

TestNG - yə giriş

- TestNG Nədir?
- TestNG-nin üstünlükləri
- IntelliJ IDE-də TestNG Dependencies ilə Java Layihəsinin yaradılması
- İlk TestNG testinizi yaradırsınız
- TestNG testi həyata keçirilir

TestNG Nədir?

TestNG Java proqramları üçün test framework-dür. Bu, avtomatlaşdırılmış testlərin yazılması və icrası üçün geniş funksiyalar təqdim edən açıq mənbəli bir vasitədir. TestNG "Test Next Generation" deməkdir və JUnit kimi digər test framework-ləri ilə müqayisədə bir sıra üstünlüklər təklif edir.

TestNG Maven və Gradle kimi məşhur **build tool**-lar ilə yaxşı inteqrasiya edir, bu da test avtomatlaşdırmasını quraşdırma və yerləşdirmə proseslərinizə daxil etməyi asanlaşdırır.

TestNG-nin üstünlükleri

- Test paketlərini (**Suites**) və test işlərini (**Cases**) idarə edir
- Testlərin prioritetləşdirilməsinə kömək edir
- Testlərin qruplaşdırılmasına kömək edir
- Paralel icra (**Execution**)
- Hesabat (**Reporting**)

Demo

- IntelliJ IDE-də TestNG Dependencies ilə Java Layihəsinin yaradılması
- İlk TestNG testinizi yaradırsınız
- TestNG testi həyata keçirilir

TestNG XML

- TestNG XML faylı nədir? İstifadə...
- TestNG XML faylını necə yaratmaq olar?
- XML fayllarından testləri necə işə salmaq olar?
- TestNG Hesabatı (Report)

XML

TestNG XML, TestNG framework-də test işlərinin konfiqurasiyası və idarə edilməsi üçün istifadə olunan fayl formatıdır. XML (Extensible Markup Language) formatında yazılmış bu fayl TestNG tərəfindən oxunur və testlərin necə işlədilməsini və konfiqurasiya edilməsini müəyyən etmək üçün istifadə olunur.

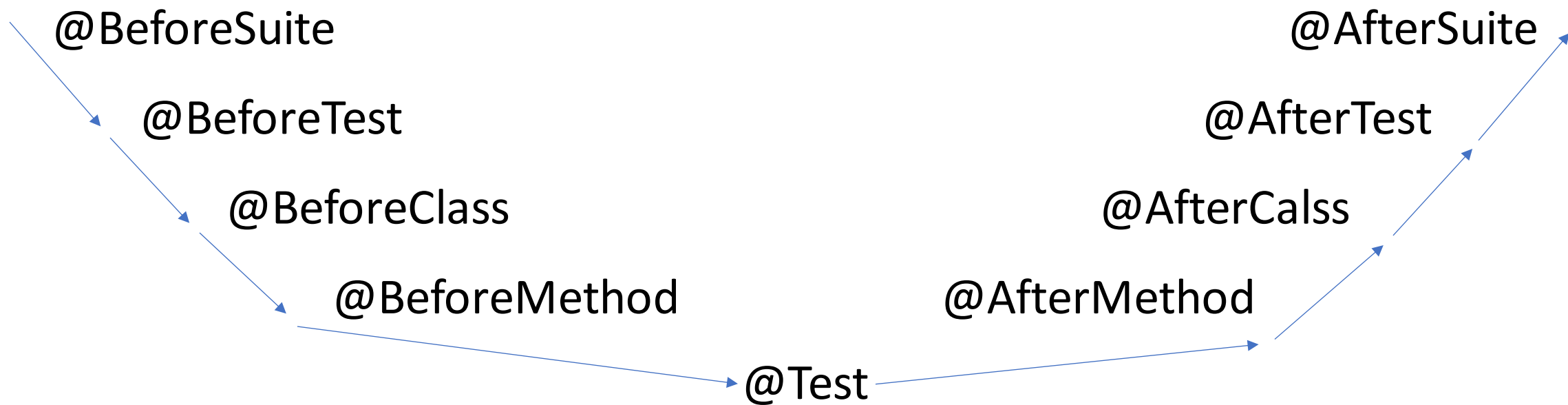
TestNG Annotasiyalar (Annotations)

