optimizes, obfuscates and encrypts manifest files, native libraries, resources, resource files and asset files. |
| Price | Free, open source tool. | commercial, enterprise-grade product |

**References:**

[**https://www.guardsquare.com/en/blog/dexguard-vs-proguard**](https://www.guardsquare.com/en/blog/dexguard-vs-proguard)

<https://www.guardsquare.com/en/proguard/manual/introduction>