Android Performance Tool

Software Performance Laboratory 09
Mattia Gustarini
Introduction

• Android application development
• Google gives a set of guidelines to follow
  • Designing for Performance
  • Designing for Responsiveness
  • Designing for Seamlessness
Introduction

• The developer need to
  • know the guidelines by heart
  • remember to apply them in its code
  • ...

• Need of an automatic tool to
  • check if the guidelines are respected
  • easily find the interesting pieces of code
Designing for performance

- Prefer Virtual Over Interface
- Prefer Static Over Virtual
- Avoid Internal Getters/Setters
- Declare Constants Final
- Use Enhanced For Loop Syntax With Caution
- Avoid Enums
- Use Package Scope with Inner Classes
- Avoid Float
Android Performance Tool

- The tool performs the analysis on class files of an Android Application project
  - ASM library to deal with the bytecode
- It returns
  - the not respected guidelines with a short explanation of the reasons
  - the location of the problem in the source code
Android Performance Tool

- The tool from command line
  - input
    - an Android project folder
  - output
    - verbose
    - quite with the output of an xml
Android Performance Tool

• The tool as eclipse plugin
• input
  • a Java Eclipse project with the Android nature
• output
  • info markers in the Eclipse Java editor
The anatomy of the tool

Class
Crawler
The anatomy of the tool

Android Project

Class Crawler
The anatomy of the tool

Android Project

Class Crawler
The anatomy of the tool

Class Crawler
The anatomy of the tool

Class Crawler

Analyzer
The anatomy of the tool

Class Crawler → Analyzer
The anatomy of the tool

Class Crawler → Analyzer → Enum → Float → InnerClass → ...
The anatomy of the tool

Analyzer

Enum
Float
InnerClass...

Android Performance Tool
The anatomy of the tool

Analyzer

- Enum
- Float
- InnerClass
- ...

SpecificAnalyzer

abstract

Basic and shared behavior

Problem
Future work...

• Add the code of the tool to the Android open source project

• Add the last three challenging guidelines
  • Avoid Creating Objects
  • Use Native Methods
  • Cache Field Lookups
Questions?