JUnit & Eclipse

201260053 Abbos Shomurodov
201260058 채숭흠

TEAM 1
200711437 성하진
200511355 정용구
200911436 조성완

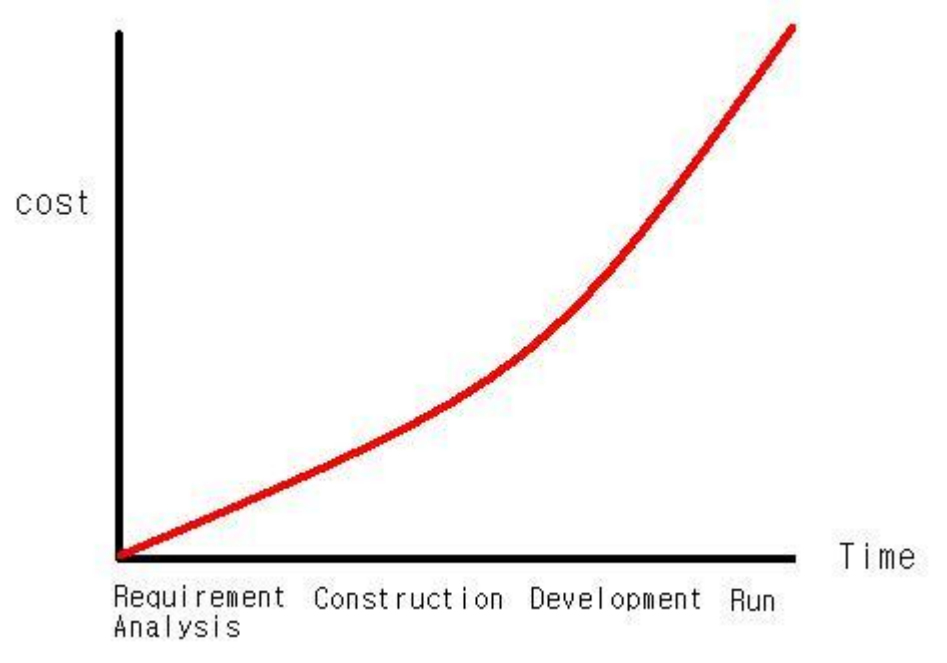
Contents

1. Software Testing Theory
2. Eclipse Plug-in
3. JUnit Practice

Software Testing Theory

1.1 Importance of testing
1.2 Test Driven Development(TDD)
1.3 Extreme Programming(XP)
1.4 Unit Test
1.5 Structural Test
1.6 JUnit

Importance of Testing



- 프로그램 개발 도중, 결함을 발견한 시기가 늦어질수록 수정 비용이 늘어난다.
- 프로그램의 결함 유무를 점검하면서 개발하는, 유동적이며 변화에 기민한 개발 방법론이 필요

As find problems lately, the cost of modification grows.

Importance of Testing(cont.)

- **Timings of software testing**
  - During development
  - After organize the requirements
  - Complete timing of coding

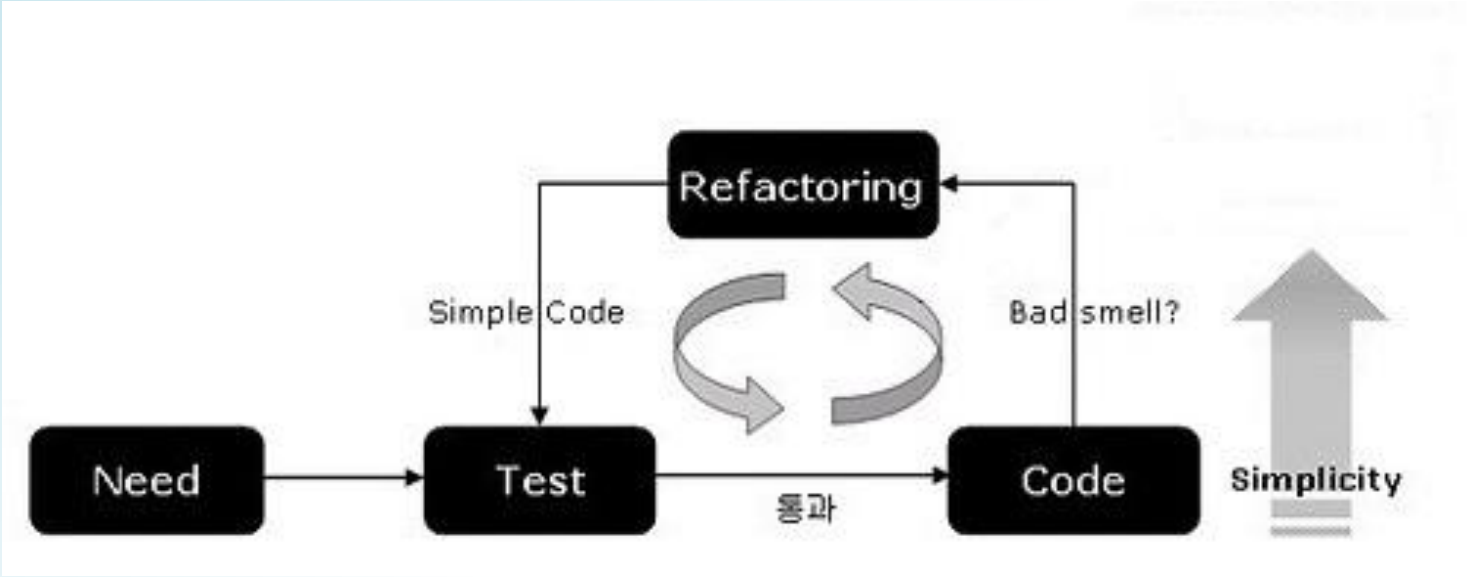
⟹ **즉, 개발 내내 반복적으로 테스트**

Test Driven Development

- Not a test technique, analysis technique
  - 테스트 기법이 아닌 분석, 설계 기술
- Related to the test-first programming concepts of XP
  - 테스트 중심의 빠른 개발 방법인 Extreme Programming과 관련
- Relies on the repetition of a short development cycle
  - 테스트-개발-테스트의 반복

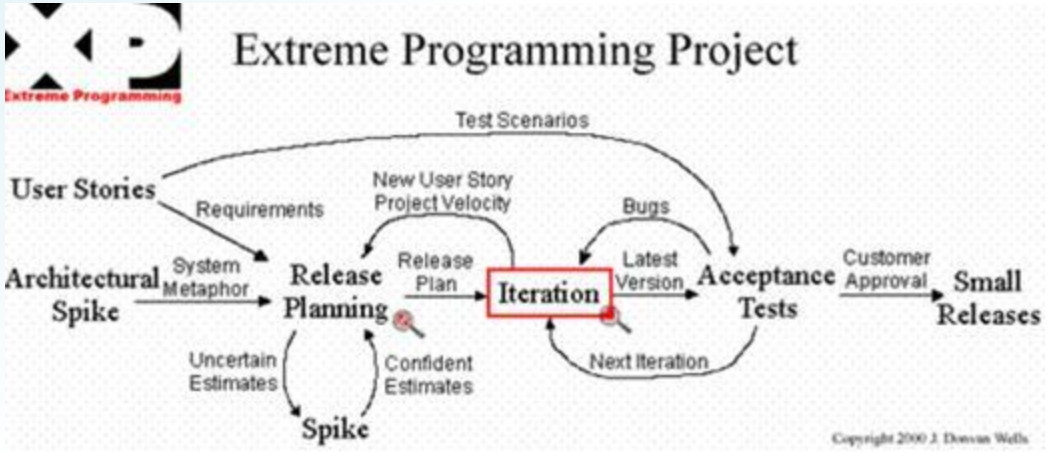
Test Driven Development(cont.)

