

به نام خدا

درس تست نرم افزار

پروژه 1

محمد سعید صدیقی (810100179) / نیما تاجیک (810100104)

آخرین هش گیتهاب: ca95d749a4652f0a357f56932c37f0e45c002c77

آدرس گیت هاب پروژه: <https://github.com/nimaCod/Software-Test-Course-projects/tree/main/CA01>

1. برای بررسی شرایط اجرای تست است. یعنی `fixture` ما باید آماده باشد تا تست اجرا شود در غیر اینصورت اجازه اجرای ادامه تست را نمی دهیم. اگر شرایط وجود نداشت به این معنی نیست که کد دارای مشکل است بلکه به این معنی است که تست در آن شرایط معنای خاصی ندارد.
2. با استفاده از ابزار هایی مثل `ConcurrentUnit` می توانیم `Stress Test` بر روی قسمت ها مختلف کد که منابع مشترک دارند پیاده سازی کنیم. یعنی تعداد زیادی `thread` همزمان قسمت مشخصی از کد را اجرا کنند و ما `thread-safety` را بررسی کنیم.
3. با این روش عملاً تست اتومات نداریم و مجبوریم هر بار به شکل کاملاً دستی خروجی تست ها را بررسی کنیم و موفق یا ناموفق بودن آنها را مشخص کنیم. همچنین در تعداد تست بالا بررسی خروجی ها ممکن نیست. (بسیار زمان بر و دشوار است)
4. الف) در حالت کلی تست ها نباید به `exception` برخورد کنند برای بررسی این مورد باید از تابع `AssertThrows()` استفاده کنیم تا نتیجه نهایی اجرای آن قطعه کد را بررسی کنیم. (تست ها باید یا `pass` شوند یا `fail` و نباید با ارور مواجه شوند). در تست آورده شده در صورتی که کد ما درست باشد و `exception` بیاندازد تست `fail` می شود در صورتی که باید `pass` شود.
ب) تست ها نباید به یکدیگر وابسته باشند زیرا لزوماً به ترتیب اجرا نمی شوند و باید مستقل از هم قابلیت اجرا شدن داشته باشند. می توانیم از `BeforeEach` یا `BeforeAll` برای ست کردن یک `fixture` مشخص استفاده کرد و باقی تست ها را طبق آن `fixture` نوشت.