Flint

A Programmable Style and Documentation Linter for Java

Ofek Inbar, Daniel Wang, Elliott de Bruin, Jessica Chen

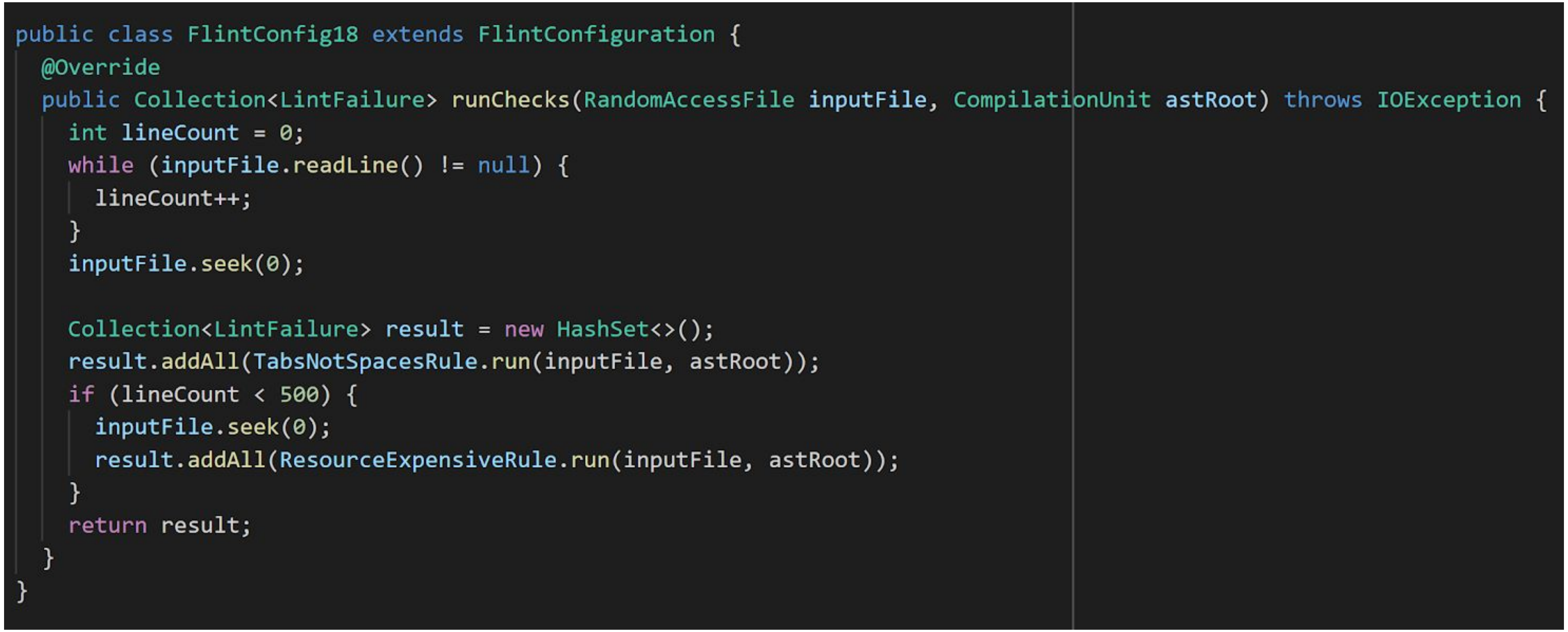
Motivation

- ➔ Developers want tools that are easy to learn and begin using in a short amount of time
- ➔ Using Java for configuration gives the user more control than using XML
- ➔ A linter IDE plugin increases productivity, streamlining the development process
- ➔ Developers want to be able to create and develop clean and readable codebases without needing to manually check their code against their style guide

Approach

- ➔ One configuration file defines the linter’s behavior
  - ◆ One file is easy for users to understand, makes setup easier
  - ◆ Configuration is written in Java, this gives the user total control over the linter’s behavior, from defining which rules are checked in what order and under what conditions
- ➔ Offer both Command Line Interface and IDE extension
  - ◆ Command line is useful for people who don’t use IDE’s, and provides more control over when to lint
  - ◆ IDE extension speeds up workflow (don’t need to switch applications or contexts to run linter)

