

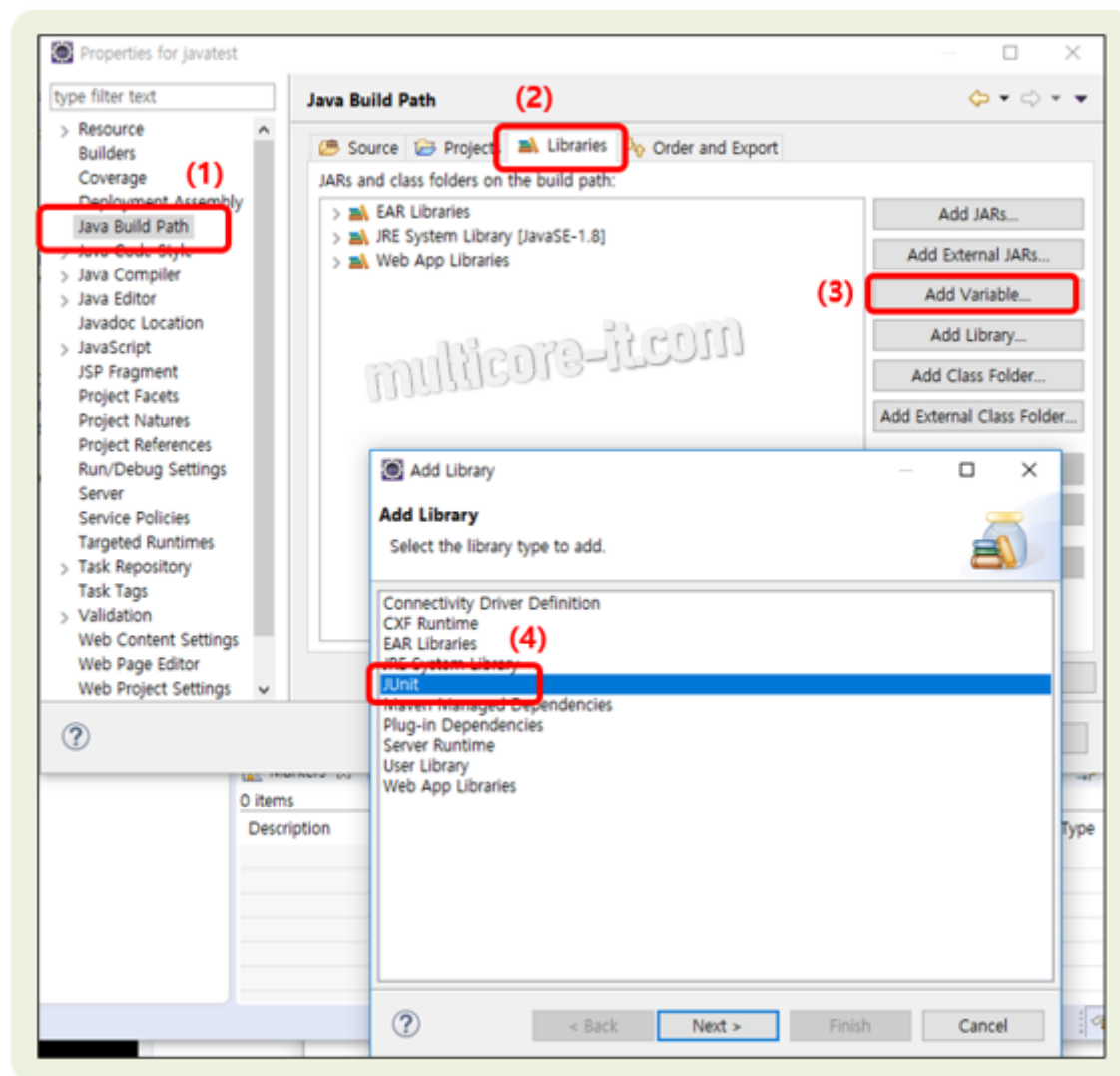
대표적인 단위테스트 도구 JUnit

2018.11.19 15:25

#JUnit

□ 대표적인 단위테스트 도구 JUnit

JUnit은 이클립스로 개발하는 자바 환경에서 가장 많이 사용되는 단위 테스트 도구이다. 테스트 함수를 단일 패키지로 묶을 수 있으며 다양한 검증 함수를 지원해 프로그램이 반복적으로 수정되는 프로그램 개발 과정에서 테스트 케이스를 반복적으로 적용할 수 있다. 단위 테스트 시간을 획기적으로 줄일 수 있을 뿐만 아니라 테스트 품질도 향상시킬 수 있다.