TestNG annotasiyaları testləri konfigurasiya etmək və onların davranışını təsvir etmək üçün TestNG test freamework daxilində istifadə olunan xüsusi işarələrdir. Bu annotasiyalar Java kodunun müəyyən nöqtələrində yerləşdirilir və TestNG bu nöqtələrdə xüsusi əməliyyatlar həyata keçirir.

TestNG-də Annotations

- **@Test**: Bir metodu test metodu kimi qeyd edir. Bu işarələmə test case-lərin yaradılmasında istifadə olunur.
- **@BeforeSuite**: Test paketi (**suite**) başlamazdan əvvəl yerinə yetirilməli olan hərəkətləri müəyyən edir.
- **@AfterSuite**: Test paketi (**suite**) tamamlandıqdan sonra görülmək tədbirləri müəyyənləşdirir.
- **@BeforeTest**: Test başlamazdan əvvəl görülməli olan hərəkətləri müəyyən edir.
- **@AfterTest**: Test tamamlandıqdan sonra görülmək tədbirləri müəyyənləşdirir.
- **@BeforeClass**: Test Class-ı başlamazdan əvvəl görülməli olan tədbirləri müəyyən edir.
- **@AfterClass**: Test Class-ı başa çatdıqdan sonra görülmək tədbirləri müəyyənləşdirir.
- **@BeforeMethod**: Hər bir test metodu işə başlamazdan əvvəl görülməli olan tədbirləri müəyyən edir.
- **@AfterMethod**: Hər bir test metodu işlədikdən sonra görülmək tədbirləri müəyyənləşdirir.
- **@BeforeGroups**: annotasiya edilmiş metodu qrupa aid olan bütün test işlərinin icrasından əvvəl qrup üçün yalnız bir dəfə işləyir.
- **@AfterGroups**: annotasiya edilmiş metodu həmin qrupa aid olan bütün test işlərinin icrasından sonra qrup üçün yalnız bir dəfə işləyir.
- **@DataProvider**: Test metodlarına dinamik məlumat vermək üçün istifadə olunur.

TestNG annotations iyerarxiya (**hierarchy**)



Mövzular

- Testlərin prioritetləşdirilməsi (**Prioritizing Tests**)
- Testləri söndürmək (**Disabling Tests**)

Prioritizing Tests: TestNG, testlərin yerinə yetirilmə sırasını müəyyən edərək, onlara üstünlük verməyə imkan verir. Testlərin prioritetləşdirilməsi müəyyən kritik və ya asılı testlərin digərlərindən əvvəl yerinə yetirilməsini təmin etmək istədiyiniz zaman faydalıdır. TestNG testləri prioritetləşdirməyin bir neçə yolunu təqdim edir:

Disabling Tests: TestNG-də siz testin icrası prosesindən xaric etmək üçün xüsusi testləri söndürə bilərsiniz. Müəyyən testləri müvəqqəti olaraq istisna etmək istədiyiniz zaman, məsələn, onlar inkişaf mərhələsində olanda və ya əlavə araşdırmaya ehtiyac olduqda testləri söndürmək faydalıdır.

Mövzular

- TestNG-də Asılılıq (**Dependency**)
- AlwaysRun xüsusiyyəti

Dependency: TestNG-də **dependentOnMethods** atributu test metodları arasında asılılıqları müəyyən etmək üçün istifadə olunur. Bu, müəyyən bir test metodunun yalnız bir və ya bir neçə müəyyən edilmiş test metodu uğurla keçdiyi halda yerinə yetirilməli olduğunu müəyyən etməyə imkan verir.

