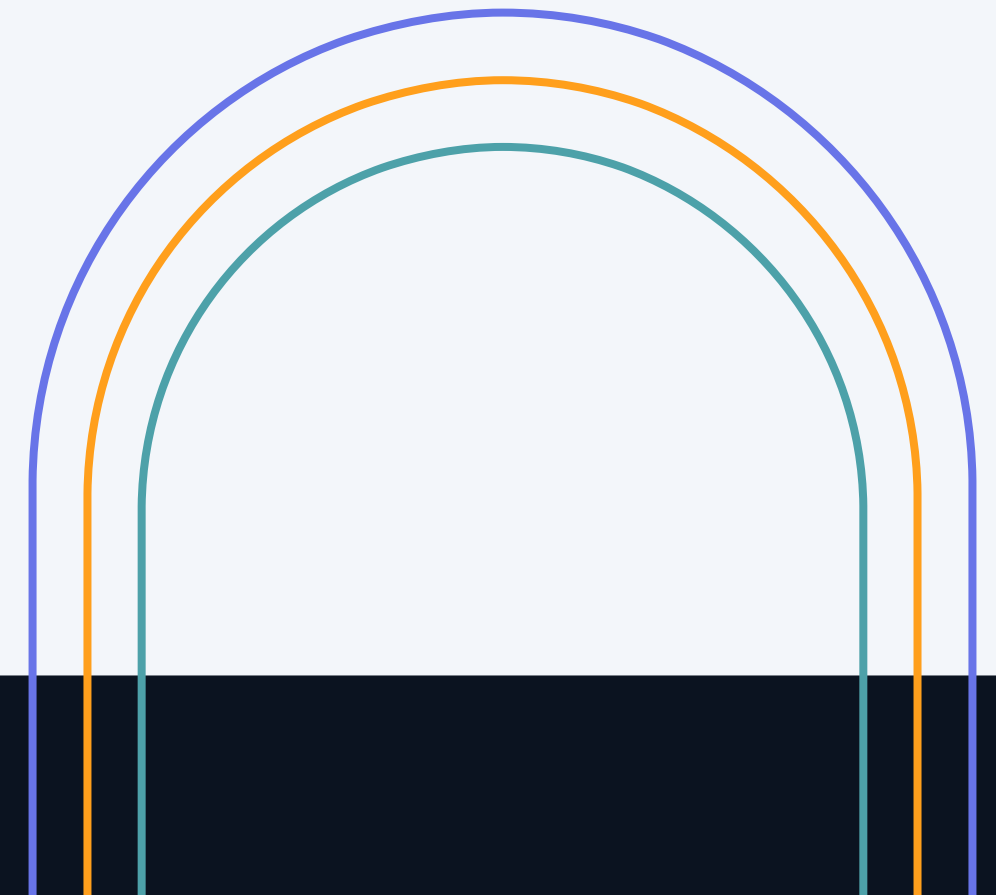


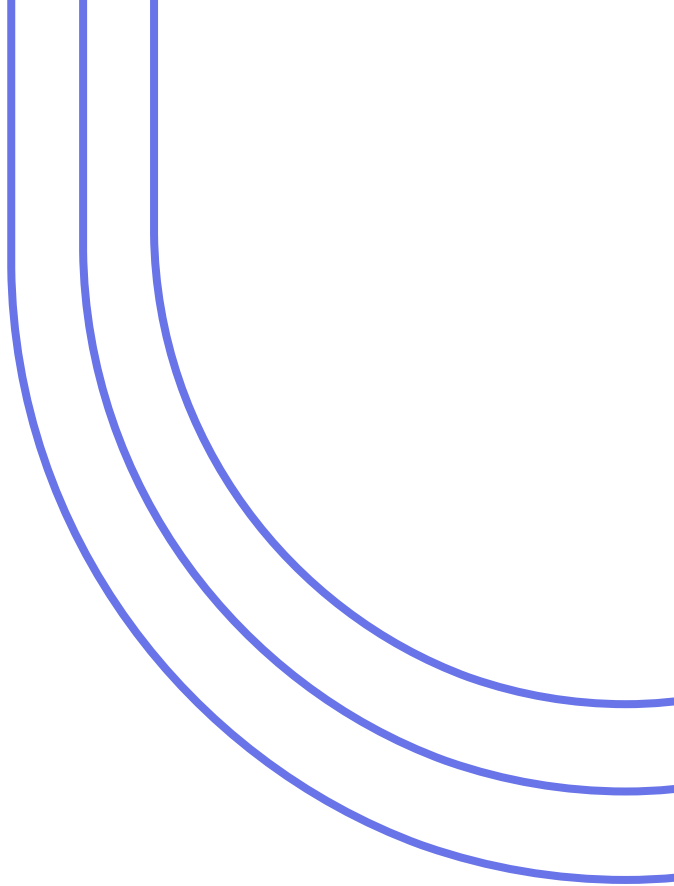
---



# Лекция 8. Тестовый фреймворк для Java TestNG.

- Системы автоматической сборки
- Gradle
- Пример написания теста с использованием TestNG



- 
- 01.** Определение
  - 02.** Системы автоматической сборки
  - 03.** Gradle
  - 04.** Интеграция проекта с IDE IntelliJ IDEA
  - 05.** Пример написания теста с использованием TestNG

# Содержание





---



# ОСНОВНЫЕ СВЕДЕНИЯ



01.



Определение

# 01. Определение

## TestNG

фреймворк для тестирования, написанный на языке Java. Он взял многое из таких фреймворков, как JUnit и NUnit, а также внедрил новые инновационные функции, которые делают его мощным и простым инструментом для тестирования.

TestNG предназначен для:

- Unit тестирования;
- Функционального тестирования;
- Интеграционного тестирования и т.д.

02.



# Системы автоматической сборки

## 02. Системы автоматической сборки

Включает в себя такие действия, как:

- сборка бинарного кода в исполняемый файл;
- выполнение тестов;
- разворачивание программы на производственной платформе;

Преимуществами систем автоматической сборки являются:

- Улучшение качества продукта;
- Ускорение процесса компиляции и линковки;
- Избавление от излишних действий;
- Минимизация некорректных сборок;
- Избавление от привязки к конкретному человеку;