JUnit, 설정

JUnit을 이클립스에 기본적으로 내장되어 있지만 사용을 위해서는 라이브러리를 추가해 줘야 한다. 이클립스 프로젝트 환경설정에서 Java Build Path > Library > Add Library에서 JUnit을 추가해주면 된다.

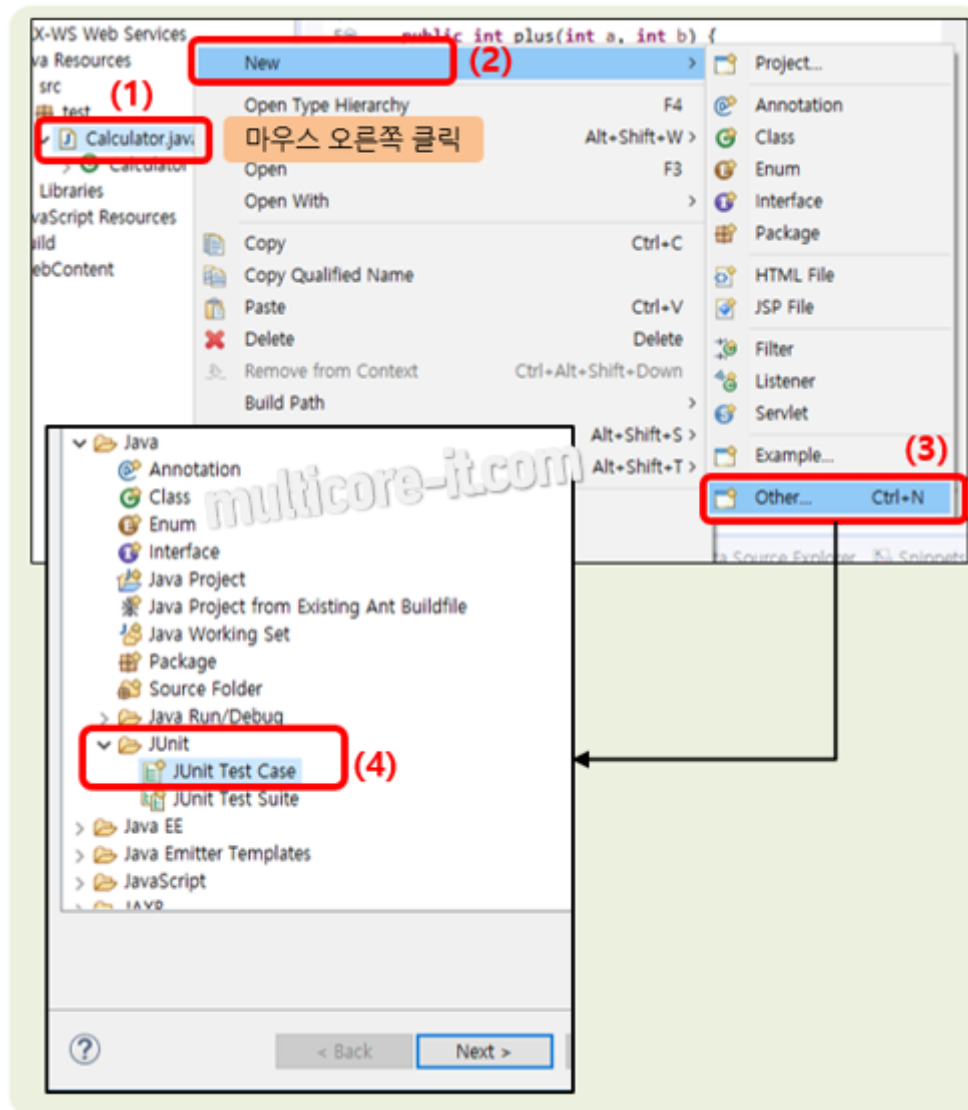
```

1 package test;
2
3 public class Calculator {
4
5     public int plus(int a, int b) {
6         return a+b;
7     }
8
9     public int minus(int a, int b) {
10        return a-b;
11    }
12
13    public int multiple(int a, int b) {
14        return a*b;
15    }
16
17    public float divide(int a, int b) {
18        return a/b;
19    }
20 }
21
22
23