- 일반적인 개발 과정
  - 디자인 → 개발 → 테스트
- TDD 개발 과정
  - 테스트 코드 작성 → 개발 → 리팩토링



Extreme Programming(XP)

- Extreme Programming is a discipline of software development based on values of simplicity, communication, feedback, and courage.
- 불필요한 작업은 최소한으로 줄여, 단기간에 가능한 좋은 제품을 고객에게 제공하는 개발 방법론
- 빠른 개발과 테스트 중심의 프로젝트 진행
- 리팩토링 과정을 통해 적절한 기능의 추가와 보완 가능



Unit Test

- What is Unit

A unit is the smallest testable part of an application. In procedural programming a unit could be an entire module but is more comm only an individual function or procedure. In object-oriented progra mming a unit is often an entire interface, such as a class, but coul d be an individual method.

- Application의 가장 작은 단위인 Unit을 테스트(모듈, 객체 등)
- TDD concept에 따라, 개발 중 반복적으로 테스트 코드 작성
- Unit Test Framework : xUnit
- Black Box Test
  - Focus on testing functional requirements

Structural Test

- Structure based test
- tests internal structures or workings of an application, as opposed to its functionality
- Coverage 작성의 대표적인 예
  - Coverage = (수행 성공 대상/총 대상) * 100
- White Box Test
  - Focus on using internal knowledge of the software

xUnit

• XP style unit testing framework. xUnit uses a protocol between a front-end to display test results and a test driver linked into applications - removing the link time dependencies between the application and the graphics, formatting libraries.

⇒ Unit Test에 사용하기 위한 Framework

▷ **xUnit 종류**

Name	JUnit	NUnit	CUnit	CppUnit	PHPUnit
Language	Java	.NET	C	C++	PHP

JUnit

- What is JUnit
  - Unit testing framework for the Java programming language.
  - JUnit has been important in the development of TDD, and is one of a family of unit testing frameworks collectively known as xUnit.

  ⇒ Java에서 쓰는 xUnit

- Automation of code execution and results after
  - 테스트 코드는 개발자가 작성
  - XML과 HTML 로 출력되는 결과를 다른 포맷으로 변환 및 저장 가능

- Eclipse에 내장(www.junit.org에서 download 가능)

JUnit 3 & JUnit 4

JUnit 3		JUnit 4
• Method는 Test로 시작		• @Test로 Method 지정가능
• TestCase를 상속받음		• 독립 클래스로 작성
• setUp(), tearDown()로 Fixture 일일이 지정		• @Before, @After로 직관적인 표현

JUnit 4

- Annotation
  - @Before: 테스트 전에 실행할 메소드 지정
  - @After: 테스트 후에 실행할 메소드 지정
  - @BeforeClass: 테스트 전에 한 번만 실행할 메소드 지정
  - @AfterClass: 테스트 후에 한 번만 실행할 메소드 지정
  - @Test(timeout=100): 제한시간 설정
  - @Ignore: 테스트 제외

- 직관적인 테스트 코드 작성 가능

```
@Test
public void verifyGoodZipCode() throws Exception{
 Matcher mtcher = this.pattern.matcher(phrase);
 boolean isValid = mtcher.matches();
 assertEquals("Pattern did not validate zip code", isValid, match);
}
```

JUnit 4(cont.)

- JUnit Methods
  - assertEquals - 같은지 비교
  - assertNull - null값을 리턴하는지 비교
  - assertNotNull - 인자로 넘겨받은 객체가 null인지 판정하고 반대인경우 실패로 처리
  - assertSame - expected 와 actual이 같은 객체를 참조하는지 판정하고 그렇지 않다면 실패로 처리
  - assertNotSame - expected 와 actual이 서로 다른 객체를 참조하는지 판정하고, 만약 같은 객체를 참조한다면 실패로 처리
  - assertTrue - boolean 조건이 참인지 판정
  - assertTrue - boolean 조건이 거짓인지 판정
  - fail - 테스트를 바로 실패 처리

Eclipse Plug-in

1.1 Ant
1.2 Subclipse
1.3 TPTP

Ant

- 빌드 자동화 툴
  - Eclipse에 내장
  - XML형태의 Ant문서 작성(*.ant)
  - 컴파일, 폴더 분류, FTP업로드, 파일 압축 등의 작업 일괄 처리 가능

- FTP사용
  - Apache에서 ftp라이브러리 download
  - 해당 라이브러리를 ant Class path에 등록
  - Xml파일에 ftp 정보 작성

Subclipse

- SVN(Subversion)
  - Software versioning and revision control system
  - 버전 관리 시스템

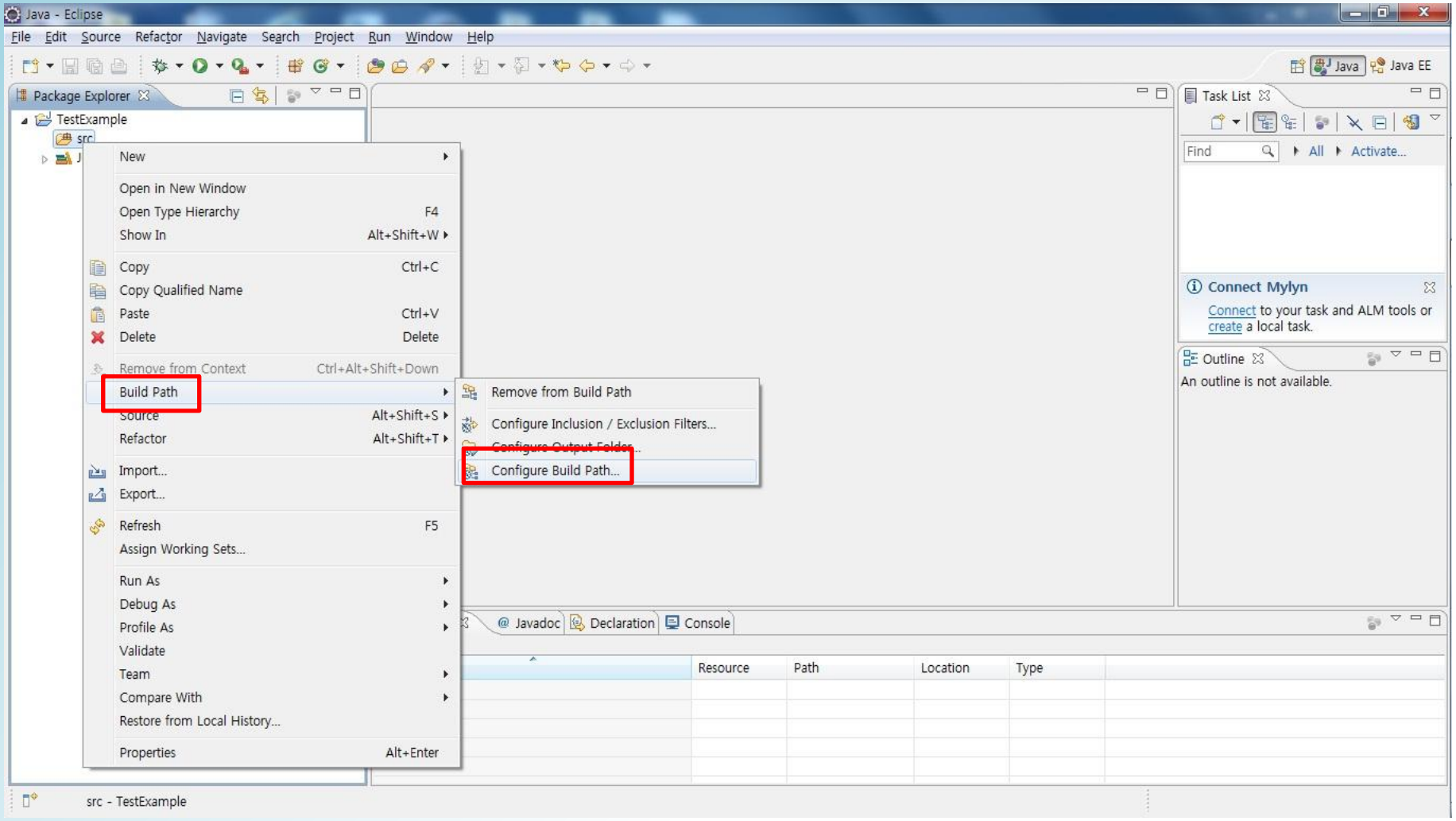
- Eclipse IDE에서 SVN을 사용하기 위한 Eclipse Plug-in