- Ведение истории сборок и релизов для разбора выпусков;
- Экономия времени и денег благодаря причинам благодаря вышеперечисленной оптимизации;

### Автоматизация сборки

Этап процесса разработки программного обеспечения, заключающийся в автоматизации широкого спектра задач, решаемых программистами в их повседневной деятельности.

03.



Gradle



## 03. Gradle

### Определение

система автоматической сборки, построенная на принципах Apache Ant и Apache Maven, но предоставляющая DSL (язык описания данных, domain-specific language) на языке Groovy вместо традиционной XML-образной формы представления конфигурации проекта.

### Создаваемые файлы

```
plugins {  
    id 'java'  
}  
  
group 'radik.labs'  
version '1.0-SNAPSHOT' // версия проекта  
- начальная  
  
repositories {  
    mavenCentral() // подключение  
    репозитория с зависимостями  
}  
  
buildDir = "target"  
  
dependencies {  
    testCompile 'org.testng:testng:6.14.3' //  
    зависимости для тестов  
}
```

04.

Decorative orange lines and dots are present in the top right and bottom left corners of the slide. A horizontal white line extends from the '04.' text towards the right. Three orange dots are arranged horizontally in the top right corner. Several orange curved lines sweep across the slide, starting from the top right and bottom left edges.

