

SVN, MANTIS, JUNIT

사용
방법

TEAM 1.

200911391 박준모

200911388 박미관

200911412 이영준

contents



1. SVN

2. Mantis

3. JUnit

SVN

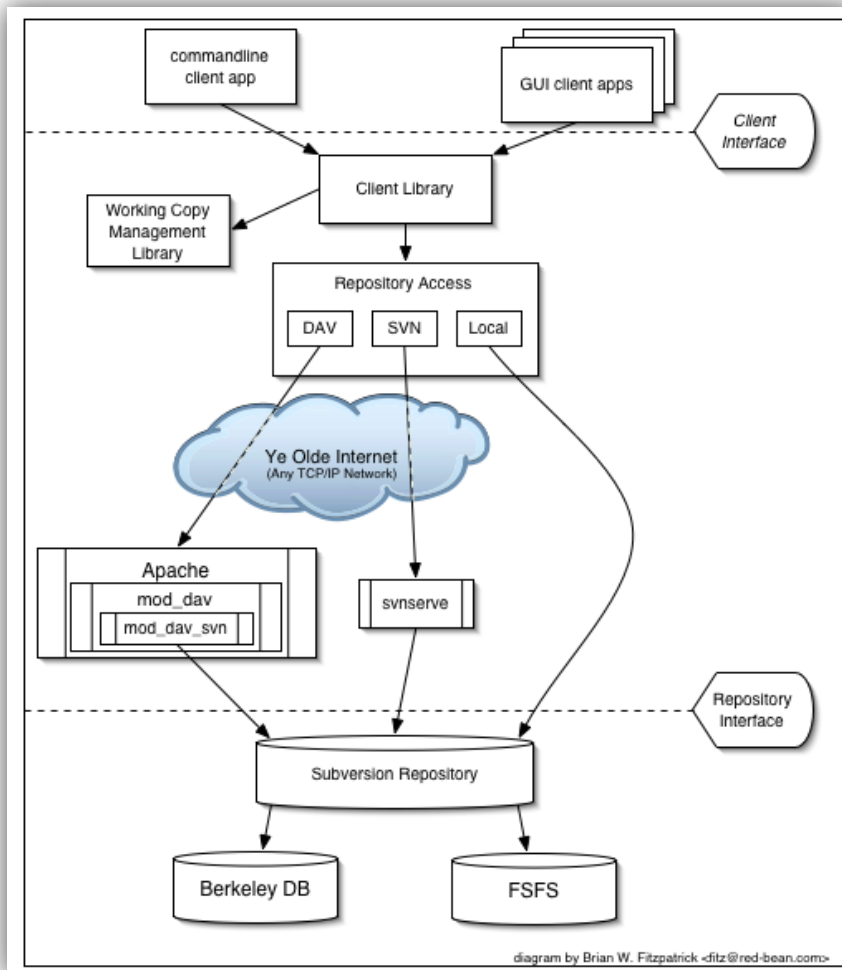
SUBVERSION



SVN_What is SVN?

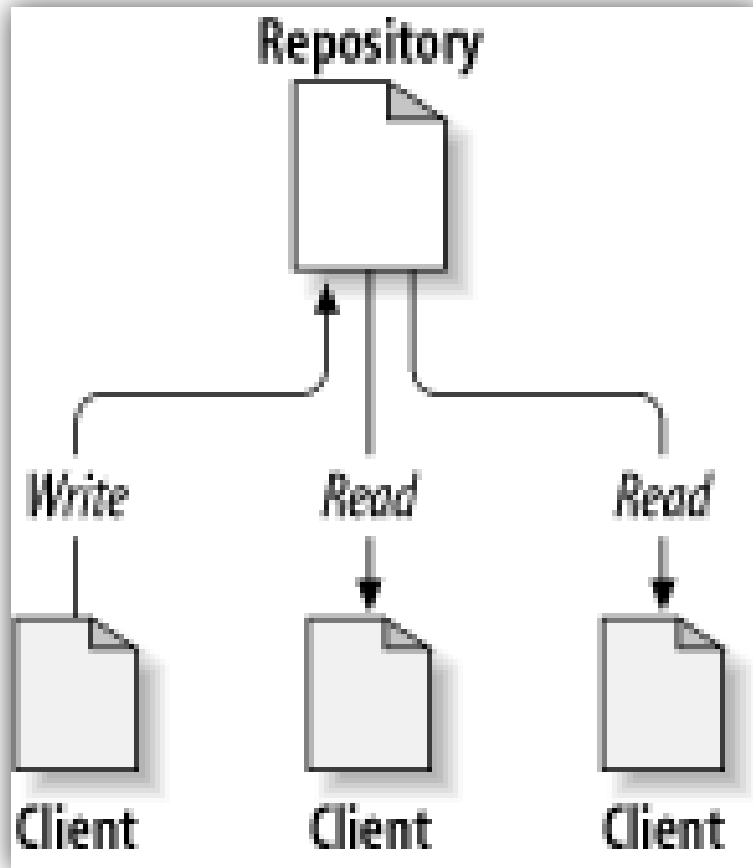
- 소스를 버전 별로 관리 할 수 있는 프로그램
- 충돌 없이 팀 단위의 동시 개발이 가능(Copy-Modify-Merge 패턴)
- 소스를 백업하거나 롤백(이전 버전으로 돌아가기) 기능
- 소스 수정, 추가, 삭제 등 기록이 로그로 남음

SVN_What is SVN?



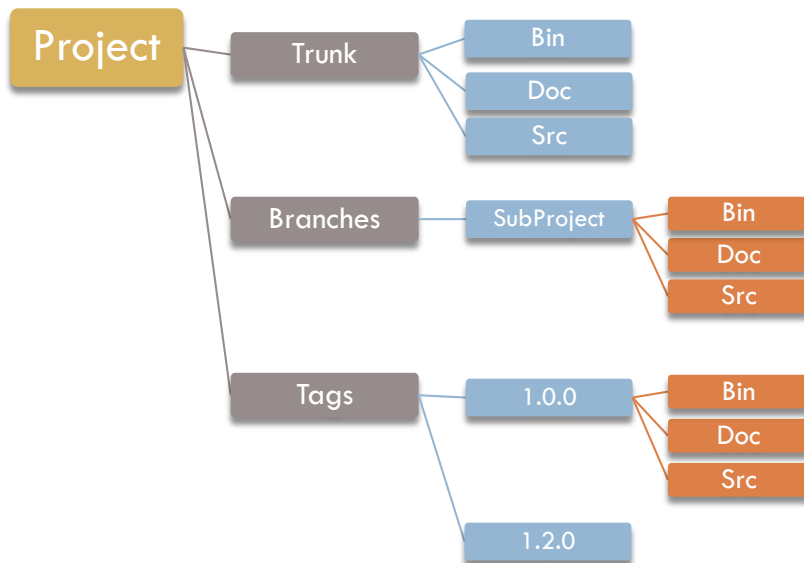
- 서버와 클라이언트로 구성
- 서버는 네트워크나 로컬로 접근 가능
- 서버 내부에 Repository가 존재

SVN_Repository



- 서버 내부에 존재
- 프로젝트 소스와 수정 내역 로그를 저장
- 각각의 클라이언트들은 하나의 저장소에 접근하여 소스를 읽거나 갱신
- 버전 관리 프로젝트마다 하나의 저장소가 필요

