**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:

A screen shot of a computer program

Description automatically generated

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 :

A graph showing the results of a report

Description automatically generated

A graph showing the number of coverage

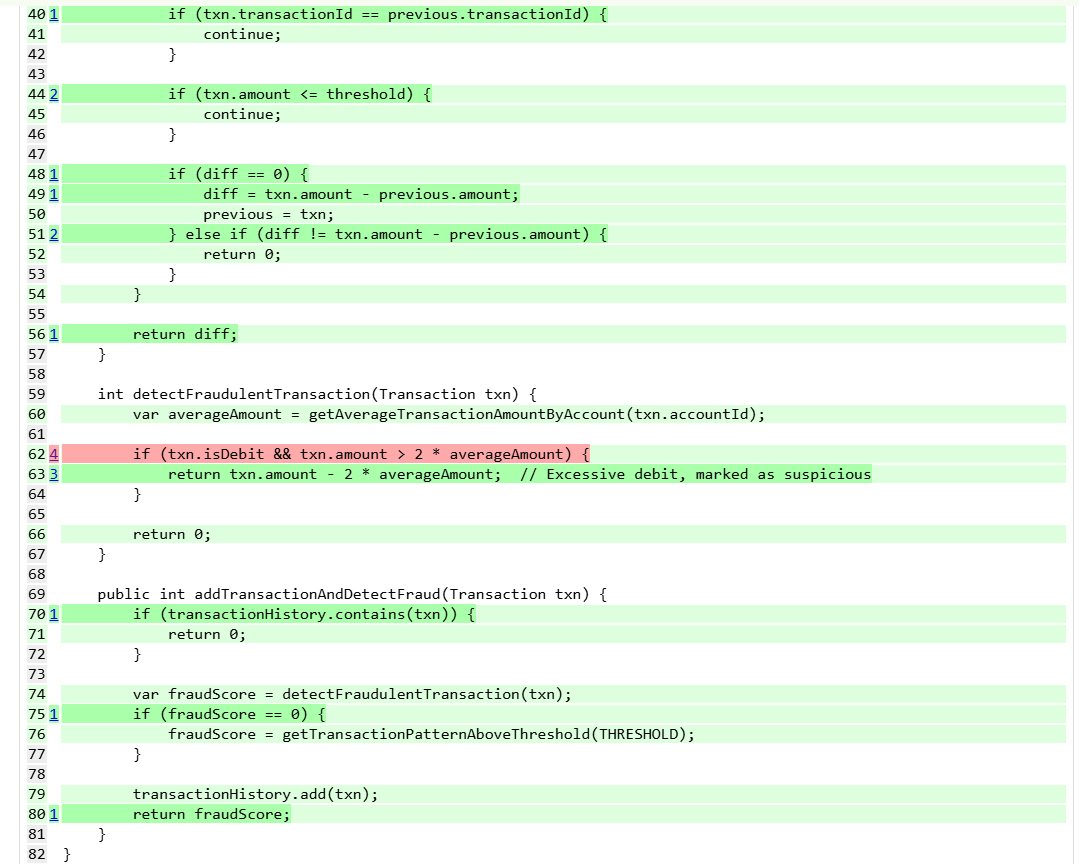
Description automatically generated

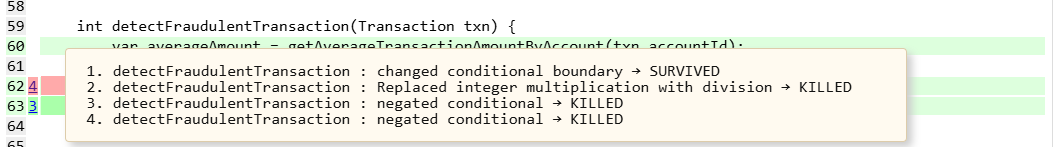
در کلاس Transaction داریم:



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

در کلاس TransactionEngine داریم :





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 زمان Refactoring، 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**