**Step 1: Create a New Java Project in Eclipse**

1. **Open Eclipse**.
2. Click on **File** > **New** > **Java Project**.
3. **Enter the project name** (e.g., ConcurrentTesting), and click **Finish**.

**Step 2: Add JUnit 5 Library to Your Project**

1. **Right-click on your project** in the **Package Explorer**.
2. Choose **Build Path** > **Add Libraries**.
3. Select **JUnit** and click **Next**.
4. **Select JUnit 5** and click **Finish**.
5. This will add the necessary JUnit 5 libraries to your project.

**Step 3: Create a Test Class for Concurrent Testing**

1. **Right-click on the src folder** in your project.
2. Choose **New** > **JUnit Test Case**.
3. **Name your test class** (e.g., ConcurrentTests), and ensure the **Test kind** is set to **JUnit 5**.
4. Click **Finish**.

This creates a test class for you. By default, it will add @Test methods.

**Step 4: Enable Concurrent Execution**

JUnit 5 allows for concurrent testing using @Execution(ExecutionMode.CONCURRENT).

**Add Concurrent Test Code to Your Class**

Replace the default test methods in your test class with the following code:

java

code

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.parallel.Execution;

import org.junit.jupiter.api.parallel.ExecutionMode;

import static org.junit.jupiter.api.Assertions.assertEquals;

@Execution(ExecutionMode.CONCURRENT)

class ConcurrentTests {

@Test

void test1() throws InterruptedException {

Thread.sleep(1000); // Simulate delay

System.out.println("Test 1 completed");

assertEquals(1, 1);

}

@Test

void test2() throws InterruptedException {

Thread.sleep(1000);

System.out.println("Test 2 completed");

assertEquals(2, 2);

}

@Test

void test3() throws InterruptedException {

Thread.sleep(1000);

System.out.println("Test 3 completed");

assertEquals(3, 3);

}

}

* The @Execution(ExecutionMode.CONCURRENT) annotation ensures that the tests run in parallel.
* Each test has a 1-second sleep to simulate work, and you can see concurrent execution.

**Step 5: Configure JUnit Platform for Parallel Execution (Optional)**

To enable more advanced parallel execution configurations, you can create a junit-platform.properties file:

1. **Right-click on src of the project**.
2. Select **New** > **File**.
3. Name the file junit-platform.properties.
4. Inside the file, add the following configuration:

properties

code

junit.jupiter.execution.parallel.enabled = true

junit.jupiter.execution.parallel.config.strategy = dynamic

junit.jupiter.execution.parallel.config.dynamic.factor = 4

This configuration will enable dynamic parallelism with up to 4 concurrent tests running at once.

**Step 6: Run the Tests**

1. **Right-click on your test class** (ConcurrentTests.java) in **Package Explorer**.
2. Select **Run As** > **JUnit Test**.
3. The JUnit view will show the results of your concurrent tests.

You should see output similar to the following, demonstrating that the tests are executed concurrently:

Copy code

Test 1 completed

Test 2 completed

Test 3 completed