

30 **D**AKİKADA **S**ELENİUM-5 **J**UNİT **F**RAMWORK **O**LUSTURMA



Ahmet Bulutluöz

Junit (Framework) Nedir ?

JUnit, Java programlama dili için **unit test framework**'u olarak bilinir.

Unit Test nedir ?

Unit Test, bir yazılımın en küçük test edilebilir parçasının, tek başına ve bağımsız olarak, test edilmesidir.

Unit Test yazılım testinin ilk seviyesidir ve bu testleri developer'lar kendileri yazar ve yürütürler.

Framework nedir ?

Framework, üzerine yazılım oluşturabileceğiniz bir yapıdır.

Framework, oluşturacağımız proje için bir temel görevi görür, böylece sıfırdan başlamaz, nasıl bir yapı oluşturmaliyim diye düşünmezsiniz.

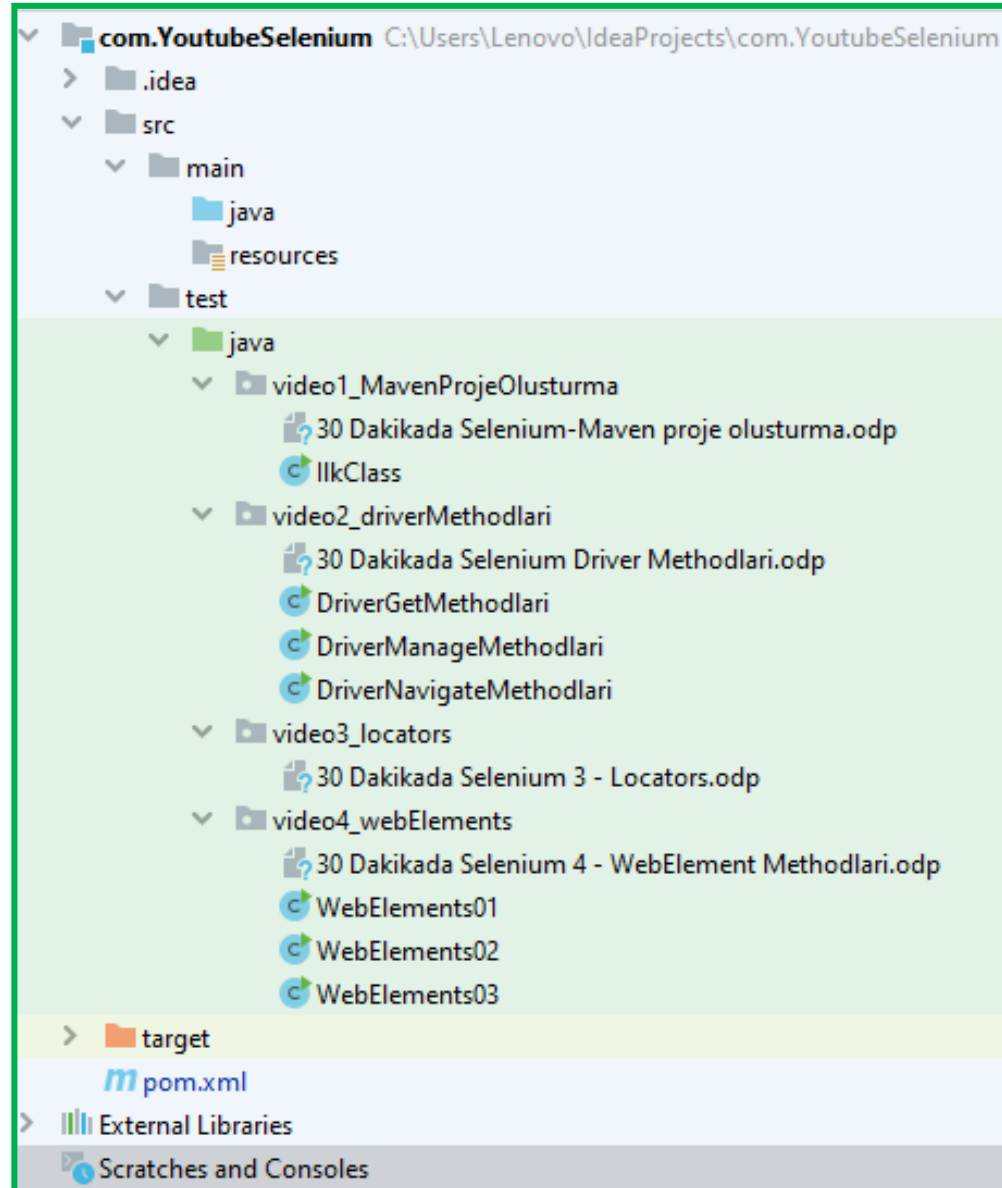
Framework'ler genelde belirli bir programlama dili veya tool ile ilişkilendirilir ve farklı görev türlerine göre şekilsel farklılıklar gösterirler.

