به نام خدا

دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر

درس آزمون نرم افزار

پروژه اول

نام و نام خانوادگی	دانیال سعیدی(810198571) محمد قره حسنلو(810198461)
تاریخ ارسال گزارش	1401/8/25
ریپو گیتهاب	
Commit IDآخرین	

فهرست گزارش سوالات

سوال 1

معیار هایی که برای تست module های مختلف استفاده کردیم، بر اساس لیست زیر میباشد:

کدهای production را برای تست های مختلف تغییر ندادیم

اجازه رفتن تست در محیط عملیاتی را ندادیم

به صورتی نوشته شده است که برای کسی که کد را میخواند و توسعه دهنده تست ها نیست، قابل فهم باشد

از overlap بین تست ها جلوگیری کرده ایم

در هر تست یک شرط خاص بررسی شده است.(سوال اول و parameterized شده انواع مختلف شرایط در سوال دوم)

بعد از تست ها از ایجاد garbage جلوگیری شده است

هزینه تا جای ممکن minimize شده است

سوال 2

در اسلایدهای درس در tests with theory از آنها استفاده شده بود که برای جلوگیری از استفاده از موارد بی ربط در ضرب کارتزین حالت های مختلف استفاده میشد. مثلا
assumeTrue(someSet!= null) که در اینجااین فرضیه را به درستی تایید میکند. اگر فرض نادرست باشد، اجرای تست لغو میشود. در واقع اگر یک عبارت با false ارزیابی شود، تست متوقف و نادیده در نظر گرفته میشود.

سوال 3

با توجه به اینکه Junit یک کد خارجی است که کد ما را کنترل میکند(در واقع دارد آن را صدا میزند)، پس به همین دلیل یک framework است. اگر کدخارجی توسط کد ما فراخوانی شود(در واقع کد ما، کد خارجی را کنترل کند که در Junit به این صورت نیست)، این یک کتابخانه در نظر گرفته میشود.

سوال 4

در تست اول، به نظر می آید مشکل خاصی وجود ندارد اما میشد از قسمت های تکراری که یک عمل را انجام میدهند، جلوگیری کرد.

در تست دوم، ما با توجه به عبارتی که در پرانتز بعد از Test@ گذاشتیم، توقع داریم که بعد از اجرا شدن تست با کلاس استثنای ConveterException رو به رو شویم اما تست به گ.نه ای نوشته شده است که این استثنا throw نمیشود.

در تست سوم، ممکن است تایم که در new Date() در خط چهارم با new Date() در خط پنجم میگیریم، یکی نباشد و به این دلیل تایم های متفاوت، این تست در اکثر مواقع fail شود(اگر به تایم های کوچکتر حساس باشد، احتمال fail شدن بیشتر میشود.)

در تست چهارم چون ما میخواهیم تا حد امکان از overlap بین تست های مختلف جلوگیری کنیم، این قاعده در این تست ها رعایت نشده است و تست اول در این قسمت با تست دوم در همین کلاس TestCalculator با هم overlap دارند و از فرض تست اول در نتیجه تست دوم استفاده شده است. با بیشتر شدن تست ها و وابستگی ها ممکن است به مشکلاتی بخوریم که debug کردن آنها کار بسیار مشکلی شود، پس تا حد ممکن(که در اینجا قابل رعایت شدن بود) باید تست ها را مستقل از هم بنویسیم.