**TestNG Attributes for Parallel Execution**

**1. invocationCount**

* This attribute allows a test method to execute **multiple times**.
* Helps in **stress testing** by running the same test multiple times.

📌 **Example:**

java

import org.testng.annotations.Test;

public class InvocationCountExample {

@Test(invocationCount = 5)

public void testMethod() {

System.out.println("Executing Test - " + Thread.currentThread().getId());

}

}

🔹 **Effect:** The test method will run **5 times sequentially**.

**2. threadPoolSize**

* Works with invocationCount to run **multiple instances in parallel**.
* Helps in **multi-threaded execution**.

📌 **Example:**

java

import org.testng.annotations.Test;

public class ThreadPoolExample {

@Test(invocationCount = 6, threadPoolSize = 3)

public void testMethod() {

System.out.println("Running Test - " + Thread.currentThread().getId());

}

}

🔹 **Effect:**

* The test will run **6 times**.
* A maximum of **3 threads** will run in parallel.

**3. timeOut**

* Sets a maximum time limit for a test.
* Useful for handling **long-running tests**.

📌 **Example:**

java

@Test(timeOut = 5000) // Test fails if it runs for more than 5 seconds

public void testWithTimeout() {

while (true); // Infinite loop (to demonstrate timeout)

}

🔹 **Effect:** If the test exceeds **5000ms (5s)**, it will fail.

**4. priority**

* Defines the **execution order** of test cases.
* Lower priority runs **first**.

📌 **Example:**

java

import org.testng.annotations.Test;

public class PriorityExample {

@Test(priority = 1)

public void testA() {

System.out.println("Test A");

}

@Test(priority = 0)

public void testB() {

System.out.println("Test B");

}

}

🔹 **Effect:** testB() executes **before** testA().

**5. dependsOnMethods**

* Ensures a test runs **only if** another test method **passes**.
* Helps maintain dependencies between test methods.

📌 **Example:**

java

import org.testng.annotations.Test;

public class DependsOnExample {

@Test

public void loginTest() {

System.out.println("Login Successful");

}

@Test(dependsOnMethods = "loginTest")

public void dashboardTest() {

System.out.println("Dashboard Loaded");

}

}

🔹 **Effect:**

* dashboardTest() runs **only if** loginTest() **passes**.
* If loginTest() **fails**, dashboardTest() will be **skipped**.

**Key Takeaways**

✔ **invocationCount** → Runs a test multiple times.  
✔ **threadPoolSize** → Enables parallel execution of multiple instances.  
✔ **timeOut** → Fails test if execution exceeds the specified time.  
✔ **priority** → Defines test execution order.  
✔ **dependsOnMethods** → Ensures dependent tests run **only if** the prerequisite test passes.