TPTP

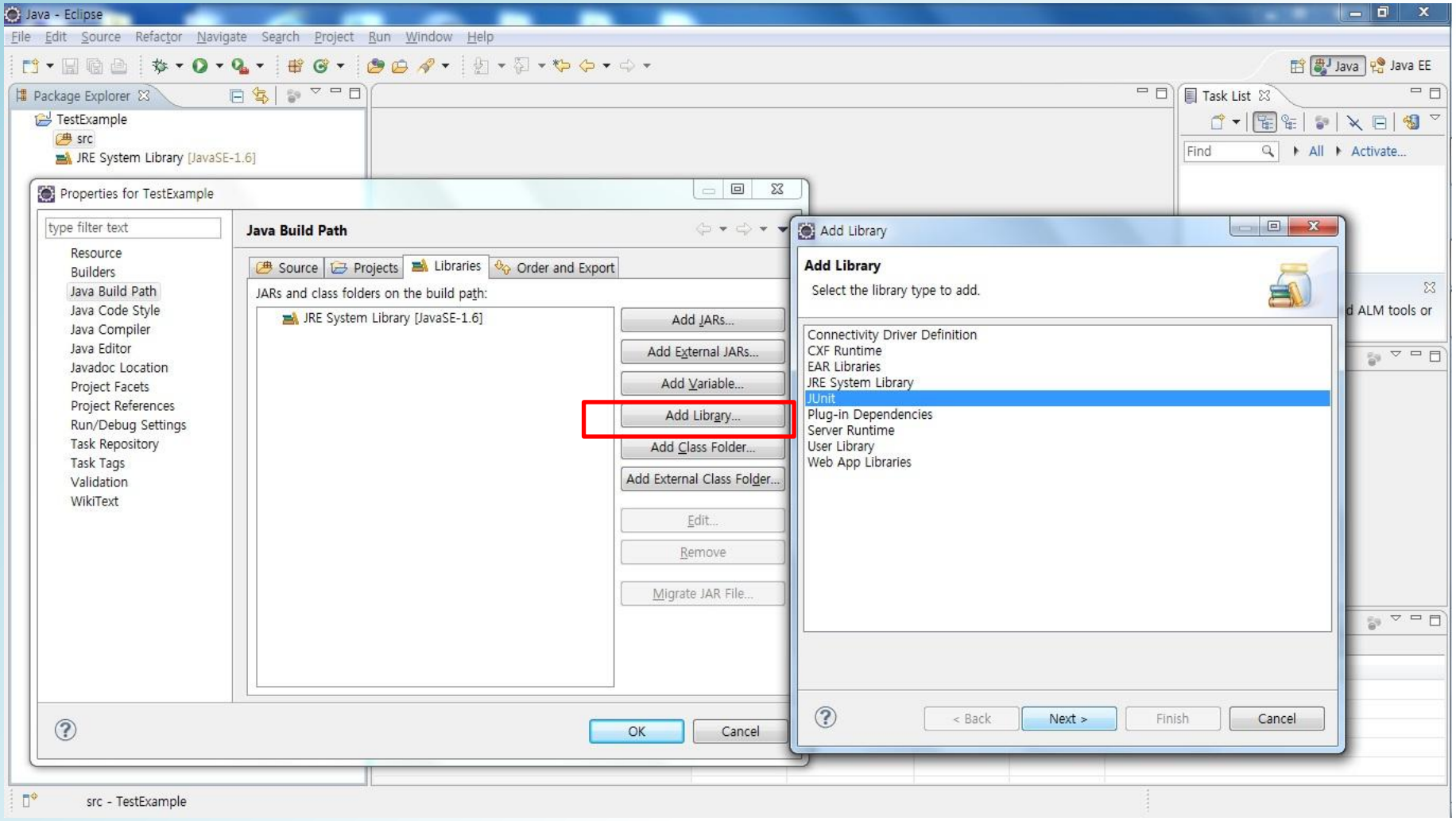
- Test & Performance Tools Platform
  - Eclipse에서 진행중인 프로젝트 및 성능 관련 툴을 위한 플랫폼
- 모니터링, 테스팅 자동화, 프로파일 등의 기능
- Garbage Collector로 제거되지 않는 객체를 수정 가능
- 높은 메모리 점유율 및 지나친 수행시간을 가지는 소스를 찾아 수정 가능

