آزمون نرم افزار

تمرین شماره 5 21 آذر 1402

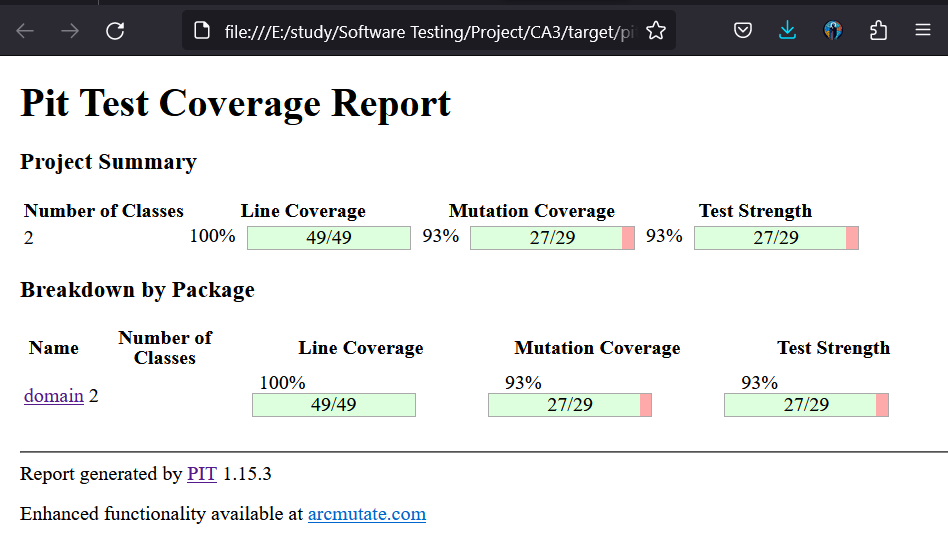
محمدرضا بخشایش – 810199381

امین باهنر – 810199548

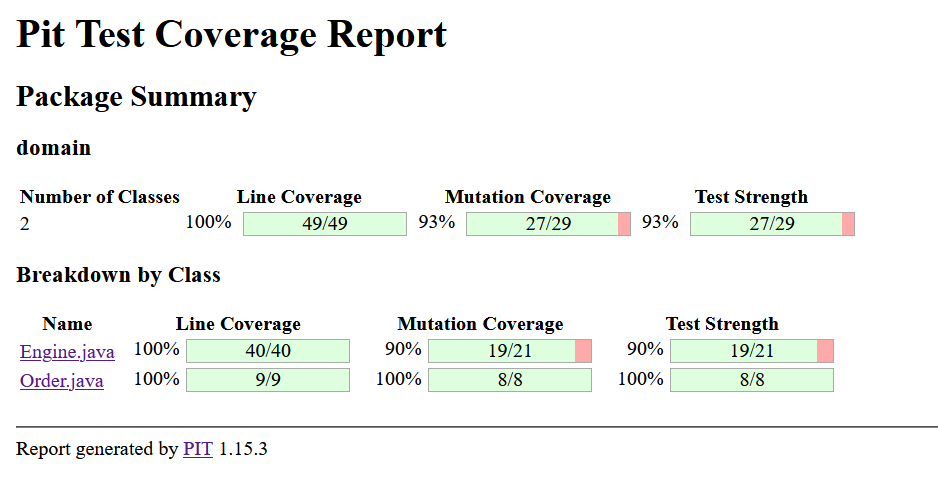
Commit: b1335b3e304566e8b8fe7fa352da7c5890390b36

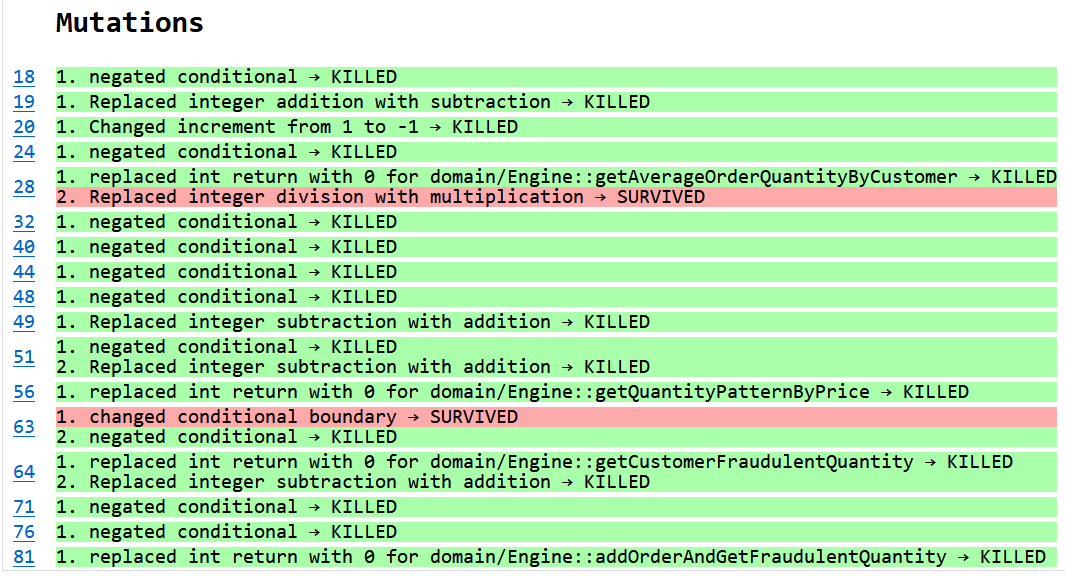
تحلیل گزارش ابزار PIT:

مطابق خواسته صورت پروژه، PIT به پلاگین های پروژه سوم اضافه و اجرا شد. نتیجه گزارش این پلاگین در ادامه آمده است:



صفحه اول گزارش که کلیت نتیجه اجرای پلاگین را نشان می دهد. الاعاتی نظیر Line Coverage و Mutation Coverage برای کل پروژه مشاهده می شود.

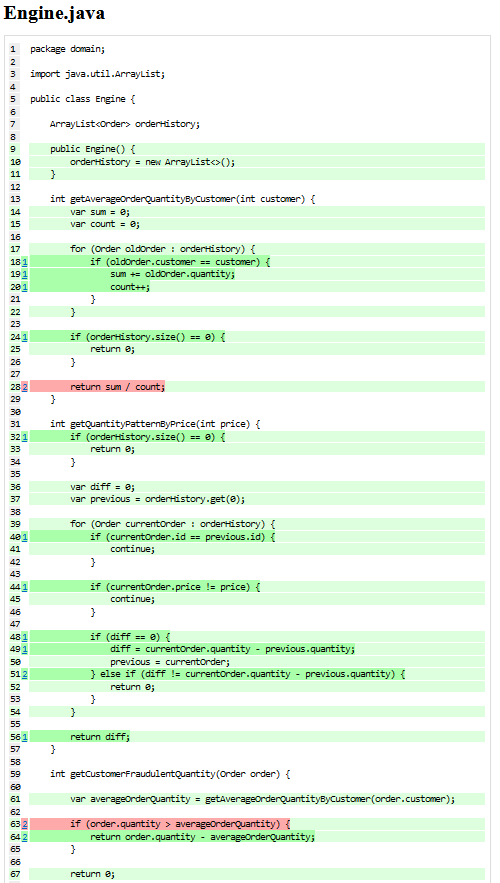


در صفحه بالا، متریک های بیان شده به تفکیک فایل مشخص شده. با توجه به گزارش، در کل پروژه 29 mutation ساخته شده که تعداد 27 مورد توسط تست ها کشته شده و 2 مورد از mutation ها توسط تست ها کشف نشده و زنده مانده اند.

از آنجا که کلاس Order پیچیدگی خاصی ندارد، از آن گذر می کنیم. در صفحه مربوط به کلاس Engine، جزئیاتی از mutation های انجام شده قابل مشاهده است. برای مثال، میتوان به تبدیل تقسیم به ضرب در خط 28 و تبدیل علامت کوچکتر به کوچک تر مساوی اشاره کرد که توسط هیچ تستی کشف نشده اند.



در تصویر بالا، گزارشی از mutation های انجام شده و لیست تست های انجام شده مشاهده می شود. در تصویر بعدی، mutation های زنده و کشته شده روی کد کلاس مشخص شده اند



تاثیر Mutation Coverage بالا بر میزان خطر Refactoring:

اساس کار سنجش تست ها با کمک Mutation Coverage، سنجیدن توانایی مجموعه تست ها برای مقابله با تغییرات ناخواسته در کد است. در فرایند refactoring احتمال بالایی برای به وجود آمدن خطا های این چنینی، (برای مثال، تبدیل مثبت به منفی، جابجایی نوع فرزند یک کلاس، تغییر operator یا نام متد و ...) وجود دارد. بنابراین میتوان نتیجه گرفت، در صورتی که مجموعه تست های ما امکان رسیدن به Motation Coverage بالا را برای ما فراهم کند، احتمال میرود اگر در فرایند Refactoring به خطا های مذکور بر خوردیم، تست ها متوجه آن شده و با fail کردن آنها، بتوانیم خطای ایجاد شده را شناسایی کنیم