آزمون نرم‌افزار

تمرین شماره ۴

مدرس: دکتر احسان خامس‌پناه

طراحان: علی هدایی، آوا میرمحمدمهدی، امیرعلی وحیدی

مهلت تحویل: یکشنبه ۴ آذر ۱۴۰۳، ساعت ۲۳:۵۹

مقدمه

هدف این تمرین، آشنایی با مفاهیم logic based testing، input space partitioning و API testing می‌باشد. logic based testing تمام حالاتی که شامل متغیری با مقدار True می‌شوند را بررسی می‌کند. اهمیت آن این است که مطمئن می‌شویم در بخش‌های مربوط به منطق کد مانند شرط‌ها، اتفاق ناخواسته‌ای رخ نمی‌دهد. همچنین input space partitioning بخش بسیار مهمی از آزمون‌ها است. در هر برنامه‌ای ورودی‌های ما می‌توانند در بازه‌هایی مشخص باشند که مقدارهای مختلفی از این مقادیر می‌توانند تاثیراتی بر روی یکدیگر بگذارند. با انجام ISP، ما مطمئن می‌شویم که تمامی حالات ممکن از ورودی‌ها در برنامه بدون مشکل اجرا می‌شوند، به خصوص مقادیر مرزی!

بخش پیاده‌سازی

برای کلاس‌های RestaurantController و TableController با استفاده از @WebMvcTest یا @SpringBootTest آزمون API بنویسید. توجه کنید که نحوه آزمون نباید به شکل فراخوانی مستقیم متدهای کنترلر باشد و آزمون‌ها باید به صورت وب تست پیاده‌سازی شوند. همانطور که در فایل آشنایی با پروژه درس هم ذکر شده بود، ممکن است برخی توابع پروژه به همراه ایرادات جزئی باشند، در هنگام نوشتن آزمون‌های خود در صورتی که به ایرادی برخورد کردید، سعی کنید آن را رفع کرده و در تحویل پروژه موارد اصلاح شده را ذکر کنید.

بخش تئوری

سوال اول

به صورت مختصر تفاوت بین دو انوتیشن @WebMvcTest و @SpringBootTest را بیان کنید.

سوال دوم

با در نظر گرفتن predicate زیر به سوالات پاسخ دهید.

$$p = (\neg a \wedge b) \vee (b \wedge c) \vee (\neg b \wedge \neg c)$$

الف) جدول درستی را بنویسید.

ب) تمام جفت ردیف‌هایی که Correlated Active Clause Coverage (CACC) را برآورده می‌کند بنویسید.

پ) تمام جفت ردیف‌هایی که Restricted Active Clause Coverage (RACC) را برآورده می‌کند بنویسید.

آیا جفت‌های نوشته‌شده در قسمت ب، زیرمجموعه جفت‌های نوشته‌شده در این قسمت هستند؟ چرا؟

ت) در predicate داده‌شده، آیا برآورده کردن clause coverage می‌تواند predicate coverage را نتیجه دهد؟ درستی آن را نشان دهید یا برای آن مثال نقض بزنید.

سوال سوم

با توجه به تابع زیر که قیمت نهایی یک خرید را پس از اعمال تخفیف بر اساس قیمت کالا، نرخ تخفیف و حداقل خرید محاسبه می‌کند، ورودی‌های آن را به بلاک‌هایی تقسیم کرده و سپس پوشش pair-wise را پیاده‌سازی کنید. برای هر مورد آزمون (test case)، با توجه به مسئله و مقادیر داده شده، assertion مناسب را انجام دهید.

```
public static String calculateDiscountedPrice(double price, double discountRate,
double minPurchase) {
    if (price <= 0 || discountRate < 0 || discountRate > 1 || minPurchase <= 0) {
        return "Invalid input";
    } else if (price < minPurchase) {
        return String.valueOf(price);
    } else {
        double discountedPrice = price * (1 - discountRate);
        return String.valueOf(discountedPrice);
    }
}
```

نکات پایانی

- پروژه در قالب گروه‌های **حداکثر دو نفره** انجام می‌شود.
- برای پیاده‌سازی ابتدا پروژه را از این لینک clone کرده و سپس یک مخزن[1] در صفحه شخصی خود به صورت خصوصی[2] 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؛ لطفا 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.

[1] Repository

[2] Private