Use of Languages in Android Applications

Asan Urazimanov

27 November 2018

Software Composition Seminar
University of Bern
Android smartphone market share > 85%

Used for sensitive applications
(e-banking, healthcare, etc.)

Languages influence app development
MOTIVATION

LEVERAGING INFORMATION

Not much information is available about the use of languages

"Missing generalization" of languages in Android development

IMPROVING CODE QUALITY

Problems may arise due to the use of complex languages, or their improper integration

Support of practical use
### ASSESSMENT TOOL

**cloc**

github.com/AlDanial/cloc v 1.76  T=0.10 s (667.6 files/s, 57257.8 lines/s)

<table>
<thead>
<tr>
<th>Language</th>
<th>files</th>
<th>blank</th>
<th>comment</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>23</td>
<td>442</td>
<td>478</td>
<td>2262</td>
</tr>
<tr>
<td>XML</td>
<td>40</td>
<td>221</td>
<td>564</td>
<td>1736</td>
</tr>
<tr>
<td>Groovy</td>
<td>3</td>
<td>10</td>
<td>49</td>
<td>47</td>
</tr>
<tr>
<td>Markdown</td>
<td>1</td>
<td>24</td>
<td>0</td>
<td>42</td>
</tr>
<tr>
<td>YAML</td>
<td>1</td>
<td>5</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>ProGuard</td>
<td>1</td>
<td>2</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td><strong>SUM:</strong></td>
<td><strong>69</strong></td>
<td><strong>704</strong></td>
<td><strong>1113</strong></td>
<td><strong>4101</strong></td>
</tr>
</tbody>
</table>
Structure

File structures used in projects
STRUCTURE

- android
  - core
  - desktop
  - gradle
    - .gitignore
    - build.gradle
    - CHANGELOG
    - gradle.properties
    - LICENSE
    - local.properties
    - README.md
    - settings.gradle

- assets
- gradle
- libs
- res
- src
  - AndroidManifest.xml
  - build.gradle
  - local.properties
  - proguard-project.txt
  - project.properties
BUILD TOOLS

Features and differences
BUILD TASKS

Dependency resolution  Cleansing  Compilation and build

Dependencies: JUnit, Hamcrest
ivy.xml  Ivy dependencies specified in ivy.xml

build.xml  build script
pom.xml (main config) Compiles project by using parameters

pom.xml (supplemental config) Plugins
build.gradle

Custom DSL based on Groovy
Lines of code:

- Gradle: 9
- Maven: 72
- Ant: 35
BUILD TOOLS OVERVIEW

Ant
- Full control over the build process.
- Build scripts use XML.

Maven
- Flexibility.
- Strictly structured, highly standardized.

Gradle
- Provides all features.
- More intuitive and shorter build scripts.
The bar chart shows the usage of software languages. The categories are 5, 6, 7, 8, and 9+ languages. The usage percentages range from 0% to 100%. The highest usage is for 5 languages, followed by 6, 7, 8, and 9+ languages with decreasing usage percentages.
LANGUAGES

Features and differences
XML

Language package(s)

Single language

Multi-language

Design
JAVA
well-established language

KOTLIN
new programming language

- less lines of code
- data classes
- null safety
## BUILD TIME DIFFERENCE

<table>
<thead>
<tr>
<th>[s]</th>
<th>Without Gradle</th>
<th>With Gradle</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAVA</td>
<td>10.82</td>
<td>9.17</td>
</tr>
<tr>
<td>KOTLIN</td>
<td>13.12</td>
<td>11.44</td>
</tr>
</tbody>
</table>

22%  
13%
Why Python?

- Startup speed
- Lack of native look
- APK size
AMAZING DISCOVERIES
# 1
.git
.settings
.aarddict
.android
.assets
.branches
doc
.heads
.hooks
.info
.libs
.logs
.My
.objects
.refs
.remotes
.res
.src
.test
test1
.web
.pics
# 2
9,526,124$

100%

185,244 users
# 3
fragmentAbout.java
fragmentAboutAbout.java
fragmentAboutAboutAbout.java
fragmentAboutAboutAboutAbout.java
MainActivity.java
mListAdapter.java
NavigationDrawerFragment.java
fragmentAboutAboutAboutAbout.java
fragmentAboutAbout.java
fragmentAboutAboutAbout.java
MainActivity.java
mListAdapter.java
NavigationViewFragment.java
I’ve wanted to be able to run **Smalltalk** on Android and iOS for many years. I finally decided to do something about it.
# 5
Security Investigation

Automatization
SUMMARY

Structure
no consistent implementation of best practices

Build Tools
many projects lack proper use of build tools

Languages
diverse use of languages complicates application maintenance