به نام خدا

گزارش کار و پاسخ سوالات پروژه ی دوم تست نرم افزار

سارا اسعدی-810197448

مهرناز شمس‌آبادی-810196

Dummy:

از dummy عموما برای پرکردن پارامتر های ورودی استفاده می‌شود. Dummy صرفا در توابع می‌گردد ولی هیچ وقت استفاده نمیشود.

Fake:

پیاده سازی ساده تر برای اجرای برنامه است و به جای اینکه از دیتابیس اصلی استفاده شود از دیتابیس درون مموری استفاده می-شود. این روش برای production مناسب نیست.

Stub:

جایگزینی است که مقدار value مطلوب برای تابع که توسط component دیگری مقدار دهی می‌شود را بر می‌گرداند

Spy:

همانند stub است به اضافه‌ی این سرویس که اطلاعاتی در مورد نحوه‌ی صدا زده شدن و مسیر پیمایش را ثبت می‌کند.

Mock:

mock ها می‌توانند رفتار object اصلی را به خوبی شبیه سازی کنند. آن ها علاوه بر مقدار خروجی انتظاراتی در مورد مسیر و صدا زده شدن ها را نیز در خود ذخیره کنند و در نهایت مورد بررسی و تست قرار بگیرد و در صورتی که مسیر مطابق فرض نباشد می‌تواند exception ایجاد کند

Verification:

در حالت state تنها مقدار خروجی تست برای ما اهمیت دارد و تغییراتی که ممکن است بر روی SUT و collaborator ها اعمال شود چک خواهد شد

اما در حالت behavioral مسیر پیموده شده و توابع صدا زده شده نیز مهم هستند در این حالت ما بررسی می‌کنیم که آیا خروجی از مسیر درستی به دست آمده است یا خیر و همچنین بررسی می‌کنیم تا در طی مسیر، همه ی فعالیت ها به درستی اتفاق افتاده باشد

برای طراحی تست دو رویکرد وجود دارد classical و mockisty

در روش classical به طور معمول از object های واقعی استفاده می‌شود و تنها زمانی که استفاده از object اصلی دشوار است و یا محدودیت های خاصی دارد از mock استفاده می شود

ولی در رویکرد mockisty به جای تمامی object ها از mock object استفاده می‌شود

برای تابع addPet کلاس Owner در سناریوی اول که حیوان جدید اضافه می‌شود تنها به بررسی اضافه شدن حیوان جدید به لیست حیوانات بسنده شده است

اما در سناریوی دوم که مقدار isNew را false در نظر گرفته‌ایم چک می‌کنیم که owner حیوان به درستی تغییر کرده باشد.

* برای تابع get کلاس petTimedCache در سناریو اول که حیوان مورد نظر در repository پیدا می‌شود تستی به صورت state verification نوشته شده است در این حالت تنها چک میکنیم که حیوان برگشتی همان حیوان مورد نظر ما باشد و مقدار id آن با مقدار id ورودی برابر باشد

در سناریوی دوم که حیوان مورد نظر در actualMap پیدا می‌شود در این حالت دوبار پشت سر هم تابع صدا زده می‌شود که در مرتبه ی اول حیوان در لیست actualMap قرار می‌گیرد و در حالت دوم که تنها شرط اول بررسی شده و pet مورد نظر را به ما بر می‌گرداند برای صحت سنجی این سناریو در حالت behavioral نیز چک می‌کنیم که با وجود صدا شدن دوباره ی تابع تنها یکبار pet از repository فرا خوانده شده باشد