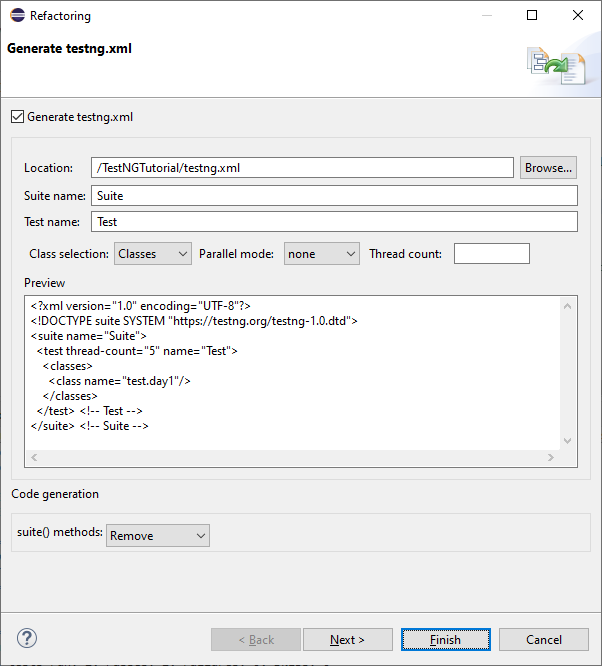
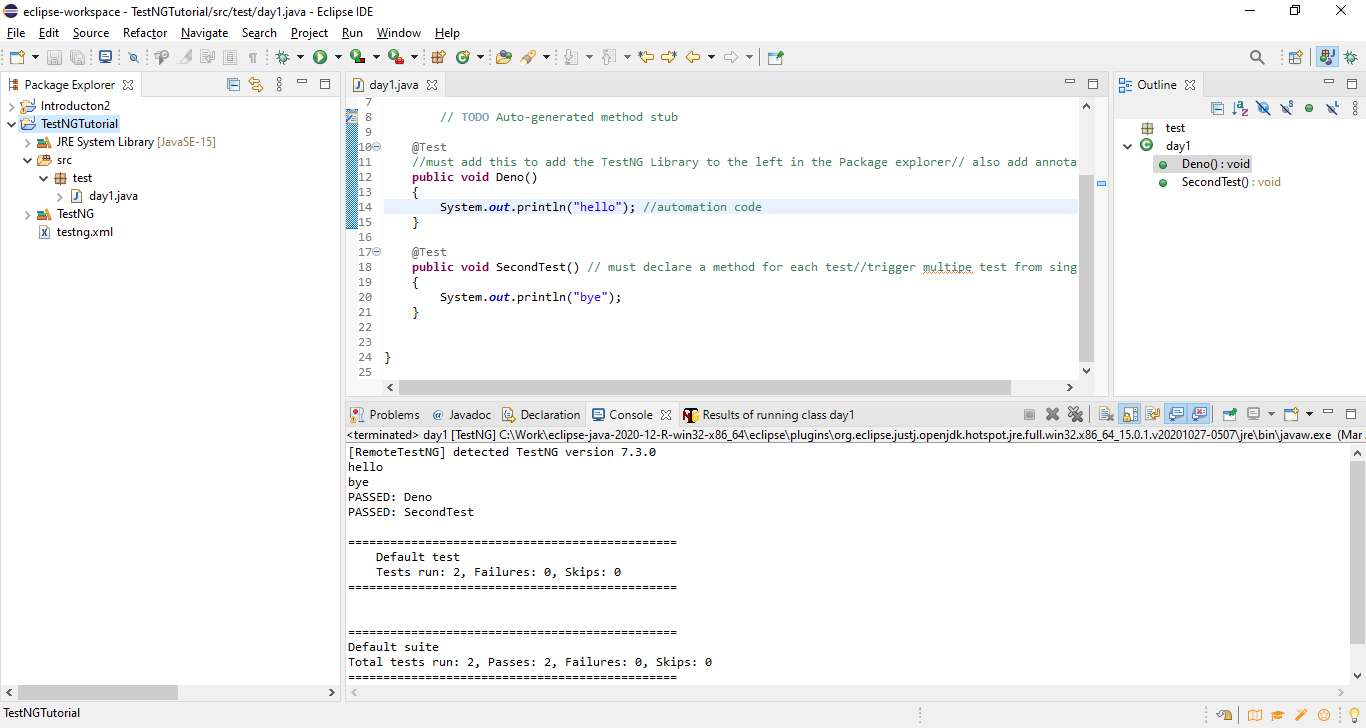
Right Click the Main Project name – TestNG – Convert to TestNG – Preview of XML – Next – Finish





**TestNG -** XML **File**

You need to have @Test Annotation followed by a method

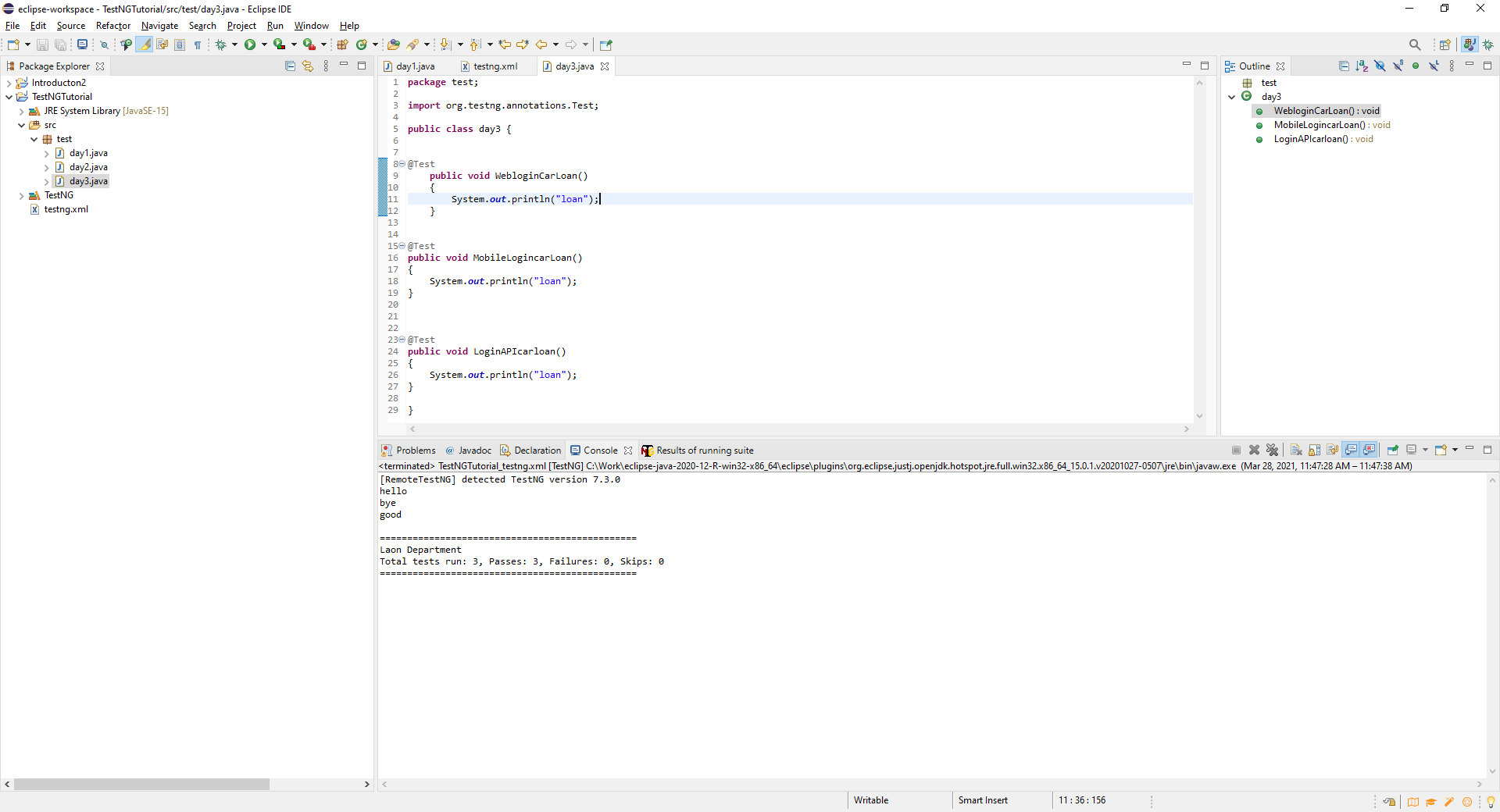
You can define multiple test cases in a single class

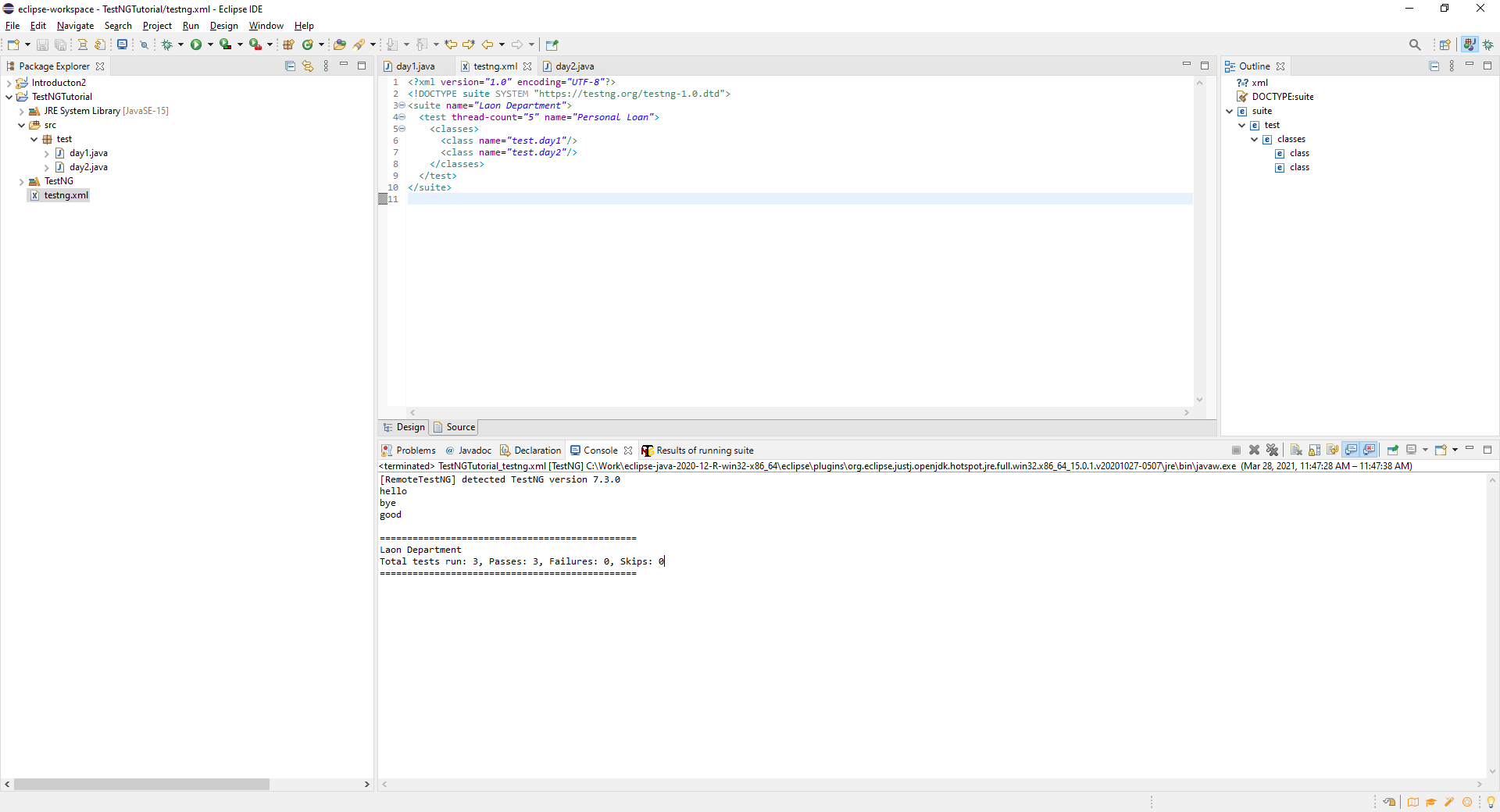
Right click on project 🡪 TestNG🡪 Convert to TestNG 🡪 XML file is created

XML Files structure - **Test Suite -> Test folder (Module)--> Test Cases** ...for every module will have test case

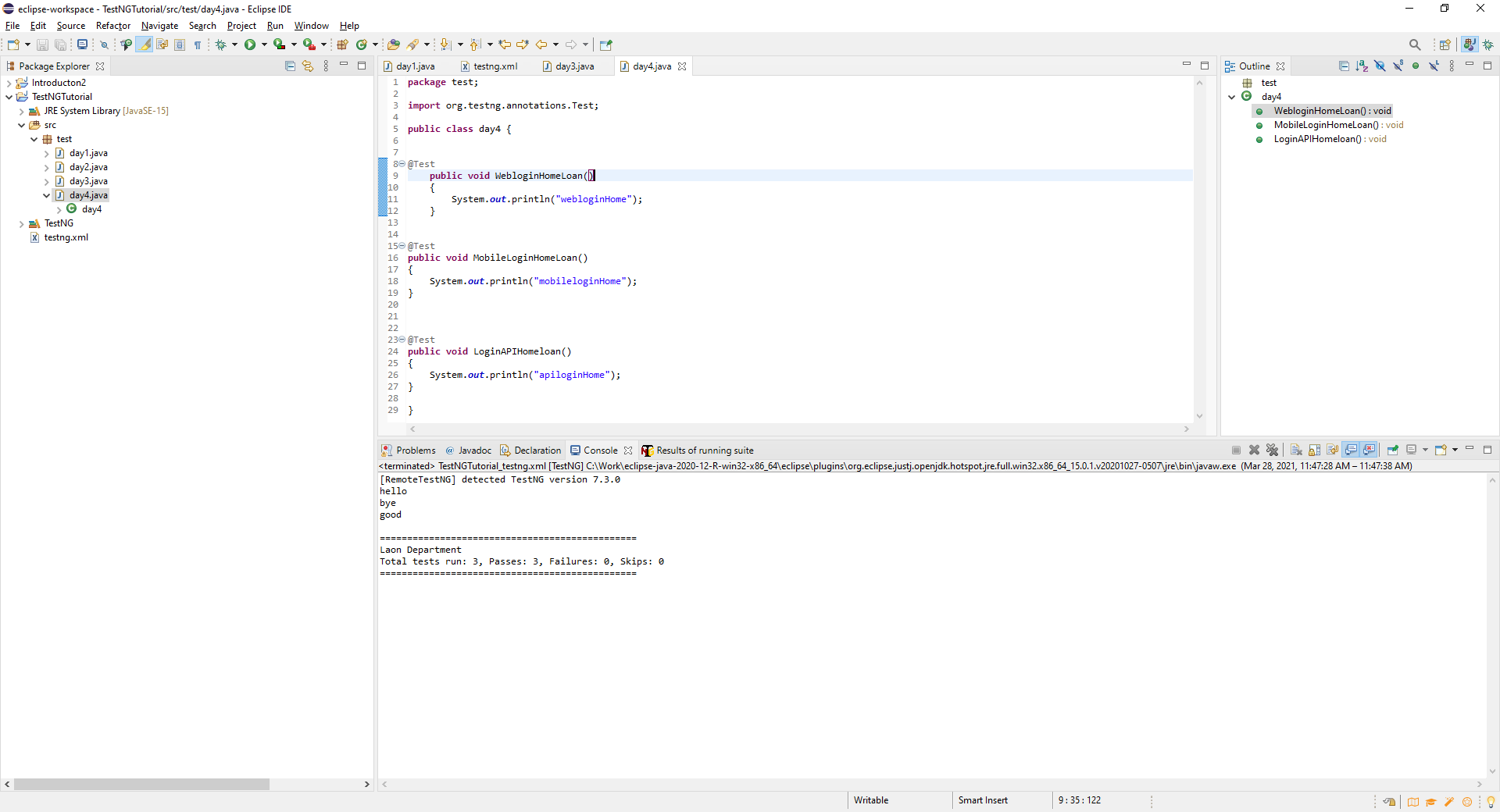
You can modularize the test cases based upon functionality and trigger them accordingly

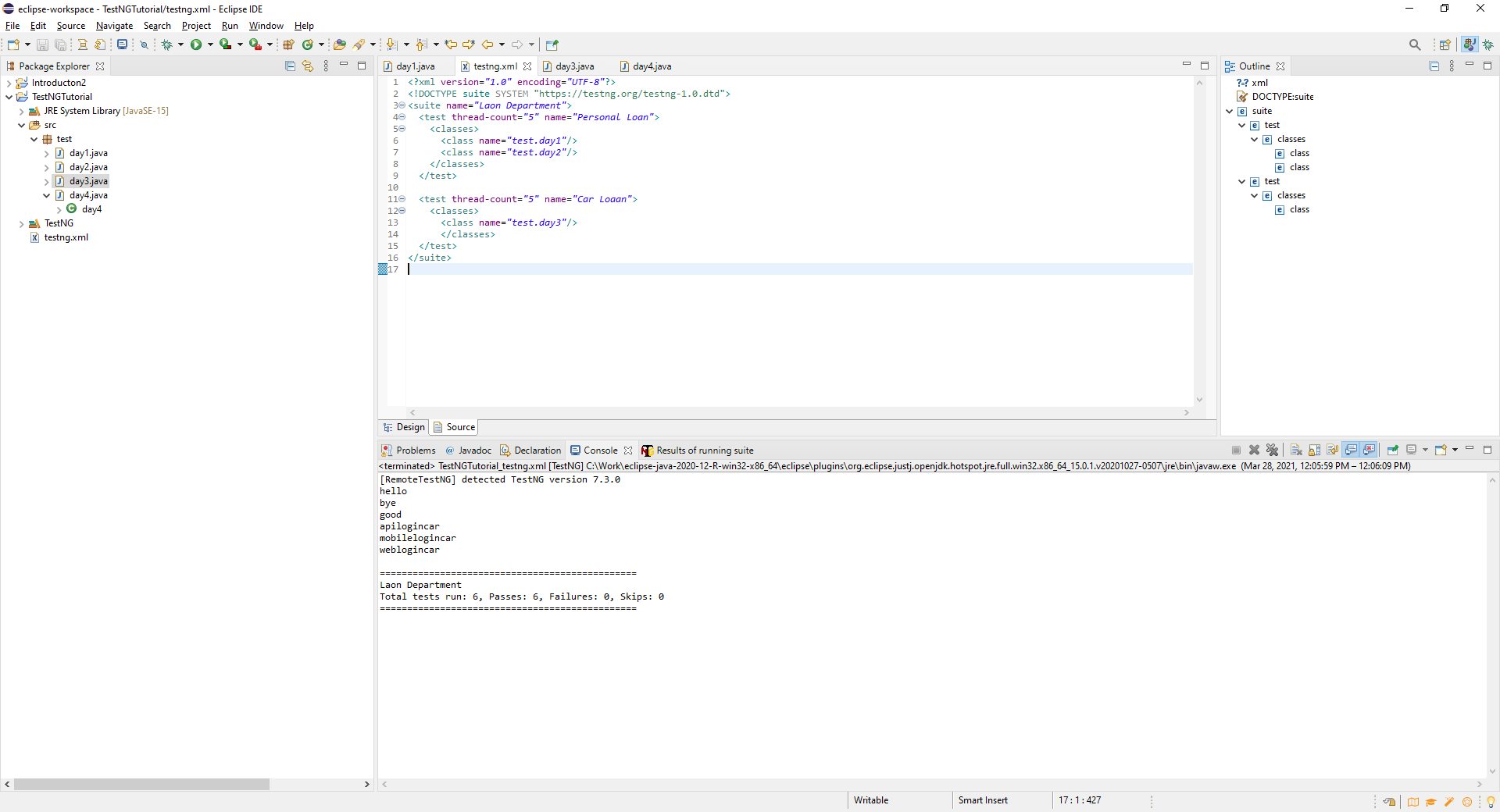
After XML run – Nice report is displayed

