[ML Kit] Assumes use of default proguard rules

DESCRIPTION js...@gmail.com created issue #1 Aug 18,

ENVIRONMENT

- Phone make and model: Pixel 3
- OS version: 11 - latest beta
- Phone make and model: Emulator
- OS version: Pie
- Google Play services version (only Android):
- ML Kit APIs and versions: smart reply 16.0.0 & 16.1.0

STEPS

1. Remove the default proguard rules from the sample app, minifyEnable = true
2. Run the app

EXPECTED

Not crash

ACTUAL

From the project I'm working on:

```
2020-08-17 09:32:28.890 19074-19239/? E/AndroidRuntime: FATAL EXCEPTION: pool-6-thread-1
    Process: com.faithlife.mobile, PID: 19074
    java.lang.UnsatisfiedLinkError: No implementation found for long MIO.c(java.nio.MappedByteBuffer, long) (tried Java_MIO_c and Java_MIO_c__Ljava_nio_Mappe
        at MIO.c(Native Method)
        at MIO.load(com.google.mlkit:language-id@@16.0.0:16)
        at pIO.zza(com.google.mlkit:common@@16.0.0:2)
        at GIO.call(Unknown Source:6)
        at FIO.run(com.google.mlkit:common@@16.0.0:4)
        at JIO.run(com.google.mlkit:common@@16.0.0:3)
        at oIO.a(com.google.mlkit:common@@16.0.0:4)
        at CIO.run(com.google.mlkit:common@@16.0.0:1)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1167)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:641)
        at java.lang.Thread.run(Thread.java:764)
```

ADDITIONAL INFORMATION

We try to be very precise with our R8 rules to avoid keeping unnecessary code in our app. We don't use the default keep rules, as they're too general.

The issue is that JNI methods aren't kept in my app's circumstances. Ideally, the library would specify its own, precise keep rules in metadata like retrofit, for example. The documentation doe rules either for using this library with proguard/r8.

In summary, libraries shouldn't assume the use of the default keep rules shipped with the SDK. People may elect to use them or not for different reasons. I'd like to request that either rules are sh specified in documentation for ML Kit libraries.

COMMENTS All comments

js...@gmail.com <js...@gmail.com> #2 Aug 18, 2(

I realized after submitting that the issue is native method names are obfuscated.

On the chance that someone else runs into the same problem, this will address it without a blanket rule.

```
-keepclasseswithmembernames class com.google.mlkit.nl.languageid.internal.LanguageIdentificationJni {
    native <methods>;
}
```

but the blanket rule may serve you well also if your project links to several native libraries

```
-keepclasseswithmembernames class * {
    native <methods>;
}
```

so...@google.com <so...@google.com> Oc

Assigned to so...@google.com.

so...@google.com <so...@google.com> #3 Oct 8, 2(

Marked as fixed.

Document change has been submitted. Closing

js...@gmail.com <js...@gmail.com> #4 Nov 13, 2(

The documentation change seems to suggest that the keep rule will be supplied as part of the library upon its next release. I'd expect the rule to be supplied as library metadata in `resources INF/proguard/`. It's helpful to keep our app's proguard rules very focused on code in our app's package.