Junit (Framework) Nedir ?

Junit bir Maven projesinde kullanabileceğimiz temel seviyedeki bir framework'dur.

Olusturdugumuz Maven projemizde JUnit framework'u kullanabilmek icin JUnit dependency'sini pom.xml'e eklememiz gerekir.

Ahmet Bulutluoz Mayıs 2022



JUnit (Framework) Nedir ?

JUnit bir Maven projesinde kullanabileceğimiz temel seviyedeki bir framework'dur.

Olusturdugumuz Maven projemizde JUnit framework'u kullanabilmek icin JUnit dependency'sini pom.xml'e eklememiz gerekir.

Ahmet Bulutluoz Mayıs 2022

JUnit Framework'unun avantajlari nelerdir ?

- 1- Açık kaynaklıdır.
- 2- Test method'larini bagimsiz olarak tanımlamak ve calistirmek için notasyonlar(**annotations**) sağlar.
(Boylece main method ihtiyacimiz kalmaz)
- 3- Expected Result ve Actual Result'i karsilastirarak Test etmemizi ve testin sonucunu otomatik olarak almamizi saglayan hazır assertion method'lari barindirir eder ve anında geri bildirim sağlar.
(Boylece if-else ile test yapma, sonuclari yazdirma gibi islemlere ihtiyacimiz kalmaz)
- 4- JUnit testleri, kodları daha hızlı yazmanıza olanak tanır ve bu da kaliteyi artırır.
- 5- JUnit, test düzgün çalışıyorsa **yeşil** olan bir çubukta test ilerlemesini gösterir ve bir test başarısız olduğunda **kırmızıya** döner
- 6- JUnit testleri, test senaryolarını ve hatta diğer test takımlarını içeren test takımları halinde organize edilebilir

JUnit dependency Ekleme?

JUnit framework'u kullanabilmek için, Maven ile olusturdugumuz bir projenin pom.xml'ine JUnit dependency eklememiz yeterlidir.

- 1- mvnrepository.com'dan uygun dependency'yi bulup kopyalayin
- 2- pom.xml'de dependencies taginin icine junit dependency'yi ekleyin.

```
<dependencies>
<!-- Ekleyecegimiz tum dependency'leri dependencies taginin icine eklemeliyiz
aksi takdirde projemiz calismaz -->
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>4.1.3</version>
</dependency>
<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>webdrivermanager</artifactId>
  <version>5.1.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/junit/junit -->
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.13.2</version>
  <scope>test</scope>
</dependency>
</dependencies>
```

Junit Annotations

JUnit, kodların daha anlaşılır olması ve framework'un daha iyi çalışabilmesi için küçük bilgiler içeren(meta-data) notasyonlara sahiptir.

- 1- Notasyonlar variable, method, class veya package için kullanılabilir.
- 2- Notasyonlar kullanıldığı bileşene ilave özellikler kazandırır.
- 3- Notasyonlar ile derleyiciye(compiler) talimatlar verebiliriz.
- 4- JUnit ile en çok kullanılan notasyonlar;

@Test

@BeforeClass @AfterClass

@Before , @After

@Ignore

Bu notasyonları kodlarla anlamaya çalışalım