گزارش‌کار تمرین کامپیوتری اول آزمون نرم‌افزار

مجید فریدفر - 810199569
اولدوز نیساری - 810199505

لینک ریپازیتوری: https://github.com/maj2idfar/Baloot
هش آخرین کامیت: dc3d287

سوال 1

عده ای معتقدند که متد های پرایوت نباید تست شوند و برای نظر خود استدلال های زیر را ارائه میدهند:

1) متد های پرایوت بخشی از پیاده سازی درونی متد ها هستند و به صورت مستقیم قابل دسترسی نیستند.این مسئله در کنار این که encapsulation یکی از اصول برنامه نویسی شی گراست و بخش های پرایوت باید پنهان بمانند باعث می شود عده ای معتقد باشند که متد های پرایوت نباید تست شوند.
2) متد های پرایوت اکثرا به جزئیات توجه دارند و تست کردن آن ها باعث می‌شود که به جزئیات بیش از اندازه توجه کنیم و توان زیادی برای آن ها بپردازیم . مثلا اگر بخشی از این متد های پرایوت تغییر کنند تمام تست ها باید تغییر کنند ، که این سختی و هزینه زیاد موجب می‌شود که تغییر کد دشوار تر شود و آزادی عملمان برای تغییر کد از دست برود.
3) متد های پرایوت به صورت غیر مستقیم جاهای دیگر تست می شوند ، تست کردن جداگانه آن ها احتمال انجام کار تکراری را بیشتر می کند .

البته با وجود این دلایل عده دیگری معتقدند که باید تست های پرایوت تست شوند و به عبارت دیگر معتقد هستند شرایطی وجود دارد که تست کردن متد های پرایوت می تواند مفید باشد ، پس میتوان گفت اگر در شرایط زیر باشیم باید تست کنیم :

مثلا اگر خود متد دارای پیچیدگی هایی باشد یا اگر بخواهیم کد را از اول بنویسیم و بخواهیم مشکل کد را به صورت دقیق متوجه شویم . یا در شرایطی که کل پیاده سازی ما به صورت test driven باشد .

سوال 2

با این که تست کردن کد های چند ریسه ای با استفاده از unit test دشوار است اما می‌توان گفت که امکان پذیر است اما مهم این است که چند نکته را رعایت کنیم :

1) تا جای ممکن واحد ها را کوچک کنیم و در بخش های موازی آن ها را تفکیک کنیم.
2) باید از شبیه ساز ها برای شبیه سازی برای شبیه سازی این وابستگی ها و ارتباطات بین ریسه ها استفاده میکنیم. این شبیه ساز ها کمک میکنند که رفتار هر یونیت را وقتی در تعامل با یونیت های دیگه است بتوانیم بررسی کنیم.
3) از ابزار هایی مثل سمافور ها یا قفل ها استفاده کنیم تا کنترل دسترسی یونیت ها را داشته باشیم.

سوال 3

1) اصلا از assertion استفاده نکرده است ، لذا برای آن که متوجه حالت هایی که fail می شوند بشود باید در آخر لاگ ها را بخواند که کار بهینه ای نیست .
   برای اصلاح میتواند یک نمونه از کلاس someclass بگیرد و سپس متد را روی آن اجرا کند و نتیجه را با نتیجه مورد انتظارمقایسه کند.
2) در این تست نیز همیشه exception ارسال میشود(یعنی عبارت expects Exception بیجا استفاده شده است .) و همیشه تست fail می شود برای اصلاح این مشکل باید از assertThrow استفاده کنیم .
3) بخش های initialize باید از قبل از اجرای هر کدام از تست ها انجام شوند و از آنجایی که تست ها ترتیب ندارند باید خط های initialization در before یا beforeall قرار بگیرند.