

# Lab 12. Unit Testing with JUnit 5

Authors: Yida Tao, Yao Zhao

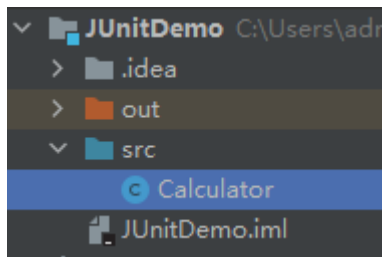
## Setup JUnit as a Maven Project

JUnit is essentially a dependency to your project, which could be downloaded and managed using Maven. Please follow this [official guide](#) of IntelliJ IDEA to create a Maven project, add a `Calculator.java`, create test cases and execute them.

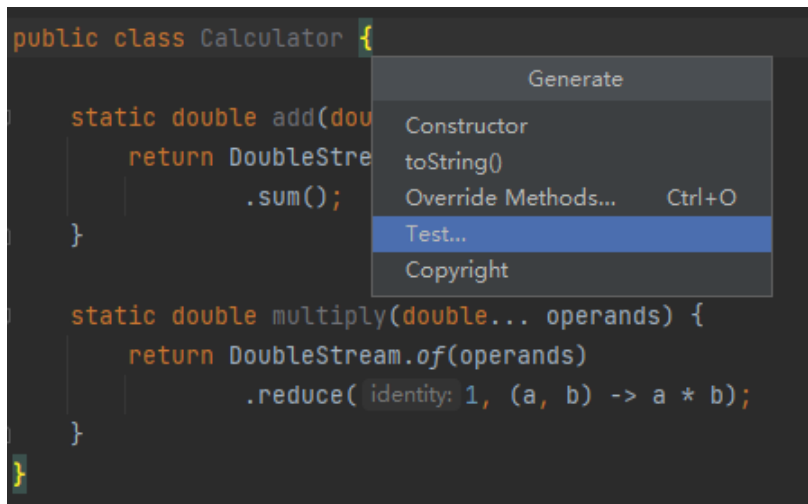
## Setup JUnit for an Existing Java Project

If you already created a Java project, not Maven project, and wanted to use JUnit on this project, you could always download necessary `.jar` files of JUnit and add them to the classpath, as shown below.

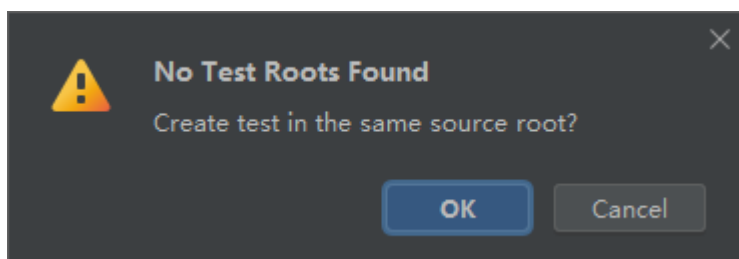
Suppose we have a Java project:



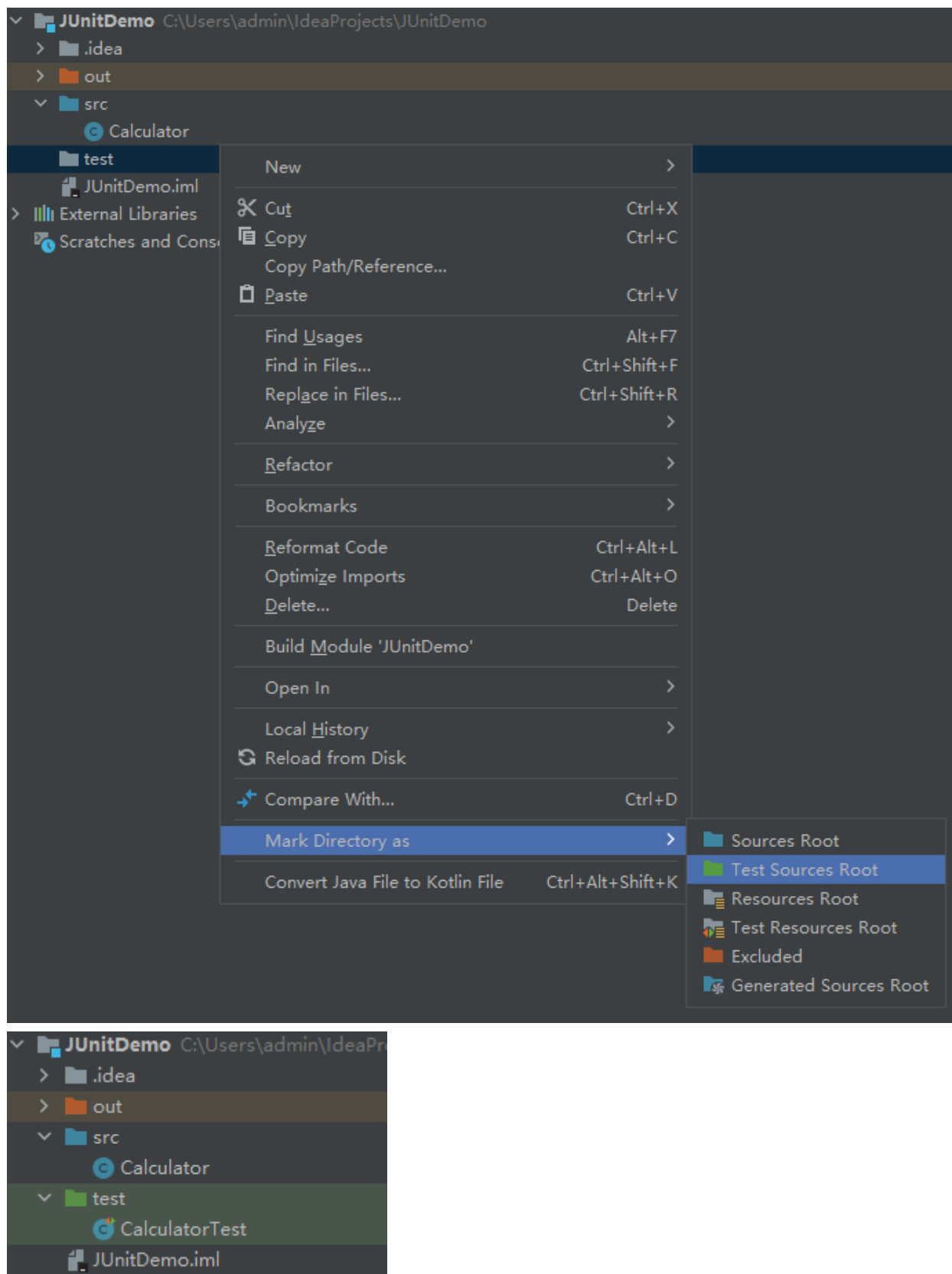
Inside `Calculator.java`, right click -> Generate -> Tests:



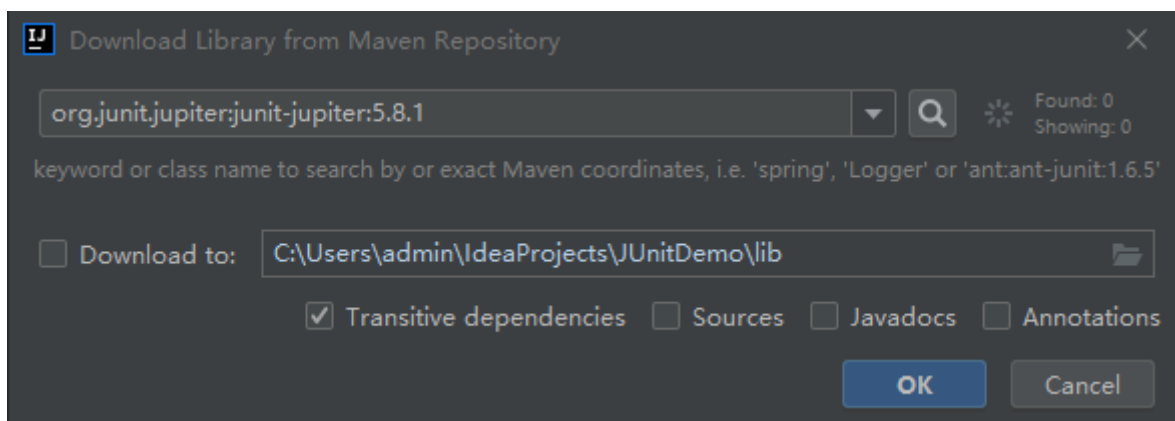
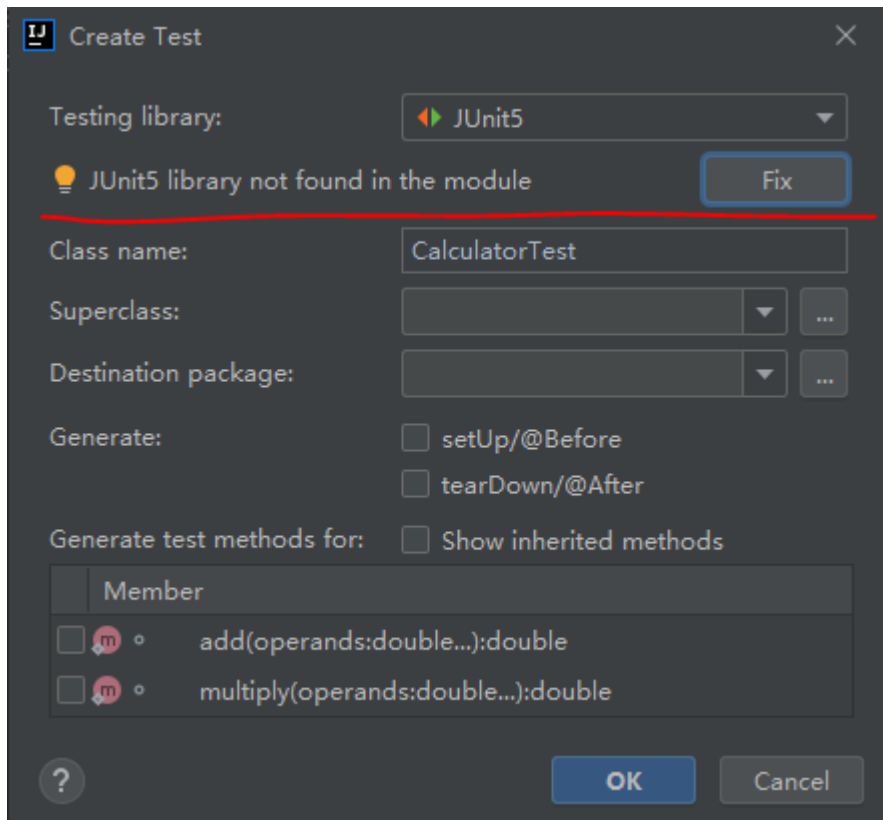
You'll probably see a popup like this. If you choose OK, the test will be put under the same directory as `Calculator.java`.



This is fine for a toy project. However, as a convention, `test` directory is typically separated from the `src` directory. We could create a `test` directory and mark it as the "test root". This way, generated tests will be put into this test root automatically, with the same package structure as those in `src`.



When you generate tests, if you haven't configured JUnit, click "Fix" and "OK" to download JUnit.



After downloading JUnit and adding it to classpath, you should be good to go.

## Executing Sample Tests

First, download `Calculator.java` and `Person.java` from Sakai. Then, download `StandardTests.java` and `AssertionDemo.java`. You should be able to execute these tests and understand their execution results.