SVN_Folders in Repository



□ Trunk

- 프로젝트의 중심이 되는 폴더
- 실제 개발 작업은 이곳에서 진행

□ Branches

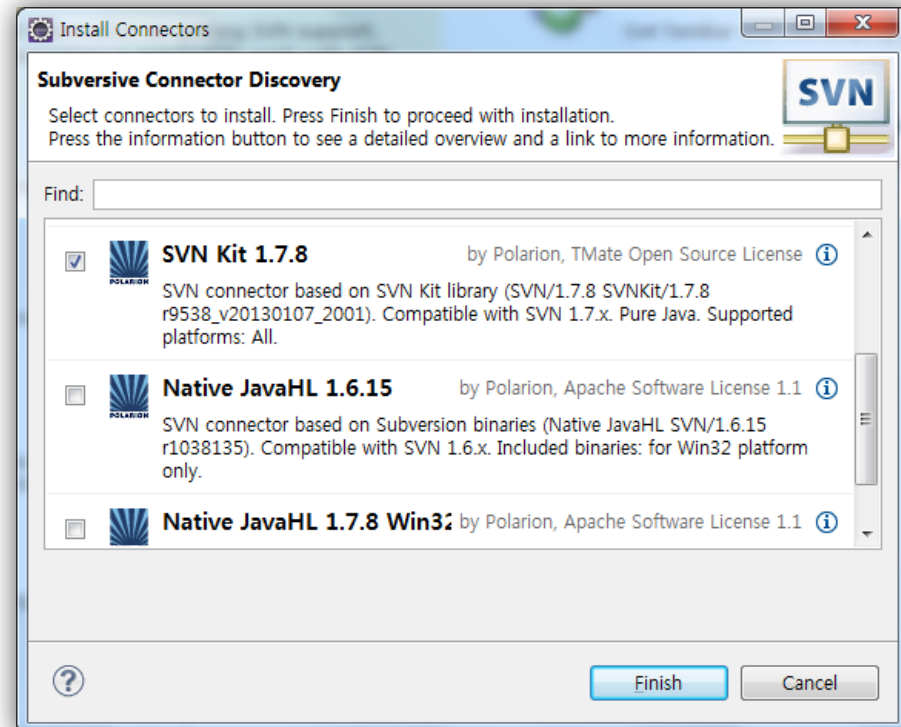
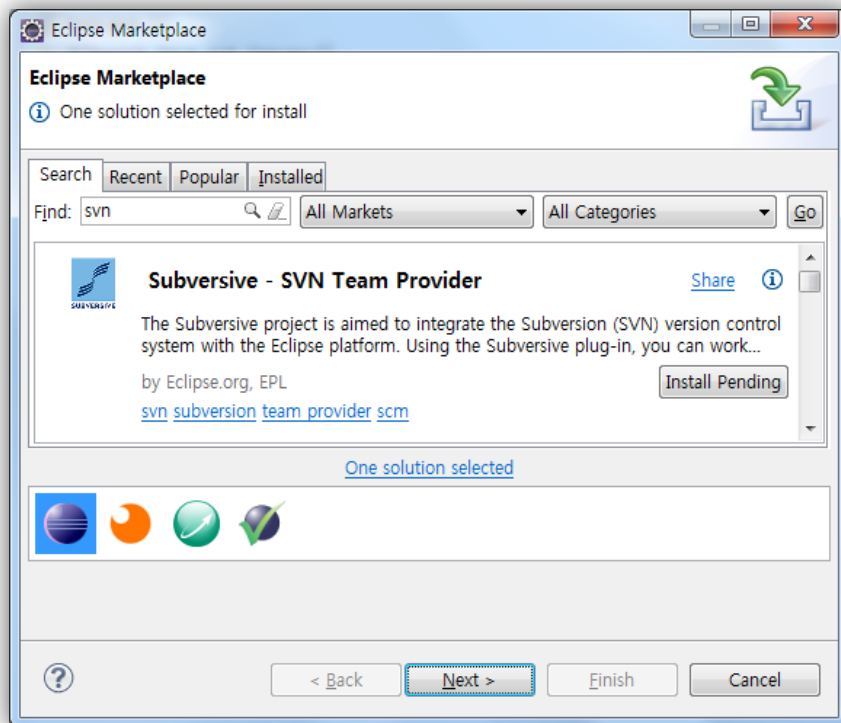
- Trunk에서 뻗어나온 또 하나의 프로젝트로서 메인 프로젝트에서 분기하여 따로 개발을 진행

□ Tags

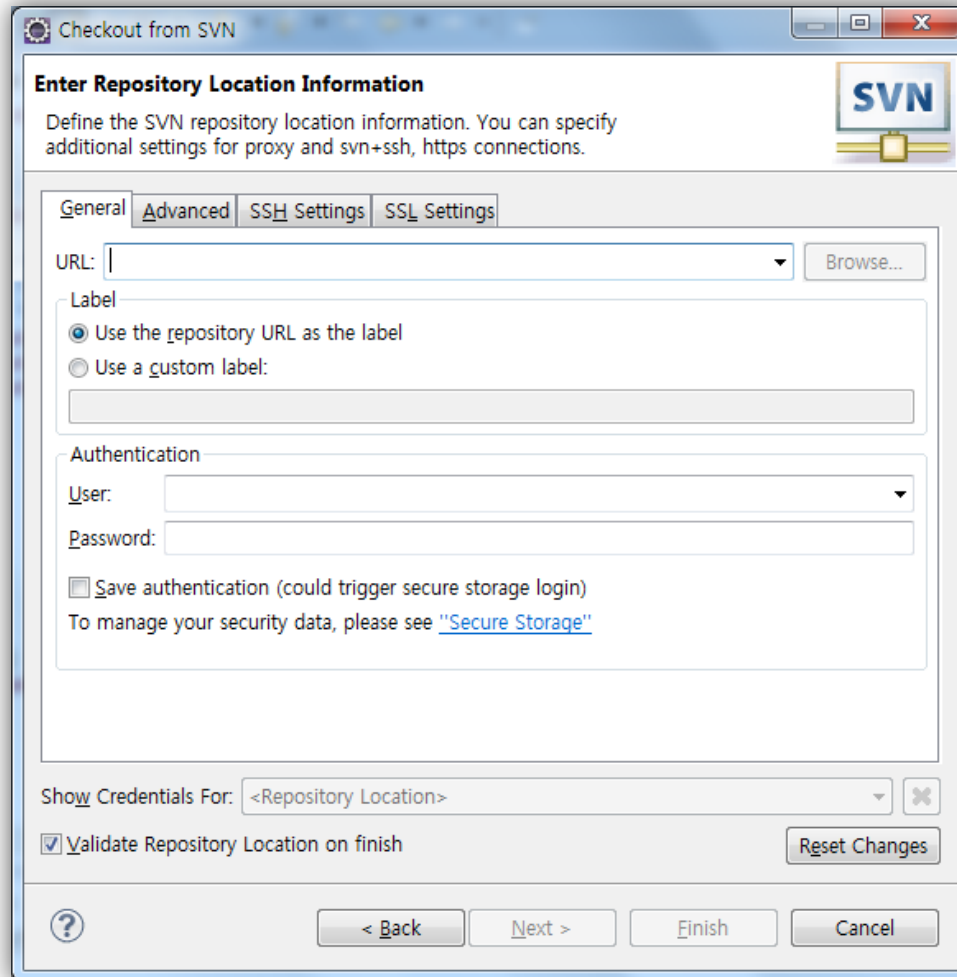
- 버전 업이나 특정 시점에서 프로젝트 전체를 복사하여 보관하는 곳

SVN_Installing SVN in Eclipse

- ❑ Eclipse Marketplace에서 Subversive – SVN Team Provider 설치한다.
- ❑ Eclipse 재시작 후 SVN kits 1.7.8 설치한다.