AlwaysRun: TestNG-də **alwaysRun** atributundan asılı metodların və ya konfigurasiyaların uğur və ya uğursuzluğundan asılı olmayaraq, test metodunun həmişə icra edilməli olduğunu göstərmək üçün istifadə olunur.

Qruplaşdırma Testləri (Grouping Tests)

TestNG-də qruplaşdırma testləri müəyyən meyarlar və ya atributlar əsasında testləri müxtəlif qruplara bölmək təcrübəsinə istinad edir. TestNG bir və ya daha çox qrupa testlər təyin etmək funksiyasını təmin edir, bu qruplar əsasında testlərin seçmə icrasına, atlanmasına və ya ardıcılığına imkan verir.

TestNG-də testlərin qruplaşdırılması `@Test` annotasiyasının `groups` atributundan istifadə etməklə həyata keçirilir.

Qruplar əsasında testləri həyata keçirmək üçün siz ya XML konfiqurasiya faylından, ya da TestNG-nin program API-sindən istifadə edə bilərsiniz. XML konfiqurasiya faylında siz hansı qrupların icra olunacağını müəyyən etmək üçün `<groups>` teqindən istifadə edə bilərsiniz.

```
@Test(groups = "sanity")  
public void test1() {  
    // Test logic here  
}
```

```
@Test(groups = {"sanity", "regression"})  
public void test2() {  
    // Test logic here  
}
```

TestNG-də təsdiqləmə (Assertions)

TestNG-də təsdiqləmələr (**Assertions**) test hadisələrinizin gözlənilən davranışını yoxlamaq üçün istifadə olunur. Onlar sizə müəyyən şərtlərin testin icrası zamanı doğru olduğunu təsdiq etməyə imkan verir və uğursuzluqları və ya gözlənilməz nəticələri aşkar etmək və bildirmək üçün bir yol təqdim edir.

TestNG-də tez-tez istifadə olunan assertions method-lar bunlardır:

- `assertTrue(condition)`: Verilmiş şərtin doğru olduğunu yoxlayır.
- `assertFalse(condition)`: Verilmiş şərtin yanlış olduğunu yoxlayır.
- `assertEquals(expected, actual)`: Gözlənilən dəyərin faktiki dəyərə bərabər olduğunu yoxlayır.
- `assertNotEquals(expected, actual)`: Gözlənilən dəyərin faktiki dəyərə bərabər olmadığını yoxlayır.
- `assertSame(expected, actual)`: Gözlənilən və faktiki obyektlərin eyni nümunəyə istinad etdiyini yoxlayır.
- `assertNotSame(expected, actual)`: Gözlənilən və faktiki obyektlərin eyni nümunəyə istinad etmədiyini yoxlayır.
- `assertNotNull()`: Verilmiş obyektin null olmadığını yoxlayır.
- `assertNull(object)`: Verilmiş obyektin boş olduğunu yoxlayır.
- `fail()`: Testi uğursuzluq kimi qeyd edir.
- `assertAll()`:

Mövzu

- TestNG-də parametrlər (**Parameterization**)

TestNG-də DataProvider

TestNG-də **@DataProvider** annotasiyası test məlumatlarını test metodlarına təqdim etmək üçün istifadə olunur. Bu, müxtəlif məlumat dəstlərini test metoduna ötürməyə və onu müxtəlif giriş qiymətləri ilə dəfələrlə yerinə yetirməyə imkan verir. **@DataProvider** annotasiyası test məlumatlarını təmin edən metoda tətbiq oluna bilər və test metodu daha sonra həmin məlumat provayderindən məlumatları istehlak edə bilər.

TestNG-də Parallel Tests

TestNG-də siz testlərin daha sürətli icrasına nail olmaq və ümumi test ötürmə qabiliyyətini artırmaq üçün paralel olaraq testlər keçirə bilərsiniz. TestNG müxtəlif səviyyələrdə paralel icra üçün daxili dəstək təqdim edir, bu da testləri paket səviyyəsində, sinif səviyyəsində və ya metod səviyyəsində paralel olaraq həyata keçirməyə imkan verir.