```

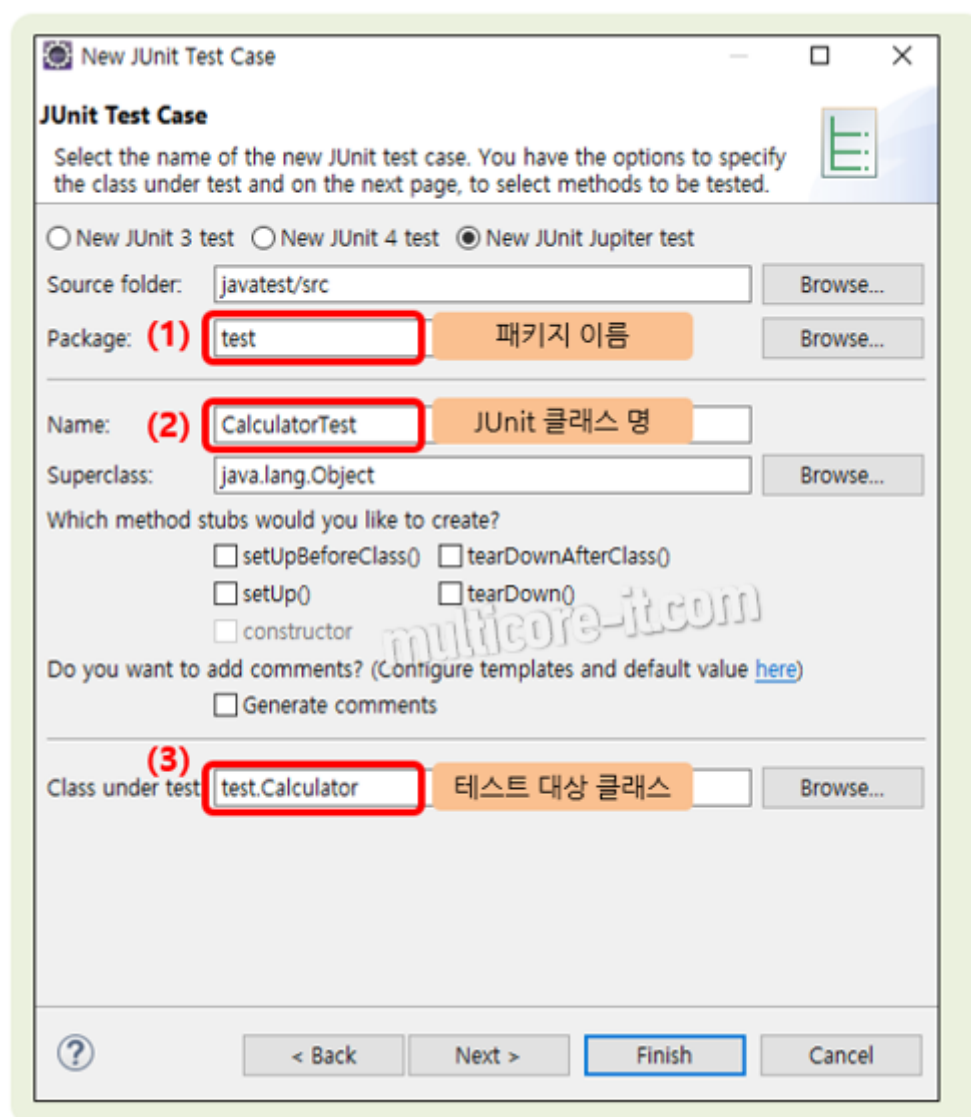
사칙연산 프로그램 개발

본격적으로 JUnit을 사용하기 전에 테스트를 위해 사칙 연산하는 클래스 Calculator.java를 만들어보자. Calculator.java는 plus, minus, multiple, divide 이렇게 모두 4 개의 메소드를 가지고 있다.



