ToiletPaper #132

Mutation testing with Pitest

Author: Christof Huber / Software Developer / Business Division Automotive Bavaria

✗ Problem

Common test metrics, such as line coverage and branch coverage, only indicate whether code has been executed during testing and not necessarily if it is working correctly.

This makes it possible to write test code that merely satisfies the metrics without testing anything (test cases without assertions).

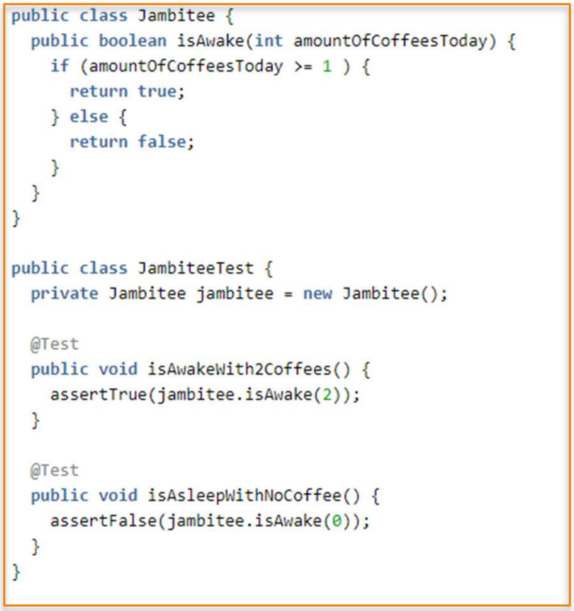
✓ Solution

To get an idea of the quality of the tests, mutation testing can be used.

A mutation test performs unit tests on different mutations of the code to be tested.

There are many different types of mutations, e.g. changing conditions (>= becomes >) or inverting boolean return values.

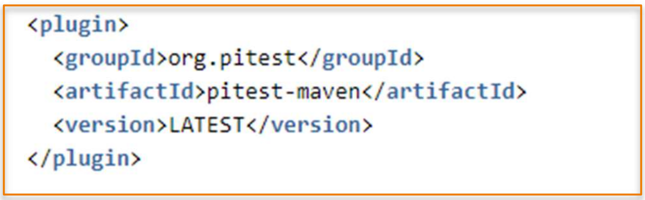
→ Example



```java
public class Jambitee {
  public boolean isAwake(int amountOfCoffeesToday) {
    if (amountOfCoffeesToday >= 1 ) {
      return true;
    } else {
      return false;
    }
  }
}

public class JambiteeTest {
  private Jambitee jambitee = new Jambitee();

  @Test
  public void isAwakeWith2Coffees() {
    assertTrue(jambitee.isAwake(2));
  }

  @Test
  public void isAsleepWithNoCoffee() {
    assertFalse(jambitee.isAwake(0));
  }
}
```

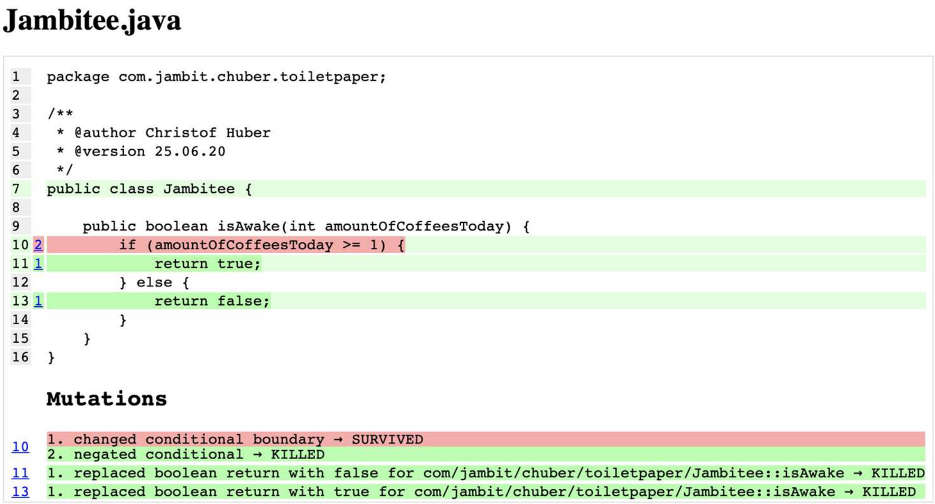
→ Although the code here has 100% line and branch coverage, testing does not secure the ">=1" condition.

In Maven, you only need to include the plugin.



```xml
<plugin>
  <groupId>org.pitest</groupId>
  <artifactId>pitest-maven</artifactId>
  <version>LATEST</version>
</plugin>
```

After that, you can perform the mutation testing by calling the mutationCoverage Goal.
`mvn org.pitest:pitest-maven:mutationCoverage`



Jambitee.java

```
1     package com.jambit.chuber.toiletpaper;
2
3     /**
4      * @author Christof Huber
5      * @version 25.06.20
6      */
7     public class Jambitee {
8
9         public boolean isAwake(int amountOfCoffeesToday) {
10 2          if (amountOfCoffeesToday >= 1) {
11 1              return true;
12            } else {
13 1              return false;
14            }
15        }
16    }

Mutations

10  1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
11  1. replaced boolean return with false for com/jambit/chuber/toiletpaper/Jambitee::isAwake → KILLED
13  1. replaced boolean return with true for com/jambit/chuber/toiletpaper/Jambitee::isAwake → KILLED
```

→ Pitest runs the tests on different mutations and detects mutations that are not covered by the tests.

✚ Further Aspects

- https://pitest.org/
- https://pitest.org/quickstart
- https://github.com/hcoles/pitest