SVN_Checkout



- 소스 가져오기 기능
- 저장소의 소스 폴더의 복사본을 받고, 이에 대한 버전 관리를 시작함 (-> 저장소와 동기화)
- File -> Import -> SVN -> Project from SVN

SVN_Update & Commit

□ Update

- 저장소의 소스폴더와 비교하여 변경된 부분을 검사하고 최신 버전으로 동기화를 맞춤

□ Commit

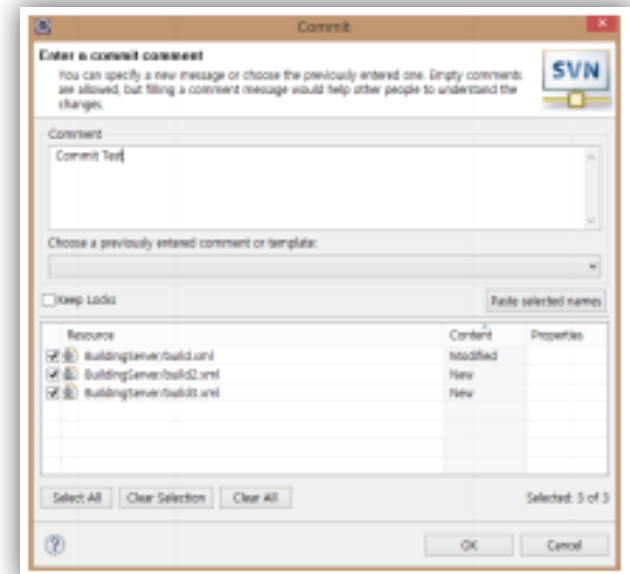
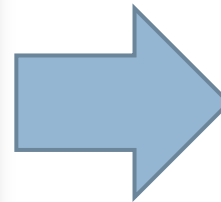
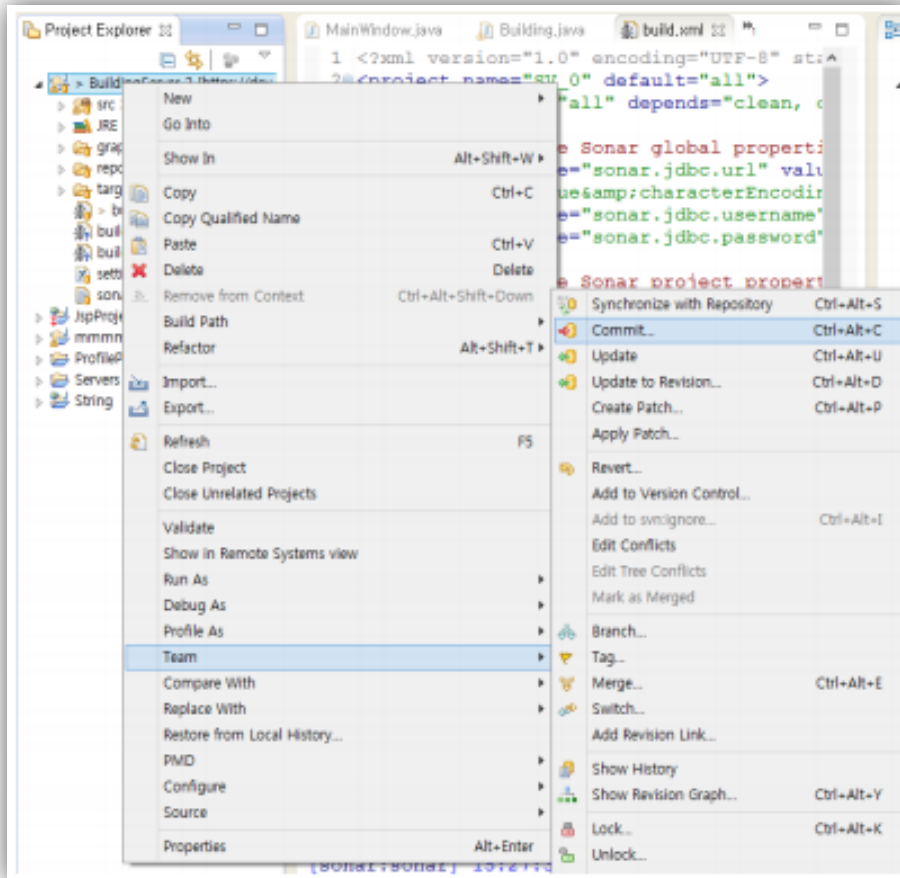
- 소스를 변경한 후 저장소에 반영
- Commit할 때마다 버전이 증가하여 변경사항이 기록이 남음

SVN_Update & Commit

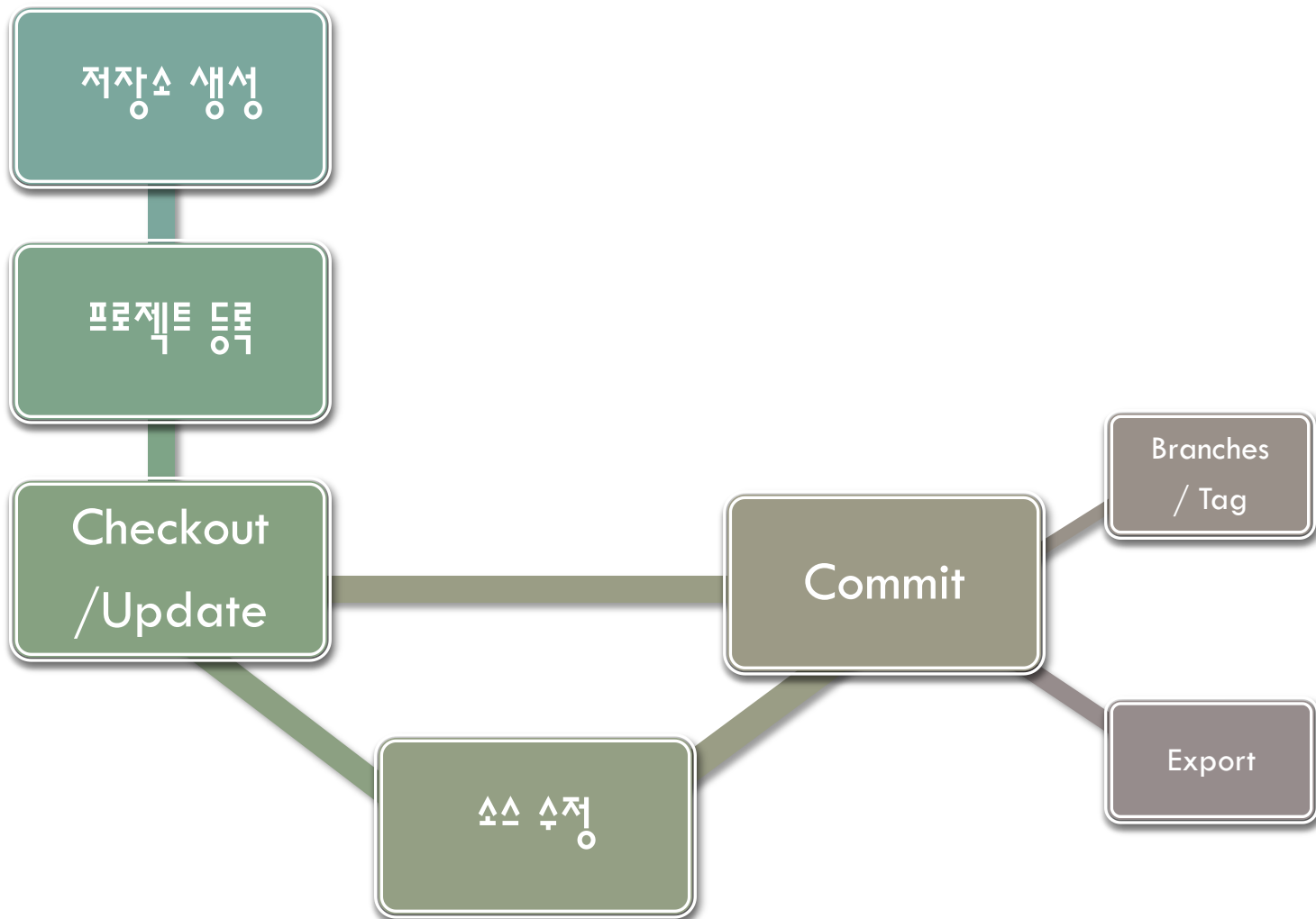
- 먼저 작업을 끝낸 개발자가 Commit하게 되면, 이후에 Commit하는 작업자들은 Commit 실패
- Commit 실패한 개발자들이 Update하게 되면 최신버전의 소스코드와 자신의 소스코드를 Merge
- Update하게 되면 그 소스파일의 버전이 바뀜