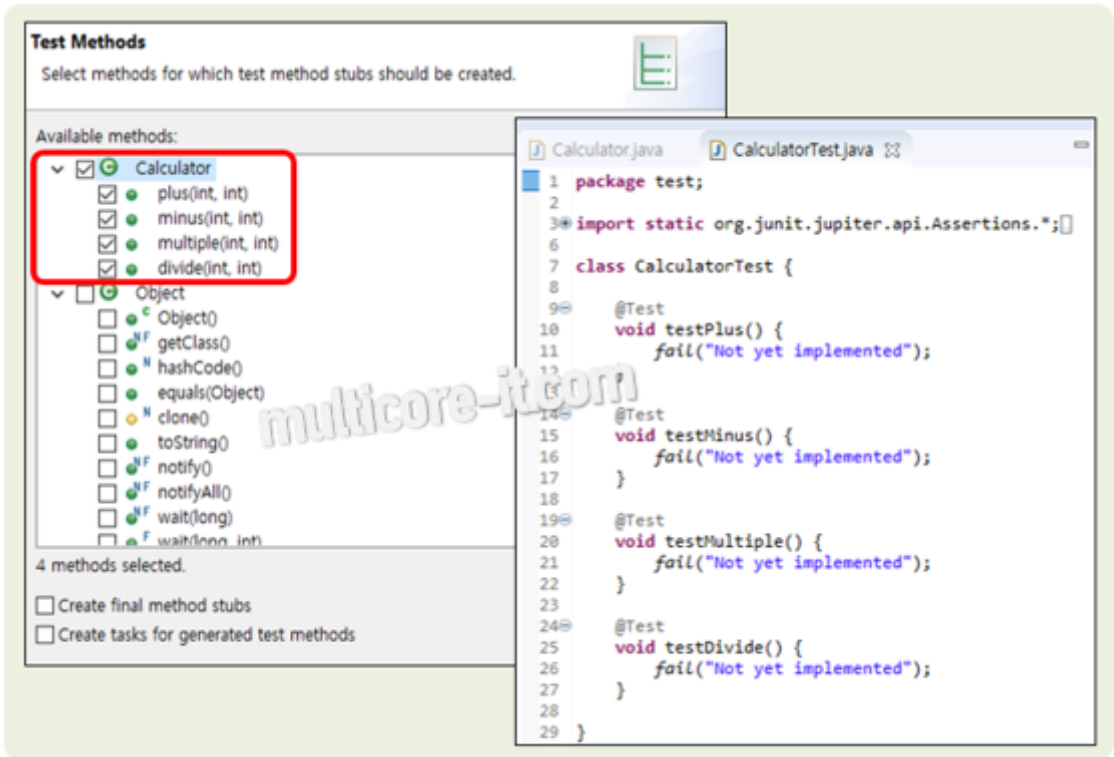
JUnit 클래스 생성 1/3

이제 JUnit 테스트 클래스를 만들어보자. 좌측 Project Explorer 영역에서 테스트할 클래스를 선택하고 마우스 오른쪽 버튼을 클릭해 [New > Other] 메뉴를 선택하면 새로운 팝업 창이 새로 나온다. 아래로 좀 내려오면 JUnit 영역이 나오고 한번 더 클릭해 JUnit Test Case를 선택하자.