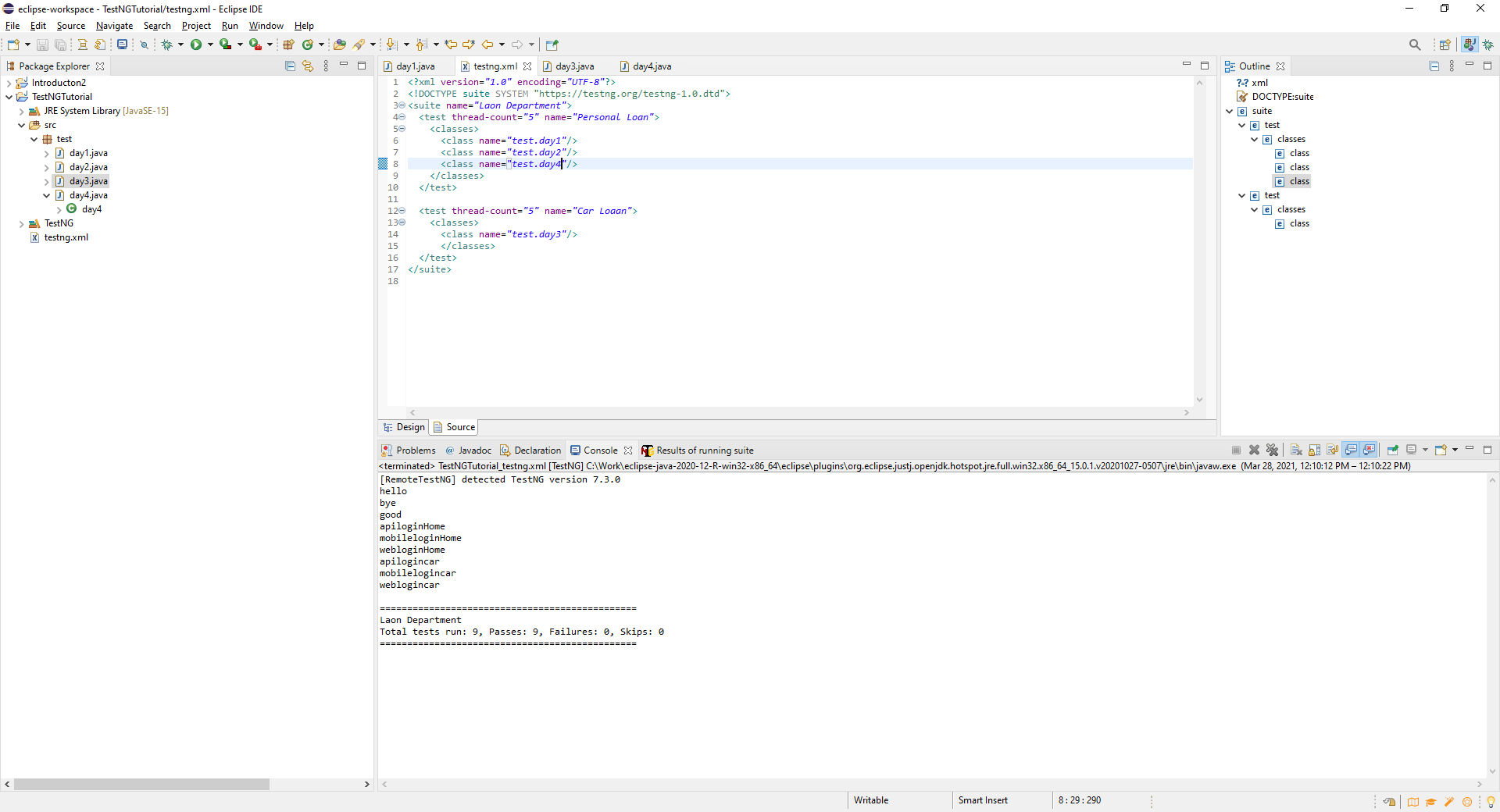




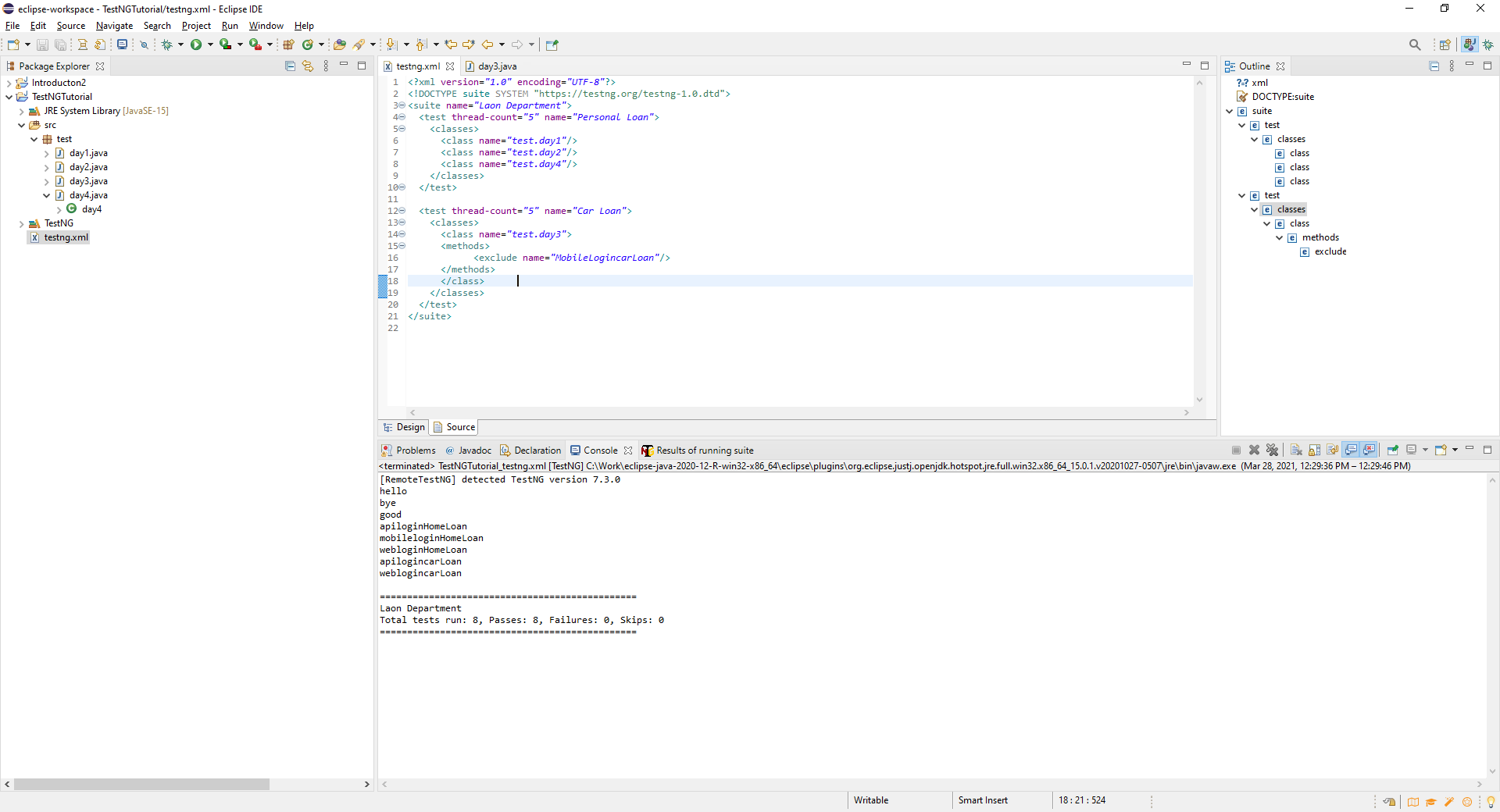
Run only Day 4 module



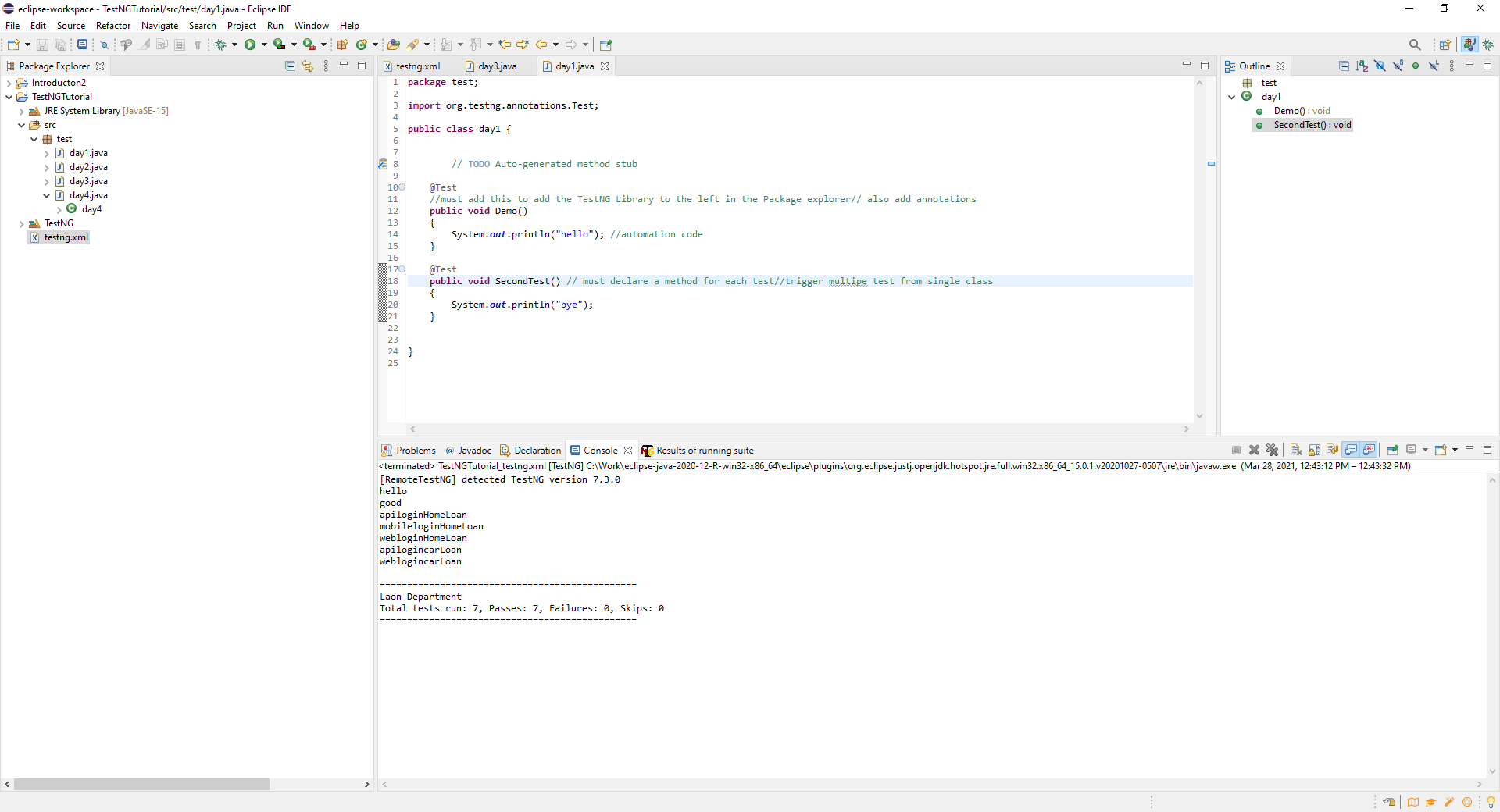


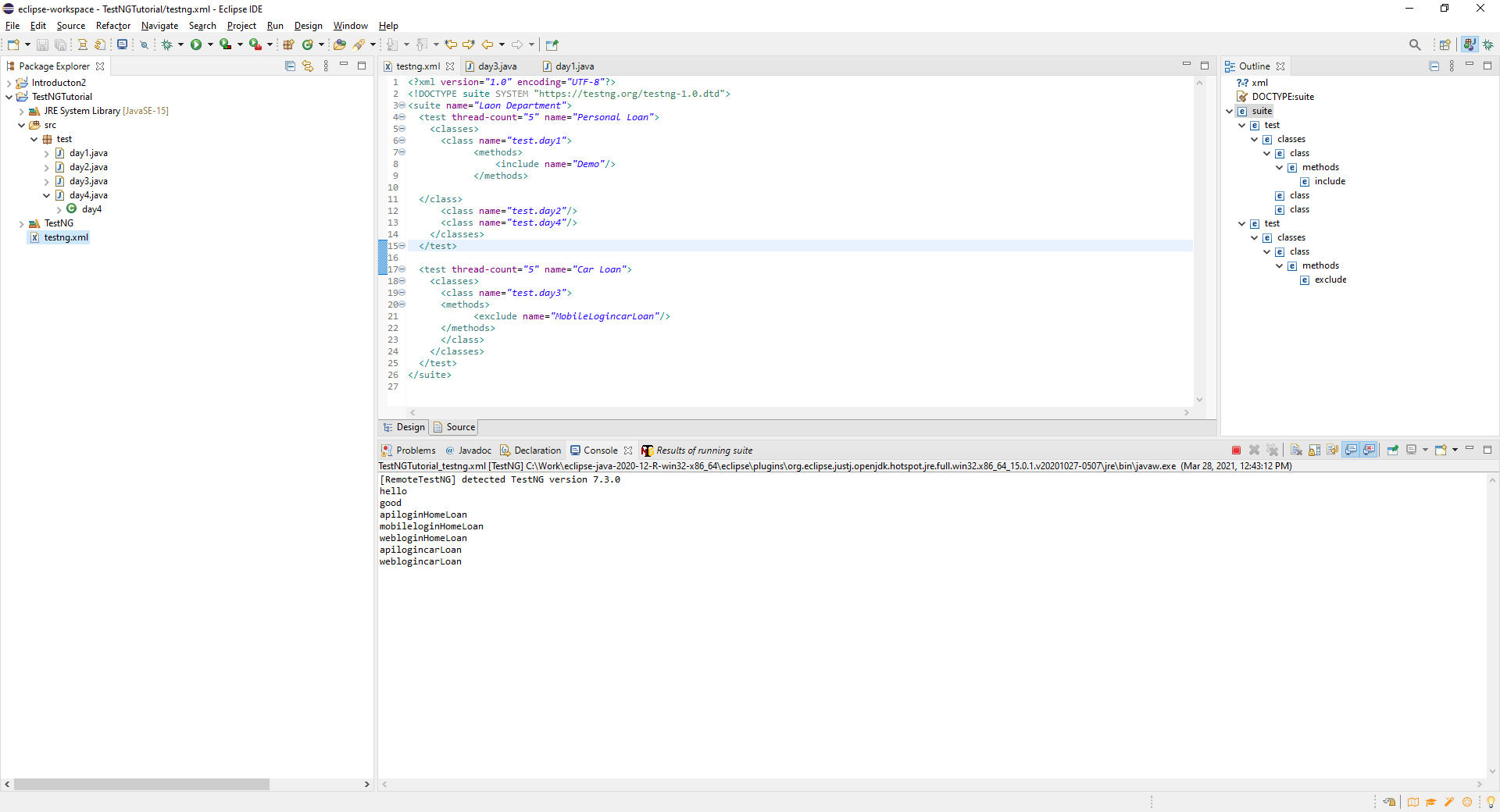


Use **Exclude** tag to skip module for Test Cases to only run specific methods from test execution. Remove method/exclude tags to put the test back in the main execution run. Good for regression test execution.

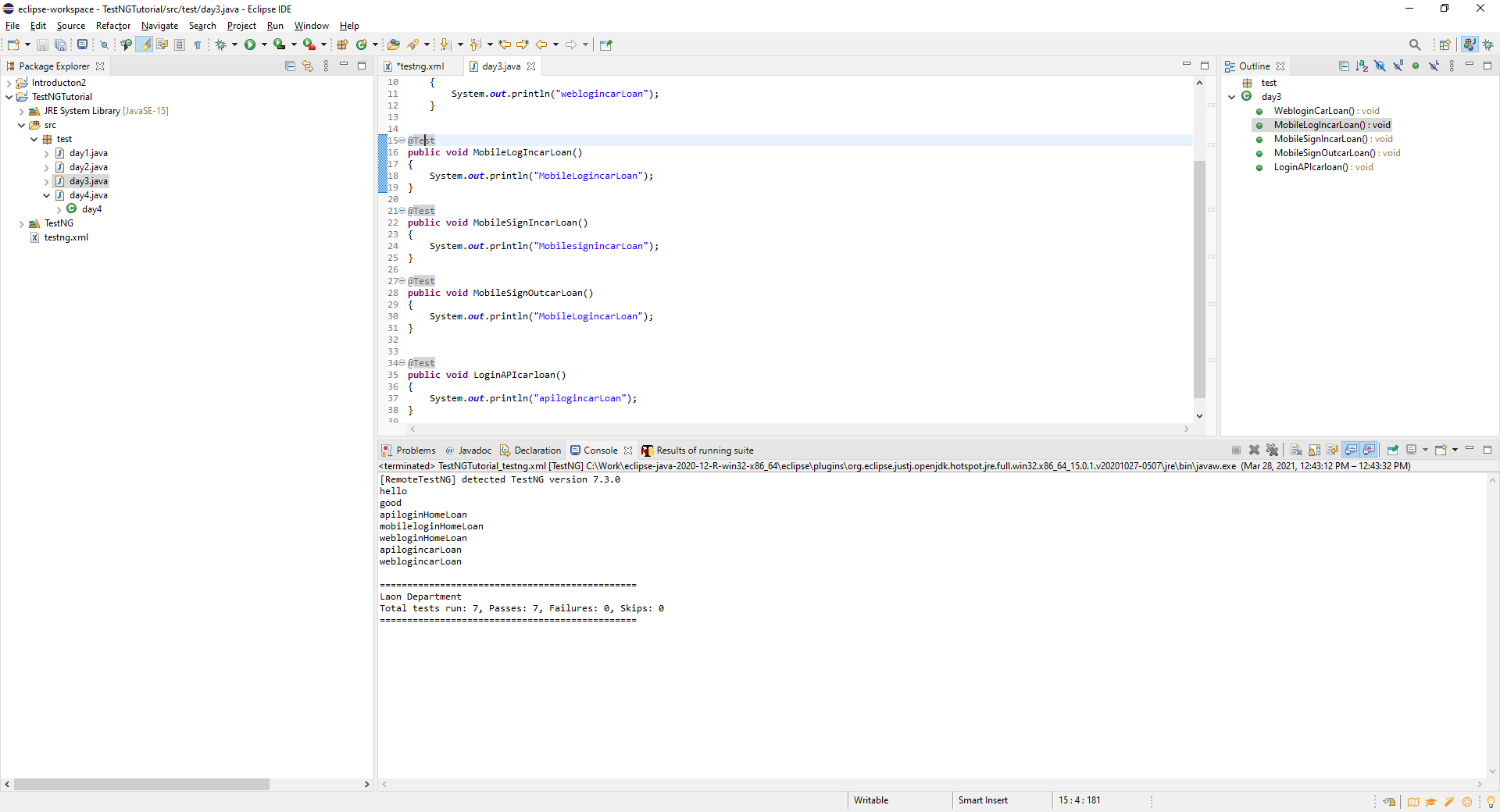


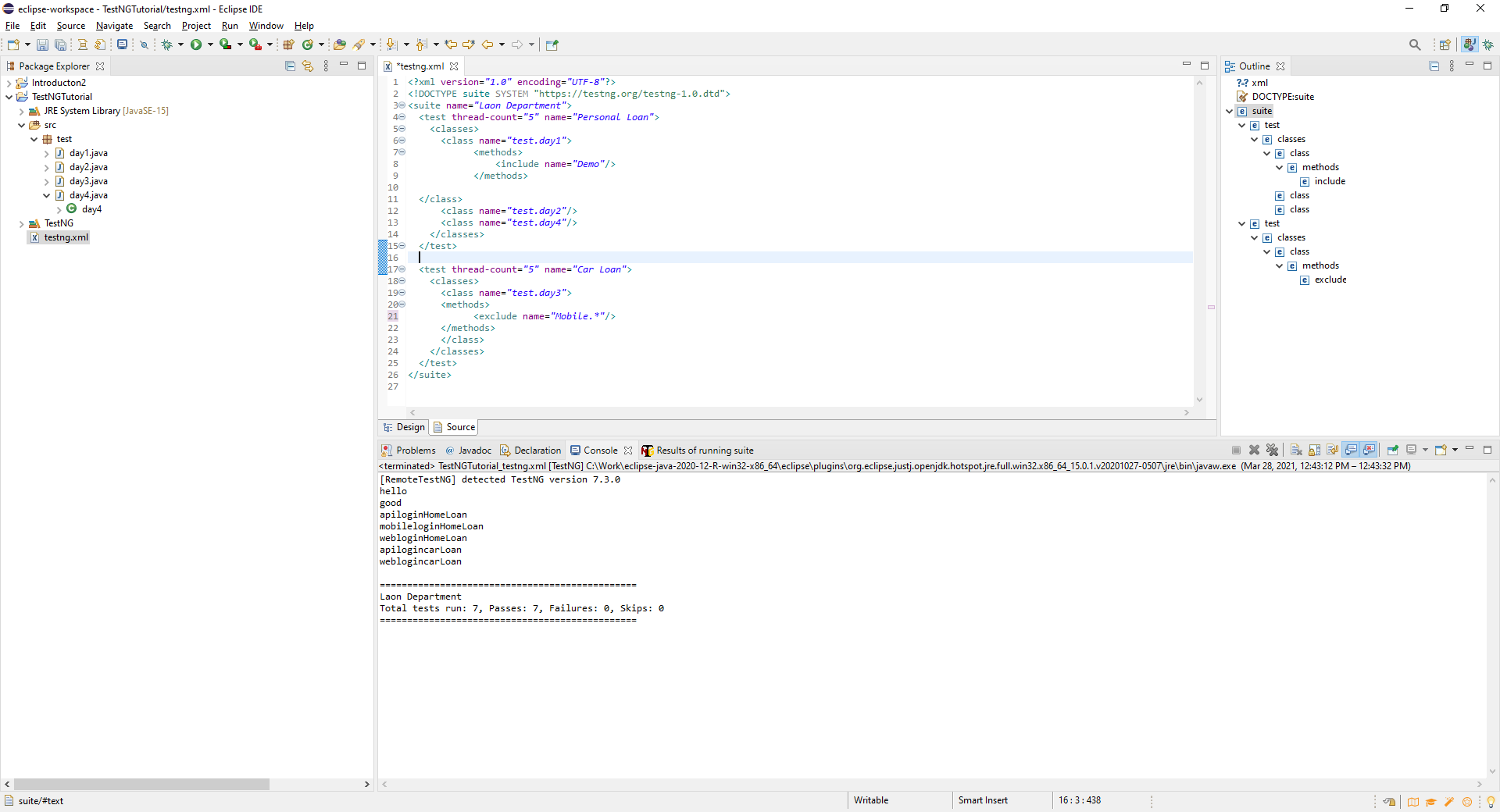
Can also **Include** tag to add specific methods for test execution. Only include the **Demo** module from day1in the XML run. Although there are two test in the day one module, only the Demo module will be run.

