

Chapter 8

Geçici Fixture Yönetimi

Test Fixture Terminolojisi

Fixture düzenlemeden bahsetmeden önce fixture tanımını yapmak gerekir.

Fixture Nedir?

Herşeyin kendisine bağlı olduğu SUT' u oluşturduğumuz çalışması için gerekli olan elemanların yerleştirildiği yere **fixture** denir.SUT'un çalışması için gerekli olan herşeye **test fixture** ve bu bilgilerin ayarlanması işlemine **fixture setup** denir.

Kitap diyor ki; **text fixture** ; SUT alıştırması yapmadan önce ayarladığımız şeylerdir.

Fresh Fixture Nedir?

Fresh Fixture stratejisinde, çalıştıracağımız her bir test için yeni bir fixture oluştururuz.Her Testcase nesnesi SUT çalışmadan önce kendi fixture'ını oluşturur.Her çalışmada böyle olur.Bu fixture'ı taze (fresh) yapar.

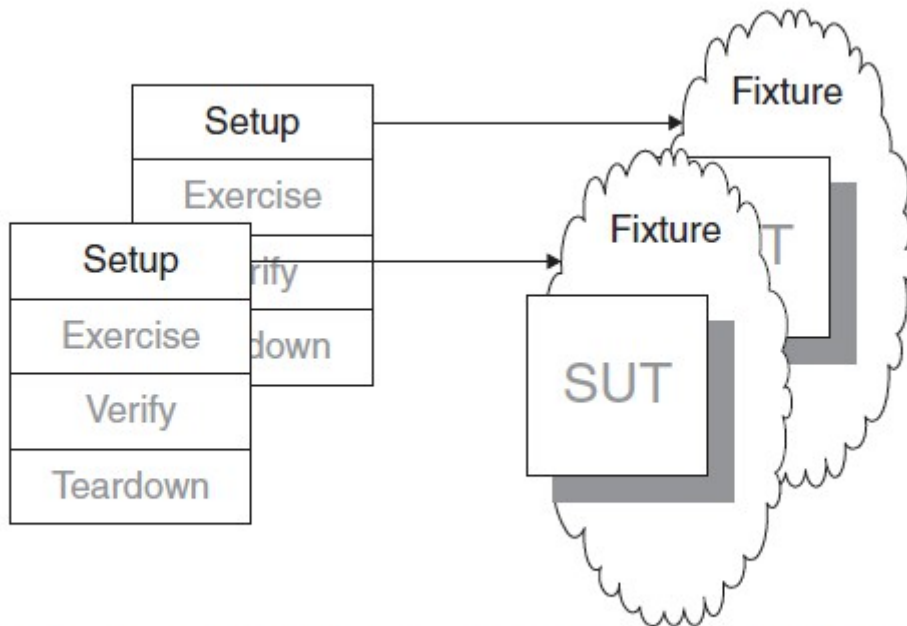


Figure 8.2 A pair of Fresh Fixtures, each with its creator. A| Fresh Fixture is built specifically for a single test, used once, and then retired.

Geçici Fresh Fixture Nedir?

Her testin sonunda *Garbage-Collected Teardown* tarafından silinen fixture'lardır.

Fresh Fixtureların Oluşturulması

Geçici veya Kalıcı Fresh Fixture'lar oluştururken mümkün olduğunca benzer fixturelar oluşturmalıyız.

Fresh Fixture için en iyi yol *In-line Setup* kullanmaktır, tüm fixture setup mantığı *Test Method*'tadır. Bu tür fixture'ı *Delegated Setup* kullanılarak da yapabiliriz, *Delegated Setup* mantığı *Test Utility Method*ları çağırımı yapar. Son olarak, *Implicit Setup* kullanabiliriz, *Test Atomasyon Frameworkü Testcase Class*'ımızda özel bir *setUp* methodu çağırır. Ayrıca biz bu üç yöntemin kombinasyonlarını kullanabiliriz.

In-line Fixture Setup

In-line Setup'ta , test fixture setup'ın hepsini *Test Method*'un içinde kullanır.

In-line Setup *Test Code Duplikasyonuna* sebep olur çünkü her *Test Method* kendi SUT'ini ayarlamaya çalışır ve birçok *Test Method*'u benzer fixture'a ihtiyaç duyar.

Delegated Fixture Setup

Dublikasyonu önlemek için *Test Utility Method* sayesinde *setup fixture*'ın yeniden kullanılmasını sağlar. Ayrıca nesne verilerinin doğruluğunun kontrol edilmesini sağlar.

Implicit Fixture Setup

Testcase Class'ki tüm *Test Method*'ların paylaştığı fixture setup mantığıdır, tüm *Test Method*ları fixture oluşturmak için erişebilmelidir.

```
Airport departureAirport;
Airport destinationAirport;
Flight flight;
public void testGetStatus_initial() {
// Implicit setup
// Exercise SUT and verify outcome
assertEquals(FlightState.PROPOSED, flight.getStatus());
}

public void setUp() throws Exception{
super.setUp();
departureAirport = new Airport("Calgary", "YYC");
destinationAirport = new Airport("Toronto", "YYZ");
BigDecimal flightNumber = new BigDecimal("999");
flight = new Flight( flightNumber , departureAirport,
destinationAirport);
}
```

Bu metodu yanı sıra tüm *Test Method*'ları aynı fixture'a ihtiyaç duyuyorsa kullanabiliriz.

SetUp Methodun Yanlış Kullanımı

setUp metodu xUnit'in özelliğidir. The *setUp* method is xUnit's inheritance. *setUp* method'ta *General Fixture* oluşturmak için her biri bir *Test Method*'a ait farklı parçalar kullanılır ve bu da

testin yavaşlamasına sebep olur. SetUp Metodu kullanılırken test metotların ortak kısımlarını kullanmalıyız.

Hybrid Fixture Setup

Üç setup yönteminin kombine edilmesidir.

Geçici Fresh Fixtureların Silinmesi

Garbage-Collected Teardown metodu bunu bizim için halledecektir.