

## ONAT SUBAŞI

Unit test ve integration test kavramları yazılım geliştirme sürecinin önemli birer parçasıdır. İkisi de farklı amaçlar ve yaklaşımlarla, yazılımın beklenen şekilde çalıştığını doğrulamak için kullanılır.

## UNIT TEST

Unit test, yazılımın en küçük test edilebilir parçaları olan "unit"lerin, yani genellikle fonksiyonlar veya metodlar gibi bireysel bileşenlerin, bağımsız bir şekilde test edilmesidir. Unit testlerin amacı, bu küçük parçaların her birinin beklenen davranışı gösterip göstermediğini kontrol etmektir. Java'da unit test yazımı için en yaygın kullanılan çerçevelerden biri JUnit'tir.

Unit testler yazılım geliştirme sürecinin erken aşamalarında hataları bulmaya, kodun daha sağlam ve bakımı kolay hale gelmesine yardımcı olur. Ayrıca, yazılımın farklı parçalarını geliştiren ekipler arasında güven oluşturur.

Örnek: Bir hesaplama işlemi yapan basit bir metod ve bu metodu test eden bir unit test.

```
/ Hesaplama işlemi yapan metod
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }
}

// Unit test
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;

public class CalculatorTest {

    @Test
    public void testAdd() {
        Calculator calculator = new Calculator();
        int result = calculator.add(5, 3);
        Assertions.assertEquals(8, result);
    }
}
```

Bu örnekte, `Calculator` sınıfındaki `add` metodunun doğru çalışıp çalışmadığını test eden bir unit test görüyoruz. Test, metodun 5 ve 3'ü doğru şekilde toplayıp toplayamadığını kontrol eder.

## INTEGRATION TEST

Integration test, birbirleriyle etkileşim halinde olan iki veya daha fazla bileşenin (sınıf, modül vb.) birlikte doğru çalışıp çalışmadığını kontrol etmek için yapılan testlerdir. Bu testler, unit testlerin aksine, izole edilmiş bileşenler yerine bileşenler arasındaki arayüzleri ve etkileşimleri test eder. Integration testler, sistemdeki farklı parçaların birlikte çalışabilme yeteneğini doğrulayarak, entegrasyon hatalarını erken aşamada tespit etmeye yardımcı olur.

Integration testler, genellikle daha karmaşık yapıdadır ve test ortamının kurulumu daha zor olabilir. Bu testler, API'ler, veritabanları, dosya sistemleri ve diğer dış sistemler gibi dış bağımlılıkları içerebilir.

Java'da integration testler yazmak için `JUnit` ile birlikte `Spring Test`, `TestNG` gibi araçlar kullanılabilir.

Örnek: İki sınıfın birbirleriyle entegre bir şekilde çalışmasını test eden basit bir integration test.

```
public class Database {
    public boolean save(String data) {
        return true;
    }
}

public class DataService {
    private Database database;

    public DataService(Database database) {
        this.database = database;
    }

    public boolean saveData(String data) {
        return database.save(data);
    }
}

// Integration test
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;

public class DataServiceIntegrationTest {

    @Test
    public void testDataServiceWithDatabase() {
        Database database = new Database();
        DataService dataService = new DataService(database);
        boolean result = dataService.saveData("test data");
        Assertions.assertTrue(result);
    }
}
```

Bu örnekte, `DataService` sınıfının `Database` sınıfı ile birlikte doğru çalışıp çalışmadığını test eden basit bir integration test görmekteyiz. Test, `DataService` sınıfının `saveData` metodunun, veriyi `Database` aracılığıyla doğru bir şekilde kaydedip kaydedemediğini kontrol eder.

Integration testler, yazılımın farklı parçalarının bir araya geldiğinde beklenen şekilde çalışıp çalışmadığını görmek için hayati öneme sahiptir. Bu testler, genellikle daha geniş bir test stratejisinin bir parçası olarak, unit testlerin üzerine inşa edilir ve yazılımın genel kalitesini artırır.

Unit test ve integration test arasındaki temel fark, unit testlerin bireysel bileşenleri bağımsız bir şekilde test etmesi, integration testlerin ise bileşenler arası etkileşimi ve entegrasyonu test etmesidir. Her iki test türü de yazılımın sağlıklı ve hata içermeyen bir şekilde geliştirilmesine katkıda bulunur.