Software Quality & Testing – Graded Assignment 12

Claudio Maggioni

Exercise 1 and 2

The necessary edits have been performed in the test class `src/test/java/comments/CommentsTest.java`.

Exercise 3

I use EvoMaster to generate test cases and find the bug in the source code. I run EvoMaster with the following command from the project root folder `comments-api`:

```
java --add-opens java.base/java.net=ALL-UNNAMED \
  --add-opens java.base/java.util=ALL-UNNAMED \
  -jar ./testing-tools/evomaster.jar --blackBox true \
  --bbSwaggerUrl http://localhost:8080/docs/swagger.yaml \
  --outputFormat JAVA_JUNIT_4 --maxTime 5m
```

EvoMaster correctly finds a bug in the `GET /comments` route, causing a 500 error when the `limit` query parameter is set to `0` and the `offset` query parameter is not given.

The generated tests are included in the directory `src/test/java/em`.

Exercise 4

To run Randoop without a test oracle I run the following commands from the project root folder:

```
echo "comments.repository.MapCriticsRepository" > file.txt
java -classpath testing-tools/randoop-4.3.2/randoop-all-4.3.2.jar:target/classes \
  randoop.main.Main gentests --classlist=file.txt --time-limit=120
```

To run Randoop with the test oracle `jdoctor-oracles.json` I run the following commands from the project root folder:

```
echo "comments.repository.MapCriticsRepository" > file.txt
java -classpath testing-tools/randoop-4.3.2/randoop-all-4.3.2.jar:target/classes \
  randoop.main.Main gentests --classlist=file.txt --time-limit=120 \
  --specifications=jdoctor-oracles.json
```

The generated tests without the oracle are included in the directory `src/test/java/randoopWithoutOracle`, while the ones generated with the oracle are included in the directory `src/test/java/randoopWithOracle`.

The difference between the test generated with or without oracle lies in the assertions over the `MapCriticsRepository.addComment(Comment)` method. The oracle file `jdoctor-oracles.json` specifies that this method should throw a `IllegalArgumentException` when its parameter is null. In the tests without oracle, this exception throwing is treated as a possible bug and tests that cause it are placed in `ErrorTest*` classes since by not expecting the exception they fail. However, in the tests generated with oracles the exception throwing behaviour is correctly tested, thus making tests fail when no exception is thrown and the parameter is indeed `null`. Even with the oracle the tests still fail, since a `NullPointerException` is thrown instead of the `IllegalArgumentException` expected in the oracle (and the documentation of the method), thus uncovering a new bug in `MapCriticsRepository`.

This `NullPointerExc`