JUnit Practice

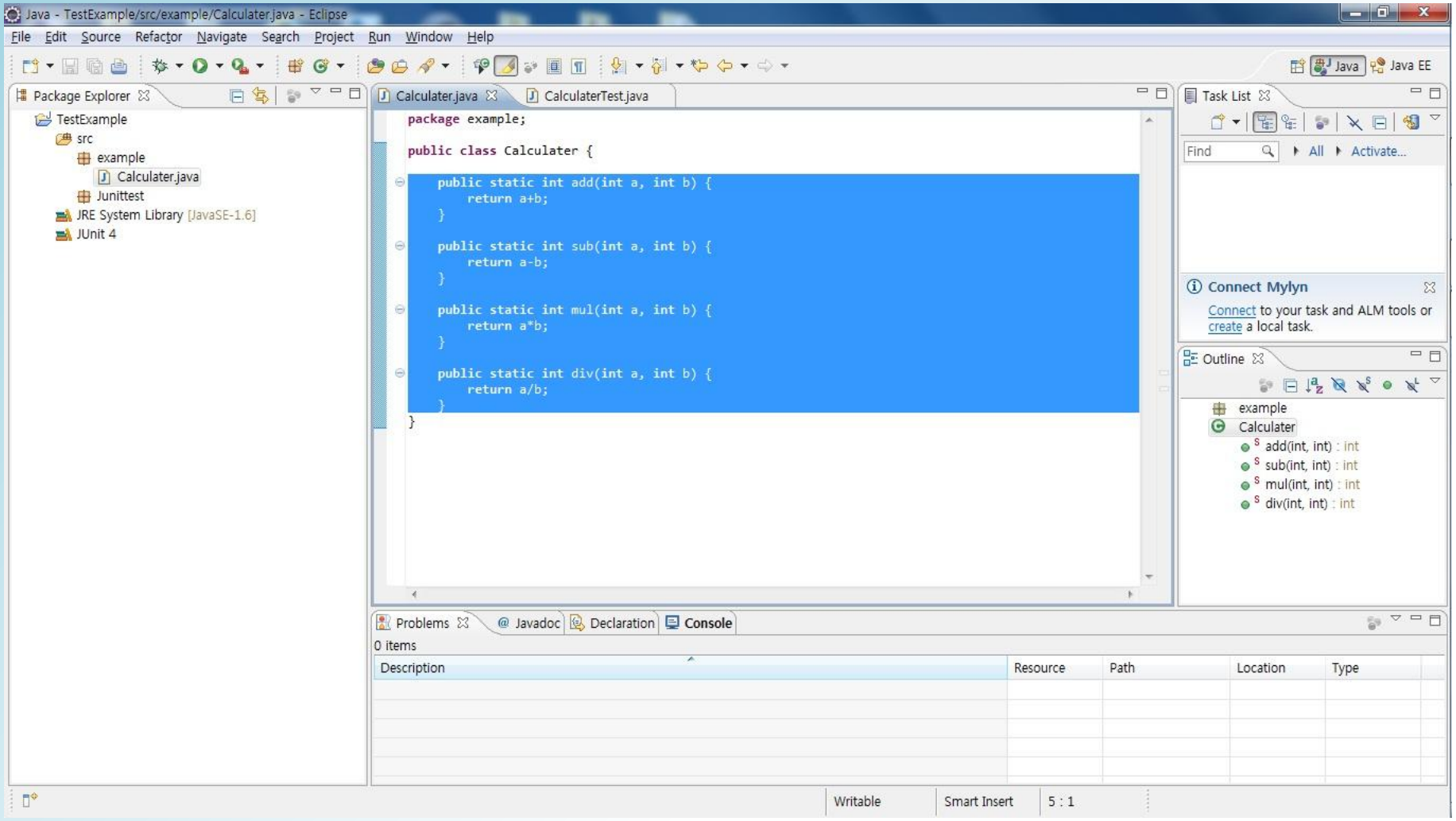
JUnit Set-up


Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Java EE
Package Explorer
TestExample
src
New
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Compare With
Restore from Local History...
Properties Alt+Enter
Remove from Build Path
Configure Inclusion / Exclusion Filters...