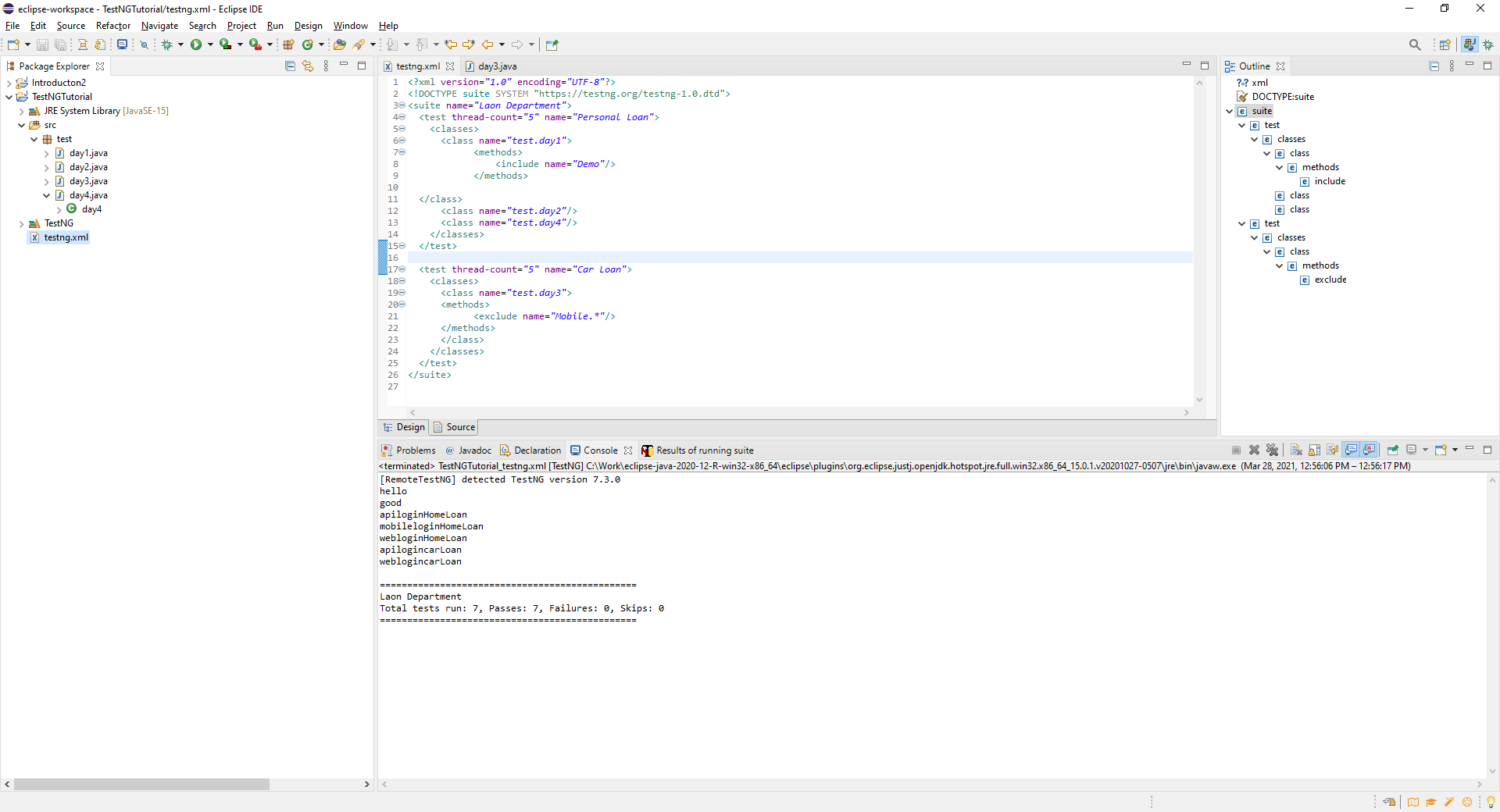


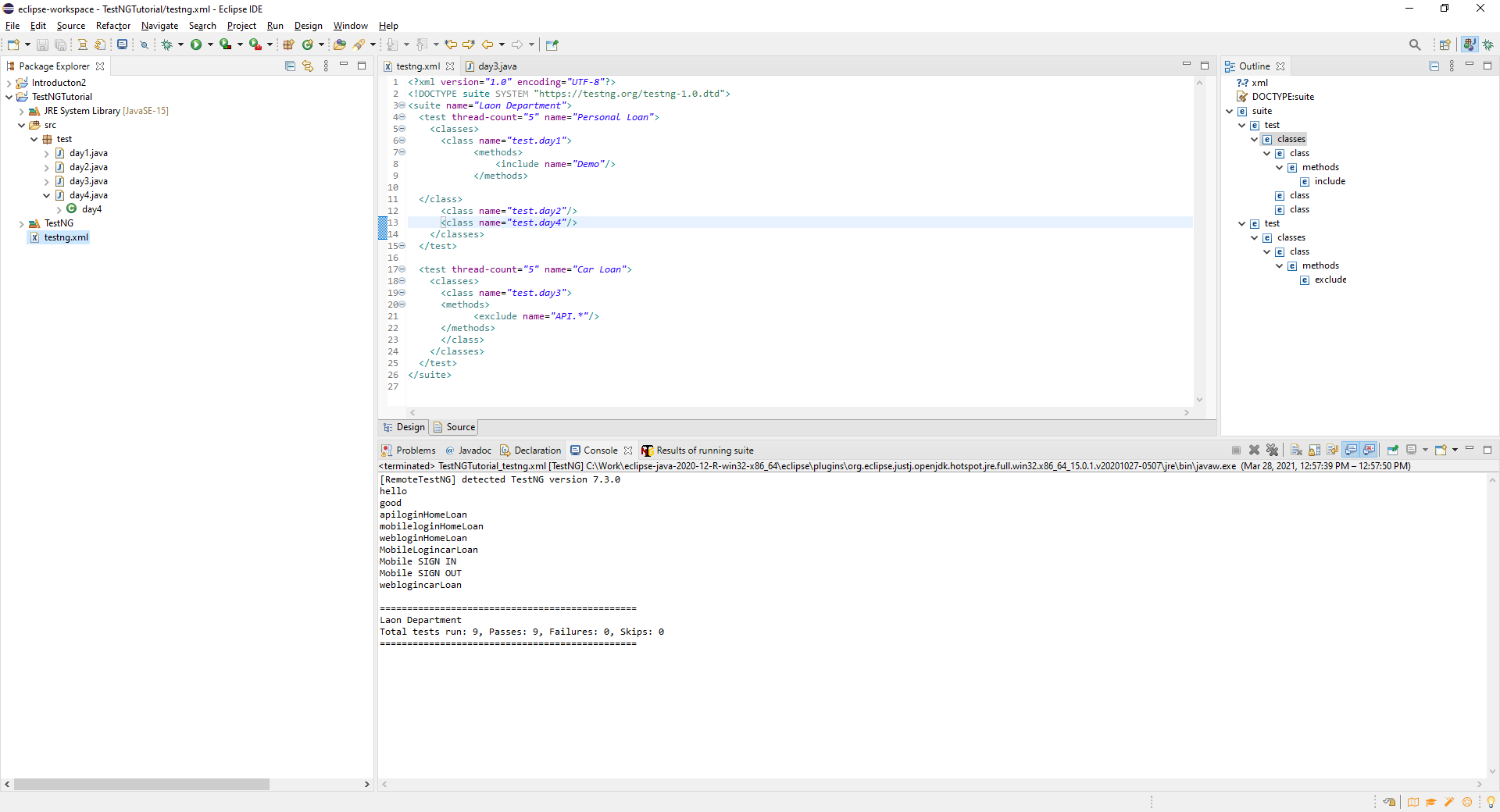


Using wildcards in Exclude. Can exclude categories of test based on your naming convention for the tests. Standard naming conventions very helpful in automation to help control and segregated test cases.



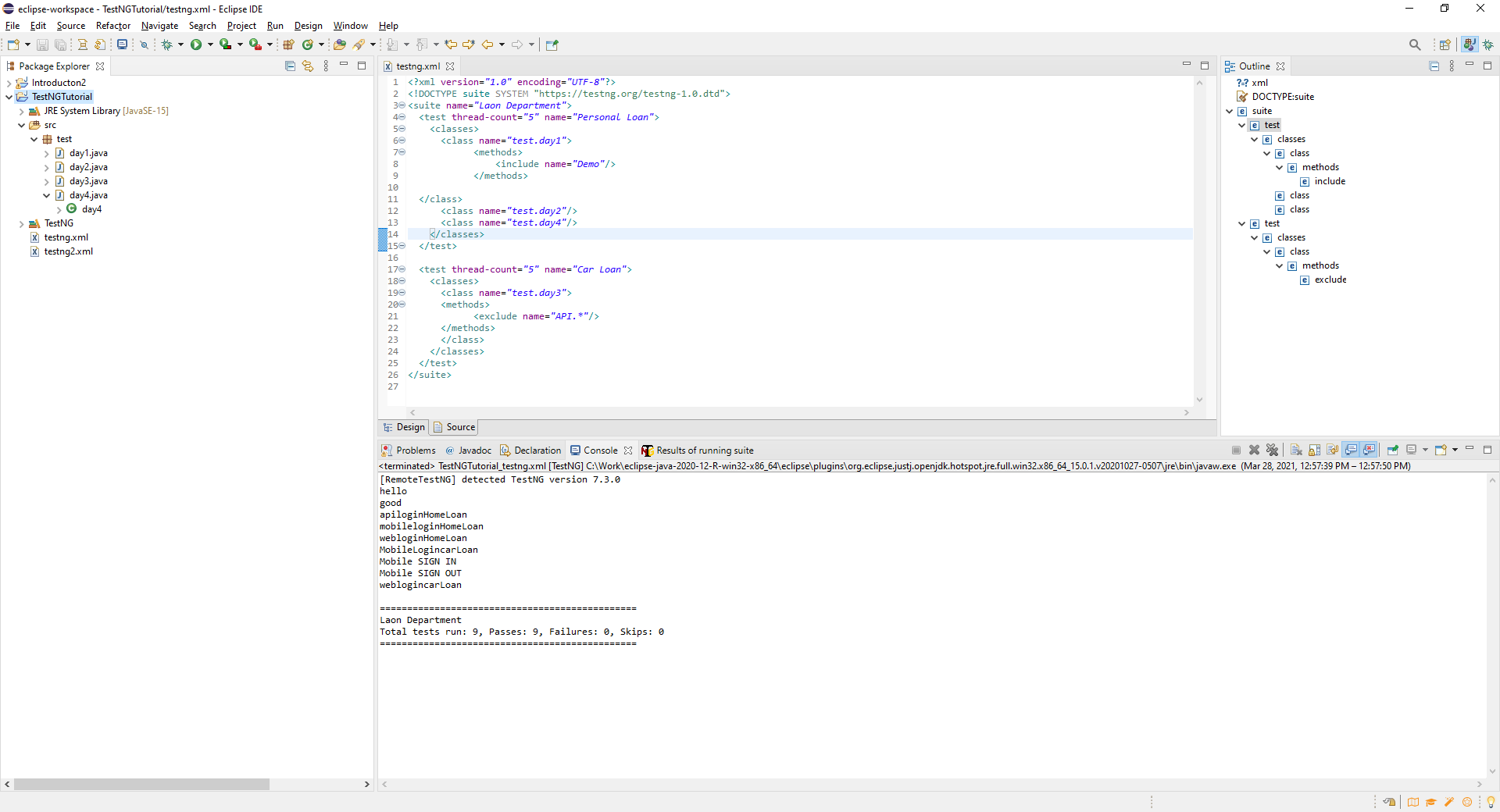


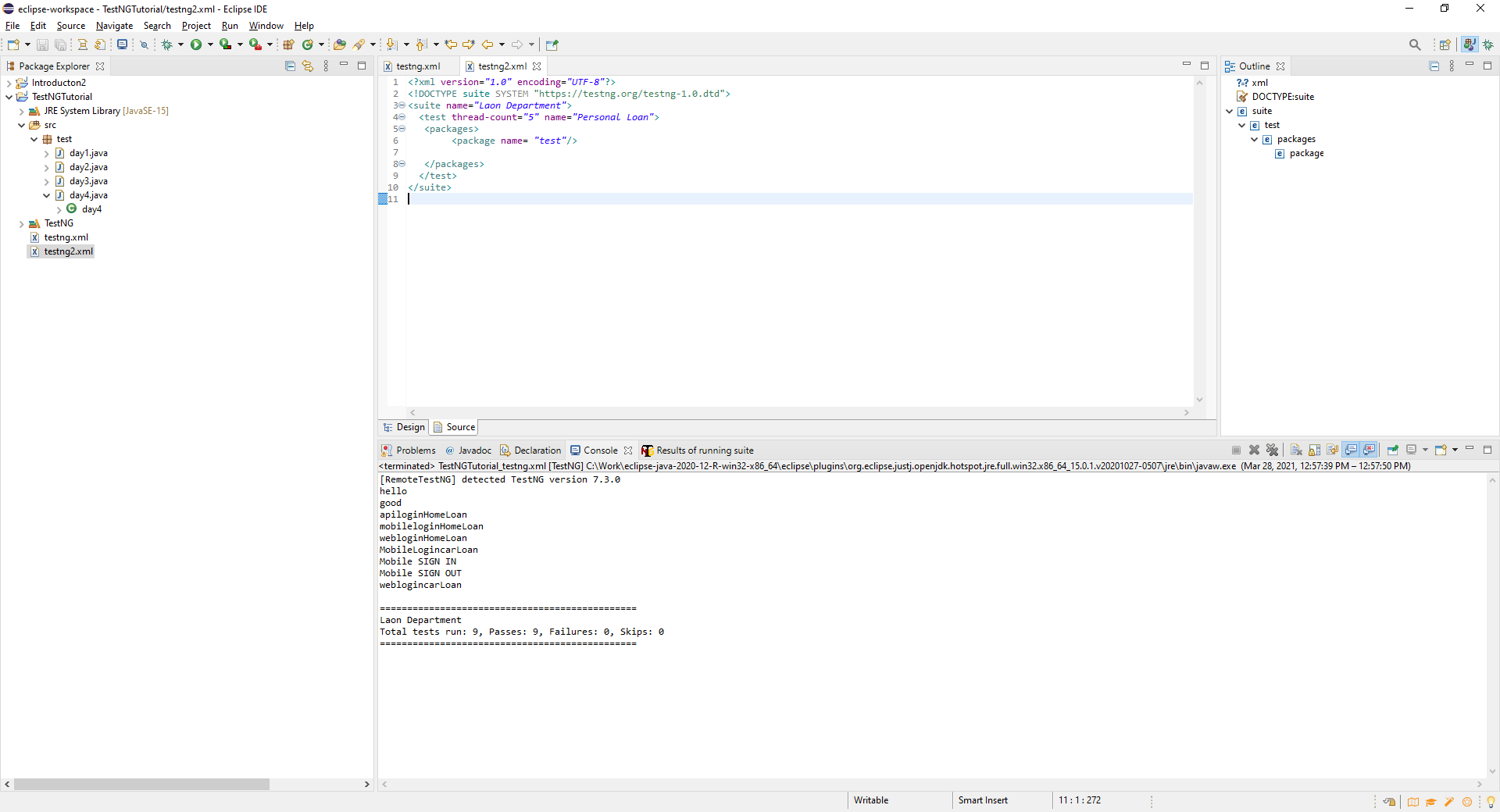


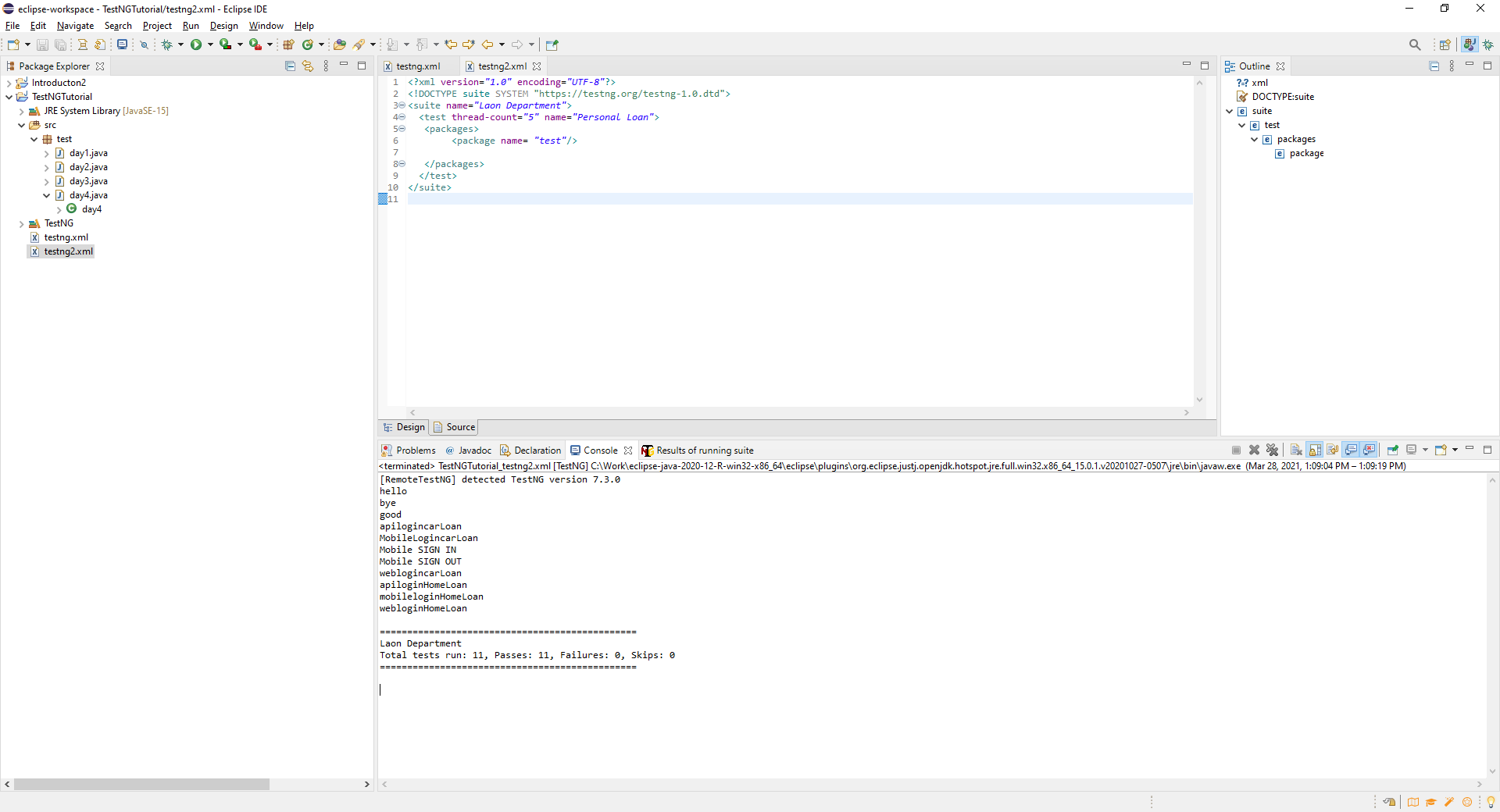


**Test Case can run from the package level**

Copy the XML file then go to the Package level for the test and Right Click and Paste. This will crate a second XML. This will be helpful when running regression tests.



