Exercise-3
JUnit

海大資工 馬尚彬

Goals

- Learn how to design unit test cases using JUnit 4.
- Learn how to conduct unit tests.
- Learn how to revise code based on the testing results.

AUT (Application Under Test)

- A Java class (MySorter.java) that sorts an array of double-typed numbers.
  - Negative numbers are not allowed.
  - Empty array is allowed.
- Examples:
  - Input: {23.1, 12.9, 99.2, 7.12}
  - Output: {7.12, 12.9, 23.1, 99.2}

Step 1: Design Test Fixtures

- Please design test fixtures, such as
  - {{1.0}, {1.0}} //test single element set
  - {{1.0, 1.1, 2.0, 2.1, 10.10, 100}, {1.0, 1.1, 2.0, 2.1, 10.10, 100}}, //test set with an incremental order
  - {{100, 10.10, 2.1, 2.0, 1.1, 1.0}, {1.0, 1.1, 2.0, 2.1, 10.10, 100}}, //test set with a decremental order
  - {{2.0, 2.1, 1.1, 10.10, 100, 1.0}, {1.0, 1.1, 2.0, 2.1, 10.10, 100}}, //test set with random order
  - {{1,2,5,4,3}, {1,2,3,4,5}} //test set with all integer value
  - {{1,2,-99}, {1,2,-99}} //test set with negative number
- Are these test fixtures enough to verify the AUT (please check the specification carefully)?
  - At least another test case should be added.

Step 2: Design JUnit Test Cases1

- Create a new project
  - File: [New] →[Java Project]→[TestSort]
  - Copy "MySorter.java" to the "src" folder
- Configure the library of JUnit4
  - Right-click the project→[Build Path]→[Configure Build Path]
  - Switch to [Libraries] and click [Add Library]
  - Select [JUnit] → [Next]
  - Select [JUnit4] →[Finish]

Step 2: Design JUnit Test Cases2

- Design your test cases
  - Right-click the project→[New]→[JUnit Test Case]
    - Fill in the name "MySorterTest“ (or any name you like) and check [setUp] and [tearDown]→[Finish]
  - Develop methods of setup, teardown and your test cases, such as testSort1, testSort2, and so forth.

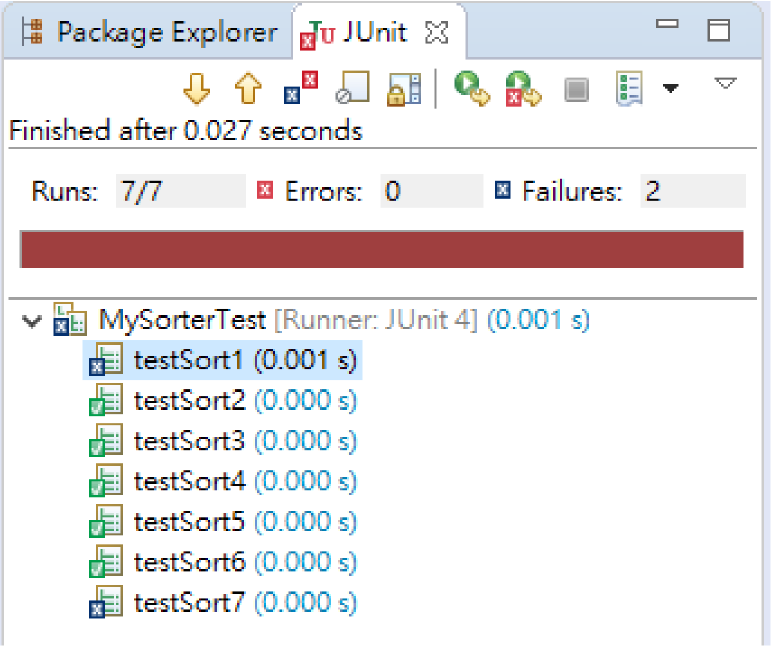
Step 2: Design JUnit Test Cases$_3$

- □ Because the JUnit4 does not provide the API of *assertDoubleArrayEquals*, you may need the method below:

```
private void assertDoubleArrayEquals (double[] expected,
double[] actual)
{
        assertEquals(expected.length, actual.length);
        for (int i = 0; i < expected.length; i++)
        {
                assertEquals(expected[i], actual[i], 0.0001);
        }
}
```

Step 3: Run JUnit Test

- □ Right-click [MyTestSorter] and click [Run As: JUnit Test]
- □ You can check the test results like the figure below:



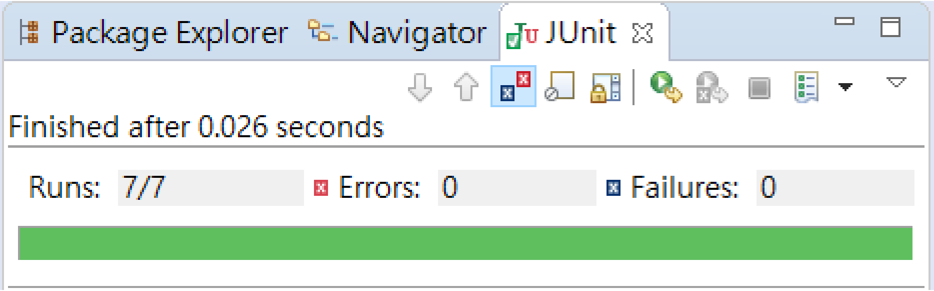
```
Finished after 0.027 seconds

Runs: 7/7    Errors: 0    Failures: 2

MySorterTest [Runner: JUnit 4] (0.001 s)
    testSort1 (0.001 s)
    testSort2 (0.000 s)
    testSort3 (0.000 s)
    testSort4 (0.000 s)
    testSort5 (0.000 s)
    testSort6 (0.000 s)
    testSort7 (0.000 s)
```

Step 4: Revise MySorter.java

- Please revise MySorter.java (the AUT) until all test cases pass.

結果上傳

- 請將結果上傳至 TronClass，包含以下三部分
  - 你修改的MySorterTest.java
  - 你修改的MySorter.java
  - 通過所有單元測試之截圖(類似下圖)



Finished after 0.026 seconds

Runs: 7/7 Errors: 0 Failures: 0