Configure Output Folder...
Configure Build Path...
Task List
Find
All
Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
An outline is not available.
Javadoc
Declaration
Console
Resource
Path
Location
Type
src - TestExample

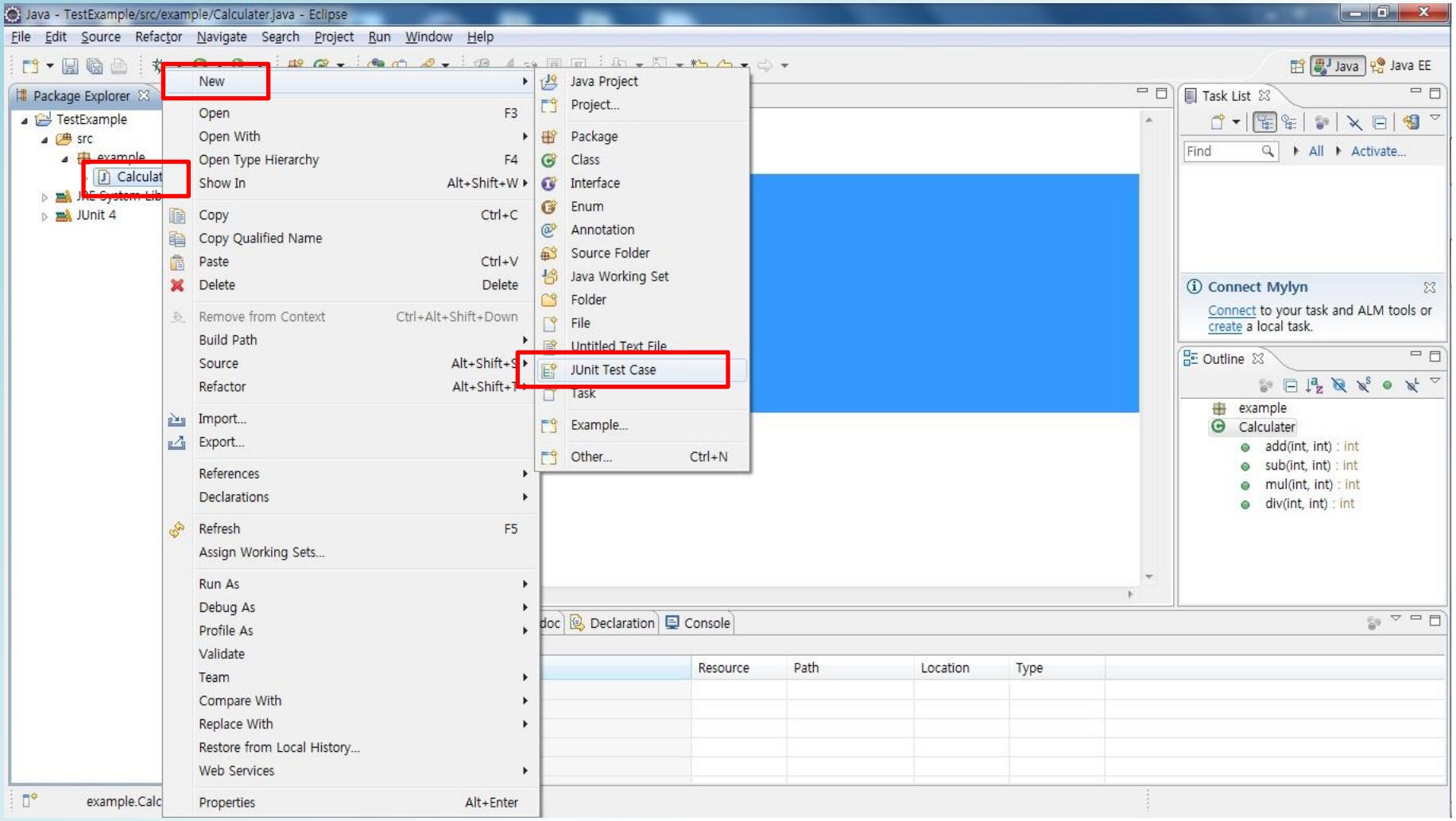
JUnit Set-up(cont.)



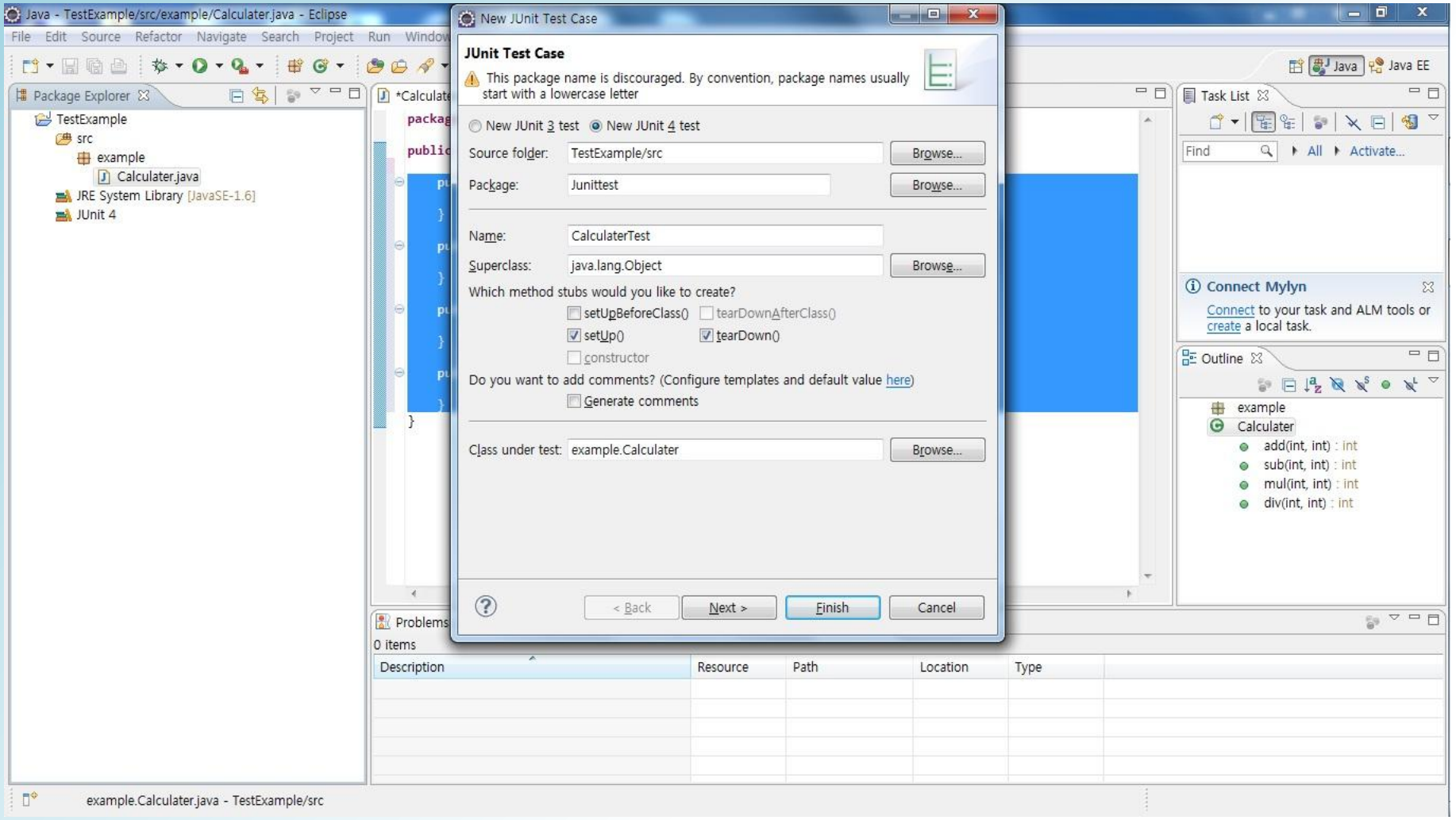
Coding(class 작성)