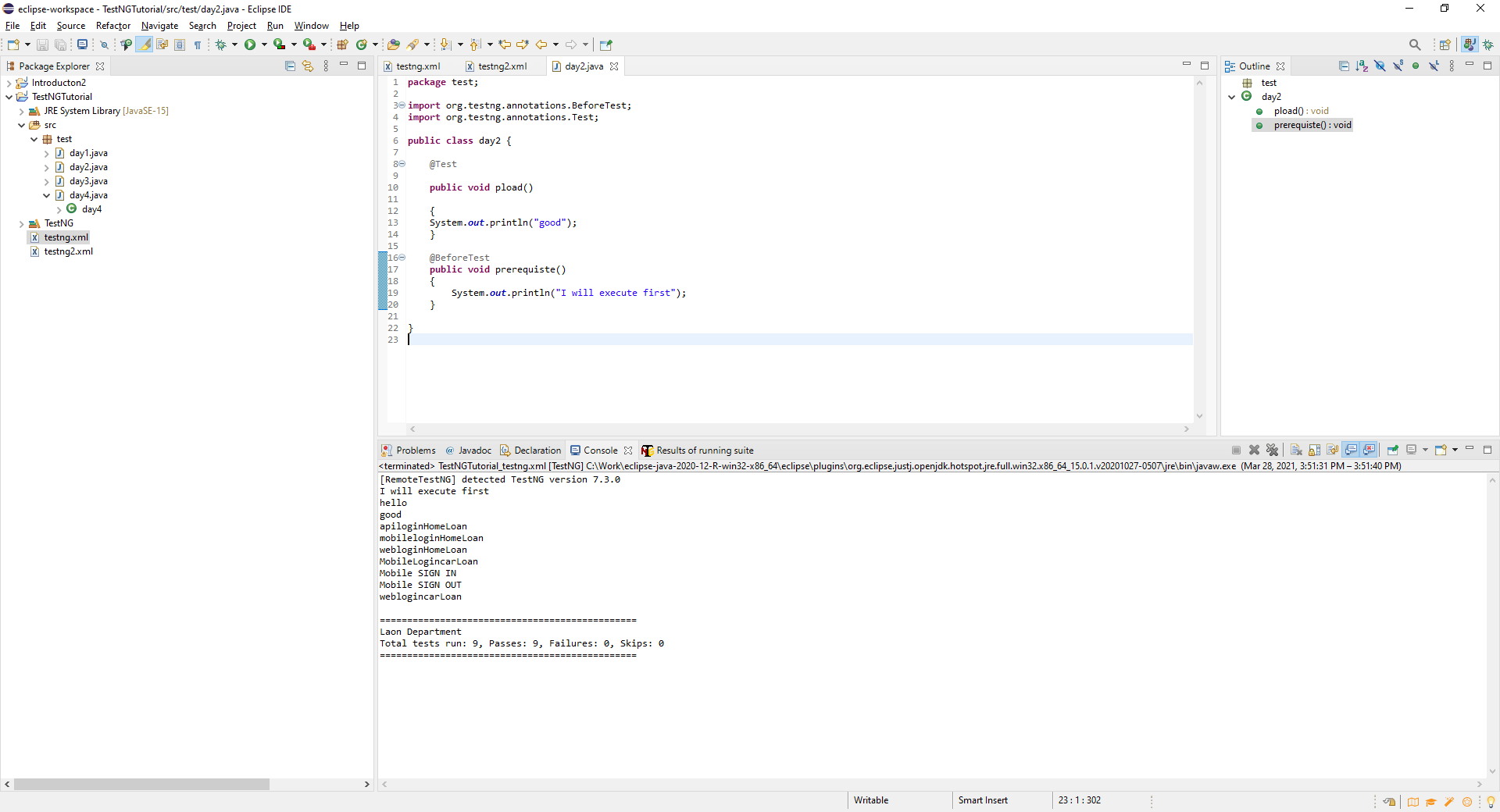




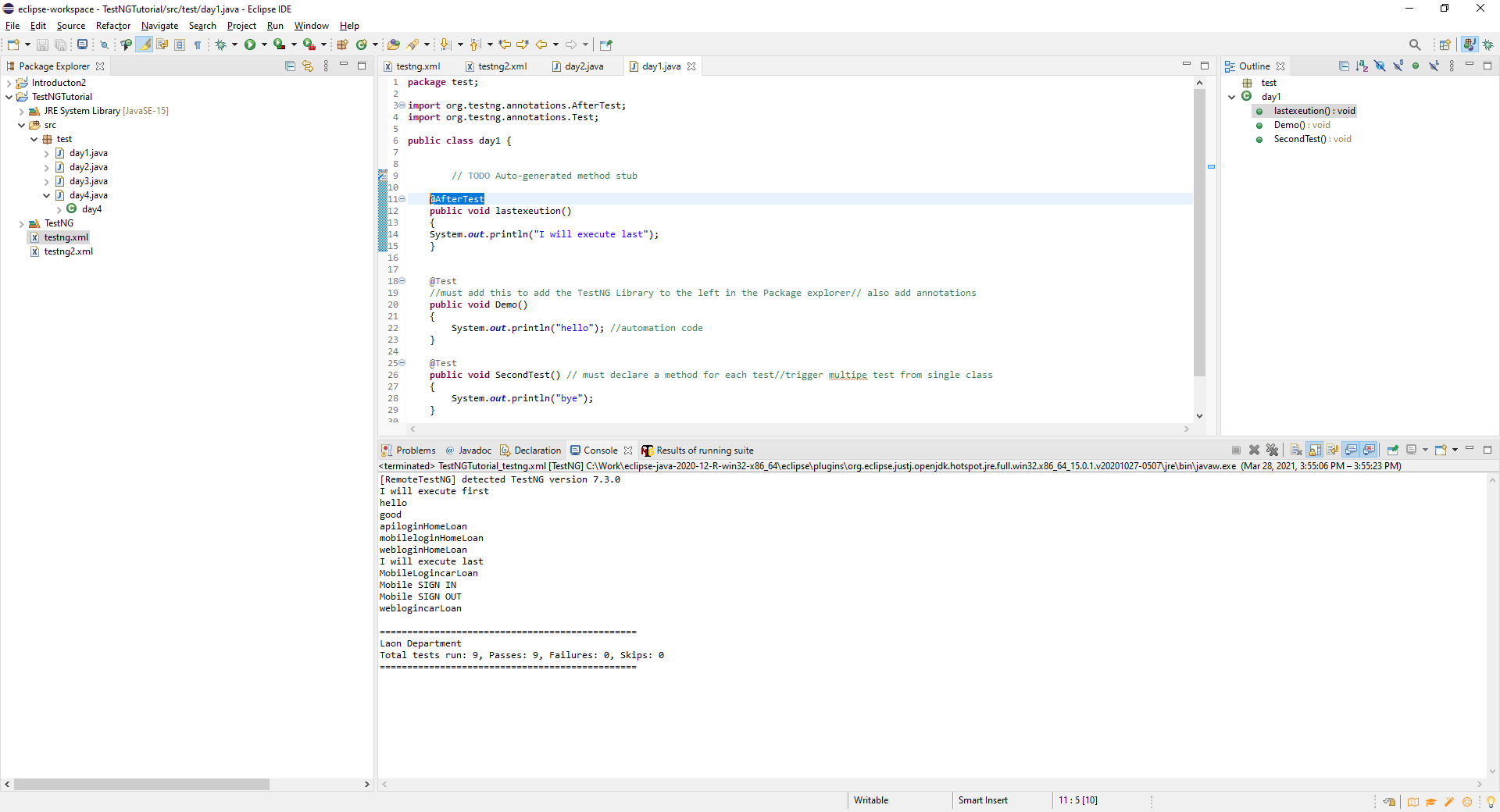
**Test Annotations Part 1 (Section 142)**

Clean data prior to executing the test in the automation suite

**@BeforeTest** to redirect to run these set of test cases before running other test



**@AfterTest** – Executed after the test in the module/folder that you place the annotation. This is why at this is why the last execution is not at the end of the result list…ie…ust this to delete cookies, close connections from service calls for Rest APIs.

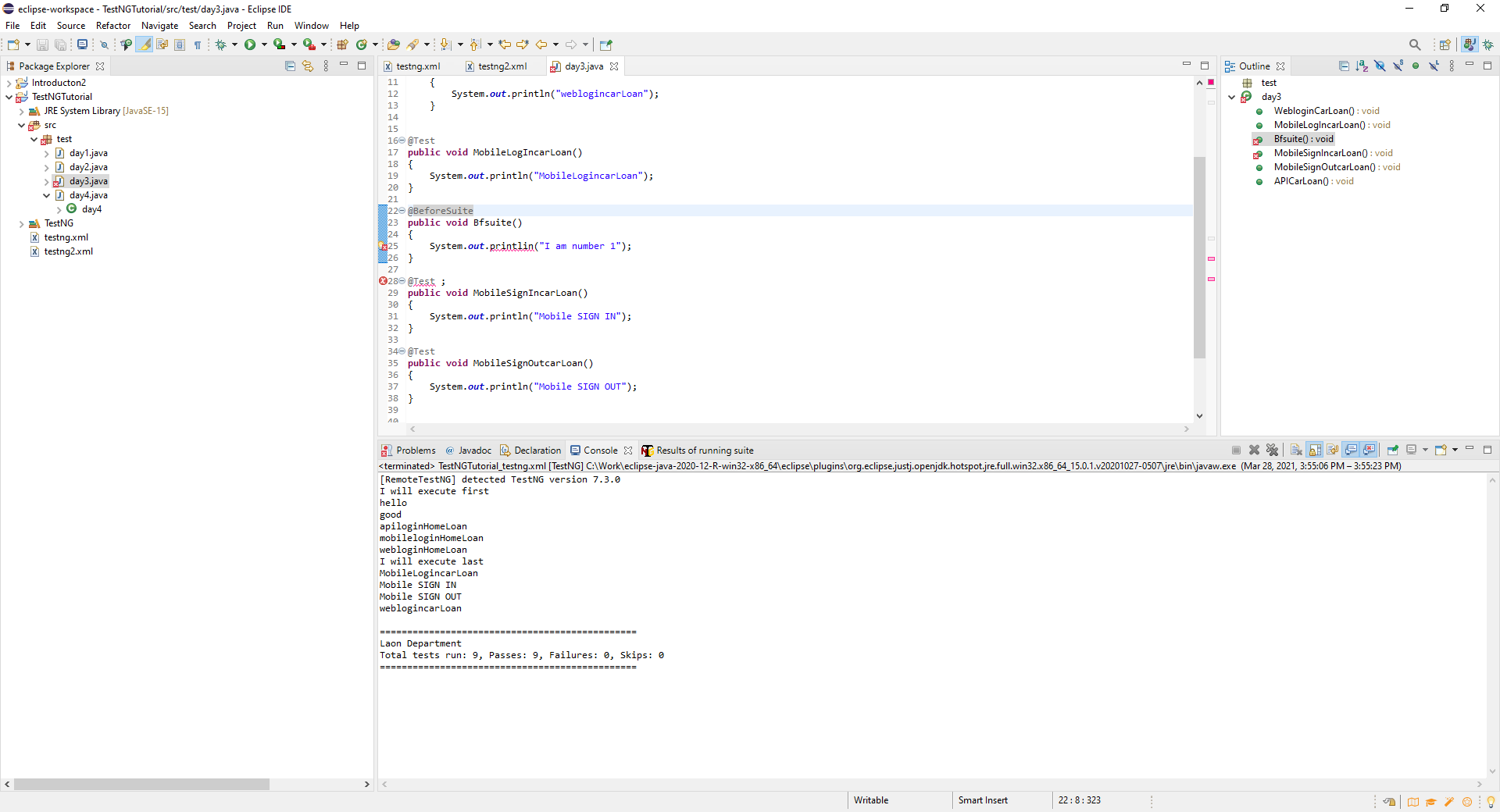


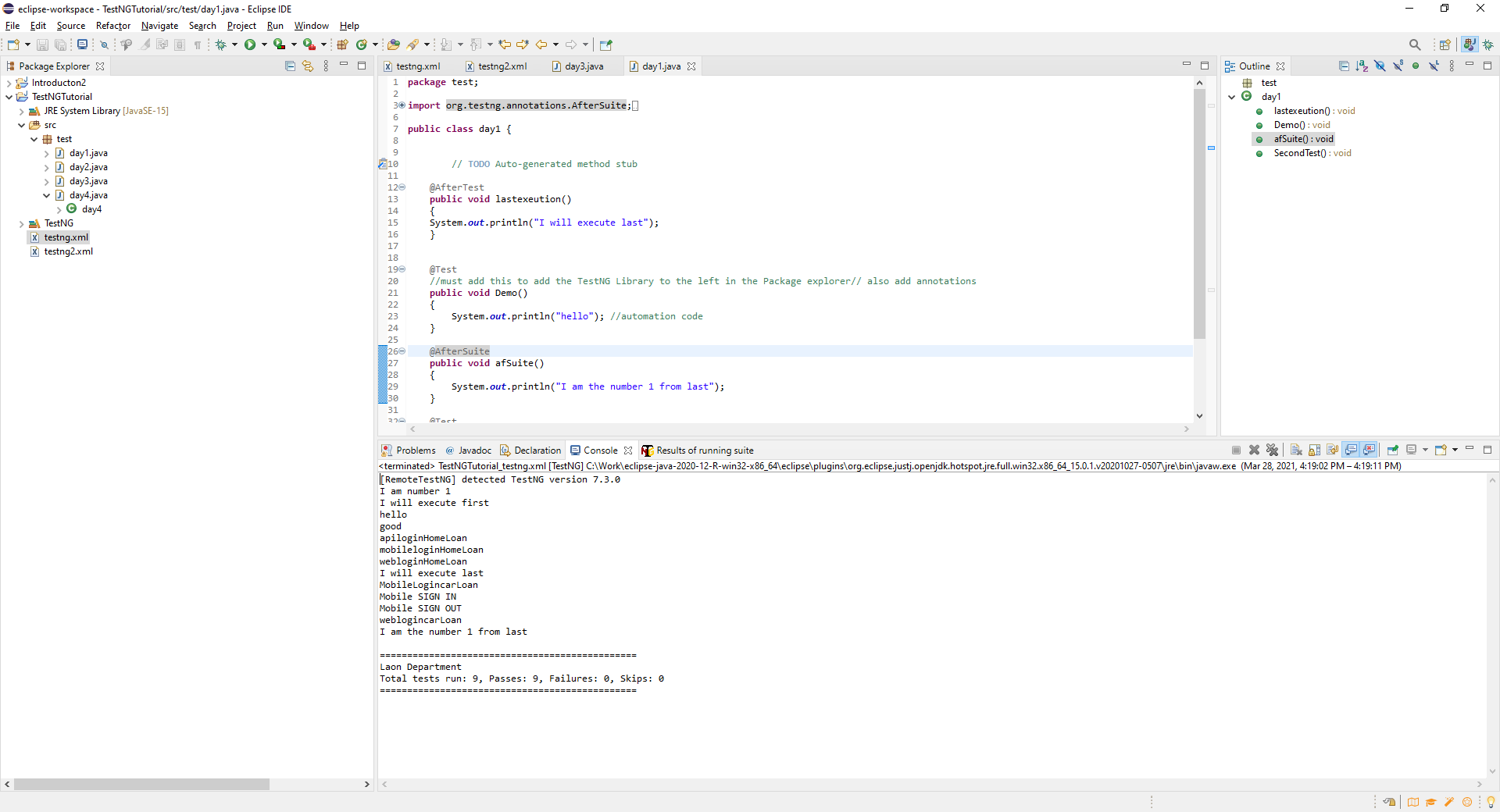
**Test Annotations Part 2 (Section 142)**

If use **before or after** suite, then the test that you have written before or after the suite is run.

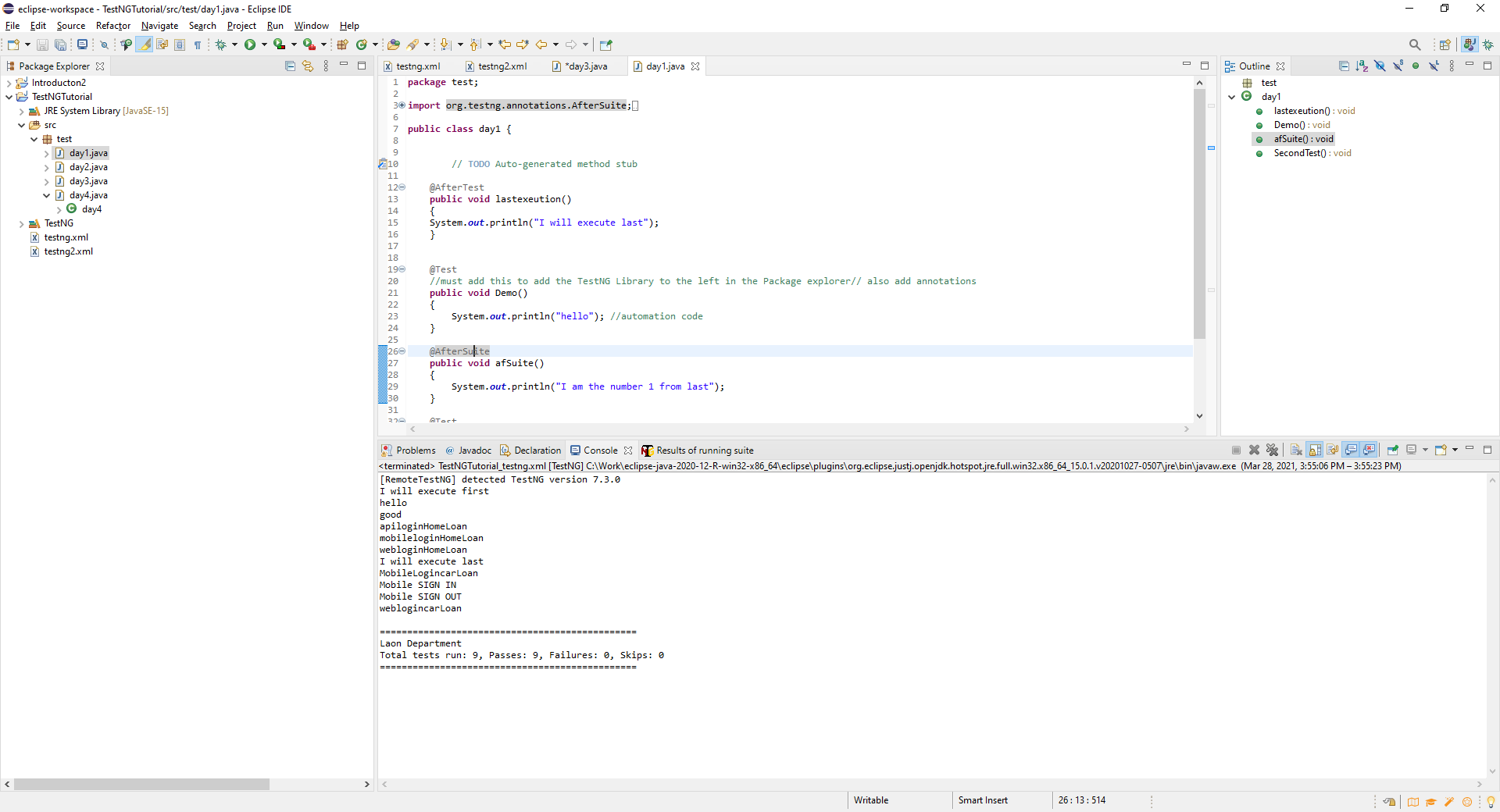
**BeforeTest** prior to the suite code, then use for example setting loading global variables like URL links to QA environment details. Rest API use to set up various security keys.

