Contracts in the Wild: A Study of Java Programs (Artifact)

Jens Dietrich[1], David J. Pearce[2], Kamil Jezek[3], and Premek Brada[4]

1 **School of Engineering and Advanced Technology, Massey University**
Palmerston North, New Zealand
j.b.dietrich@massey.ac.nz
2 **School of Engineering and Computer Science**
Victoria University of Wellington, Wellington, New Zealand
djp@ecs.vuw.ac.nz
3 **NTIS – New Technologies for the Information Society**
Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Republic
kjezek@kiv.zcu.cz
4 **NTIS – New Technologies for the Information Society**
Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Republic
brada@kiv.zcu.cz

Abstract

This artefact contains a dataset of open-source programs obtained from the Maven Central Repository and scripts that first extract contracts from these programs and then perform several analyses on the contracts extracted. The extraction and analysis is fully automated and directly produces the tables presented in the accompanying paper. The results show how contracts are used in real-world program, and how their usage changes between versions and within inheritance hierarchies.

1998 ACM Subject Classification D.1.5 Object-oriented Programming, D.2.4 Software/Program Verification, D.3.3 Language Constructs and Features

Keywords and phrases verification, design-by-contract, assertions, preconditions, postconditions, runtime checking, java, input validation

Digital Object Identifier 10.4230/DARTS.3.2.6

Related Article Jens Dietrich, David J. Pearce, Kamil Jezek and Premek Brada, "Contracts in the Wild: A Study of Java Programs", in Proceedings of the 31st European Conference on Object-Oriented Programming (ECOOP 2017), LIPIcs, Vol. 74, pp. 9:1–9:29, 2017.
http://dx.doi.org/10.4230/LIPIcs.ECOOP.2017.9

Related Conference European Conference on Object-Oriented Programming (ECOOP 2017), June 18-23, 2017, Barcelona, Spain

1 Scope

The artefact is designed to support repeatability of all experiments reported in the companion paper. The artefact contains scripts that automatically performs all experiments and directly produce the (LaTeX sources of the) tables presented in the paper.

The provided scripts first extract contracts from the programs in the data set and then performs several analyses on those contracts. The analysis and extraction scripts and the data sets (programs and extracted contracts) are all designed to be reusable.

Note that the scripts used to extract the initial dataset from the Maven repository are not part of this artefact. This is because this was a snapshot taken at a certain point in time (3 August 16) that cannot be reproduced by executing the same script later. Instead, the artefact contains a physical copy of those programs.

Table 1 Results Location.

File	Description	Location in Paper
dataset.tex	Data set metrics	Table 2
topusers-lvonly.tex	Top programs using contracts (latest versions only)	Table 3
contractsbytype.tex	Contract elements by type	Table 4
contractsbyclassification.tex	Contracts by classification	Table 5
evolution.tex	Contract evolution data result summary	Table 6
hierarchy.tex	Contract hierarchy data result summary	Table 7
gini-annotations.tex	Gini for annotations	Gini in Section 4.1
gini-apis.tex	Gini for API	Gini in Section 4.1
gini-assertions.tex	Gini for assertions	Gini in Section 4.1
gini-runtimeexceptions.tex	Gini for runtime ex.	Gini in Section 4.1
gini.tex	Gini for all data	Gini in Section 4.1
contraints_across_versions.csv	Number of constraints across versions	Data for Section 4.2
postconditions_removed.log, preconditions_added.log, contracts_cannot_be_classified.log	Textual contracts classification	Discussion in Section 4.3
preconditions_hierarchy_added.log	Textual contracts classification for LSP	Discussion in Section 4.4

For convenience, the artefact contains a master scripts `run-all.sh` that invokes all extraction and analysis scripts, and places the results in the `~/contractstudy/results` folder. The files generated in this folder are mapped to the tables in paper as described in Table 1. The output files are actually those tables, as the output format is LaTeX. An exception is the spreadsheet data in csv format containing data used in Section 4.2. This is processed as follows: scaled contract counts for the first / last version are computed by dividing the values from column 3 (6) by values from column 5 (7). These values are then copied into the spreadsheet `~/doc/notes/box-plot.xlsx` in order to create the boxcharts included in the paper (Figure 2).

The various scripts are implemented as executable Java classes. Each Java class performs certain data analysis as denoted in the Table 2. The package names `contractstudy.scripts` are omitted for brevity. The source code of those classes is included in the artefact in the `~/contractstudy/src/` folder.

2 Content

The artefact is distributed as a VirtualBox image. The virtual machine is pre-installed with Ubuntu Server 16.04.2 LTS, the account used has the user name **artefact** and the password **artefact**, please note the British spelling. The VM was installed with following additional packages:

```
sudo apt install openjdk-8-jdk
sudo apt install maven
sudo apt install mercurial
```

The structure of the VM file system and the location of the various data files and scripts is described in Table 3.

Table 2 Script Details.

Executable Java Class	Output
CollectContracts	Contracts in JSON files in `out/contracts/`
AnalyseContractEvolution	Table `results/evolution.tex`
AnalyseContractUsage	`contractsbytype.tex`, `contractsbyclassification.tex`, `topusers-lvonly.tex` and Gini in `results/`
CollectDataSetStats	Dataset statistics in `results/dataset.tex`
AnalyseContractUsageAcrossVersions	CSV files with program version statistics in `results`
AnalyseHierarchyContracts	Table `results/hierarchy.tex`

Table 3 Directory structure

Directory	Content
~/contractstudy/	Project home
~/mvn-data/	Data corpus according to Section 3.1
~/contractstudy/results	Results
~/contractstudy/out/contracts/	Extracted contracts in JSON format
~/contractstudy/out/struct/	Class structures in JSON format
~/contractstudy/run-all.sh	Script to run all experiments. It generates "results" and "out"
~/contractstudy/run-all-continue.sh	Script to run all experiments. It skips all already generated results
~/contractstudy/src	Project sources
~/contractstudy/target	Project binaries
~/contractstudy/pom.xml	Maven build script

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS).

The latest version of our code is available on Bitbucket: `https://bitbucket.org/jensdietrich/contractstudy`.

4 Tested platforms

The artefact is known to work on any platform running Oracle VirtualBox version 5 (`https://www.virtualbox.org/`) with at least 6 GB of free space in RAM.

5 License

MIT License (`https://opensource.org/licenses/MIT`)

6 MD5 sum of the artifact

69b699de9dc72cc01e27b20df6307830

7 Size of the artifact

7.3 GB