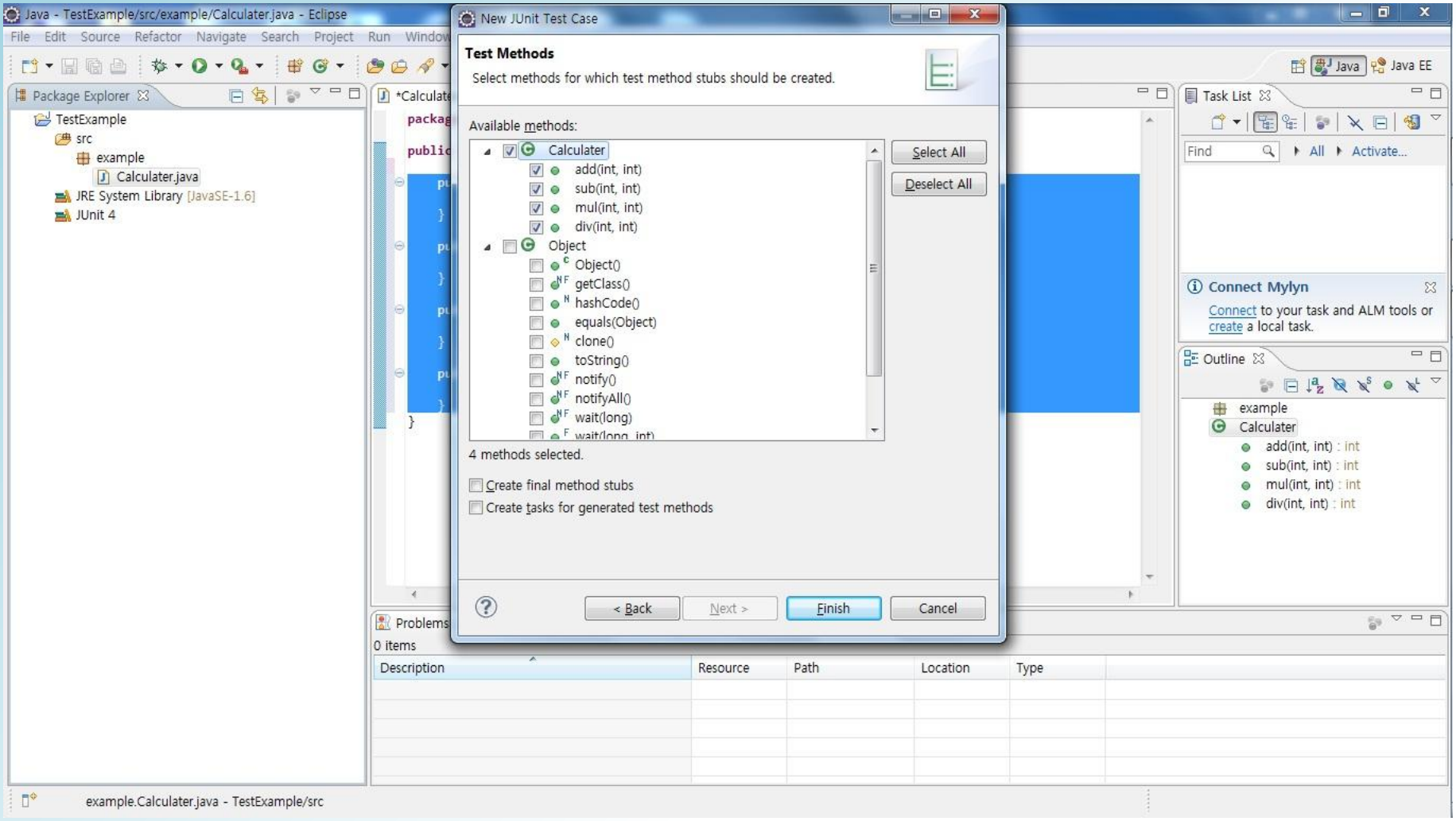
JUnit Test Case 작성



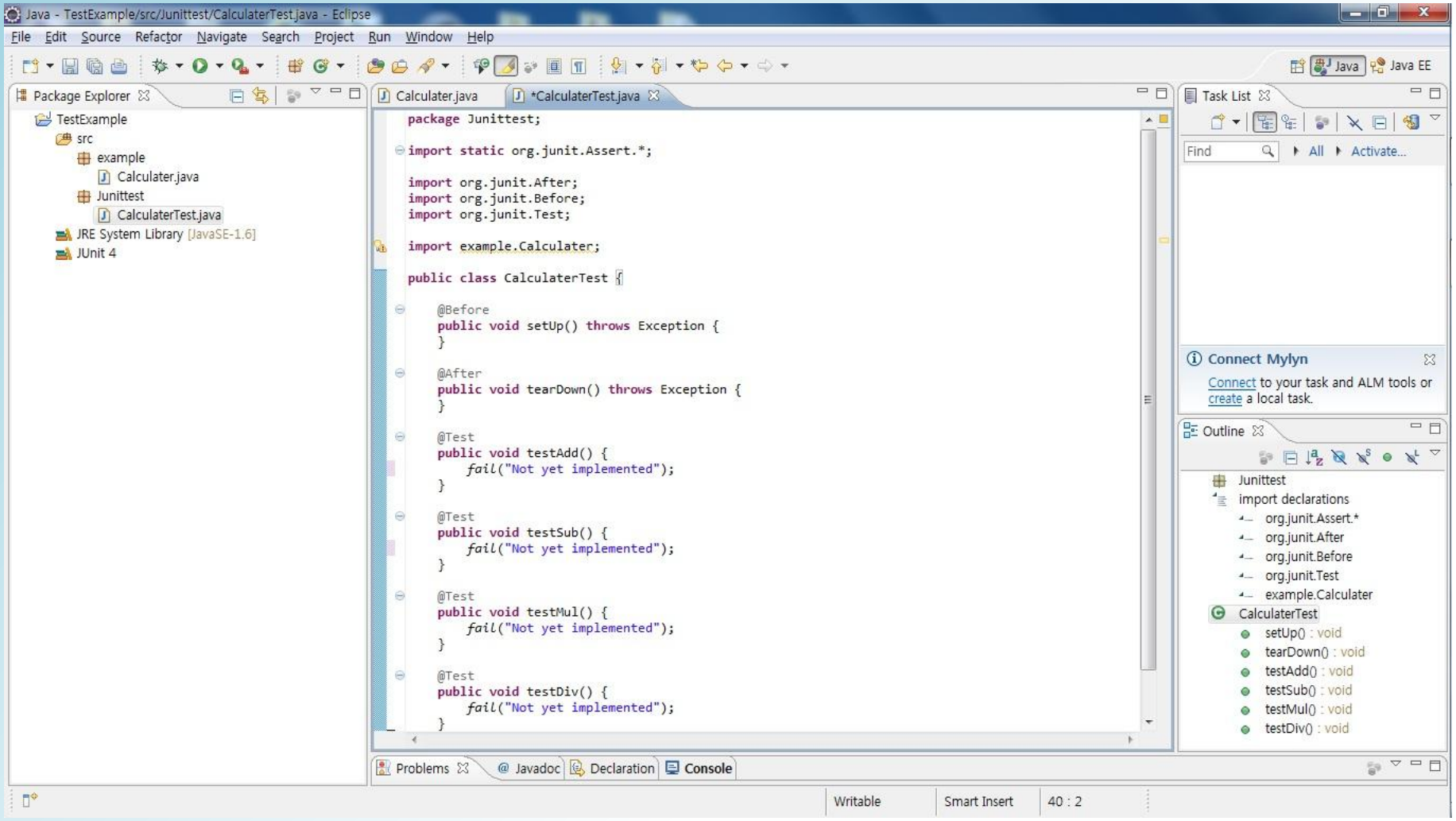
JUnit Test Case 작성(cont.)



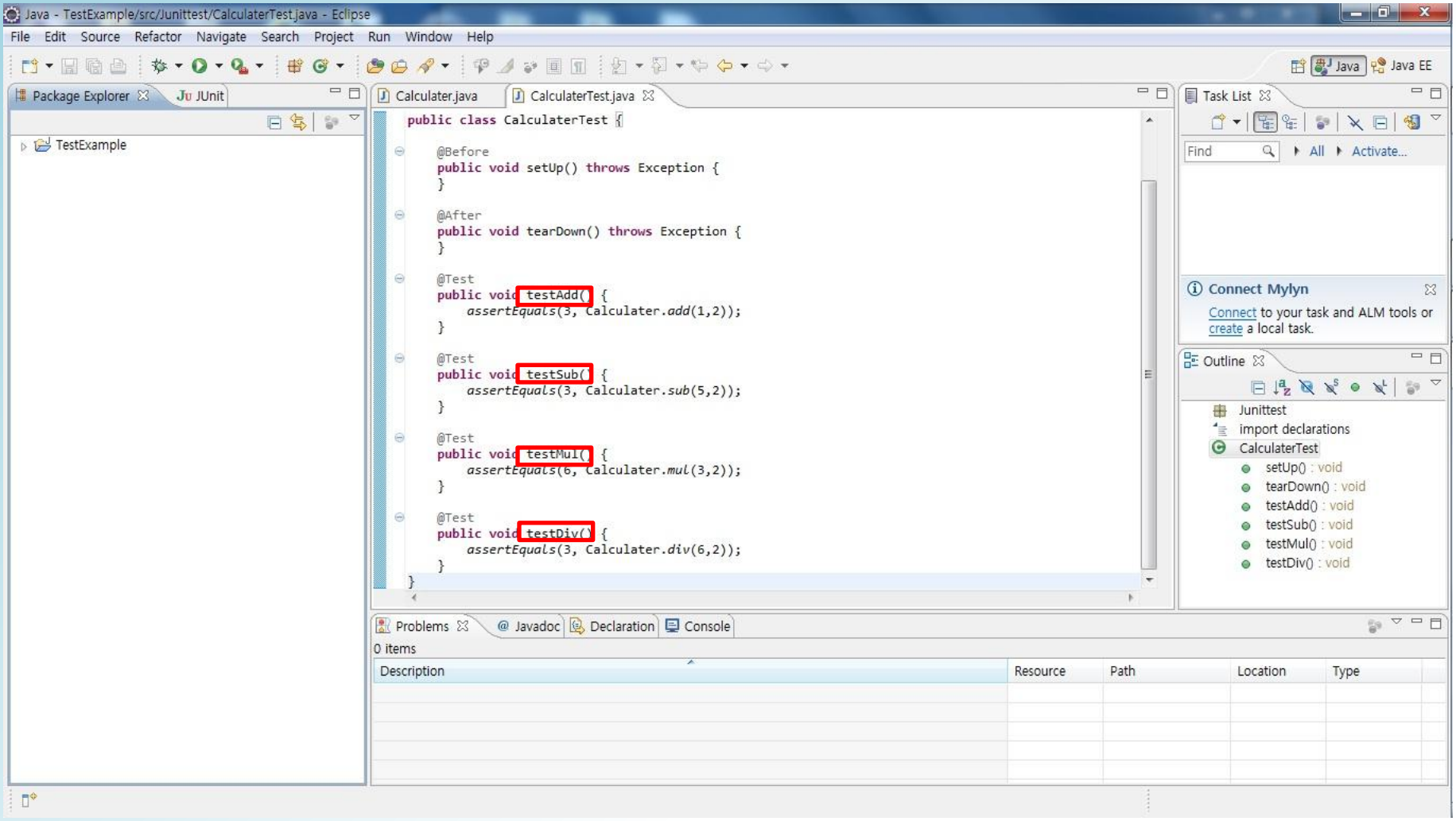
JUnit Test Case 작성(cont.)