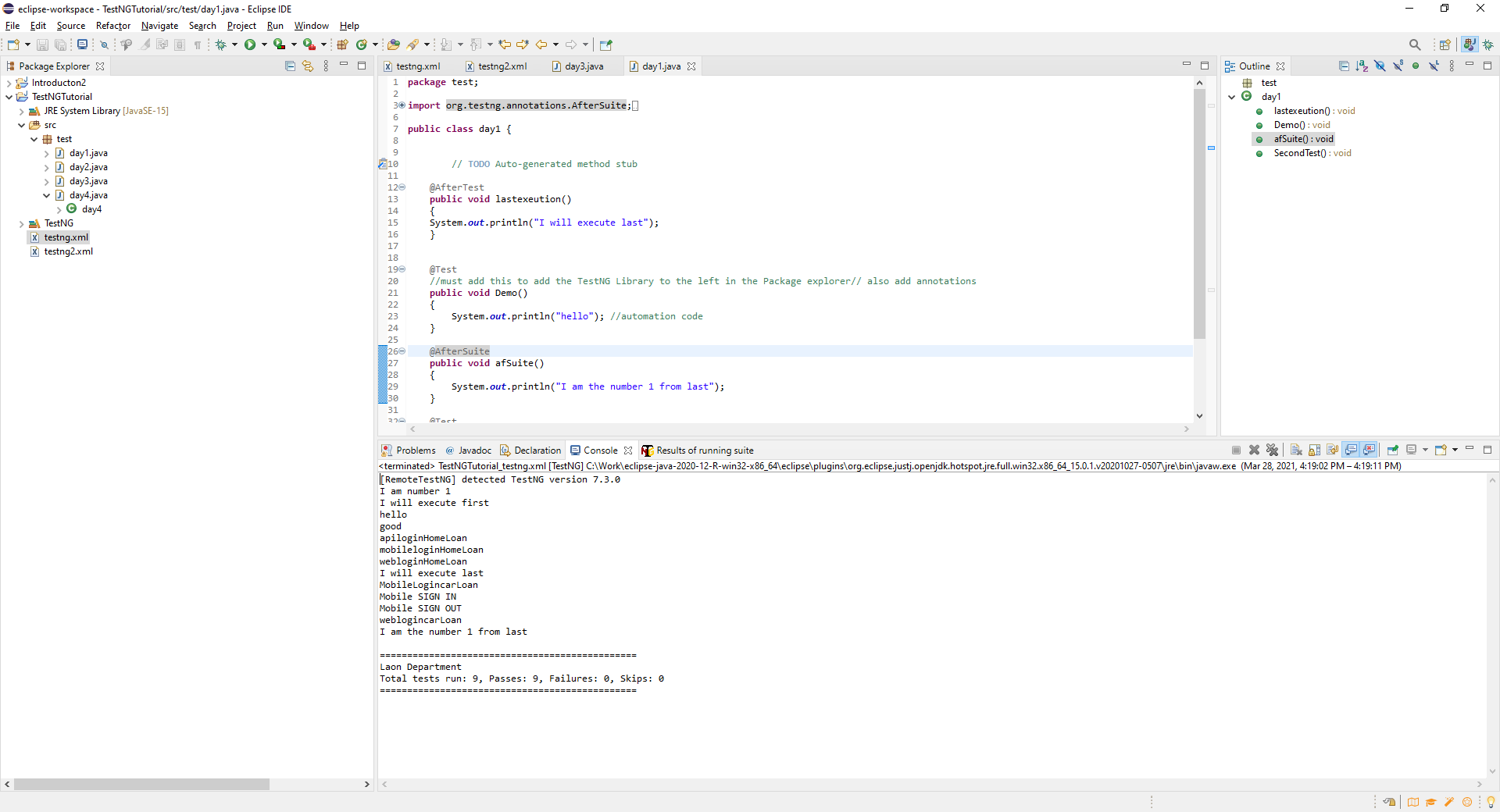
Using @Before Test





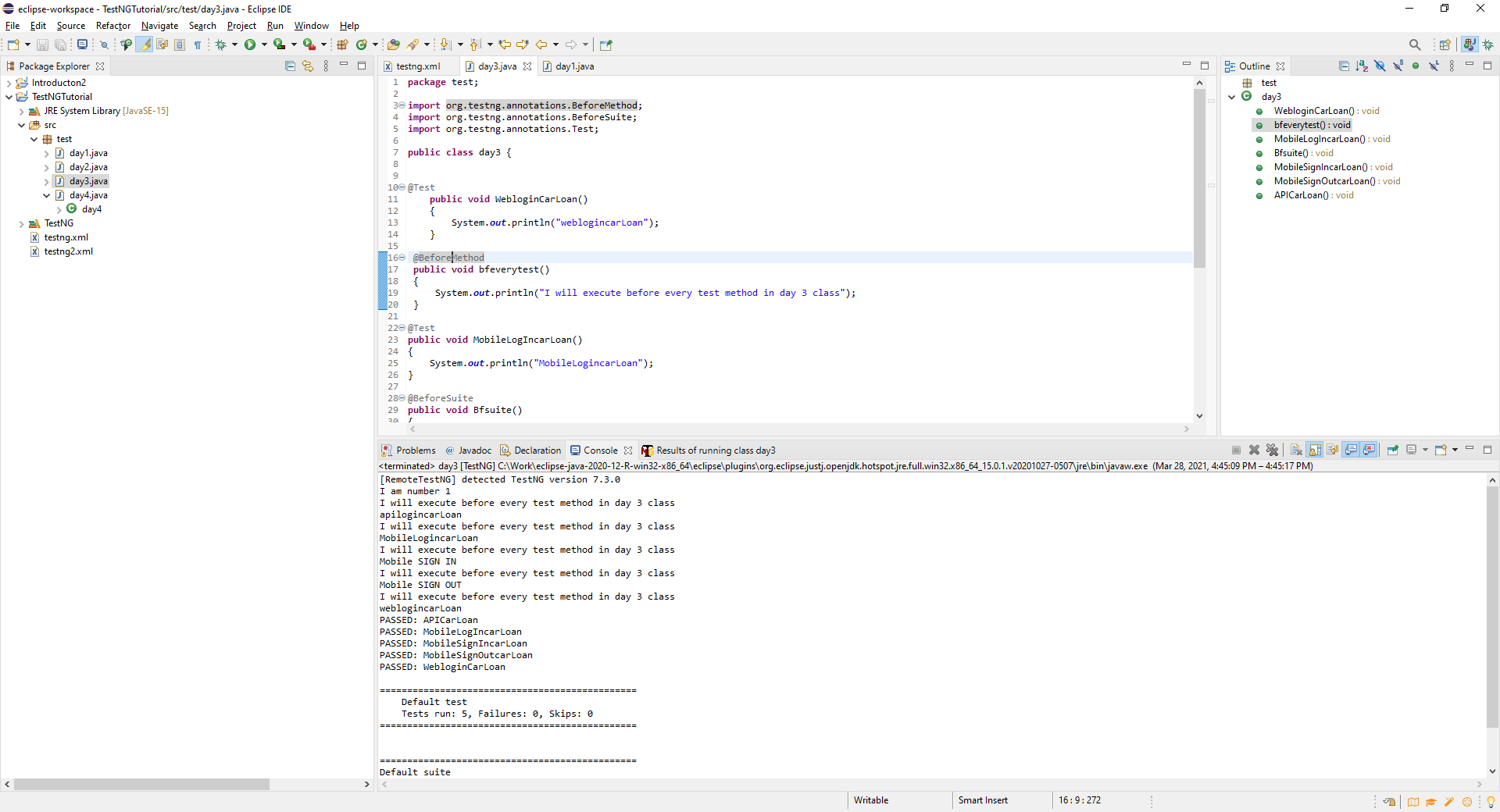
**Using @AfterSuite**





**@BeforeMethod and @AfterMethod**

This annotation applies to a class. Before executing any of the test blocks the method will run. Ie….If there are 4 methods in the class file and you runt the **BeforeMethod** an notation this will execute first and then the first test, then run again before running the next test etc……

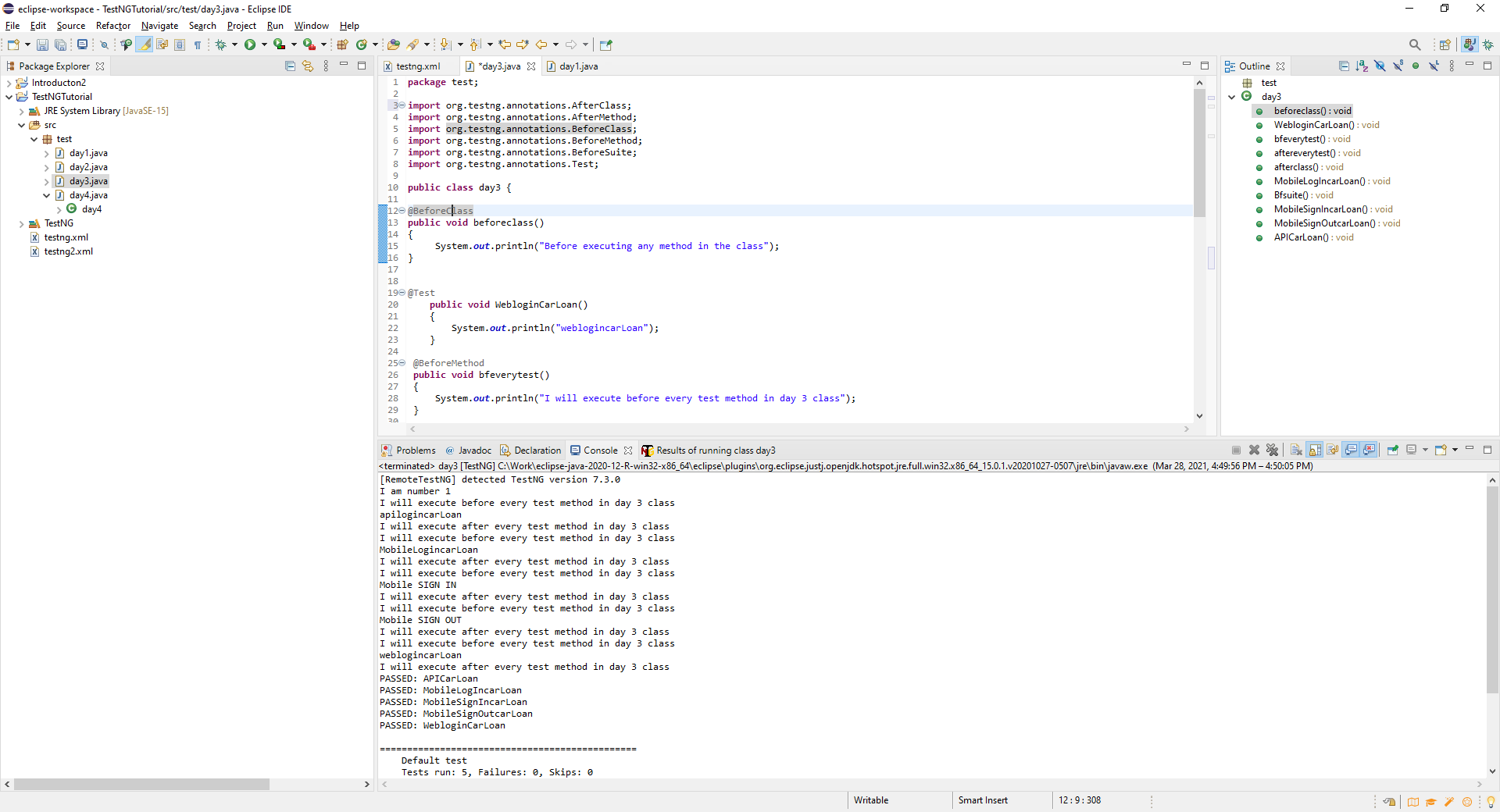


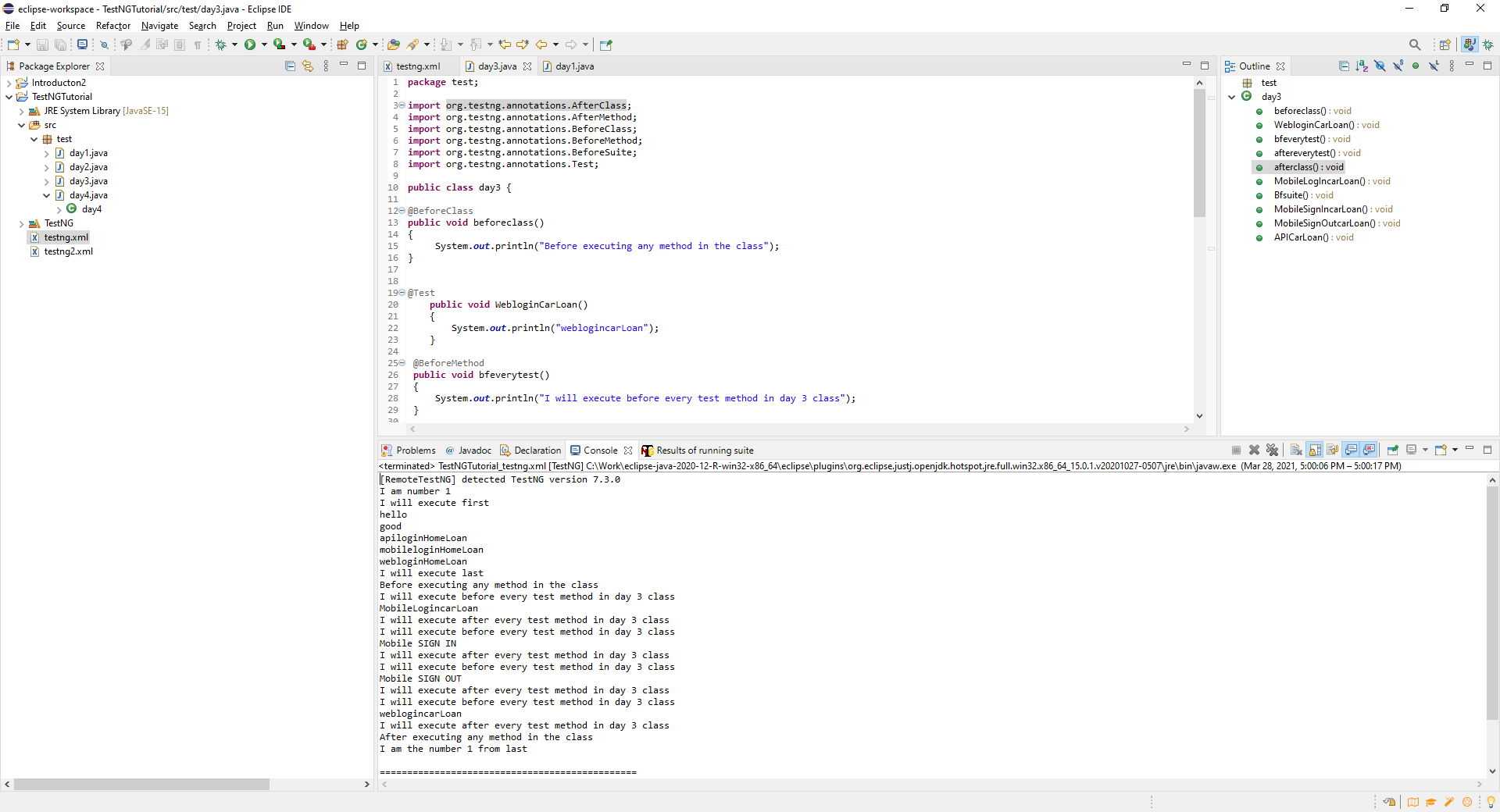


**@BeforeClass and @AfterClass annotation**

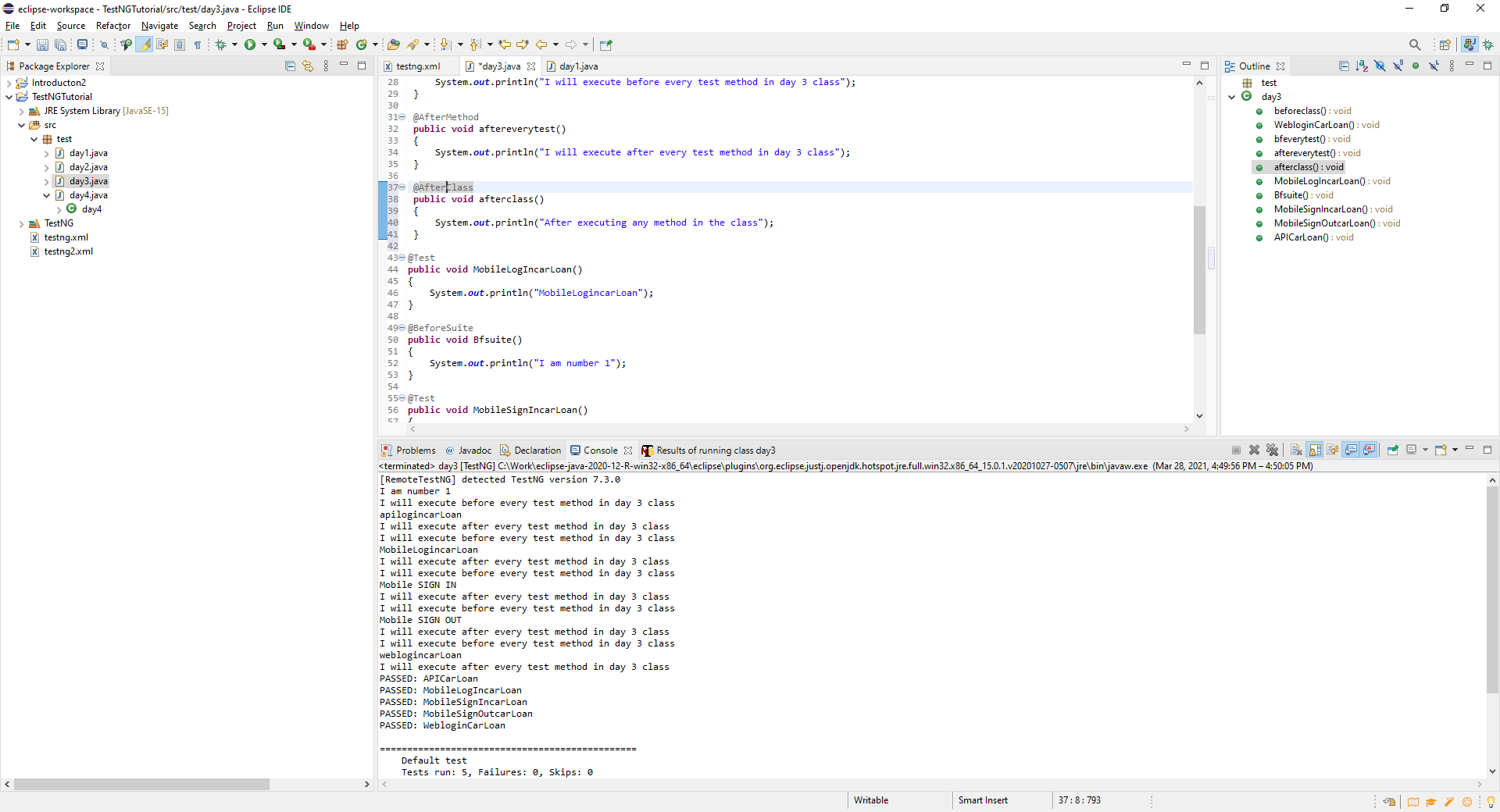
The scope is before class specific to class only. Before executing anything in the class the **@BerforeClass** is executed. Ie….if have 6 methods in test case, before class execute and then the test/methods will executed . Test are executed in alphabetical order of the method name.

