پروژه اول درس تست

آرین حدادی

امیر پور محمدی علی

سوال 3) در رابطه با تست کردن متد های private می توان گفت که یونیت تست متد های private در کل توصیه نمی شود و در سایت <http://shoulditestprivatemethods.com> هم که در صورت پروژه به آن اشاره شده است(که Kent Beck آن را توییت کرده است) هم صراحتا در جواب سوال  
 Should I test private methods گفته است نه.

در حالت کلی هم نیازی به تست آن ها نخواهد بود زیرا میتوان با تست متد های پابلیک که خودشان برای پیاده سازی منطق خود متد های private را صدا میزنند، از عملکرد درست آن متد های private نیز اطمینان حاصل کرد.

یونیت تست در واقع مانند مشتری است.مشتری با متد های privateکاری ندارد بلکه از طریق رابط های public به برنامه دسترسی دارد و لذا همان ها هم تست می شوند.و اگر تست کردن یک بخشی از منطق برنامه از طریق رابط های public سخت است احتمالا هنگام عرضه نرم افزار هم استفاده از آن سخت خواهد بود و این نشان می دهد که نیاز است در طراحی نرم افزار بازبینی داشته باشیم.همچنین اگر تست کردن متدی public به قصد تست کردن متد های private به گونه ای باشد که متد public متد های private زیادی را صدا می زند و دقیق نمیتوانیم متد های private را یکی یکی تست کنیم و این باعث شود که بخواهیم متد های پرایوت را جداگانه تست کنیم، این حالت هم می تواند نشان دهنده این باشد که متد های ما از یکی از قوانین clean coding که بیان می کند یک متد نباید تعدادی زیادی responsibility داشته باشد و باید یک کار انجام دهد را در طراحی کد نقض کرده ایم و نیاز به اصلاحاتی در نحوه نوشتن کد داریم.

و لذا در حالت کلی تست کردن متد های پرایوت توصیه نمی شود و نیازی هم به آن ها نیست.

اما اگر برخلاف توصیه ها کسی بخواهد آن را تست کند یا باید آن را پابلیک کند یا از ریفلکشن استفاده کند.

ما در حین زدن پروژه در متد getVisitsInternal() به یک متد private برخوردیم اما آن را تست نکردیم.زیرا از این متد در متد public به نام getVisits() استفاده می شود و زمانی که این متد تست شده و درست عمل می کند لذا میتوان اطمینان داشت که این متد private هم به طور صحیح کار می کند که این حالت همان حالتی است که بالاتر هم ذکر شد که از طریق تست متد های public به صحت عملکرد متد های private پی ببریم.