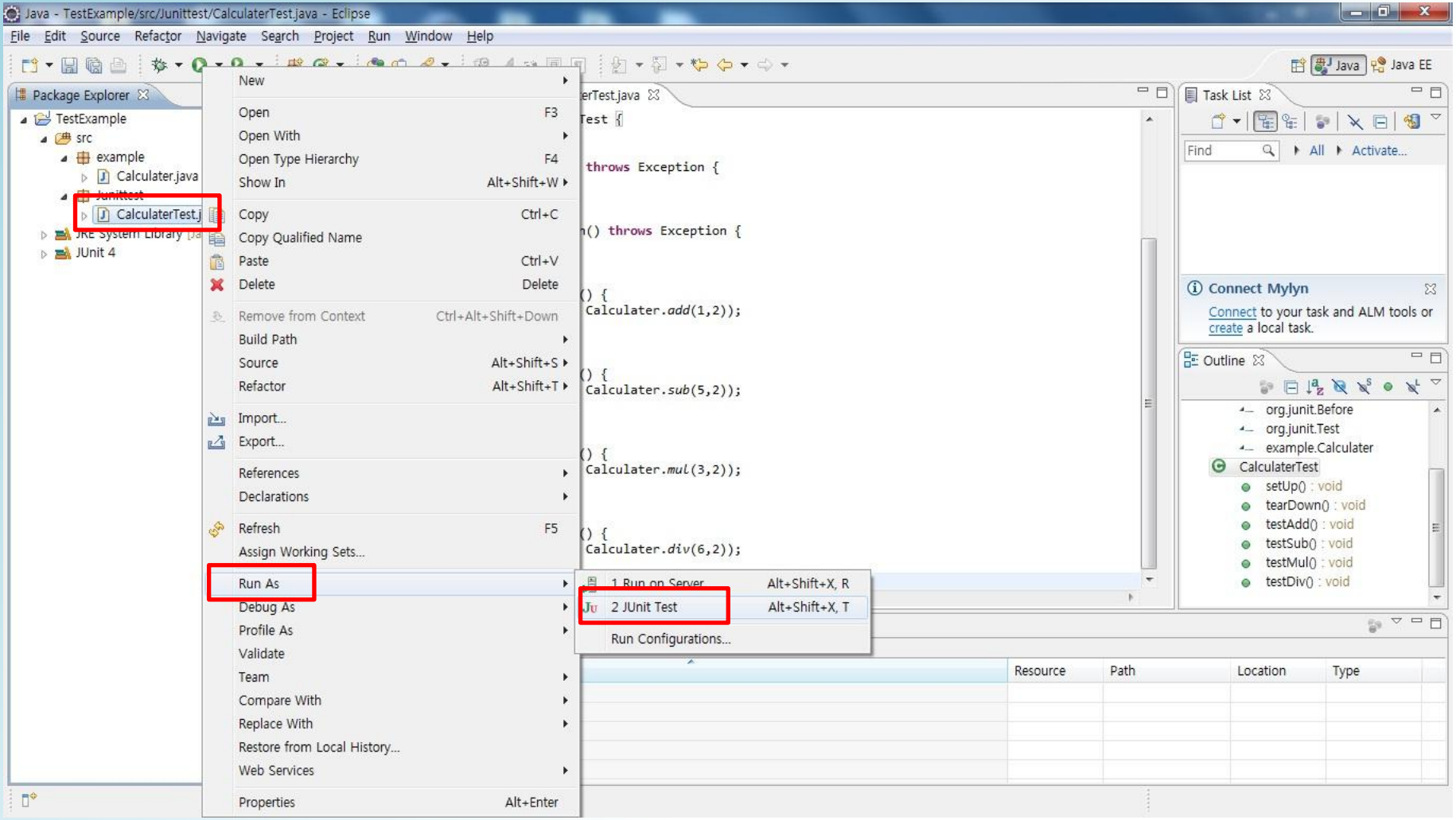
JUnit Test Case 작성(cont.)



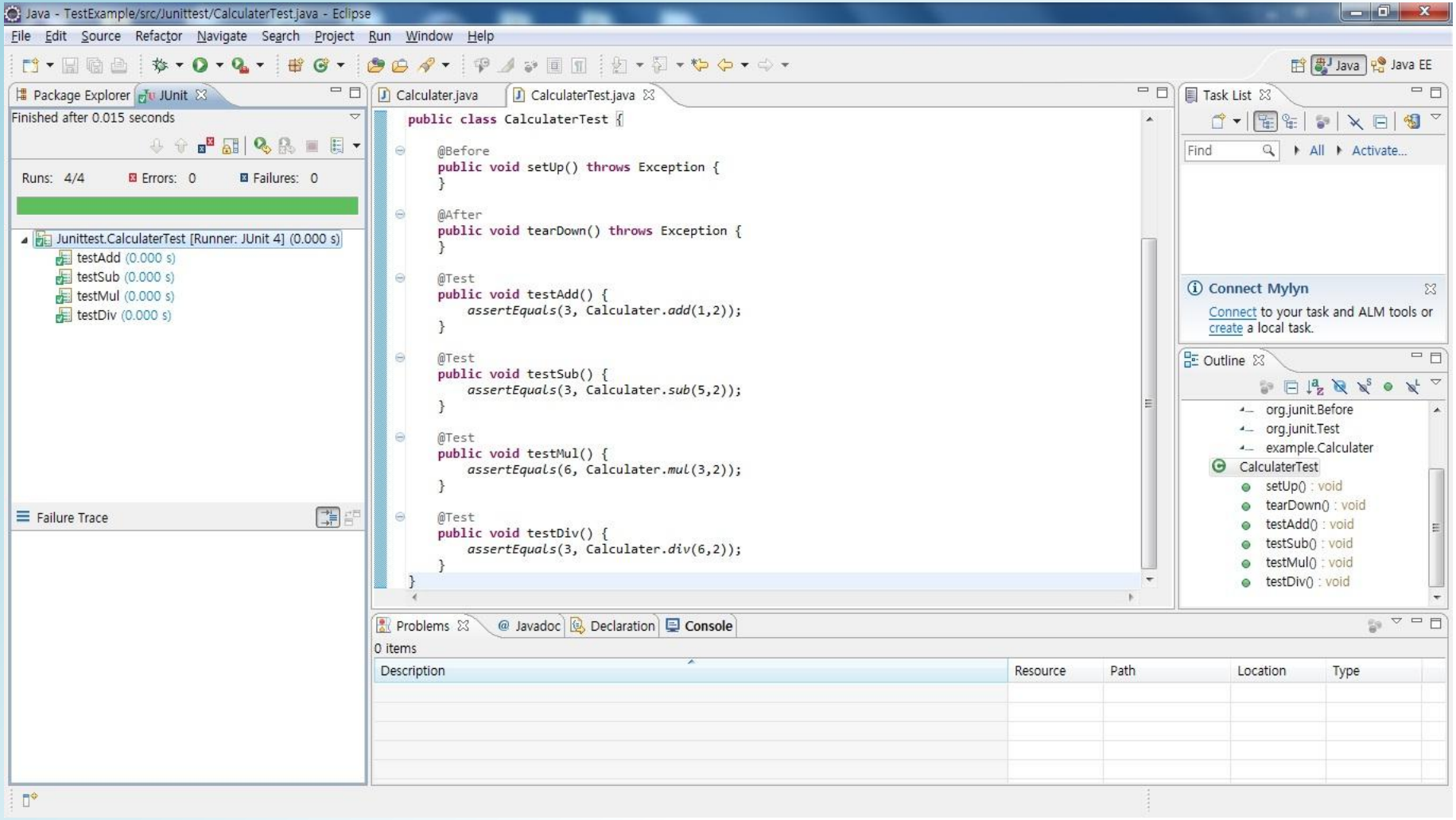
JUnit Test Case 작성(cont.)