SVN_Update & Commit

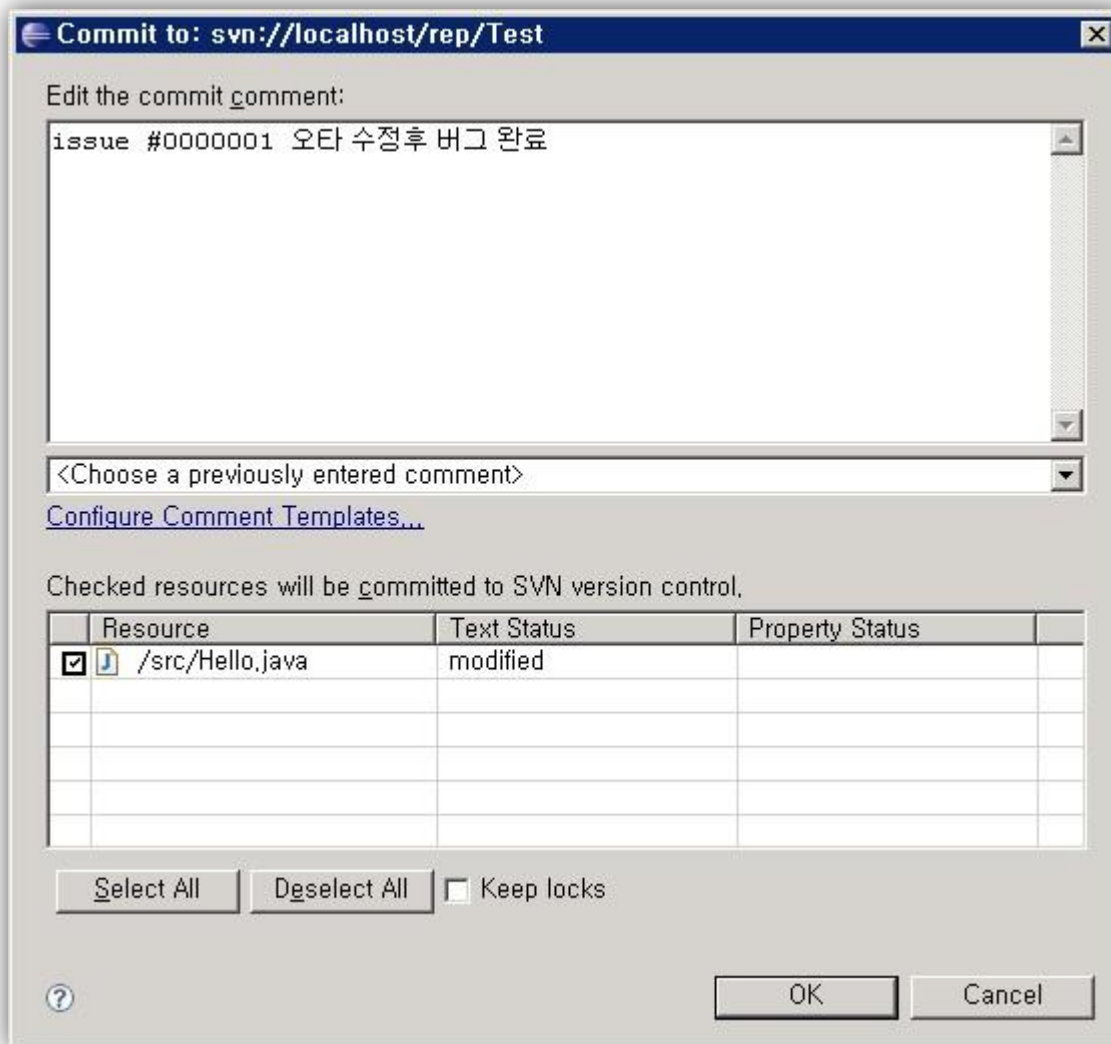
□ Project 우클릭 -> Team -> Commit / Update



SVN_Development Flow



SVN_with Mantis



- 정공식 표현에 맞추어 작성
- issue : 이슈
- #0000001 : 이슈번호
- 내용

SVN_with Mantis

0000001: 실행은 아무나 하나. - Mantis - Microsoft Internet Explorer

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

주소(1) http://localhost/mantis/view.php?id=1

0000001 [드디어실행] 보통 항상발생 2008-04-22 11:04 2008-04-22 14:35

보고자 administrator 상태 보기 공개

이슈 할당

우선 순위 낮음 해결 상태 수정됨

상태 @1@

요약 0000001: 실행은 아무나 하나.

설 명 초보 개발자의 좌측우측 mantis 실행기

추가 정보 사항

Tags No tags attached.

Attach Tags (Separate by ',') Existing tags Attach

첨부 파일

이슈 갱신 할당: [나] 상태 변경: 새로운 이슈 이슈 감시하기 내용 복사후 관련이슈추가 이슈 이동 이슈 삭제

이슈 관계

새로운 관계 현재 이슈 다음 이슈의 원인 추가

파일 업로드

파일 선택 (최대 크기: 5,000k) 찾아보기... 파일 업로드

이 이슈를 감시하는 사용자가 없습니다.

이슈 노트

(0000001)
administrator (관리자)
2008-04-22 14:35

수정 삭제 보호하기

***** Source code change *****
issue 0000001 오타 수정후 버그 완료

SVN Revision:3
수정됨: Test/src/Hello.java

--- Test/src/Hello.java 2008-04-22 05:03:00 UTC (rev 2)
+++ Test/src/Hello.java 2008-04-22 05:35:00 UTC (rev 3)
@@ -1,6 +1,6 @@

public class Hello {
 public static void main(String[] args){
- System.out.println("Hello");
+ System.out.println("수정된 Hello");
 }
}

완료 로컬 인트라넷

MANTIS



Mantis_What is Mantis?

- Bug Tracking System

- 개발일정 관리가 가능한 tool

- 설치 사양

- MySQL 4.1.1 이상

- PHP 4.3.0 이상

- 아파치, IIS등의 웹서버

Mantis_Advantage

- 오픈 소스
- 한글화
- E-mail과 연동하여 변동사항을 즉시 알림
- 버그보고 기본정보 필드가 잘 정의 되어 있고 추가 필드의 커스터마이징이 편함
- 버그에 대한 히스토리가 모두 저장되어 진행사항 파악 편함
- 내보내기 기능으로 보고서 작성이 용이

Mantis_Project

□ 프로젝트 추가

- '관리/프로젝트 관리' 메뉴로 들어가 새로운 프로젝트를 생성합니다.
- 등록 후에 아래와 같은 상세한 설정을 할 수 있습니다.

프로젝트 추가	
* 프로젝트 이름	어린이 학습용 그림판
상태	개발버전 v
전역 분류 상속	☒
상태 보기	공개 v
설명	어린이 학습을 위한 게임 기능이 있는 그림판 up down
프로젝트 추가	

Mantis_Project

□ 서버 프로젝트

➤ 연관된 서버프로젝트를 생성하고 관리할 수 있습니다.