Approach

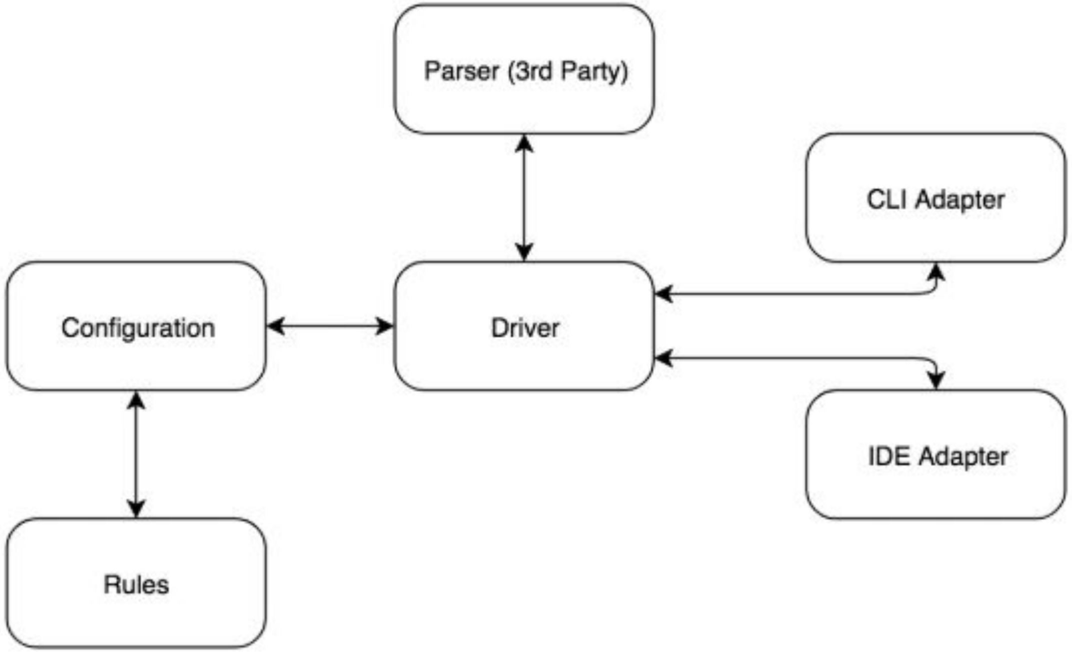

```
public class FlintConfig18 extends FlintConfiguration {
  @Override
  public Collection<LintFailure> runChecks(RandomAccessFile inputFile, CompilationUnit astRoot) throws IOException {
    int lineCount = 0;
    while (inputFile.readLine() != null) {
      lineCount++;
    }
    inputFile.seek(0);

    Collection<LintFailure> result = new HashSet<>();
    result.addAll(TabsNotSpacesRule.run(inputFile, astRoot));
    if (lineCount < 500) {
      inputFile.seek(0);
      result.addAll(ResourceExpensiveRule.run(inputFile, astRoot));
    }
    return result;
  }
}
```

Research Questions

- ➔ Tools that bring milder learning curve
  - ◆ e.g. Java vs. XML, various linter definitions
- ➔ Functionality of existing tools
  - ◆ How people love and hate about them?
- ➔ What developers would like to get out of program analysis?
- ➔ XML and Java Trade-offs

Preliminary Results



Preliminary Results

- ➔ CLI Version Almost Done
  - ◆ CLI Adapter Implemented and Tested
- ➔ Driver being able to connect rules and CLI Adapter
- ➔ CI tools are powerful and unexpectedly easy to use!

Related Tools

- ➔ Checkstyle
  - ◆ Able to customize style checks
  - ◆ Configuration must be done in XML
  - ◆ Comes with pre-written checks
- ➔ Google-java-format
  - ◆ Style checker for Google’s style guide
  - ◆ Not easily customizable
  - ◆ Able to automatically fix errors
- ➔ Error Prone
  - ◆ Hooks into your applications compile step
  - ◆ Catches common programming mistakes at runtime and provides suggested fixes

Works Cited

"Checkstyle – Checkstyle 8.17." *Checkstyle*, 27 Jan. 2019, checkstyle.sourceforge.net/.

"google/google-java-format: Reformats Java source code to comply with Google Java style." *Google*, 10 Jan. 2019, github.com/google/google-java-format/.

Christakis, Maria, and Christian Bird. "What Developers Want and Need from Program Analysis: An Empirical Study." *Microsoft.com*, Microsoft Research, www.microsoft.com/en-us/research/uploads/prod/2016/07/What-Developers-Want-and-Need-from-Program-Analysis-An-Empirical-Study.pdf.