

# JUNIT ile UNIT TEST

JUnit, testlerimizi gerçekleştirmek için geliştirilen açık kaynak kodlu bir test framework'ü olarak tanımlanabilir. Java için geliştirilen ve Unit Test olarak adlandırdığımız, birim testlerin yapılmasında geliştiricilere yardımcı olur. Birim test, tanım olarak uygulamamızı oluşturan alt parçaların, bileşenlerin test edilmesidir. Yazılan her program, kullanılan dil farkı gözetmeden girdiye karşılık bizlere çıktı üretir. Unit testler, oluşan veya oluşabilecek hataları görmemizi (test etmemizi) sağlayan önemli etkenlerdir.

JUnit4 ve JUnit5 ile birlikte geliştirdiğimiz uygulamaların birim testleri gerçekleştirilir. Bu testler sayesinde, beklenen çıktımızın kontrolü sağlanır. JUnit4 ve JUnit5 farklı anotasyonlara sahiptir. Bunu aşağıdaki tabloda görebiliriz.

İşlemler	JUnit4	JUnit5
Metodun, test olduğunu belirtmek için	@Test	@Test
Geçerli sınıfta bulunan tüm test metotlarından önce yürütüleceğini belirtmek için	@BeforeClass	@BeforeAll
Geçerli sınıfta bulunan tüm test metotlarından sonra yürütülmesini belirtmek için	@AfterClass	@AfterAll
Her test metodundan önce yürütülmesini istediğimiz yerlerde	@Before	@BeforeEach
Her test metodundan sonra yürütülmesini istediğimiz yerlerde	@After	@AfterEach
Bir test metodunu devre dışı bırakmak için	@Ignore	@Disabled
Dinamik test yapıları için	JUnit4 uygulanamaz	@TestFactory
İç içe testler için	JUnit4 uygulanamaz	@Nested
Etiket ve filtreleme için	@Category	@Tag

Yapısal olarak JUnit4, tek jar dosyası içerisinde herşeye sahiptir. JUnit5 ise JUnit Platform, JUnit Jupiter, JUnit Vintage olmak üzere 3 alt projeden oluşmaktadır.

JUnit4 tarafından, 3.parti eklentiler, IDE'ler desteklenmez. Fakat JUnit5, JUnit Platform sayesinde popüler eklenti ve IDE'leri destekler. Hali hazırda bulunan JUnit4 testlerini kolayca JUnit5 içerisinde bulunan Vintage kütüphanesi ile kolayca JUnit5 testlere çevirebiliriz.

Test olarak işaretlenmiş metotları Assert(iddaa) kavramı ile kontrol ederiz. Assert kavramını bu testler ile beraber sıkça duyarız. Mantık olarak aynı fakat yazı olarak JUnit4 ve JUnit5 arasında farklılıklar olabilir. Assert kavramları statik olarak oluşturulur ve genellikle beklenen değer, çıktı ile oluşan değer, hata mesajlarından oluşur. Böylece hatanın tespitini ve açıklayıcı olarak yazdığımız hata mesajları ile hata olan yeri kolayca düzeltebiliriz. Bu assert kavramlarının JUnit4 ve JUnit5 için içerisindeki parametrelerin yeri farklıdır. Örnek vermek gerekirse assertEquals metodunu ele alabiliriz. Bu kavram 2 değerin eşit olup olmadığını test eder. JUnit4 için assertEquals(String mesaj,

long beklenen deęer, long şuan gelen deęer) olarak ifade edilirken, JUnit5 için ise assertEquals(long beklenen deęer, long şuan gelen deęer, String mesaj) olarak ifade edilir.

JUnit5 ve JUnit4 karşılaştırıldığı zaman, JUnit5 yeni ve geliştiricileri daha özgür kıldığı için JUnit5'in öne çıktığını söyleyebiliriz. JUnit5 ile geliştirilebilir bir mimariye sahip olurken aynı zamanda geçmişte farklı versiyonlar ile yapılmış testleri kolayca JUnit5 tarafına çevirmemize olanak sağlar. Son olarak JUnit geliştiricilere TDD odaklı geliştirme yapmayı sağlar.