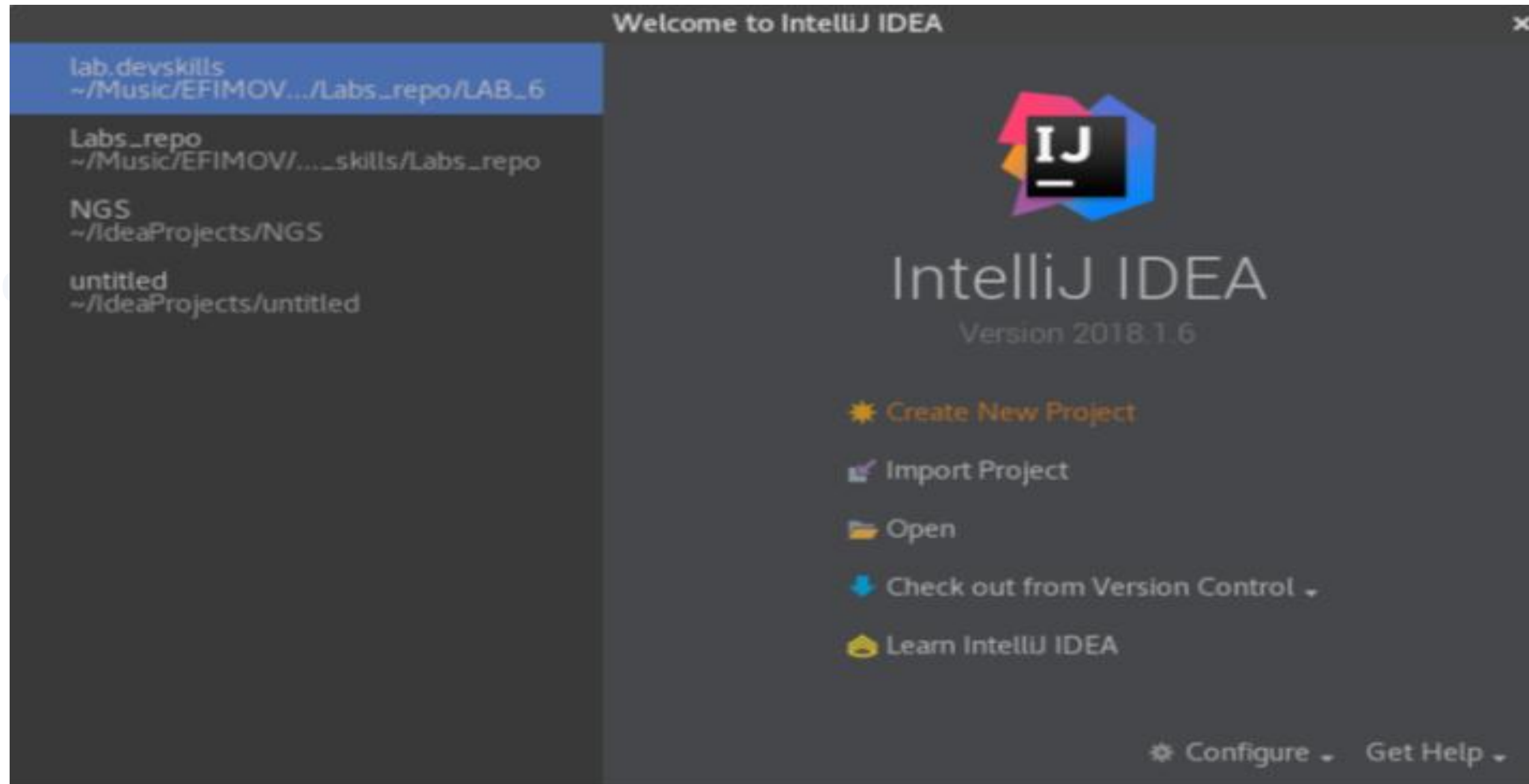
# Интеграция проекта с IDE IntelliJ IDEA

