



به نام خدا
تمرین کامپیوتری شماره ۲



810198537

ملیکا حیدری دستجردی

دانشکده مهندسی برق و

810199461

علی عطاءاللهی

کامپیوتر

بخش اول - پیاده‌سازی تست

بخش دوم - گزارش کار

سوال اول

☐ Constructor Injection:

از آنجا که این مورد باعث می‌شود موقع ایجاد instance از کلاس، فیلدهای لازم مقدارشان مشخص شود، این روش خوبی‌های خود را دارد. مثلاً مطمئن هستیم کلاس در وضعیت valid ای قرار دارد و خواهد داشت. همچنین کد قابل پیش‌بینی‌تر و نگهداری آن نسبتاً راحت‌تر می‌شود.

☐ Setter Injection:

این روش از Constructor Injection سخت‌گیری کمتری دارد چون لزوماً همه فیلدهای لازم را موقع ایجاد کلاس مقداردهی نمی‌کنیم و دستانمان بازتر است که در کدام موارد وابستگی تزریق کنیم. همچنین بعداً می‌توانیم آنها را تغییر دهیم.
بسته به شرایط الویت ما با یکی از این دو الگوی اول است.

□ Field Injection:

این روش آخرین ترجیح است. وابستگی‌ها مستقیماً به فیلدهای کلاس تزریق می‌شوند و کلاس با وابستگی‌ها coupled می‌شود. در عین حال مطمئن نیستیم که همه وابستگی‌ها ایجاد شده‌اند یا خیر. در مجموع نگهداری و تست کد سخت‌تر می‌شود. نمی‌توانیم مطمئن شویم همه وابستگی‌ها ایجاد شده‌اند.

سوال دوم

الف) چون مقدار حقیقی به فیلدها نمی‌دهند، صرفاً آنها را پر می‌کنند که تست بتواند پیش برود. یعنی انگار نقش بازی می‌کنند.

ب) انواع مختلفی دارند مثل:

- Dummy Objects:

وقتی به کار می‌رود که تابعی برای اجرا به یک پارامتر نیاز دارد اما از آن استفاده نمی‌کند. در واقع برای صحت signature تابع آنجاست.

- Stubs:

یک سری ریسپانس آماده به method call ها می‌دهند. به ما کمک می‌کنند رفتار وابستگی‌ها را حین تست کنترل کنیم.

- Mocks:

در واقع Stub های پیشرفته هستند. نه تنها ریسپانس‌های آماده تولید می‌کنند بلکه مطمئن می‌شوند کد در حال تست با این ریسپانس‌ها مطابق انتظار عمل می‌کند.

- Spies:

شبیه به Mock ها هستند ولی برای مانیتور کردن اینکه یک متد چطور و چند بار call شده است کاربرد دارند.

- Fakes:

پیاده‌سازی‌های ساده شده از یک وابستگی هستند که برای راحتی تست استفاده می‌شوند.

سوال سوم

در classical method تمرکز روی استیت نهایی است اما در mocklist آنچه برایمان مهم است interaction بین شیء‌ها و مولفه‌ها است. در classical صحت وضعیت نهایی بررسی می‌شود ولی در mocklist چک می‌کنیم اشیاء با وابستگی‌ها تعامل درستی داشته باشند.

نقاط قوت	نقاط ضعف	
classical	<p>پیاده‌سازی ساده‌ای دارد. برای سیستم‌هایی که deterministic هستند و خروجی مشخص دارند بسیار مناسب است.</p>	<p>با تعاملات داخلی کد هیچ کاری ندارد و کنترل آنها با این روش سخت است.</p>
mocklist	<p>برای سیستم‌های پیچیده‌ای که interconnected هستند مناسب است. می‌تواند با بررسی تعاملات بخش‌های داخلی، خطاها را زودتر شناسایی کرده و از ایجاد مسائل بزرگتر جلوگیری کند.</p>	<p>پیاده‌سازی آنها به خصوص برای سیستم‌های با وابستگی بالا نسبتاً سخت‌تر است.</p>

https://github.com/AliAtaollahi/Software_Testing_CA1/commit/593261efbc6177fcdde08d171e0e60841783a79d