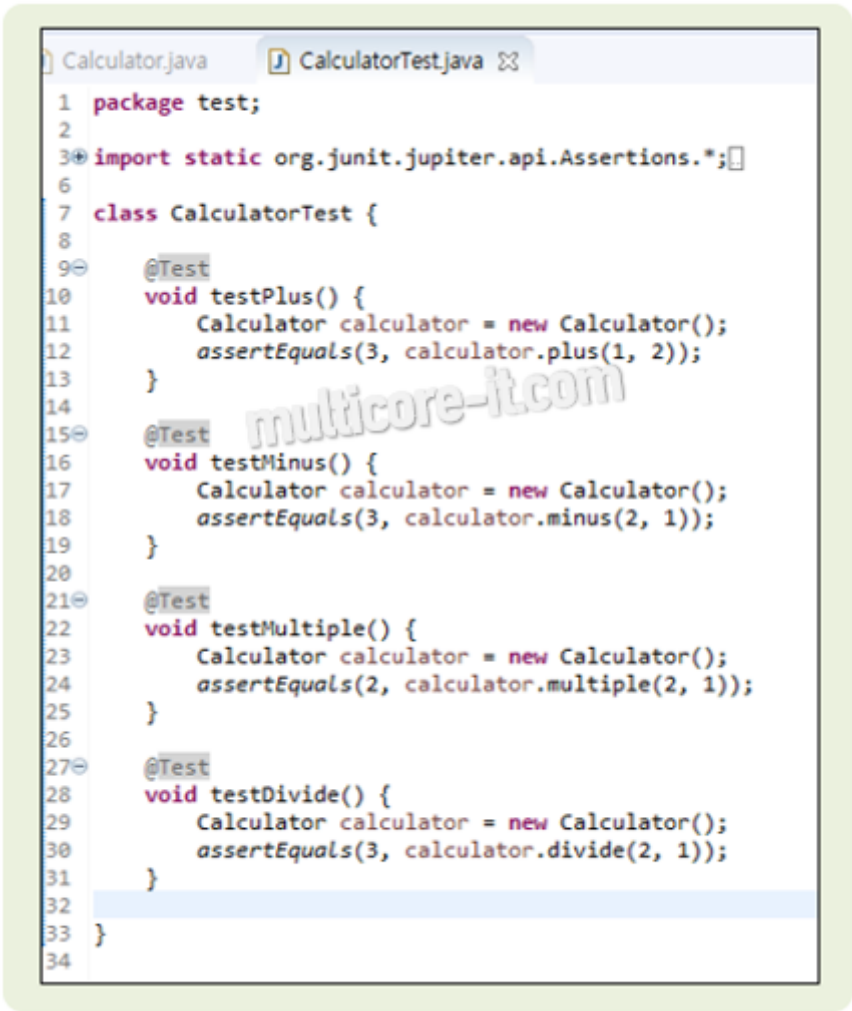
JUnit 클래스 생성 2/3

JUnit 테스트 케이스 생성화면에서 파일이 생성될 패키지 이름, 클래스 이름 그리고 어떤 클래스를 테스트할 지 정보를 확인할 수 있다. 실제로 JUnit을 사용할 때는 별도의 패키지를 만들어서 JUnit 테스트 케이스 클래스만 별도로 모아두는 것이 좋다. 운영에 적용할 때 JUnit 테스트 케이스 클래스는 필요가 없기 때문에 삭제하거나 운영으로 이관이 안되도록 설정하는 것이 좋다. 별도의 패키지에 파일을 모아 놓으면 이런 설정을 쉽게 할 수 있다. 이 책에서는 편의를 위해 테스트 대상 클래스와 같은 패키지에서 JUnit 클래스를 만들도록 한다.



