به نام خدا

تمرین کامپیوتری اول – تست نرم افزار – بخش تئوری

امیرحسین راحتی 810100144

علی ممتحن 810100213

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

شناسه آخرین کامیت: d0af7cd8bbdbdcf42c471151b535823bbb8cda7f

- ## پاسخ سوالات بخش تئوری

1) Assume در جاوا برای شرایطی استفاده می‌شوند که تست تنها در صورتی باید اجرا شود که یک شرط خاص برقرار باشد. اگر آن شرط برقرار نباشد ، فرآیند اجرای تست متوقف میشود.

: AssumeTrue()

این تابع دو فرم فراخوانی دارد:

assumeTrue(boolean b)

assumeTrue(String message, boolean b)

این متد آزمون را تنها در صورتی اجرا می‌کند که شرط condition برابر با true باشد. اگر شرط برقرار نباشد، آزمون لغو می‌شود (fail نمی شود).

2) خیر . چون ممکن است با اجرای تست در تعداد زیادی دفعات ، مشکلی نبینیم . اما در حالت بعدی مشکلی بخاطر شرایط اجرای ترد ها بوجود بیاید که تا قبل آن به وجود نیامده و دچار مشکل شویم . برای حل این مشکل ، می توان از formal methods برای آنالیز کردن برنامه و flow آن استفاده کرد که البته هزینه های زیادی از جنبه های مختلف دارد اما اطمینان ایجاد میکند.

3) این تست ، عملا تست نیست ! چون چک نمیشود که خروجی دقیقا همان مقدار 10 است که میخواهیم یا نه . اگر خروجی چیز دیگری باشد و عوض شود هم نمیتوانیم آن را به سادگی بفهمیم و باید خروجی ترمینال یا فایل لاگ را خودمان بررسی کنیم یا کد دیگری بنویسیم که آن را بررسی کند .

4) در دو تست مطرح شده:

a) ظاهرا در این تست انتظار این میرود که کد حین اجرا یک exception را throw کند . برای تست این موضوع استفاده از کلید واژه except صحیح نیست و میتوان از annotation هایی مثل annotation پایین استفاده کرد که از این موضوع مطمئن شد : @Test(expected = Exception.class)

همچنین نام تست میتواند مورد زیر تست را بهتر نشان دهد. از ظرفی ، تعریف متغیر برای تست در داخل متد تست باعث میشود که اگر بخواهیم بعدا حالات زیر تست را بیشتر کنیم ، نیاز است بدنه کد تست را تغییر دهیم و نمیتوانیم از روش های parametrized test برای تست کردن استفاده کنیم .

b) در پایان تست اول ، اثر آن پاک نمی شود و نتیجه آن روی نتیجه تست بعدی اثر می گذارد . در واقع تست دوم وابسته به تست اول است. از طرفی ، این دو تست نتیجه محاسبات را بررسی می کنند و قابلیت reuse شدن دارد . اما با این روش تعریف تست ، امکان استفاده دوباره آن وجود ندارد.

- باگ های کد

1) باگ در تابع getMaxSeatsNumber در فایل restuarant.java که باید درصورتی که هیچ میزی یافت نشده باشد 0 برگردانده شود.
2) در قسمت هایی از کد با توجه به اینکه ممکن است در برنامه چند درخواست همزمان بیاید بعضی از توابع و دسترسی ها باید کنترل همروندی داشته باشند. این مورد در فایل restaurant.java مدیریت شده است.