서버프로젝트 새로운 서버프로젝트 생성

이름	상태	사용가능	분류 상속	상태 보기	설명	실행
게임 하기	개발버전	X	☒	공개	색칠 공부 게임을 한다	[편집] [링크해제]

하위 프로젝트 상속 관계 바꾸기

▼ 서버프로젝트로 추가

□ 분류

➤ 이슈 등록 시 지정할 분류를 관리합니다.

분류

분류	할당하기	실행
[모든 프로젝트] General		

분류 추가

▼ 카테고리 복사(From) 카테고리 복사(To)

Mantis_Project

□ 버전

➤ 버전을 등록하고 관리합니다.

버전				
버전/빌드번호	릴리스함	더 이상 사용되지 않음	시간 기록	실행
0.1			2013-05-01 06:35 JST	편집 삭제
		버전 추가 버전 추가 및 수정		
버전 복사해 올 곳		버전 복사해 넣을 곳		

□ 사용자

➤ 프로젝트에 관련된 사용자를 추가하고 권한을 설정할 수 있습니다.

프로젝트에 사용자 추가		
사용자 이름	접근 레벨	
lee	보고가능	사용자 추가
사용자 복사(From) 사용자 복사(To)		

Mantis_Issue

□ 이슈 상태

상 태	설 명
신규	새로 추가된 이슈
정보 부족	정보가 더 필요하며 보고한 사람은 추가 정보를 제공해야 함
이슈 검토	이슈에 대해 조사했으나 확인이나 할당되지 않은 상태이며 해당 개발자는 이슈로 인정하지 않을 수 있음을 유저에게 알림
확인된 이슈	개발자가 이슈 내용을 확인하고 재현할 수 있음. 보통 정보를 갱신하는 사람 (updater)이나 다른 개발자가 설정
할당된 이슈	현재 개발자에게 할당되어 처리 중임
해결된 이슈	이슈를 수정한 것으로 판단하며 이에 대한 확인을 기다리고 있음
다음	이슈가 완전 해결되어 폐쇄함

신규

정보 부족

이슈 검토

확인된 이슈

할당된 이슈

해결된 이슈

다음

Mantis_Issue

중요도(심각성)	보통
우선 순위	새 기능 요구
프로필 편집	사소한
☐ 또는 다음 내용을 채웁니다.	오타
제품 번호	기능 개선
할당하기	보통
	중요함
	충돌
	장애

□ 심각도

종류	설명
새 기능 요구	새로운 기능을 요구
사소한	그래픽 정렬, 형식 등 단순 외형적 문제 등 별 거 아닌 사소한 것
오타	글, 문법, 단어 등의 오류
기능 개선 (tweak)	기능 향상 등의 개선 및 조정이 필요
보통 (minor bug)	기능상 덜 중요한 문제나 쉽게 해결할 수 있는 문제
중요함 (major bug)	기능상 중요한 문제
충돌	응용 프로그램 또는 OS 의 충돌
장애	개발 혹은 시험 작업을 더 이상 진행하기 어려움

Mantis_Issue

□ 예상 작업량

종류	설명
조정 (tweak)	곧 처리됨
작은 수정 (minor fix)	많은 시간이 걸리지 않음
새로 작성 (major rework)	일 양이 매우 많음
재설계 (redesign)	재설계가 필요함

Mantis_Issue

이슈 해결도 선택	수정됨
해결된 버전	개설
태도 추가	수정됨
	문제 제기
	재현할 수 없음
	수정할 수 없음
	중복
	이슈 아님
	보류(suspended)
	수정 계획 없음

이슈 해결 상태

상 태	설 명
개설	새로 만들어 짐
수정됨	시험 완료되었으며 이슈 상태는 해결되었다고 표시함
문제 제기	이전에 해결했지만 당시 처리 결과가 만족스럽지 않거나 정확하게 처리되지 않아 다시 발생한 상태
재현할 수 없음	이슈를 재현할 수 없으며 코드를 살펴봐도 파악할 수 없는 상태임. 추가적인 정보가 들어오면 다시 할당하도록 함
수정할 수 없음	이번 버전에서는 수정할 수 없는 문제임
중복	기존의 이슈와 중복된 것으로 중복된 이슈 번호가 필요함 ('#이슈번호'로 표시)
이슈 아님	이슈가 아님
보류(suspended)	중요하지 않은 문제로 수정을 보류함
수정 계획 없음	수정 할 계획이 없음

Mantis_Issue

- 이슈 보고
 - 화면 상단의 [이슈 보고 하기] 클릭

상세 보고 내역을 입력하세요

*분류	[모든 프로젝트] General
재발생 가능성	불규칙적
중요도(심각성)	중요함
우선 순위	긴급
프로젝트 편집	
또는 다음 내용을 채웁니다.	
제품 번호	0.1
담당자	administrator
목표 버전	0.1
*요약	요약
*설명	설명
재발생 단계	재발생 단계
추가 정보 사항	추가정보사항
파일 업로드 (총 크기: 5,000k)	찾아보기...
상태 보기	☒ 공개 ☐ 비공개
계속 보고	☐ 이슈 보고를 계속하려면 선택
 보고서 제출	

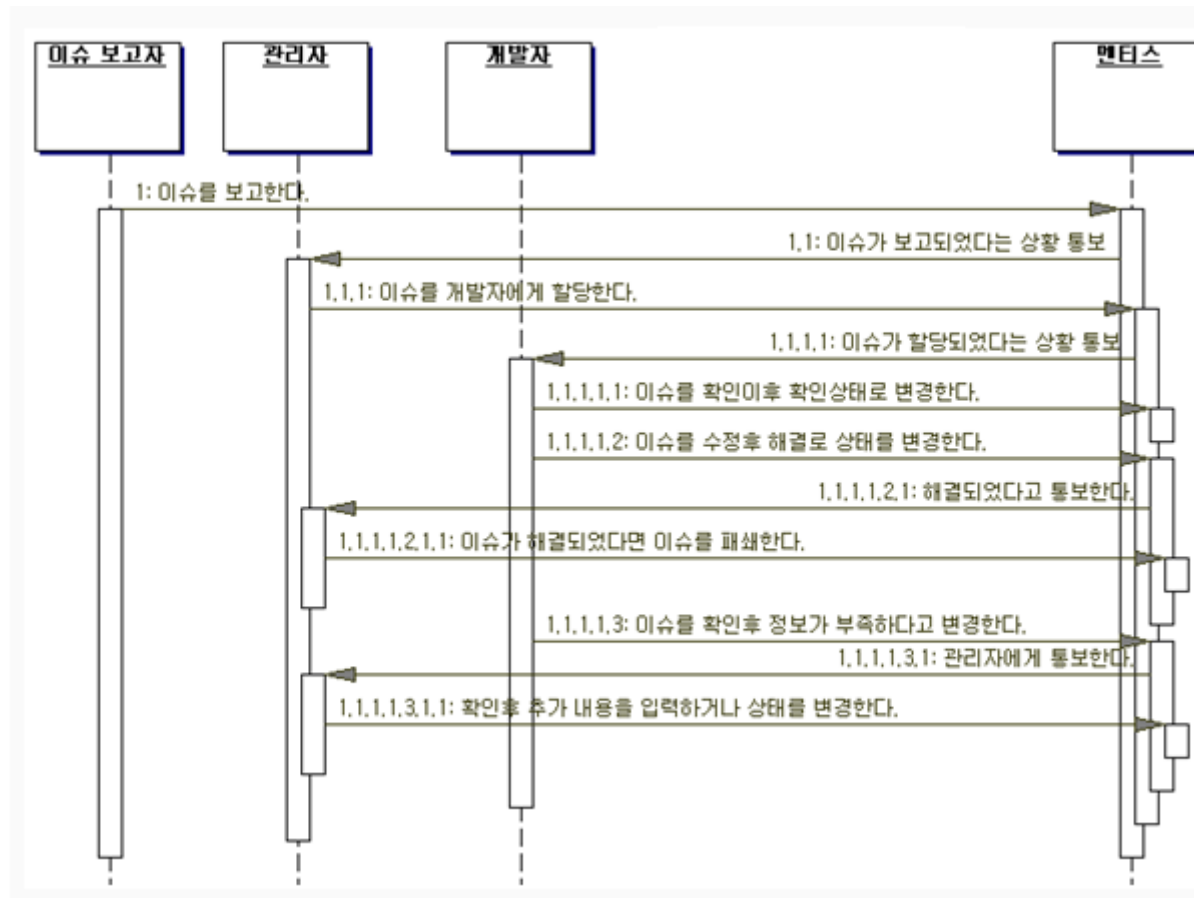
Mantis_Issue

□ **드로잉 이슈 확인**

▶ **화면 상단의 [이슈 보기] 클릭**

☐			0000001		General	중요함	활당된 이슈 (administrator)	2013-05-01	너무 잘생김
☐ 모두 선택		이동		확인					