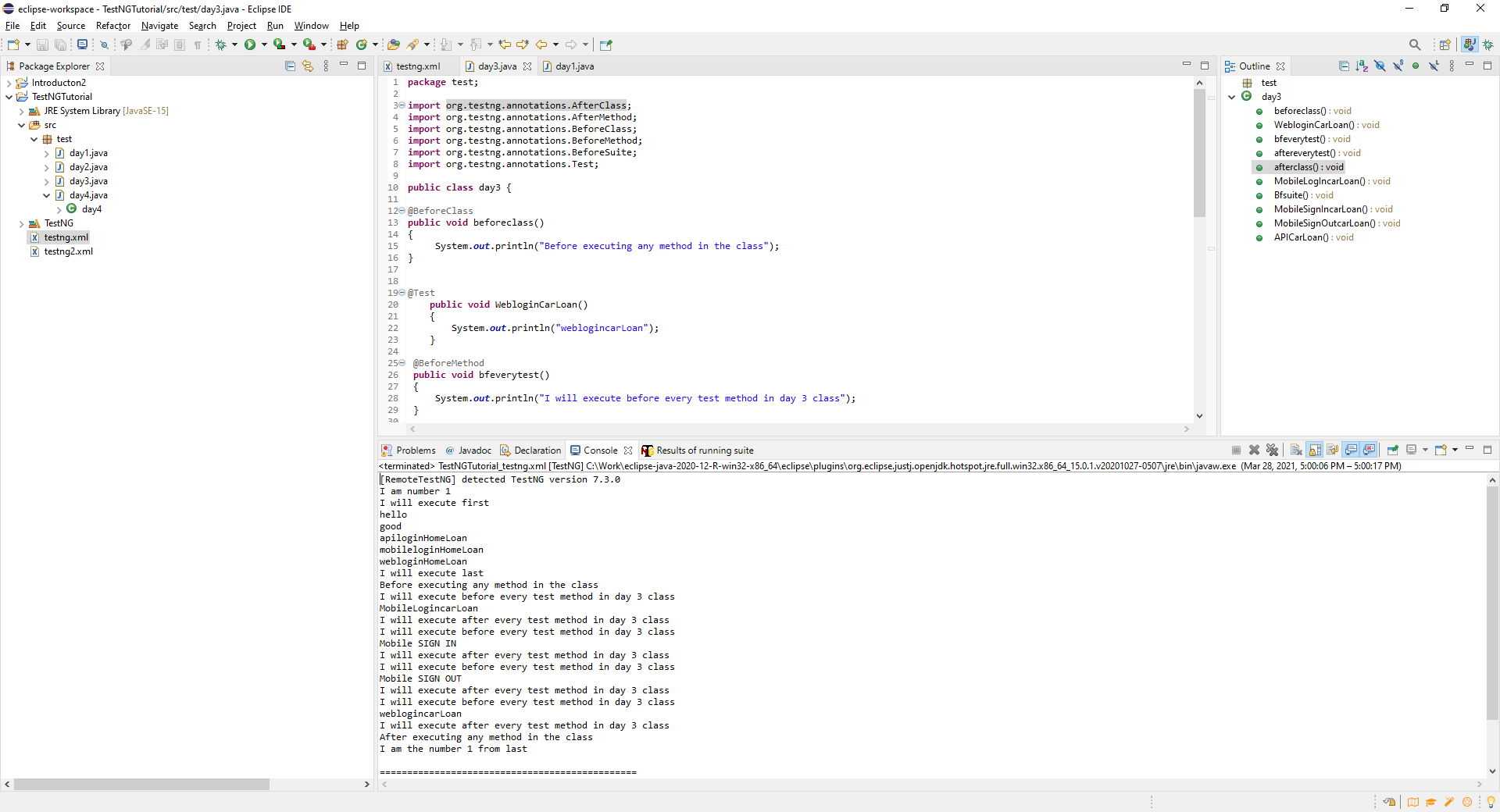
**@BeforeClass**





**@AfterClass**





Difference between **@BeforeTest** and **BeforeMethod**:

before test depends on the testing file based on the test folder scope.

before specific to each class file and execute before each and every method to test.

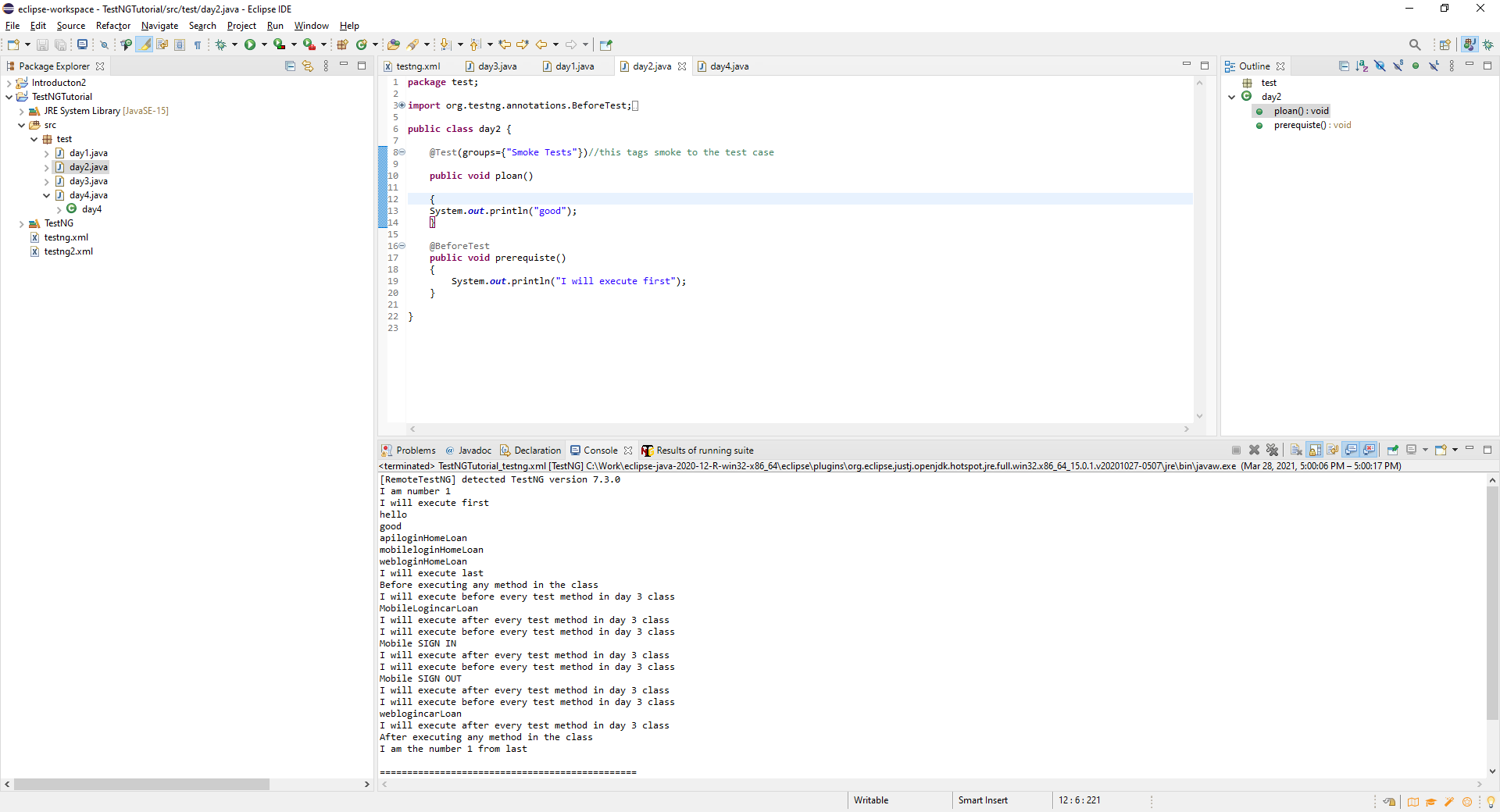
[**http://javarticles.com/2015/02/example-of-testng-configuration-xml-testng-xml.html**](http://javarticles.com/2015/02/example-of-testng-configuration-xml-testng-xml.html)

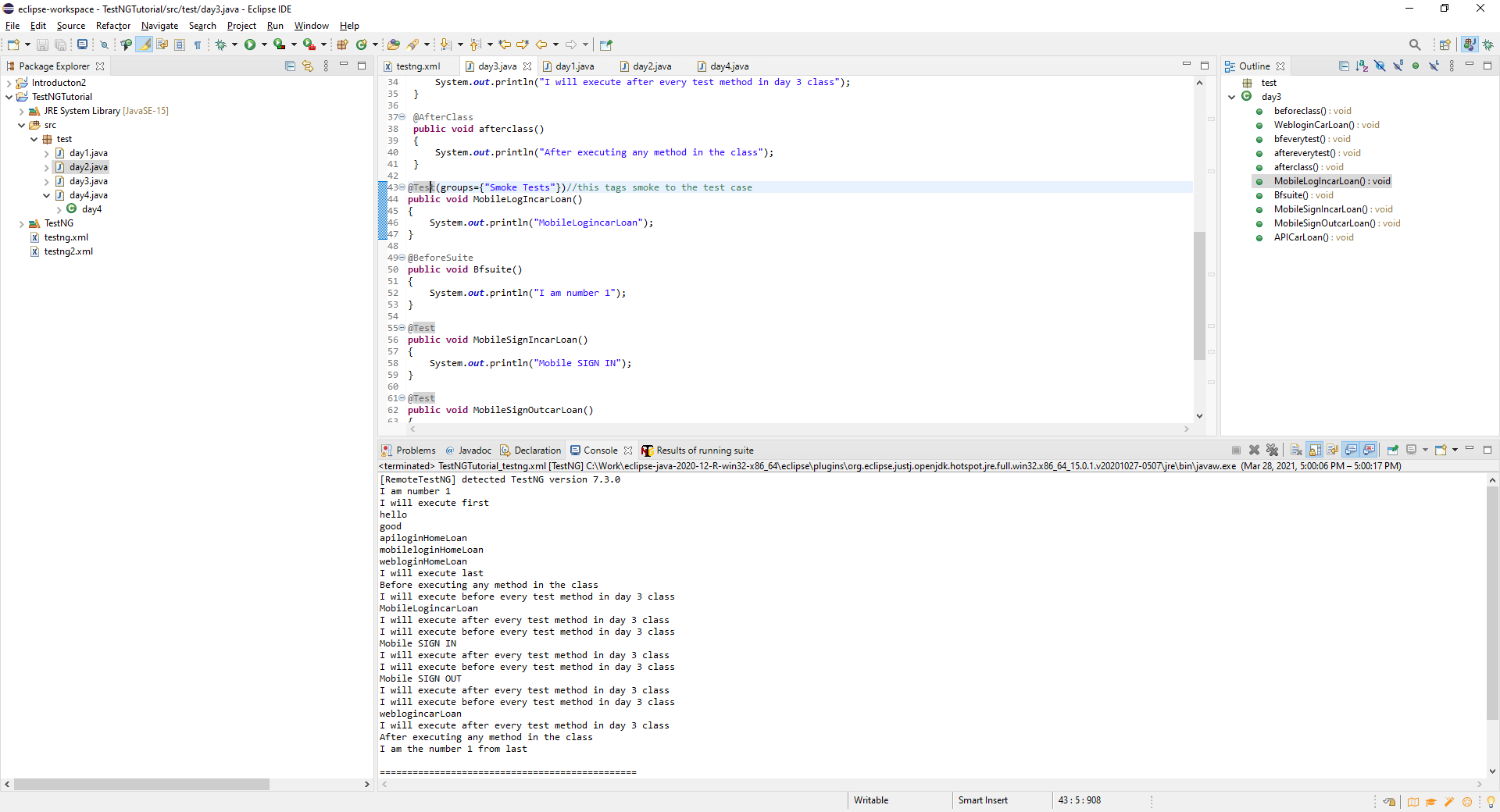
**Groups**

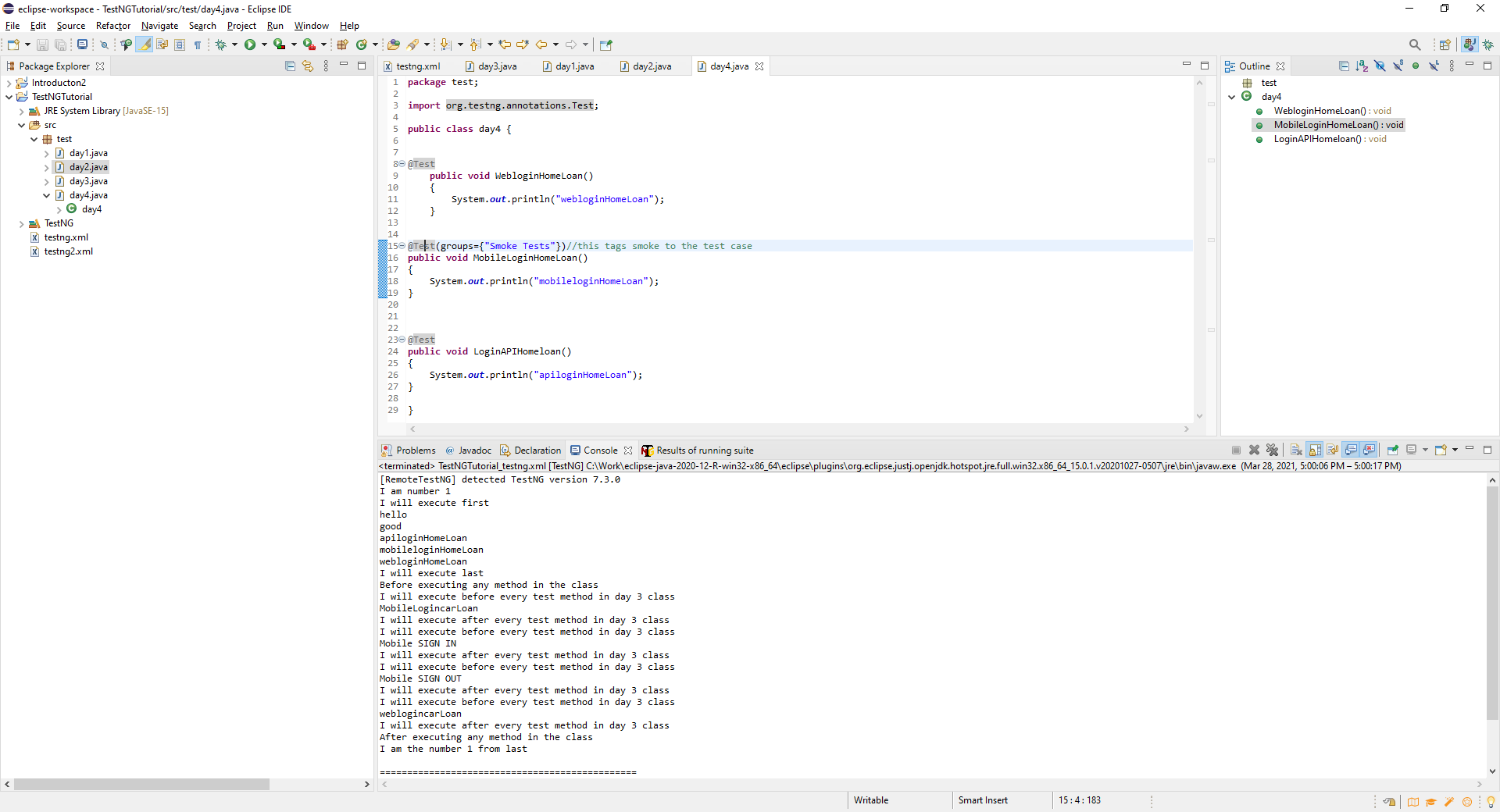
**Pick specific test cases to run after the build as smoke tests** (Random test from different class files) **. Say have 100 test cases and only want to run 40 use groups to do this.**

Must give a tag to the specific cases you want to run in this case we tag each test as a smoke test. Randomly picked test from various test class and methods.

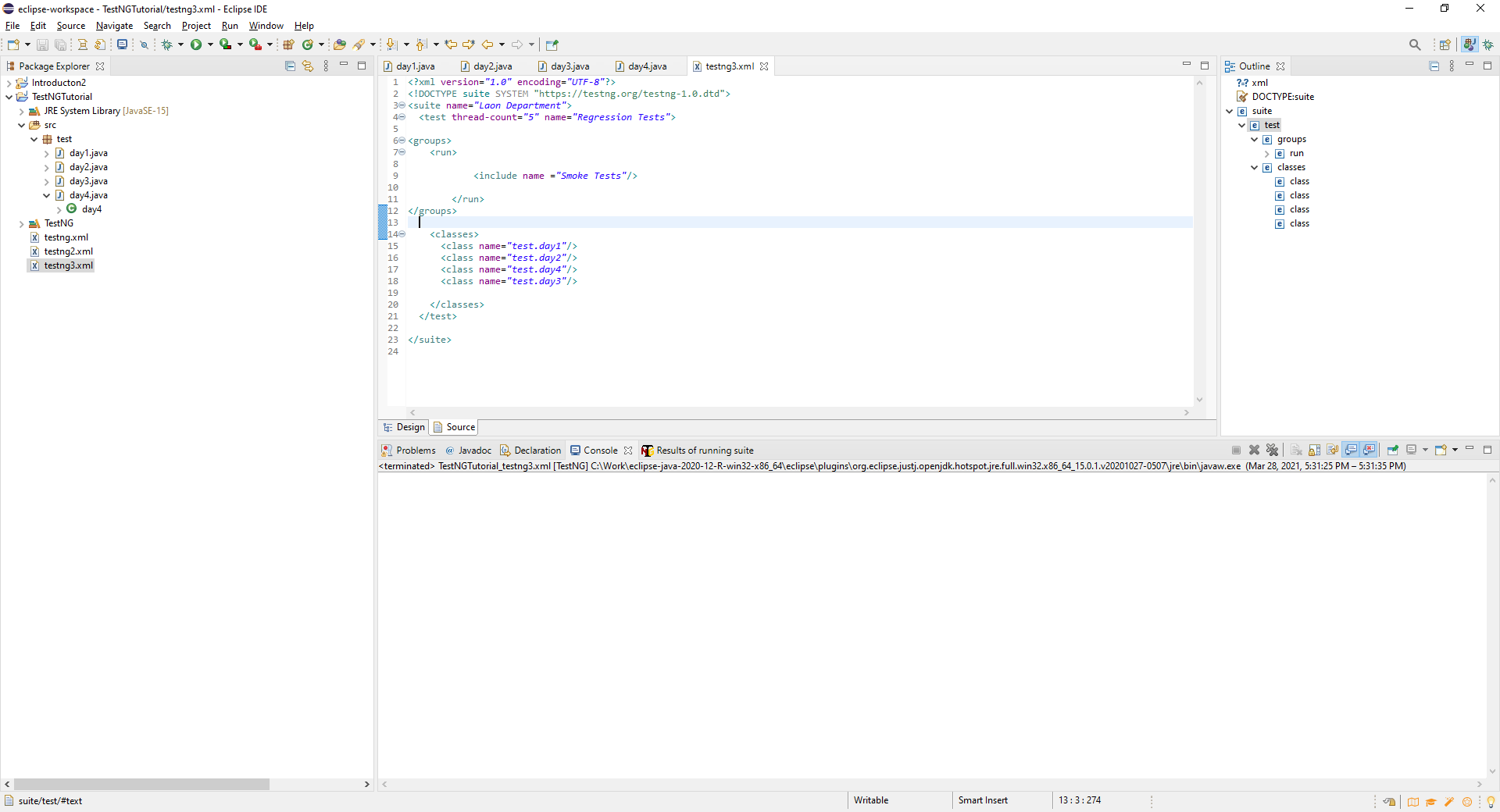
Can use <**exclude> if don’t want to run the smoke test.**