Parallel Tests (Paralel Testlər)

- Paralel testlərin aparılmasının üstünlükləri
- Test **Method-ların** paralel olaraq aparılması (**Run**)
- Test **Class-ların** paralel olaraq aparılması (**Run**)
- Test **Suite-lərin** paralel olaraq aparılması (**Run**)

Paket səviyyəsində paralel testin icrasına bir nümunə:

```
<suite name="Test Suite" parallel="tests">  
  <test name="Test 1">  
  </test>  
  <test name="Test 2">  
  </test>  
</suite>
```

Siz həmçinin sinif səviyyəsində və ya metod səviyyəsində paralel icranı təyin edə bilərsiniz.

```
<suite name="Test Suite">  
  <test name="Test">  
    <classes parallel="classes">  
      <class name="com.example.TestClass1" />  
      <class name="com.example.TestClass2" />  
    </classes>  
  </test>  
</suite>
```

TestNG Listeners

- Dinləyicilər nədir? (**Listeners**)
- Dinləyicilərin növləri
- Dinləyicilərdən necə istifadə etmək olar?

Dinləyicilər nədir? (**Listeners**)

TestNG-də dinləyicilər (**Listeners**) testin icrası zamanı baş verən müxtəlif hadisələri dinləməyə və onlara cavab verməyə imkan verən xüsusi komponentlərdir. TestNG xüsusi dinləyicilər yaratmaq və istifadə etmək üçün bir yol təqdim edir ki, bu da testin icrası zamanı giriş, hesabat və ya fərdi yoxlama kimi hərəkətləri yerinə yetirməyə imkan verir.

TestNG dinləyiciləri testin həyata keçirilməsinin həyat dövründə müxtəlif hadisələrə uyğun gələn müxtəlif üsullara (**method**) malikdir.

Dinləyicilərin növləri? (Listeners)

- onStart(ITestContext context): **Test paketi başlamazdan əvvəl çağırılır.**
- onFinish(ITestContext contex): **Test paketi bitdikdən sonra çağırılır.**
- onTestStart(ITestResult result): **Hər bir test metodu başlamazdan əvvəl çağırılır.**
- onTestSuccess(ITestResult result): **Test metodu uğurlu olduqdan sonra çağırılır.**
- onTestFailure(ITestResult result): **Test metodu uğursuz olduqdan sonra çağırılır.**
- onTestSkipped(ITestResult result): **Test method-u atlandıqdan sonra çağırılır.**
- onTestFailedButWithinSuccessPercentage(ITestResult result): **Test metodu uğursuz olduqdan sonra çağırılır, lakin paket konfigurasiyasında müəyyən edilmiş müvəffəqiyyət faizi daxilində.**
- onTestFailedWithTimeout(ITestResult result): **Test metodu vaxtı bitdikdən sonra çağırılır.**

Müsahibə sualları:

1. TestNG-nin məqsədi nədir?

Nümunə cavab: Java üçün TestNG framework-i həm testçilər, həm də tərtibatçılar üçün nəzərdə tutulmuş açıq mənbəli qabaqcıl test avtomatlaşdırma framework-dir. TestNG-nin məqsədi avtomatlaşdırılmış testlər üçün istifadəsi asan, oxuna bilən, strukturlaşdırılmış, saxlanıla bilən və istifadəçi dostu mühiti təmin etməkdir. məlumat provayderləri kimi brauzerlər arası testi asanlaşdırın, çünki birdən çox cihaz və brauzerdə sınaqdan keçirə bilərsiniz. Bundan əlavə, çərçivədə proqramın gözlənilmədən dayandırılmasının qarşısını alan istisnaları idarə etmək üçün daxili mexanizm var.