## 04. Интеграция проекта с IDE IntelliJ IDEA

---



Google Slides

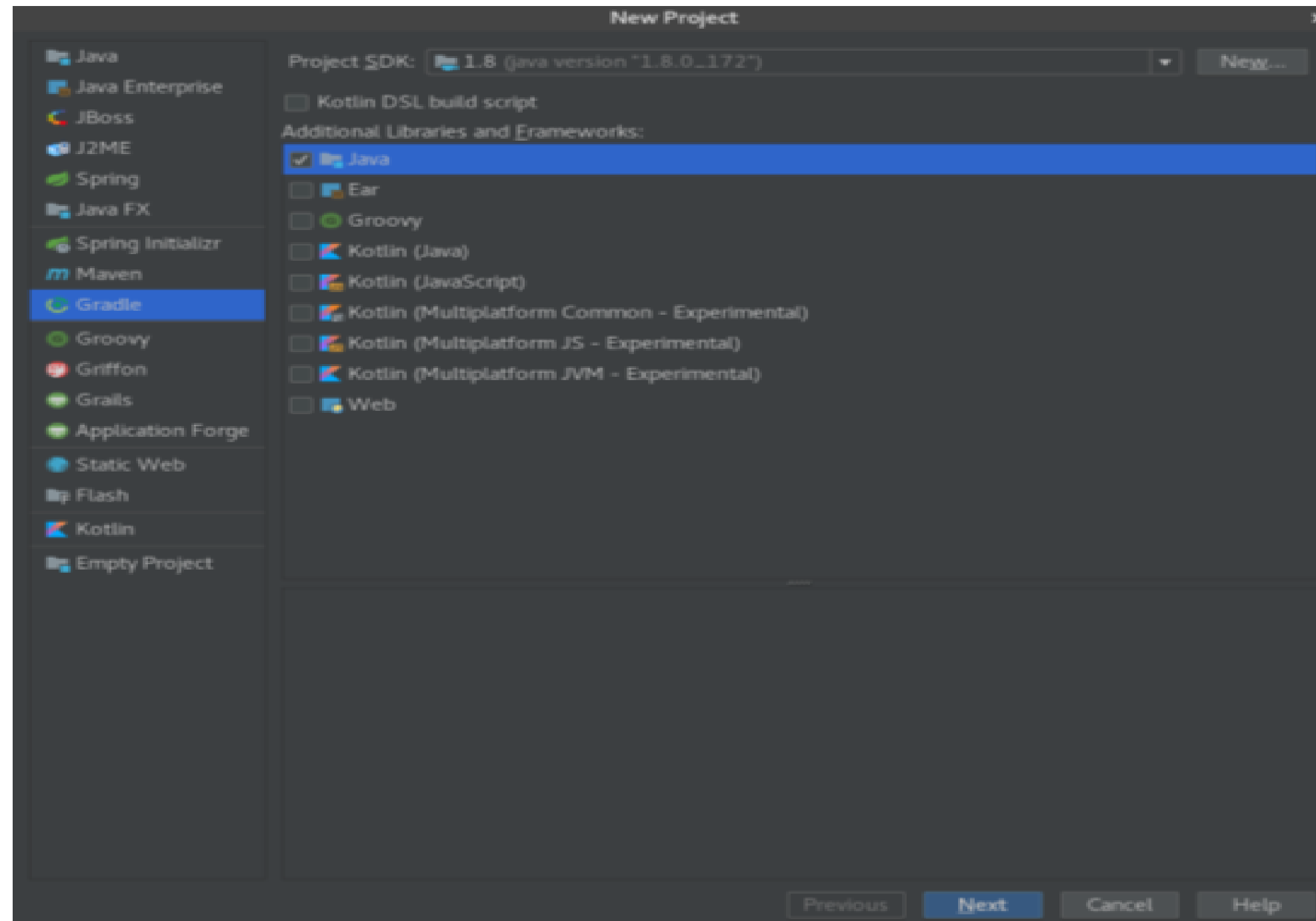


Создание проекта  
в IntelliJ IDEA

## 04. Интеграция проекта с IDE IntelliJ IDEA



Google Slides



# Создание проекта с использованием Gradle

05.



**Пример написания теста с  
использованием TestNG**

## 05. Пример написания теста с использованием TestNG



### Начало

Создаем Gradle проект и добавляем зависимости:

```
plugins {  
    id 'java'  
}  
group 'radik.labs'  
version '1.0-SNAPSHOT' // версия проекта  
- начальная  
repositories {  
    mavenCentral() // подключение  
    репозитория с зависимостями  
}  
buildDir = "target"  
dependencies {  
    testCompile 'org.testng:testng:6.14.3' //  
    зависимости для тестов  
}
```

