



مقدمه

هدف از این تمرین آشنایی با نحوه تبدیل یک نمونه برنامه به گراف و تحلیل کردن انواع نیازمندی‌های پوشش بر روی آن است. تحلیل برنامه‌ها به صورت سیستماتیک کار دشواری است و این موضوع مربوط به تنوع بسیار بالا در روش‌های استفاده و خلاقیت‌هایی است که یک برنامه‌نویس می‌تواند در برنامه خود به کار گرفته باشد. بنابراین برای تحلیل متن یک برنامه، نیازمند یک مدل میانی هستیم که پردازش آن ساده باشد و در عین حال خاصیت‌های برنامه اصلی را داشته باشد. CFG یکی از این مدل‌ها است که به کمک آن می‌توان بخش زیادی از پیچیدگی‌های متن برنامه را کنار زد و بر روی فرایند پشت آن تمرکز کرد. با داشتن این گراف می‌توانیم انواع معیارها را تعریف کنیم و کیفیت آزمون‌های خود را بسنجیم. این روش به دلیل اینکه امکان اجرا به صورت خودکار را دارد و نیازی به نیروی انسانی ندارد ارزشمند است.

بخش پیاده‌سازی

در این بخش می‌خواهیم درصد پوشش شاخه^۱ و پوشش جمله^۲ بشینیه را روی یک نمونه کد به دست بیاوریم؛ برای اندازه‌گیری درصد پوشش نیاز به یک ابزار داریم. در این تمرین از jacoco استفاده شده است و برای اینکه زمان کمتری صرف راه اندازی کنید، پروژه به همراه این مازول در اختیار شما قرار گرفته است. کدی که در اختیار شما قرار داده شده یک سامانه ساده برای کشف فساد در تراکنش‌های بانکی است؛ می‌توانید سامانه فسادیاب را از [این لینک](#) مشاهده کنید. شما باید پوشه test را تکمیل کنید تا پوشش بیشینه را برای هر دو معیار ذکر شده به دست بیاورید. حداقل درصد قابل قبول برای هر دو معیار ۹۰ درصد است. در صورتی که نتوانستید به پوشش ۱۰۰ درصدی برسید باید بتوانید موانع را مطرح کنید. برای اجرای آزمایش‌ها^۳ و ایجاد گزارش پوشش^۴ از دستور mvn test استفاده کنید. دقت کنید که پیش از اجرای این دستور، لازم است متغیر JAVA_HOME بر روی JDK 21 تنظیم شده باشد. برای تایید این مورد، می‌توانید از دستور mvn --version استفاده کنید؛ در این حالت ورژن جاوا باید برابر با ۲۱ باشد. بعد از اجرا، گزارش به صورت html در مسیر target/site/jacoco/index.html در دسترس خواهد بود.

^۱ Branch Coverage

^۲ Statement Coverage

^۳ Testcases

^۴ Coverage Report

Fesadyab

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
domain	<div><div></div></div>	26%	<div><div></div></div>	20%	16	22	32	45	3	7	0	2
Total	125 of 171	26%	24 of 30	20%	16	22	32	45	3	7	0	2

Created with JaCoCo 0.8.12.202403310830

بخش تئوری

سوال اول

با توجه به کد داده شده، به سوالات زیر پاسخ دهید.

```
int getAverageOrderQuantityByCustomer(int customer) {  
    var sum = 0;  
    var count = 0;  
  
    for (Order oldOrder : orderHistory) {  
        if (oldOrder.customer == customer) {  
            sum += oldOrder.quantity;  
            count++;  
        }  
    }  
  
    if (orderHistory.size() == 0) {  
        return 0;  
    }  
  
    return sum / count;  
}
```

- الف) برای قطعه کد بالا CFG را رسم کنید. سپس تمام Prime Path ها و DU Path ها را لیست کنید.
- ب) مجموعه آزمایشی تولید کنید که تمامی DU Path ها و Prime Path ها را پوشش دهد.
- ج) آیا کد ما خطایی دارد؟ اگر بله، توضیح دهید در کدام آزمایش این خطا دیده می شود و چگونه می توان آن را رفع کرد.

سوال دوم

قطعه کد زیر را در نظر بگیرید:

```
var1, var2 = false, false;

if (a | b) {
    var1 = true;
} else if (c) {
    var2 = true;
}

return var1, var2; // Returns the pair of (bool, bool)
```

الف) حداقل چند آزمایش برای پوشش جمله صد درصدی و چند آزمایش برای پوشش شاخه صد درصدی این تکه کد نیاز است؟ (صرفاً حالات مقداردهی متغیرهای a و b و c را بنویسید)

ب) آیا می‌توانید کد را به صورتی تغییر دهید که خروجی و منطق تابع تغییری نکند ولی بتوان با یک آزمایش به پوشش جمله صد درصدی رسید؟

ج) بر اساس CFG توابع، یک شرط کافی برای اینکه بتوان با یک آزمایش به پوشش جمله صد درصدی رسید ارائه کنید.

سوال سوم

اثبات کنید که پوشش کامل مسیرهای du لزومی بر پوشش کامل مسیرهای $prime$ ندارد.

نکات پایانی

- پروژه در قالب گروه‌های **حداکثر دو نفره** انجام می‌شود.
- برای پیاده‌سازی ابتدا پروژه را از **این لینک** clone کرده و سپس یک مخزن⁵ در صفحه شخصی خود به صورت خصوصی⁶ ایجاد کرده و تغییرات لازم را بر روی آن اعمال کنید.
- کاربر **SWT-UT** را به مخزن خود اضافه کنید.
- پاسخ سوالات بخش تئوری را در قالب یک فایل PDF در صفحه درس بارگذاری کنید. توجه داشته باشید که نیازی به ذکر کدهای بخش پیاده‌سازی در این فایل نیست؛ تنها لازم است در ابتدای این فایل، **آدرس مخزن و شناسه آخرین کامیت** خود را بنویسید.
- برای تحویل کافیست یکی از اعضای گروه فایل PDF را در صفحه درس بارگذاری نماید.
- هدف از این تمرین، یادگیری شماست؛ لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده مشابهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاست ذکر شده در کلاس، کسر خواهد شد.

⁵ Repository

⁶ Private