Mantis_IssueFlow



Mantis_Roadmap

□ 로드맵

- 목표 버전까지의 이슈들의 상태를 확인 가능

어린이 학습용 그림판 - 로드맵

[어린이 학습용 그림판](#) - [0.1](#) (릴리스 예정 2013-05-01) [[이슈 보기](#)]

=====

0%

- [0000001](#): [General] 너무 잘생김 ([administrator](#)) - 할당된 이슈.

전체 1개 중 0개 이슈를 해결했습니다. 진행 (0%).

JUNIT



JUnit_What is Junit?

□ Junit 이란?

- Java를 위한 Unit Testing Framework
- 자동화된 테스트를 위한 도구

□ Unit Test란?

- 소스 코드의 특정 모듈이 의도된 대로 정확히 작동하는지 검증하는 작업
- 함수와 메소드에 대한 테스트 케이스(Test case)를 작성하고 Test하는 과정
- 문제를 초기에 발견할 수 있게 해준다

JUnit_Term

□ 용어 정리

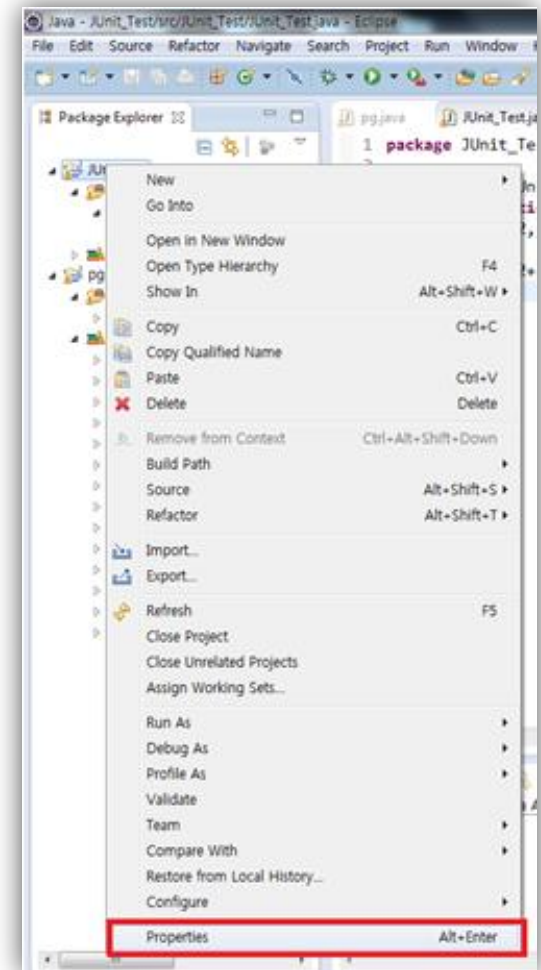
용어	설명
@Test	Unit Test를 수행할 대상 method
@Ignore	Test를 수행하지 않을 method
@After	각 UnitTest의 Method의 실행 후, 뒤에서 초기화 및 자원 정리
@Before	
@AfterClass	각 UnitTest의 class 수행 전, 후에 초기화 및 자원정리
@BeforeClass	

JUnit_How to Use?

- JUNIT은 기본적으로 eclipse를 설치 시 포함되어 있다.
- 이를 사용하려면 다음과 같이 설정해주어야 함

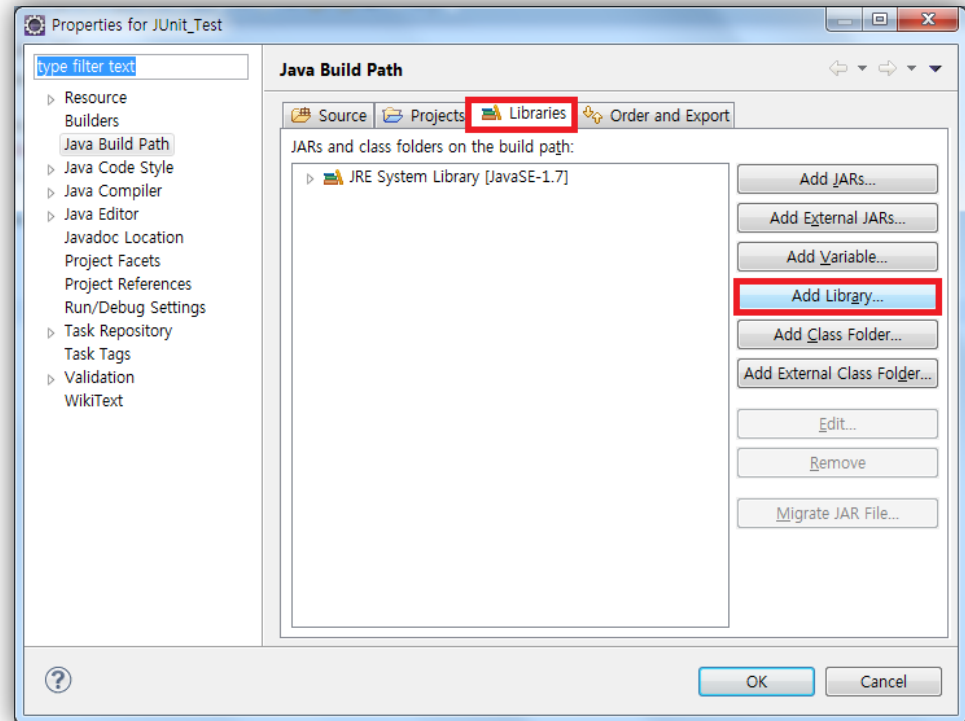
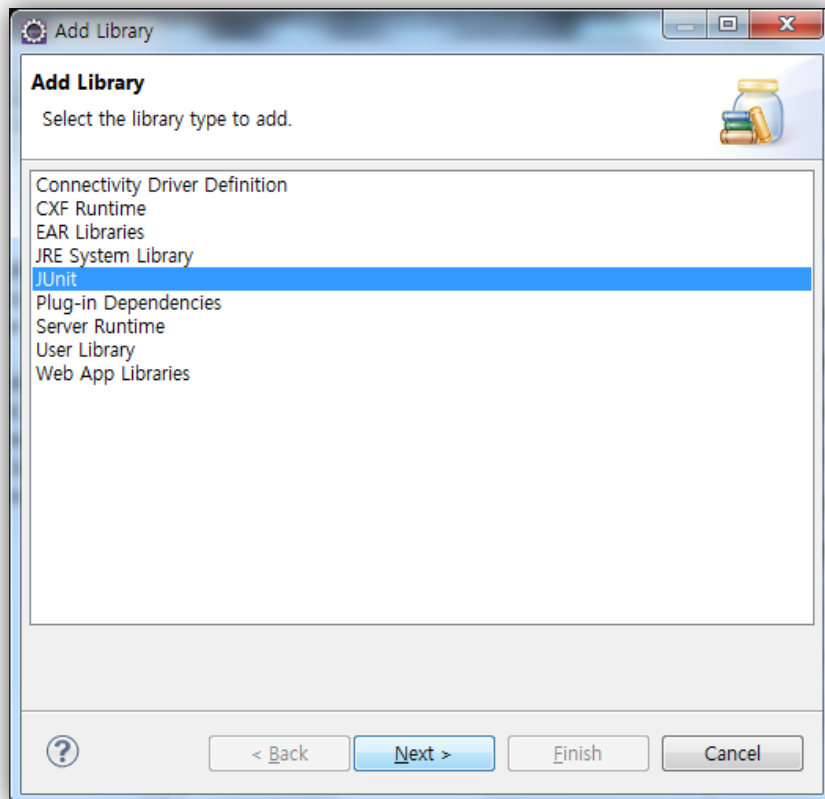
1. 개발하는 프로젝트의 properties를 설정

