به نام خدا

تمرین کامپیوتری پنجم – تست نرم افزار

امیرحسین راحتی 810100144

علی ممتحن 810100213

در این تمرین ، ابتدا pitest را به پروژه اضافه میکنیم . چون در این تمرین از استاندارد Junit5 استفاده میکنیم ، باید فایل pom به صورت زیر کانفیگ شود:

```
<plugin>
    <groupId>org.pitest</groupId>
    <artifactId>pitest-maven</artifactId>
    <version>1.16.3</version>
    <dependencies>
        <dependency>
            <groupId>org.pitest</groupId>
            <artifactId>pitest-junit5-plugin</artifactId>
            <version>1.2.1</version>
        </dependency>
    </dependencies>
</plugin>
```

سپس برای بررسی Mutation coverage از دستور زیر استفاده میکنیم :

mvn test-compile org.pitest:pitest-maven:mutationCoverage

خروجی pitest به صورت زیر خواهد بود :

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
2	98%	49/50	97%	28/29	97%	28/29

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
Fesadyab.domain	2	98%	49/50	97%	28/29	97%	28/29

Report generated by PIT 1.16.3

Enhanced functionality available at arcmutate.com

Pit Test Coverage Report

Package Summary

Fesadyab.domain

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
2	98%	49/50	97%	28/29	97%	28/29

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
Transaction.java	89%	8/9	100%	4/4	100%	4/4
TransactionEngine.java	100%	41/41	96%	24/25	96%	24/25

Report generated by PIT 1.16.3

در کلاس Transaction داریم:

Transaction.java

```
1   package Fesadyab.domain;
2
3   import lombok.Getter;
4   import lombok.Setter;
5
6   @Getter
7   @Setter
8   public class Transaction {
9       int transactionId;
10      int accountId;
11      int amount;
12      boolean isDebit;
13
14      @Override
15      public boolean equals(Object obj) {
16 1        if (obj instanceof Transaction transaction) {
17 2            return transactionId == transaction.transactionId;
18          }
19 1        return false;
20      }
21  }
```

Mutations

```
16  1. negated conditional → KILLED
17  1. negated conditional → KILLED
    2. replaced boolean return with true for Fesadyab/domain/Transaction::equals → KILLED
19  1. replaced boolean return with true for Fesadyab/domain/Transaction::equals → KILLED
```

فقط متغیر isDebit کاور نشده است و چون در متد های این کلاس از آن استفاده نشده است.

در کلاس TransactionEngine داریم :

```
40 1            if (txn.transactionId == previous.transactionId) {
41                  continue;
42              }
43
44 2            if (txn.amount <= threshold) {
45                  continue;
46              }
47
48 1            if (diff == 0) {
49 1                diff = txn.amount - previous.amount;
50                  previous = txn;
51 2            } else if (diff != txn.amount - previous.amount) {
52                  return 0;
53              }
54          }
55
56 1        return diff;
57      }
58
59      int detectFraudulentTransaction(Transaction txn) {
60          var averageAmount = getAverageTransactionAmountByAccount(txn.accountId);
61
62 4        if (txn.isDebit && txn.amount > 2 * averageAmount) {
63 3            return txn.amount - 2 * averageAmount;  // Excessive debit, marked as suspicious
64          }
65
66          return 0;
67      }
68
69      public int addTransactionAndDetectFraud(Transaction txn) {
70 1        if (transactionHistory.contains(txn)) {
71              return 0;
72          }
73
74          var fraudScore = detectFraudulentTransaction(txn);
75 1        if (fraudScore == 0) {
76              fraudScore = getTransactionPatternAboveThreshold(THRESHOLD);
77          }
78
79          transactionHistory.add(txn);
80 1        return fraudScore;
81      }
82  }
```

```
58
59      int detectFraudulentTransaction(Transaction txn) {
60          var averageAmount = getAverageTransactionAmountByAccount(txn.accountId);
61
62 4        1. detectFraudulentTransaction : changed conditional boundary → SURVIVED
63 3        2. detectFraudulentTransaction : Replaced integer multiplication with division → KILLED
            3. detectFraudulentTransaction : negated conditional → KILLED
            4. detectFraudulentTransaction : negated conditional → KILLED
64
65
```

فقط قسمت قرمز رنگ کاور نشده است . دلیل آن این است که در حالتی که averageAmout صفر است ، با تغییر علامت < به =< ، نتیجه فرقی نمیکند و عملا امکان کاور کردن آن در این حالت وجود ندارد .

تاثیر mutation بالا در refactoring : اگر تست‌های موجود نتوانند تغییرات کوچکی که در کد اعمال شده را شناسایی کنند، این نشان‌دهنده ضعف در پوشش تست‌ها است. در زمانRefactoring ، این ضعف می‌تواند باعث شود تغییرات ناخواسته یا باگ‌ها شناسایی نشوند. همچنین در Refactoring ، ممکن است تغییراتی که به نظر بی‌خطر هستند، منجر به تغییر رفتار کد شوند Mutation Coverage این تغییرات را شناسایی می‌کند.

لینک مخزن پروژه : https://github.com/AmirhosseinRHT/Software-Testing

شناسه آخرین کامیت: 8083c9b22a35920db80652c86af610cd8bbb4cd8