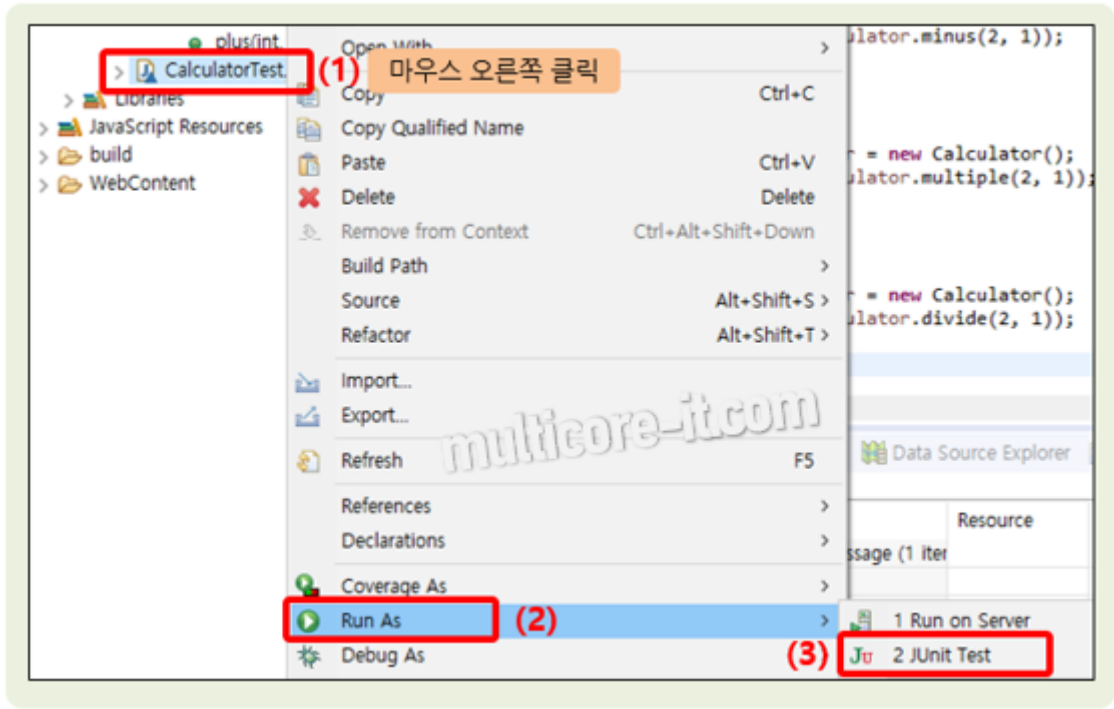
JUnit 클래스 생성 3/3

다음 버튼을 누르면 테스트할 메소드를 선택할 수 있다. 사칙 연산을 위한 모든 메소드를 선택하자. CalculatorTest.java 클래스가 생성됐고 사칙연산과 관련된 테스트 메소드 4개가 생성된 것을 확인할 수 있다. 아직 세부적인 테스트 케이스가 지정되지 않았기 때문에 “Not yet implemented”라고 지정되어 있다.