- Properties>Java Build Path>libraries 탭을 선택



JUnit_How to Use?

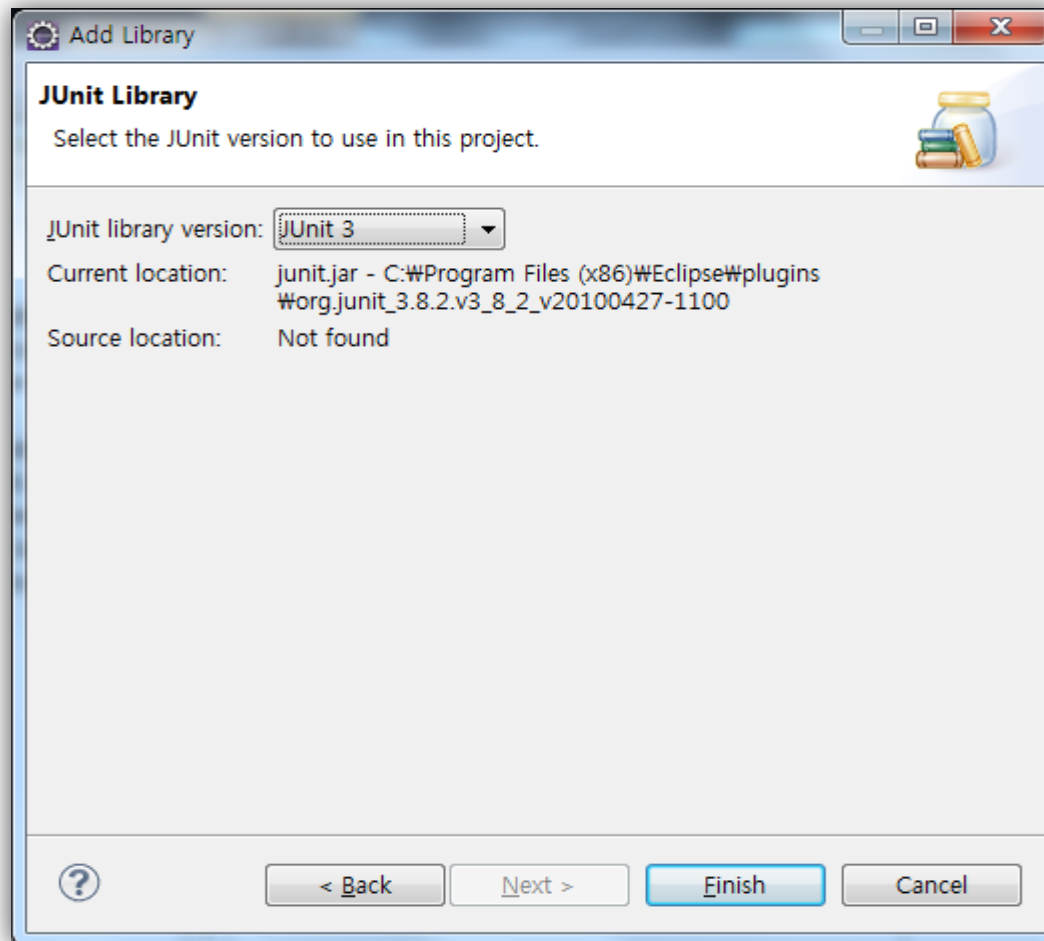
➤ Add library를 선택



➤ JUNIT을 클릭

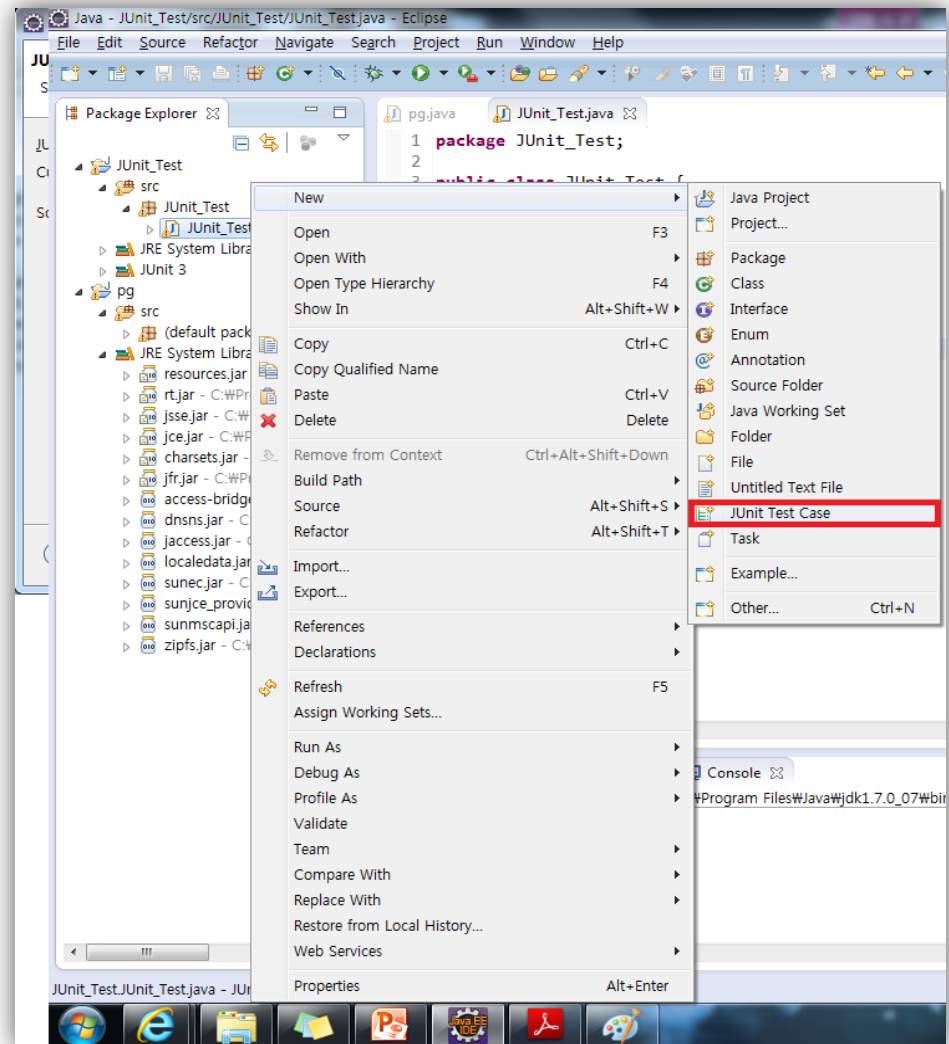
JUnit_How to Use?