### Содержимое класса Calc.java

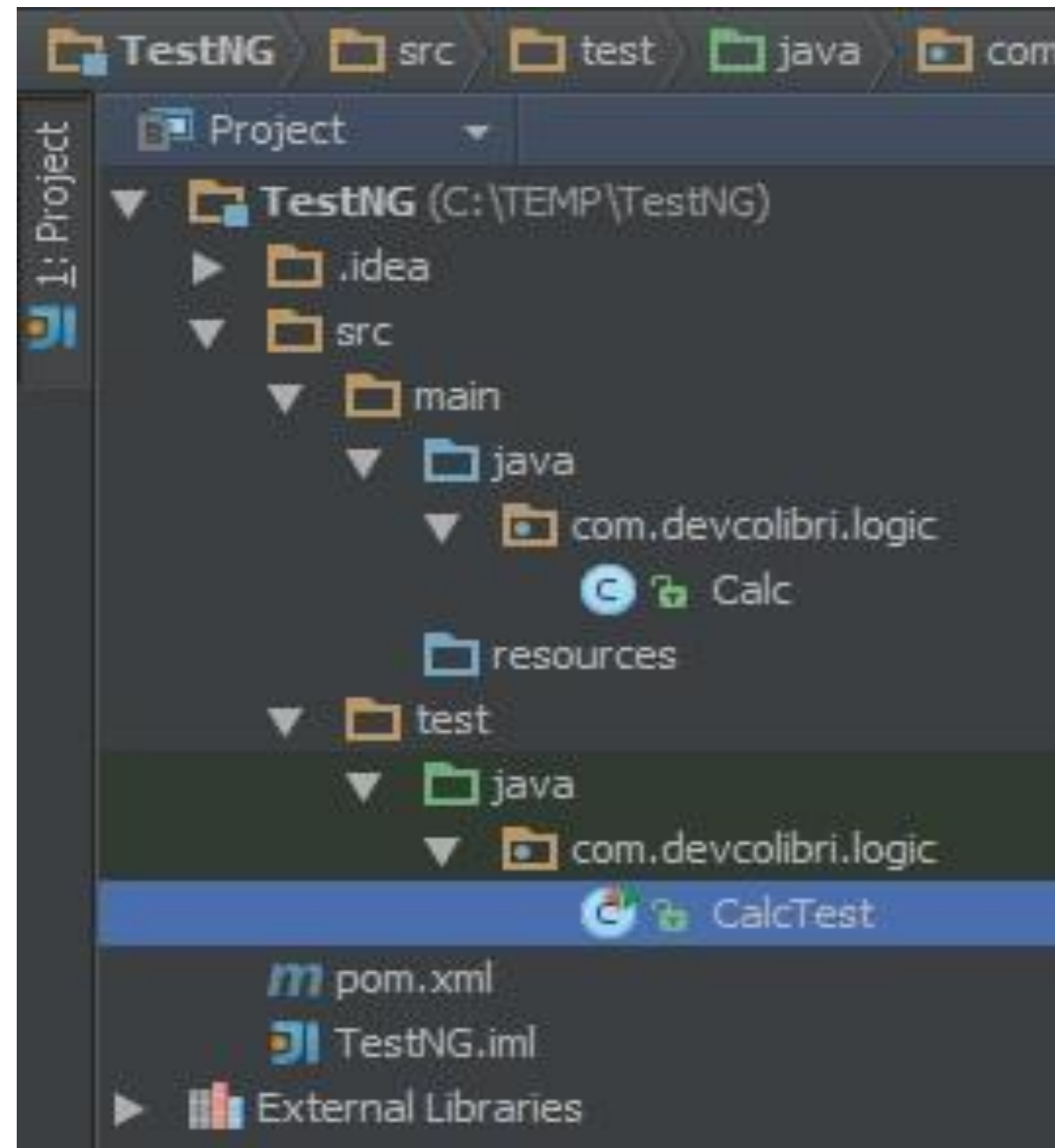
```
package com.devcolibri.logic;  
public class Calc {
```

```
    public int sum(int a, int b){  
        return a + b;  
    }  
}
```

Содержимое класса CalcTest.java:

```
package com.devcolibri.logic;  
import org.testng.annotations.Test;  
public class CalcTest {  
    private Calc calc = new Calc(); //  
    экземпляр класса который мы будем  
    тестировать  
    @Test  
    public void testSum() throws Exception {  
    }  
}
```

## 05. Пример написания теста с использованием TestNG



Вид проекта после создания класса Calc  
и CalcTest

## 05. Пример написания теста с использованием TestNG



```
package com.devcolibri.logic;

import org.testng.Assert;
import org.testng.annotations.Test;

public class CalcTest {

    private Calc calc = new Calc();

    @Test
    public void testSum() throws Exception {
        Assert.assertEquals(5, calc.sum(2, 3));
    }
}
```