2. TestNG-də metodu və ya kod blokunu keçmək mümkündürmü?

Nümunə cavab: Bəli, siz test annotasiyalarınızda 'enabled' parametrini 'false' olaraq təyin etməklə müəyyən test metodunu və ya kodu keçə bilərsiniz. Syntax: `@Test(enabled = false)`

3. TestNG-də test case-lər üçün prioritet təyin etməyin ən yaxşı yolu nədir?

Nümunə cavab: Test üsullarınızın sırasını prioritetləşdirmək prioritet sıranı müəyyən etməklə həyata keçirilə bilər. Nəticə etibarlı ilə, test prioritet dəstinə uyğun olaraq həyata keçiriləcək. Syntax: `@Test(priority=2)`

Müsahibə sualları:

4. TestNG digər test framework-lərindən nə ilə fərqlənir?

Nümunə cavab: TestNG sənayedə geniş istifadə olunan Java üçün sınaq çərçivəsidir. JUnit kimi digər çərçivələrlə müqayisədə qabaqcıl xüsusiyyətlər və funksiyalar təqdim edir. TestNG güclü annotasiyaları, çevik test konfigurasiyasını, paralel test icrasını, testin prioritetləşdirilməsini, verilənlərə əsaslanan testi və testdən asılılığın idarə edilməsini dəstəkləyir. O, həmçinin daha yaxşı hesabat imkanları və Maven və Gradle kimi **build** alətləri ilə integrasiya təklif edir.

5. TestNG annotasiyaları anlayışını izah edin.

Nümunə cavab: TestNG annotasiyaları TestNG çərçivəsinə təlimat və ya metadata təqdim etmək üçün istifadə olunur. Onlar test strukturunu müəyyənləşdirməyə və icra axınına nəzarət etməyə kömək edir.

6. TestNG-də asılılıq (**dependencies**) anlayışını izah edin.

Nümunə cavab: TestNG-də `@Test` annotasiyasında "dependsOnMethods" atributundan istifadə edərək test metodları arasında asılılıqları müəyyən edə bilərsiniz. Asılılıqları qeyd etməklə siz müəyyən test üsullarının yalnız digər müəyyən edilmiş test metodlarının uğurla icrasından sonra yerinə yetirilməsini təmin edirsiniz. Bu, testin icrası axınının idarə edilməsinə və test üsullarının istənilən ardıcılığının saxlanmasına kömək edir.

Müsaibə sualları:

9. TestNG istifadə edərək paralel olaraq testləri necə həyata keçirə bilərsiniz?

Nümunə cavab: TestNG paralel test icrasını dəstəkləyir, bu da testin icra müddətini əhəmiyyətli dərəcədə azalda bilər. Siz `<suite>` və ya `<test>` teqindəki "paralel" atributundan istifadə edərək paket, sinif və ya metod səviyyəsində paralel icranı konfigurasiya edə bilərsiniz. TestNG "metodlar", "siniflər" və ya "testlər" kimi müxtəlif paralel rejimlər təklif edir ki, bu da testlərin bir çox mövzu və ya proseslər arasında necə paylanmasına və icrasına nəzarət etməyə imkan verir.

8. TestNG-də ətraflı test hesabatlarını necə yarada bilərsiniz?

Cavab nümunəsi: TestNG ətraflı test hesabatları yaradan daxili hesabat imkanlarını təmin edir. Varsayılan olaraq, TestNG keçmiş və uğursuz testlər, icra müddəti və uğursuzluqlar üçün yığın izləri haqqında məlumatları ehtiva edən HTML hesabatı yaradır. Siz həmçinin hesabatın görünüşünü və funksionallığını artırmaq üçün TestNG-ni ExtentReports və ya Allure kimi digər hesabat alətləri ilə birləşdirə bilərsiniz.

Qeyd:

Unutmayın ki, bunlar sadəcə nümunə cavablardır. TestNG anlayışlarını və xüsusiyyətlərini hərtərəfli başa düşərək müsahibənizə hazırladığınızdan əmin olun. Uğurlar!