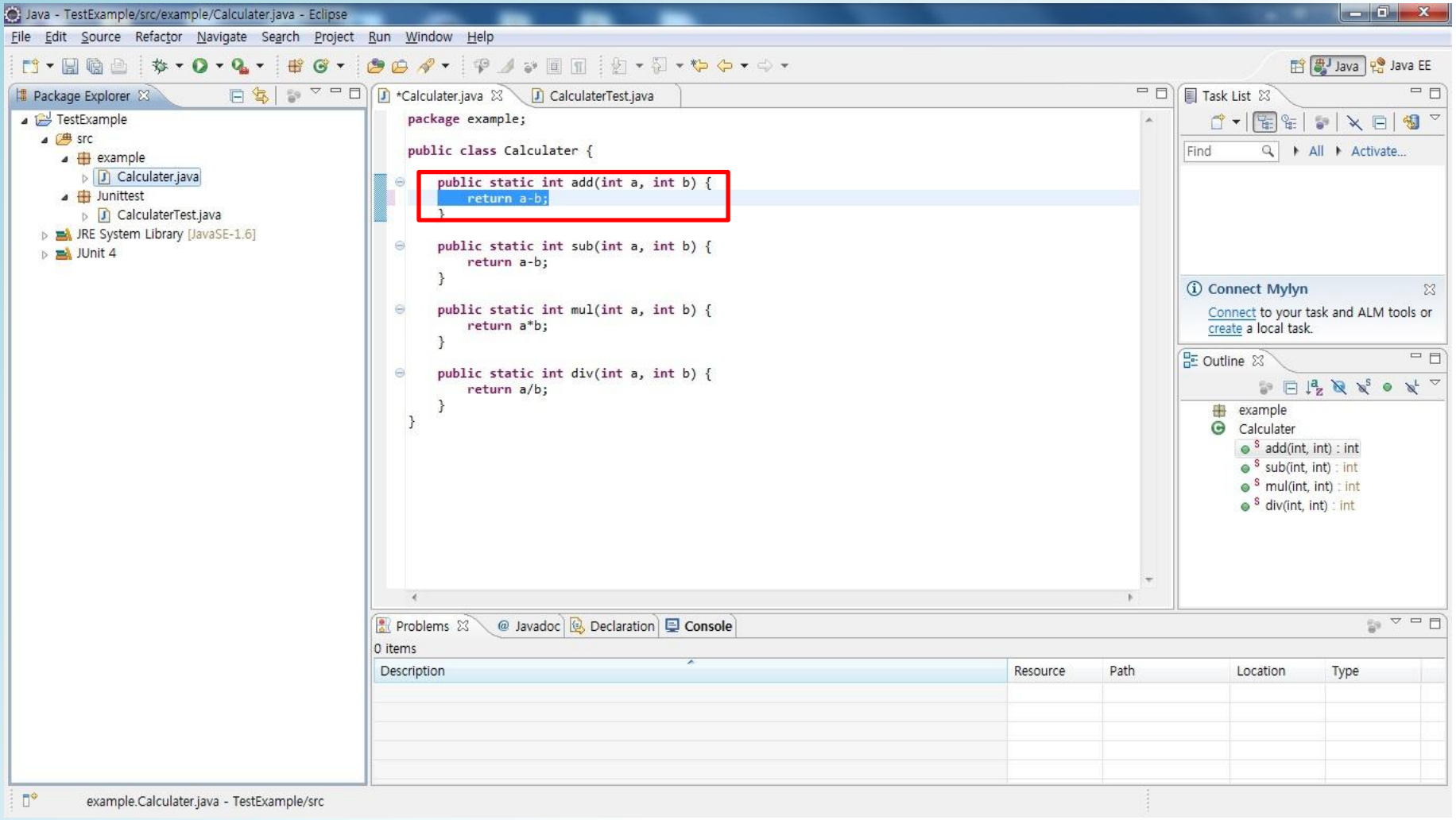
Testing


Java - TestExample/src/Junittest/CalculaterTest.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
TestExample
src
example
Calculater.java
Junittest
CalculaterTest.j
JRE System Library
JUnit 4
New
Open F3
Open With
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
References
Declarations
Refresh F5
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Compare With
Replace With
Restore from Local History...
Web Services
Properties Alt+Enter
1 Run on Server Alt+Shift+X, R
2 JUnit Test Alt+Shift+X, T
Run Configurations...
throws Exception {
() throws Exception {
() {
Calculater.add(1,2));
() {
Calculater.sub(5,2));
() {
Calculater.mul(3,2));
() {
Calculater.div(6,2));
Task List
Find All Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
org.junit.Before
org.junit.Test
example.Calculater
CalculaterTest
setUp() : void
tearDown() : void
testAdd() : void
testSub() : void
testMul() : void
testDiv() : void
Resource Path Location Type

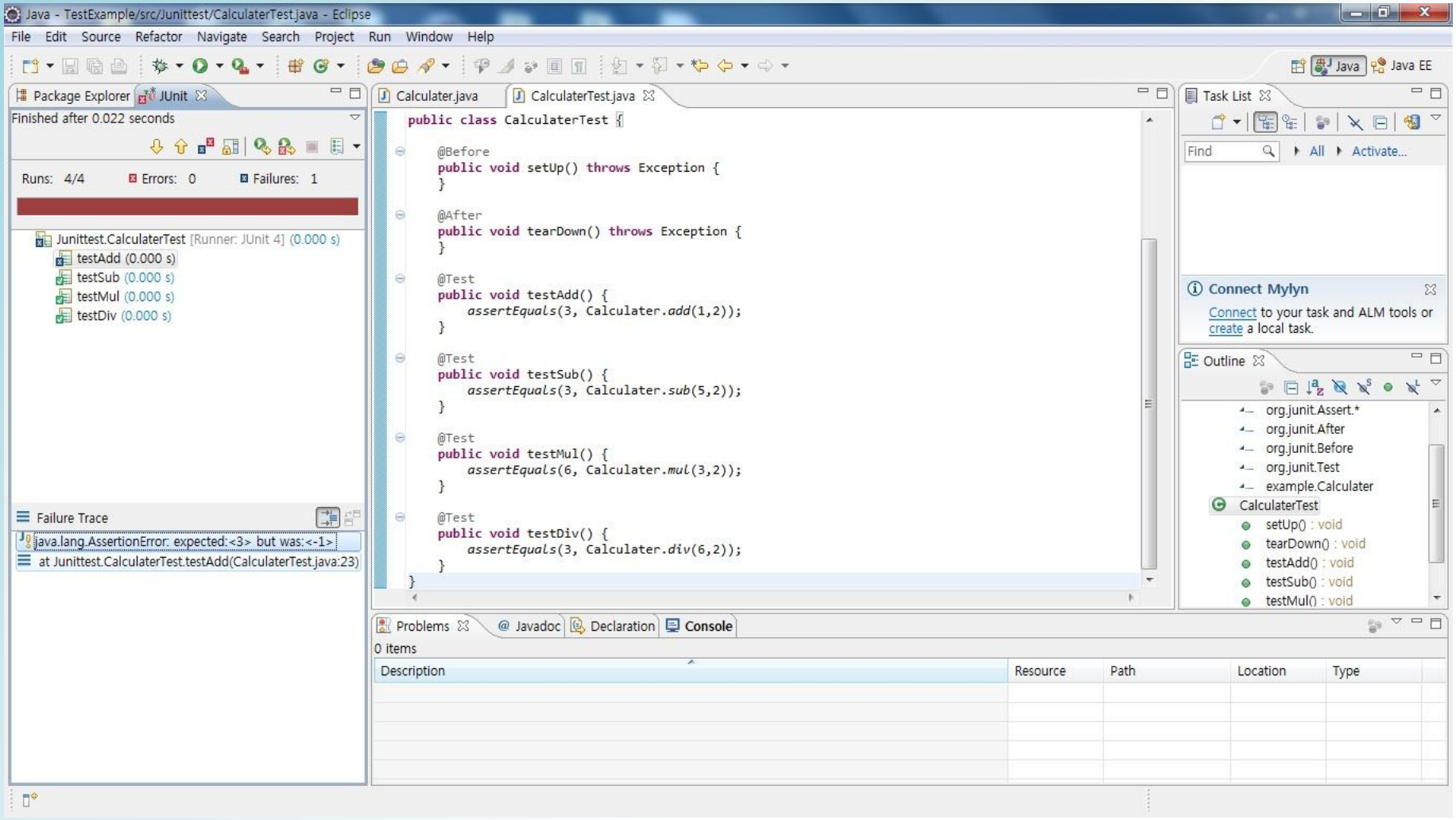
Testing(cont.)



Testing(cont.)



Testing(cont.)



References

<자바 개발자도 쉽고 즐겁게 배우는 테스팅 이야기> 이상민

http://blog.naver.com/1jongrak?Redirect=Log&logNo=50132272876

http://searchstory.tistory.com/268

http://xprogramming.com/book/whatisxp/

http://blog.naver.com/softgear?Redirect=Log&logNo=100002991166

http://en.wikipedia.org/wiki/

http://www.extremeprogramming.org/

http://www.junit.org/

http://blog.naver.com/nachaos?Redirect=Log&logNo=110030350613

http://www.ibm.com/developerworks/kr/library/tutorial/j-junit4/

http://www.eclipse.org