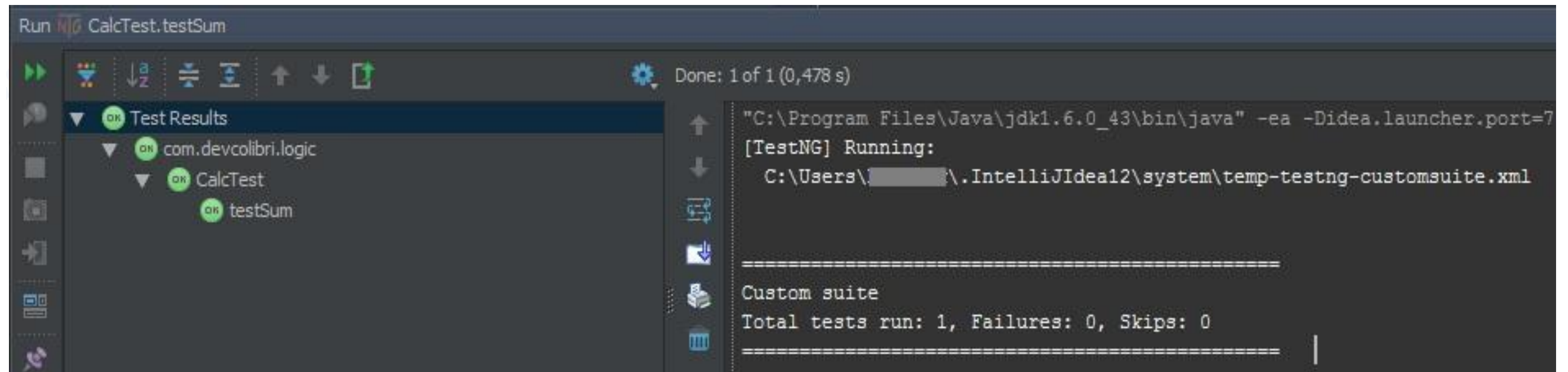
Context menu options:

- Copy Reference (Ctrl+Alt+Shift+C)
- Paste (Ctrl+V)
- Paste from History... (Ctrl+Shift+V)
- Paste Simple (Ctrl+Alt+Shift+V)
- Column Selection Mode (Alt+Shift+Insert)
- Find Usages (Alt+F7)
- Analyze
- Refactor
- Folding
- Go To
- Generate... (Alt+Insert)
- Add to Watches
- Process 'CalcTest.java' Annotations (Ctrl+Alt+Shift+F9)
- Compile 'CalcTest.java' (Ctrl+Shift+F9)
- Save 'testSum()' (highlighted with a red box)
- Run 'testSum()' (Ctrl+Shift+F10)** (highlighted with a red box)
- Debug 'testSum()'

Запуск теста



## 05. Пример написания теста с использованием TestNG



Результаты теста

## 05. Пример написания теста с использованием TestNG



Google Slides

```
"C:\Program Files\Java\jdk1.6.0_43\bin\java" -ea -Didea.launcher.port=  
[TestNG] Running:  
  C:\Users\barchuk\.IntelliJ IDEA12\system\temp-testng-customsuite.xml  
  
@BeforeSuite  
@BeforeTest  
@BeforeClass  
@BeforeMethod  
@AfterSuite  
  
=====  
Custom suite  
Total tests run: 1, Failures: 0, Skips: 0  
=====
```

Результат использования аннотаций

## 05. Пример написания теста с использованием TestNG

---



### Исключения

```
package com.devcolibri.logic;

import org.testng.Assert;
import org.testng.annotations.Test;

import java.util.List;

public class TestExpectedExceptionTest
extends Assert {

    @Test(expectedExceptions =
NullPointerException.class)
    public void testNullPointerException() {
        List list = null;
        int size = list.size();
    }
}
```

Google Slides

кто их сделал

## 05. Пример написания теста с использованием TestNG



### Timeout

```
package com.devcolibri.logic;

import org.testng.annotations.*;

public class TestNGTimeTest {
    @Test(timeOut = 1000)
    public void waitLongTime() throws
    Exception {
        Thread.sleep(1001); // заставляем
        поток заснуть на 1001 миллисекунду
    }
}
```