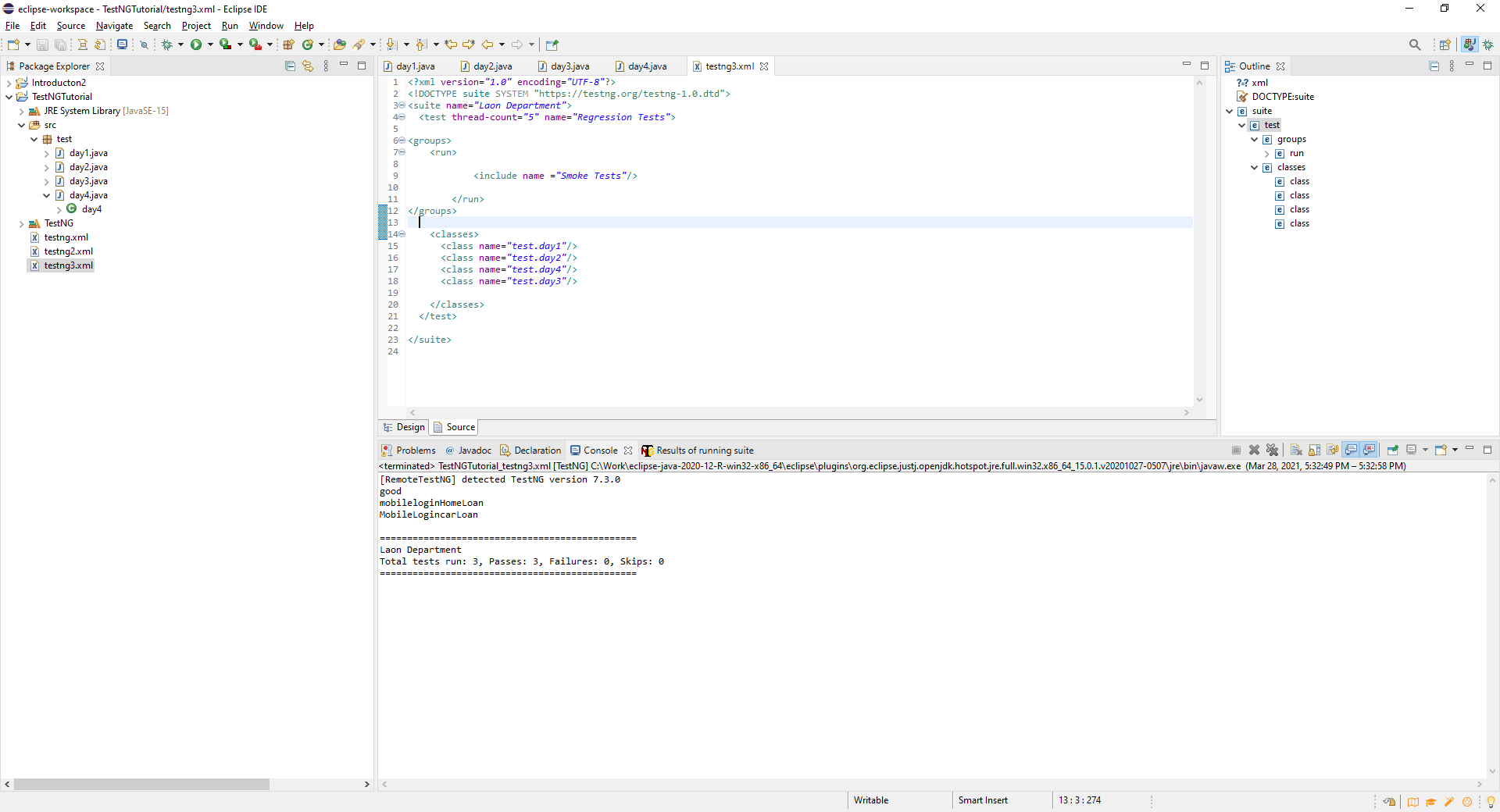




In new XML file, set up a Regression Test folder to run the tagged or selected tests

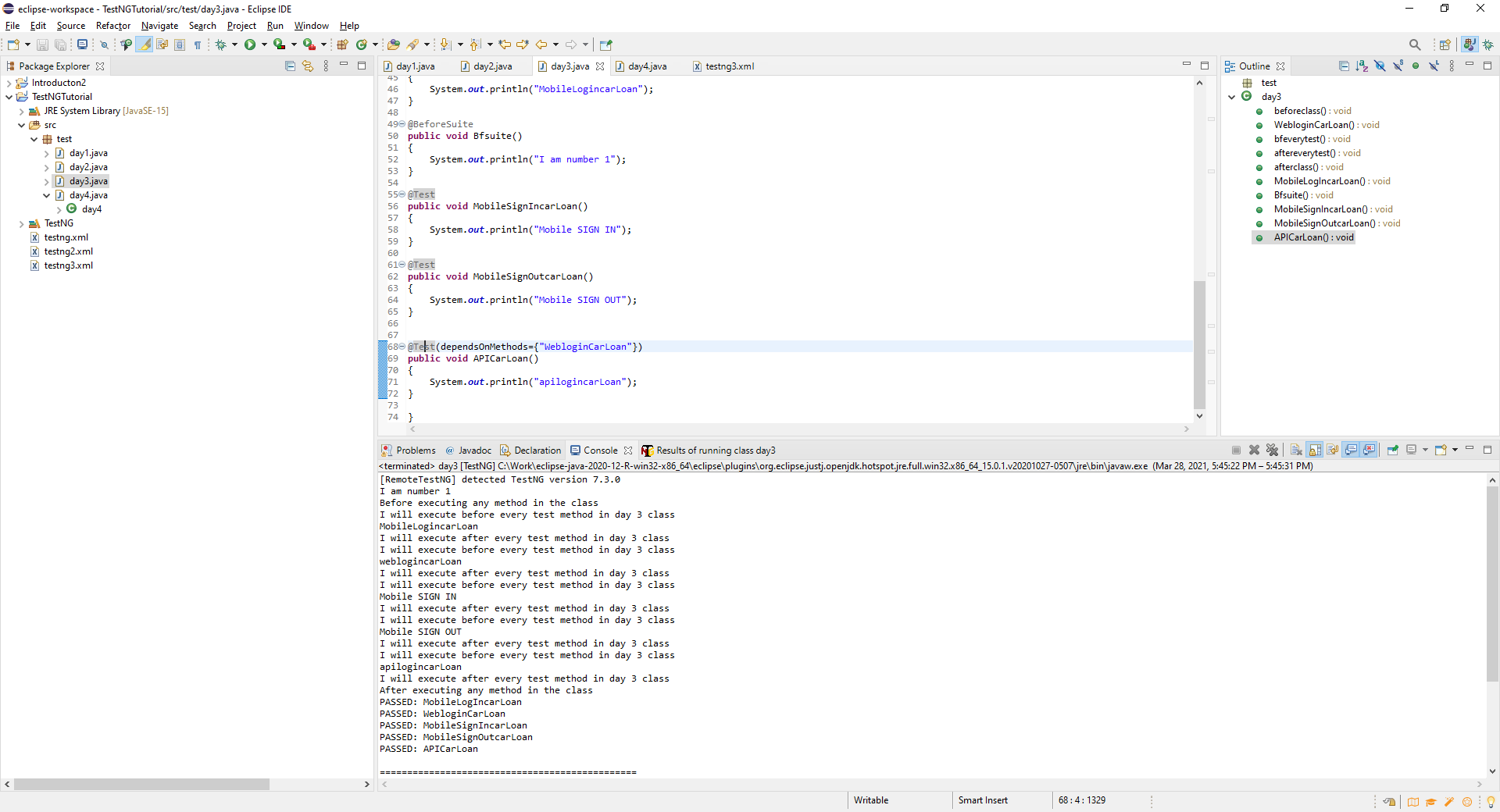


Each of the test tag as Smoke Test are run



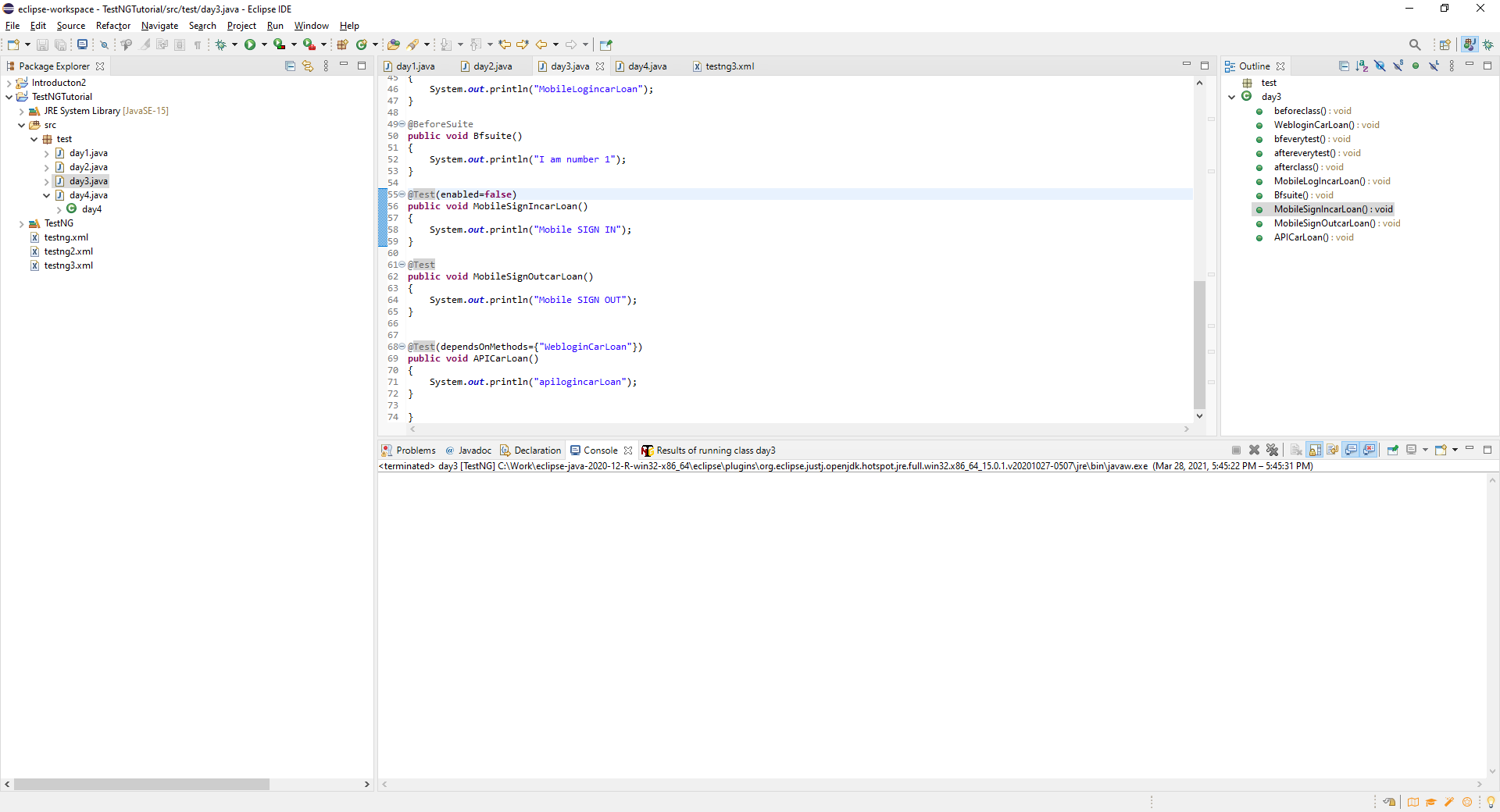
**Annotation <Depends>**

**@Test(dependsonMethods={“method,method ”})**

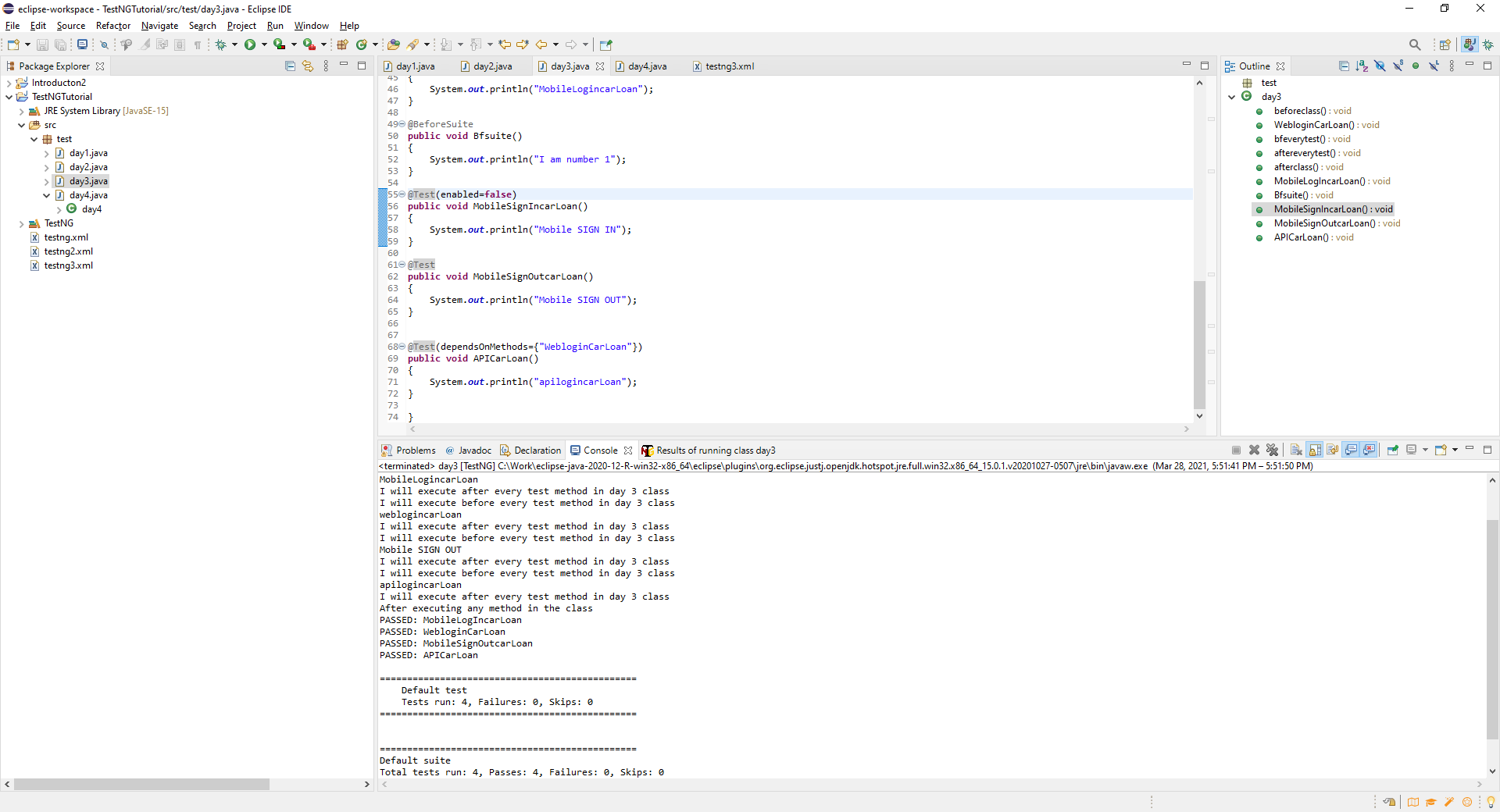


Annotation **- Enabled** = **false** – use this to skip a test case. Especially if already know a failure in the test and you don’t want framework to trigger the test case.

@Test(enabled=false change to true it will run again.

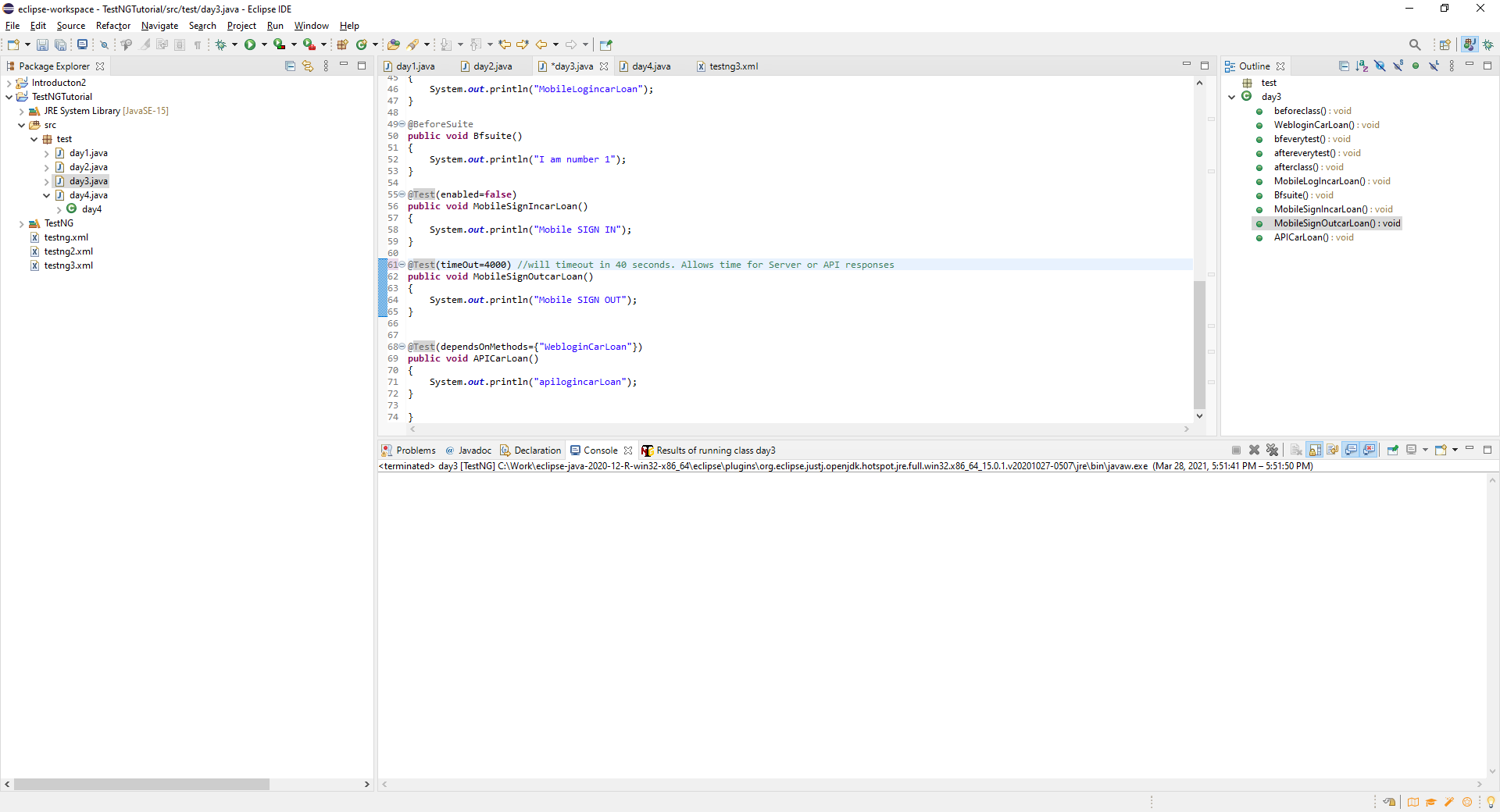


No Mobile Sign IN test because we have skipped this test.