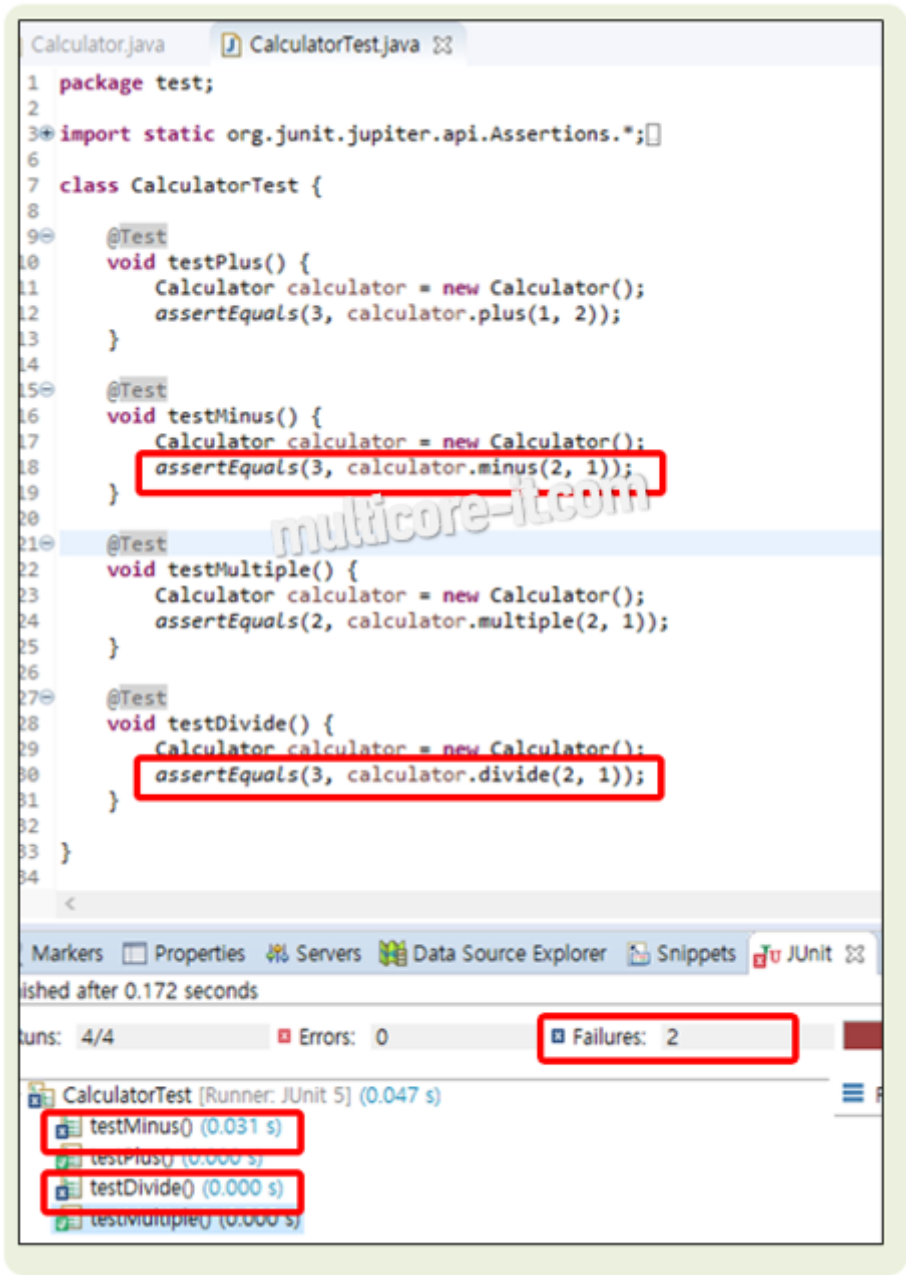
JUnit 테스트 케이스 생성

이제 테스트 함수에 테스트 케이스를 포함하고 있는 검증 함수를 작성해서 JUnit 클래스를 완성해 보자. assertEquals는 두 개의 인자의 결과가 같으면 테스트 성공을 의미한다. JUnit은 다양한 검증 함수를 제공하며 JUnit에서 제공하는 API 문서에 다양한 함수에 대한 설명을 찾을 수 있다. <http://JUnit.sourceforge.net/javadoc/org/JUnit/Assert.html> 사이트를 참고해 보자.



JUnit 단위테스트 실행

검증 함수까지 개발이 완료됐으면 이제 단위 테스트를 실행할 차례다. JUnit 클래스를 선택하고 마우스 오른쪽 버튼을 클릭하면 새 창이 나오는데 Run As > JUnit Test 를 선택하면 단위 테스트를 실행할 수 있다.



JUnit 단위테스트 결과 분석

단위 테스트를 실행하면 화면 하단에 JUnit 이라는 이름의 탭이 하나 생기고 결과가 표시된다. 단위 테스트에는 모두 2개의 실패 (Failures)가 나왔는데 아래쪽으로 좀 더 내려와보면 testMinus와 testDivide 함수 앞에 x 표시가 된 것을 확인할 수 있다. 더블 클릭하면 화면 상단에 소스코드에 실패가 발생한 위치로 바로 이동한다. 실패가 발생한 코드를 살펴보고 프로그램을 수정하고 다시 CalculatorTest 클래스를 실행해 오류가 정정됐는지 확인할 수 있다.

JUnit을 사용하는 이유는 테스트 케이스를 함수로 작성해서 관리할 수 있고 별도의 클래스에 테스트 케이스를 보관해 프로그램이 변경 되더라도 기존의 테스트 케이스를 그대로 활용할 수 있기 때문이다. 또한 다양한 검증 함수를 지원하기 때문에 복잡한 테스트 케이스를 적용할 수 있다.

웹 테스트 자동화 도구 셀레니움 (0)	2018.11.19
대표적인 단위테스트 도구 JUnit (0)	2018.11.19
프로그램 단위 테스트 (0)	2018.11.19
효율적으로 코딩하기 (0)	2018.11.19
팀으로 개발하기 (0)	2018.11.19
개발환경 구축하기 (0)	2018.11.19

NAME

PASSWORD

HOMEPAGE

http://

SECRET☐

WRITE

+ Recent posts

9. 카트폴 DQN 모델 실행
8. 카트폴 DQN 강화학습
7. 텐서플로우 심층 신경망
6. 텐서플로우 단일 신경망