### Зависимые и независимые тесты

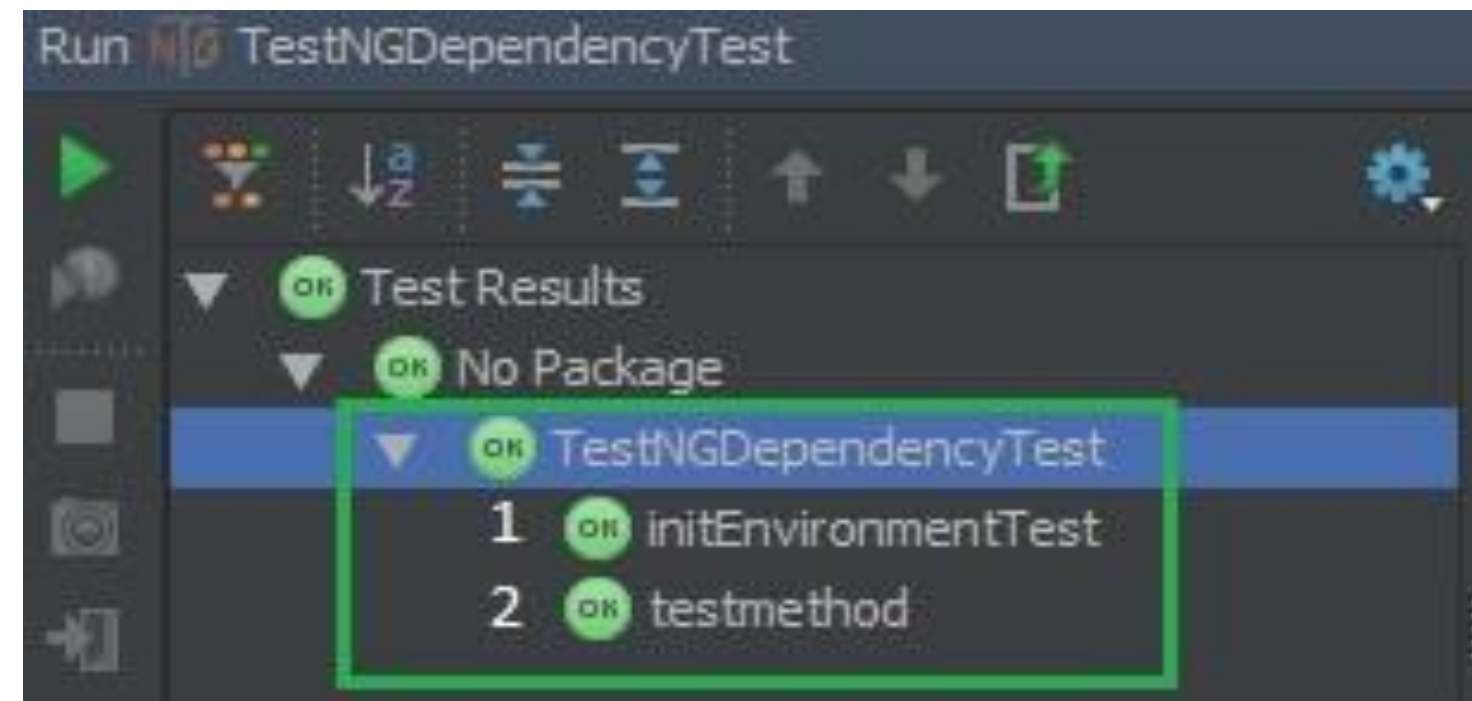
```
import org.testng.annotations.Test;

public class TestNGDependencyTest {

    @Test
    public void initEnvironmentTest() {
        System.out.println("This is
        initEnvironmentTest");
    }

    @Test(dependsOnMethods={"initEnvironm
    entTest"})
    public void testmethod() {
        System.out.println("This is
        testmethod");
    }
}
```

## 05. Пример написания теста с использованием TestNG



Как отображаются зависимые тесты

## 05. Пример написания теста с использованием TestNG



### Если нужно что – то посложнее

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="Suite">
  <test name="UnitTesting">
    <classes>
      <class
name="com.javarticles.testng.UnitLevelTest
ing" />
    </classes>
  </test>
  <test name="SomeFeature">
    <groups>
      <run>
        <include name="someFeature" />
      </run>
    </groups>
    <classes>
      <class
name="com.javarticles.testng.UITesting" />
    </classes>
  </test>
</suite>
```

### Игнорирование ТЕСТОВЫХ МЕТОДОВ

```
package com.devcolibri.logic;
import org.testng.annotations.*;
public class TestNGIgnoreTest {

    @Test(enabled = false)
    public void testsetProperty() {
        System.out.println("Этот метод
будет проигнорирован в момент
проведения тестирования!");
    }
}
```