Annotation **Timeout - CVan apply to the particular service that is takeing too much time**

**@Test(TimeOut=4000) –** This counts 40 seconds before through error that something is wrong

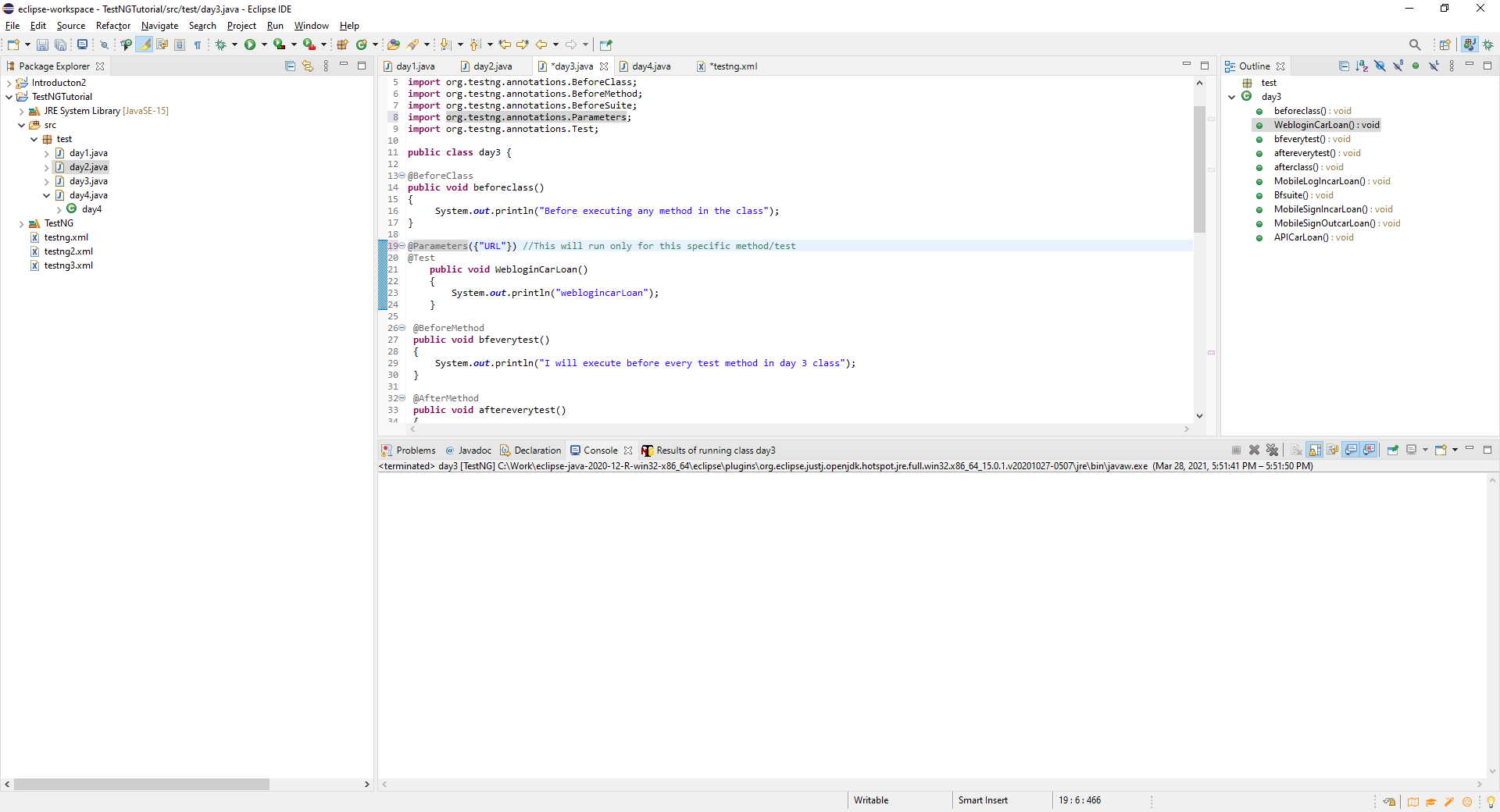


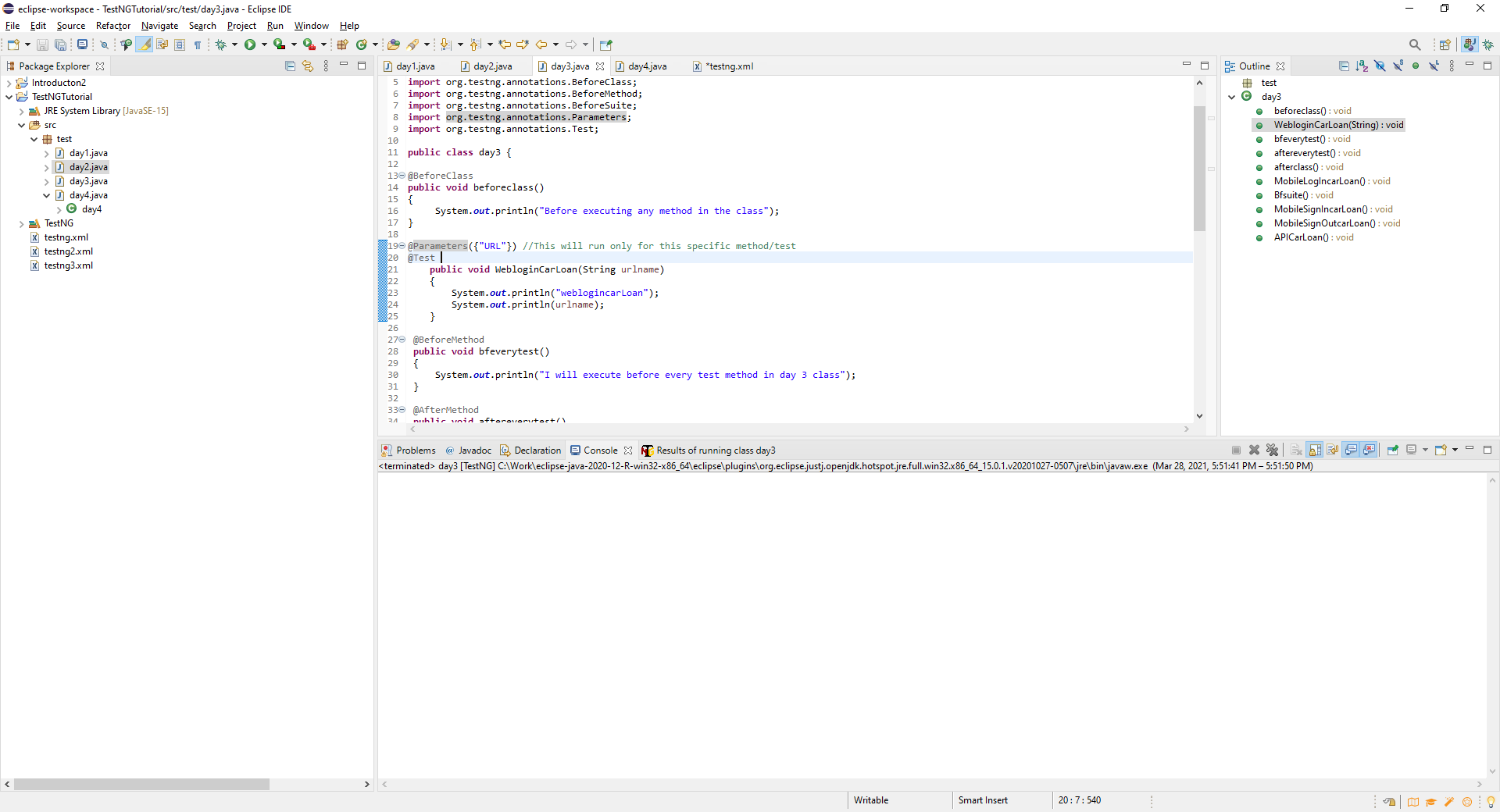
**Parameterizing from TestNG XML File**

Values common and constant across all test. Global environment variables are assigned. Allows you to avoid hardcoding stuff.

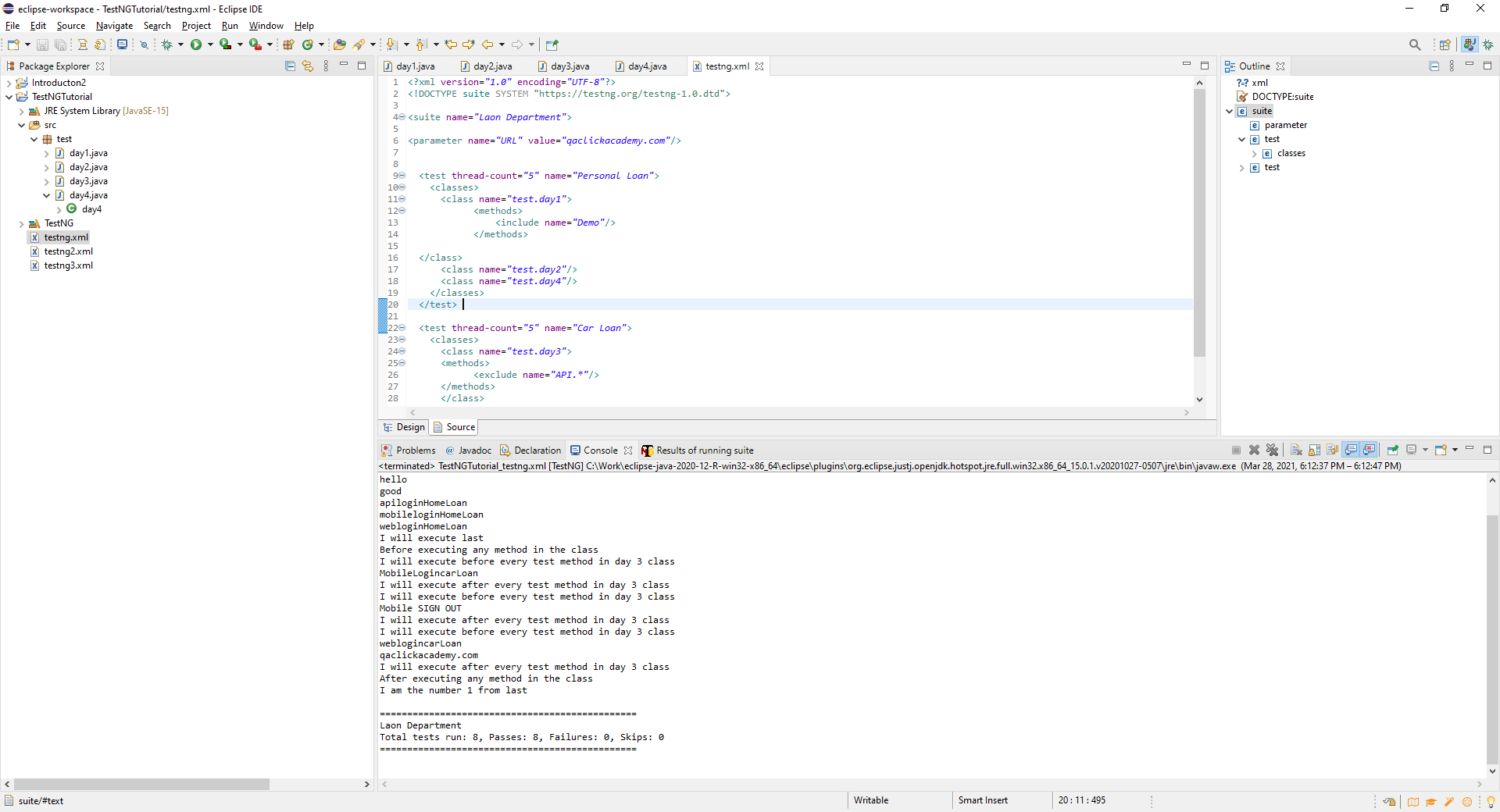
<parameter name=*"URL"* value=*"qaclickacademy.com"*/>

This is applicable to all the test cases in the Test Package

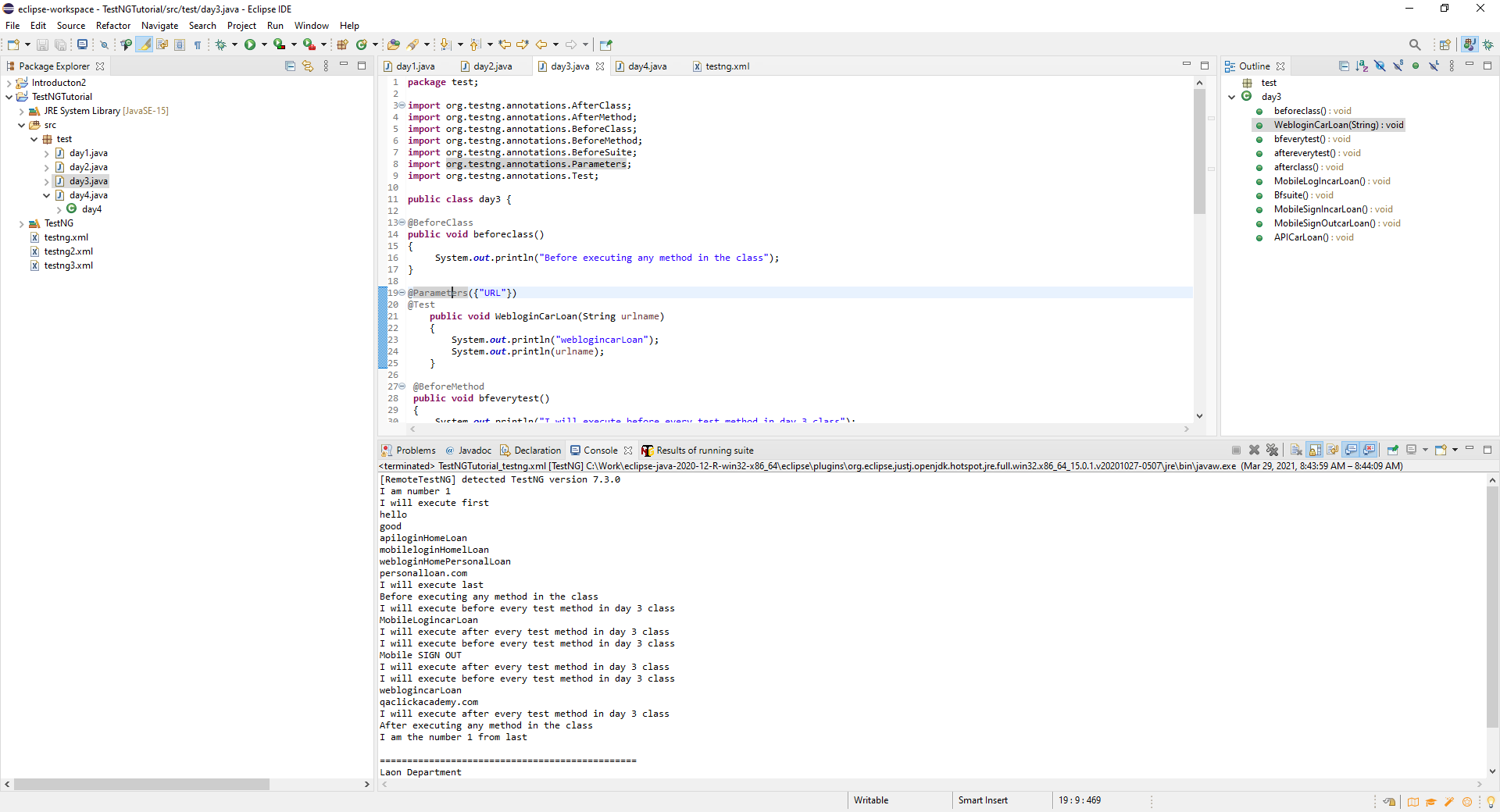




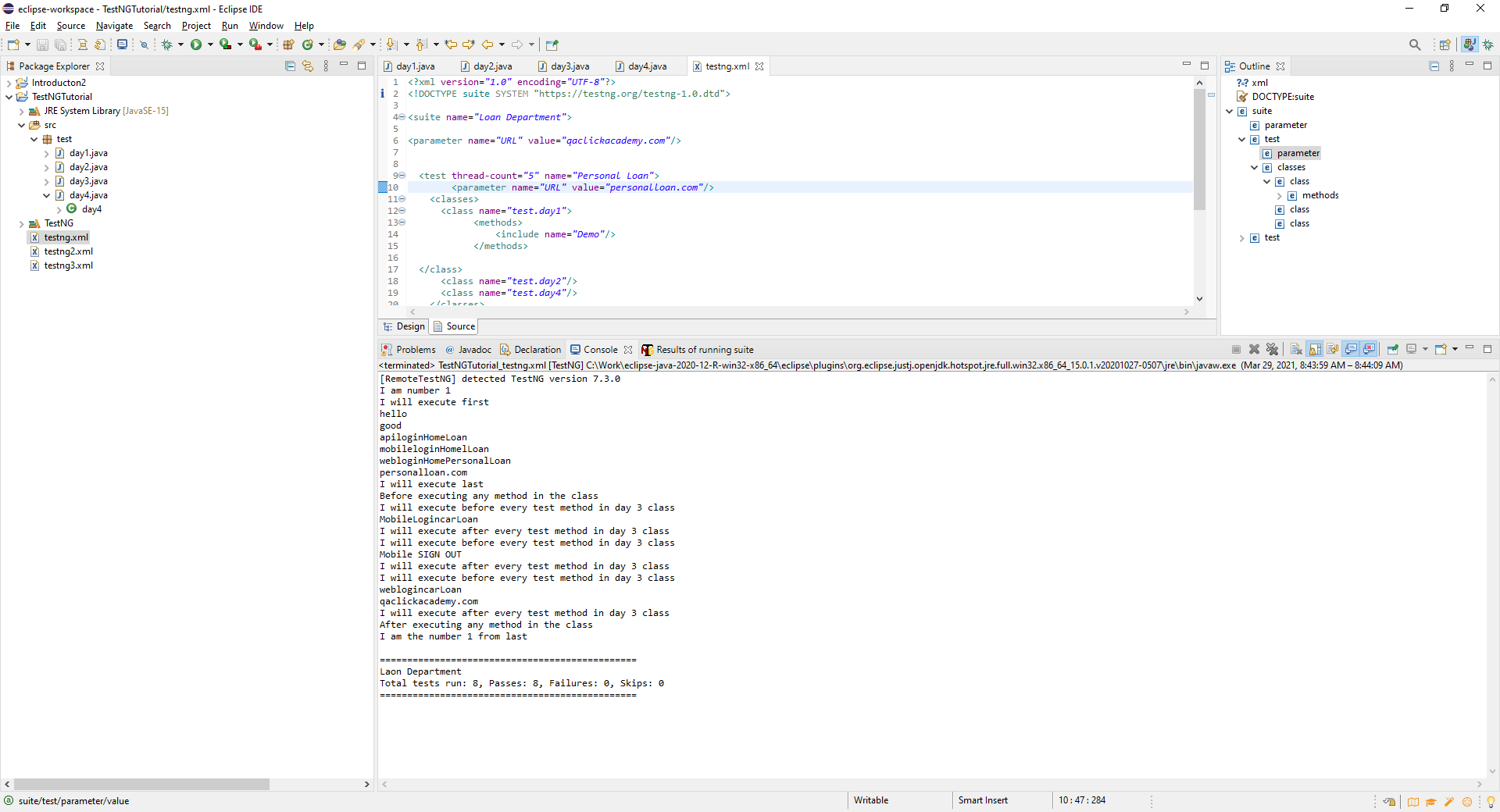
Given when then for code in Rest API will be code in the code block



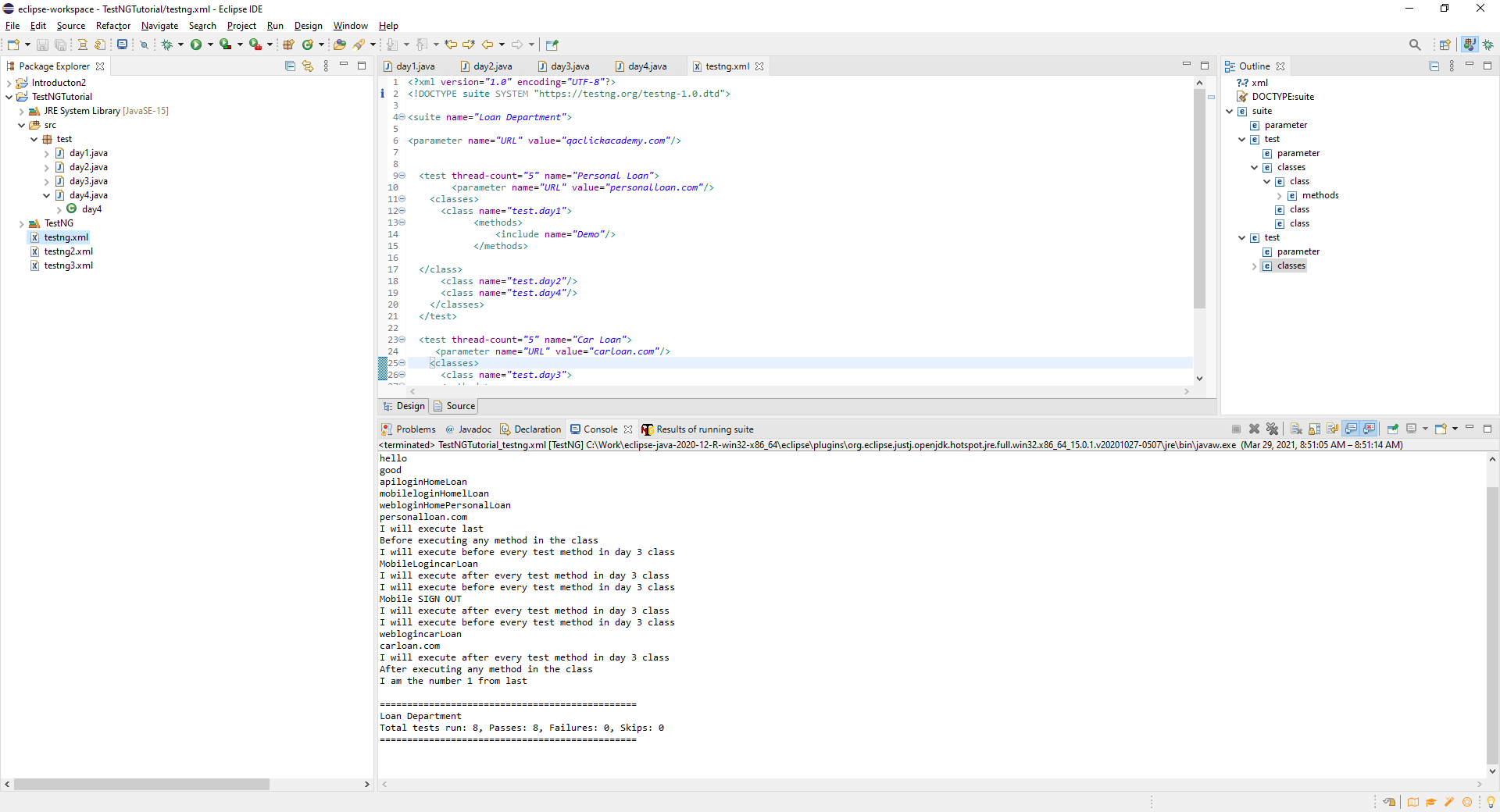
Parameters work at the test case level as well



Run the Test XML File



Updating the XML file with parmeters – Gloabla environment variables set up in the XML file



**Data Provider annotations – This is specific to Test case not defined in the XML file**

Parameterizing with multiple data sets by running test with multiple combinations. This will allow you to loop thru data and send multiple values.

1st create an array

Object [] data=new object [3][2];