- 사용할 JUnit 버전을 선택하고 Finish 클릭

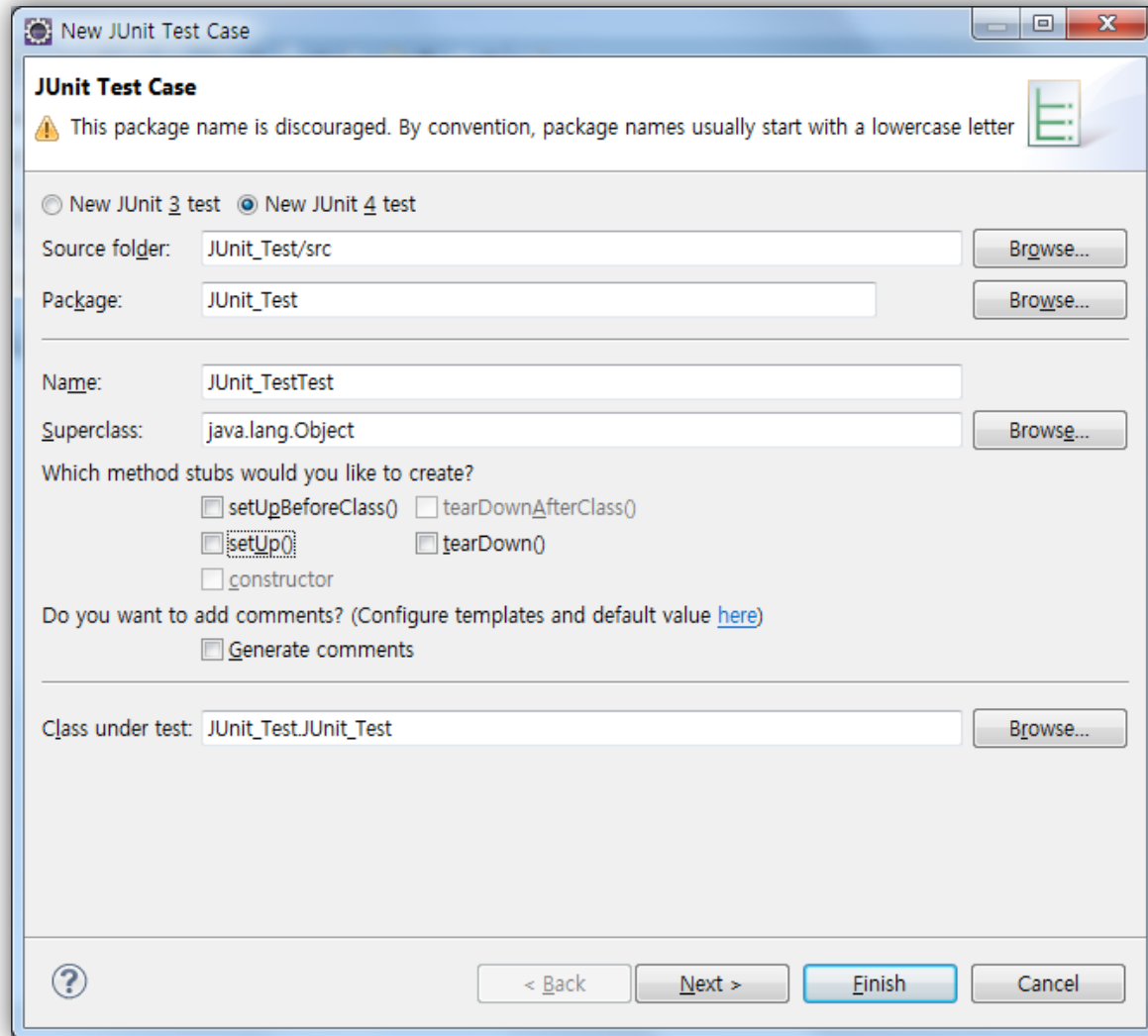


JUnit_How to Use?

2. 테스트하고자 하는 소스를 선택하고,
New>JUnit Test Case 클릭



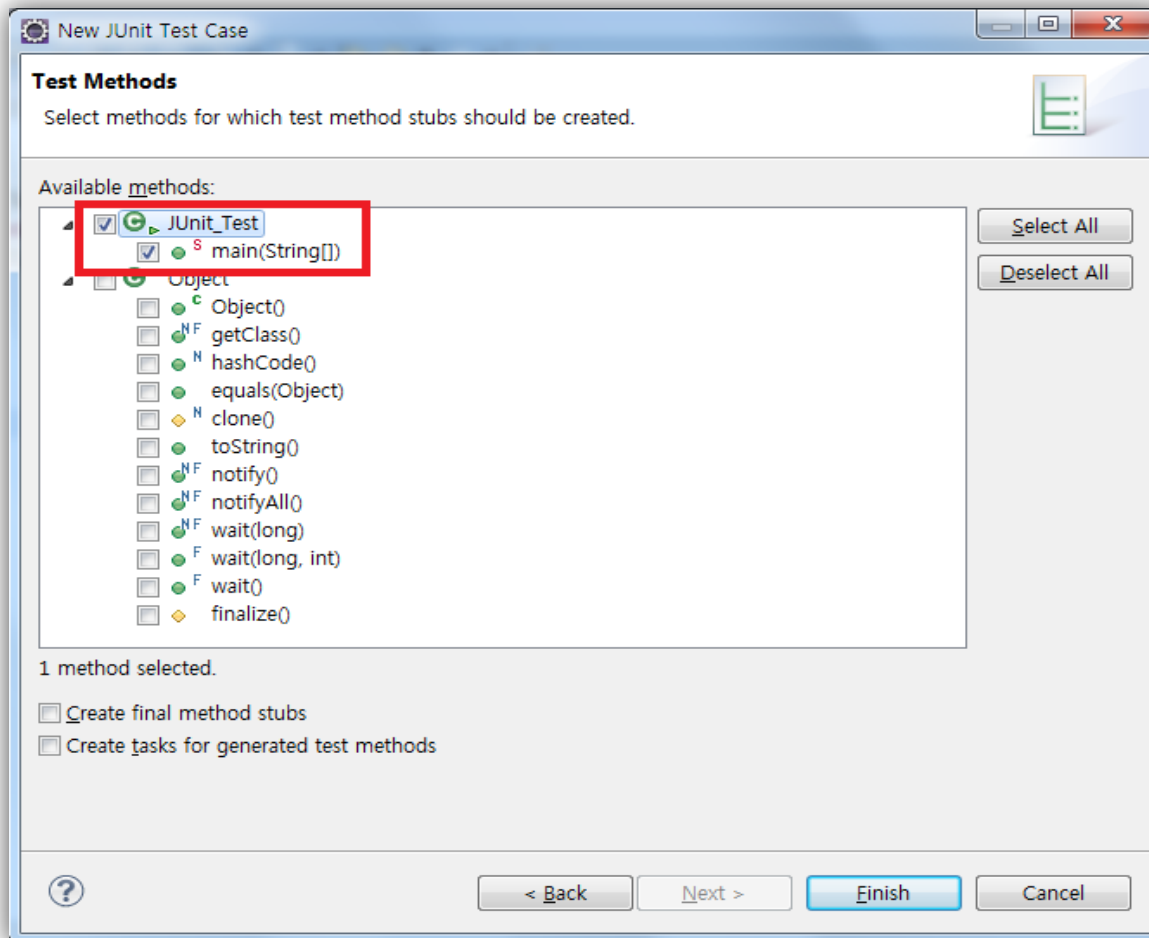
JUnit_How to Use?



- **Source folder : Test 하려는 Source code 경로**
- **setUp(): 테스트대상클래스의 객체를 생성하거나 네트워크 연결, DB 연결 작업 등을 수행.**
- **tearDown(): setUp 과정 반대의 기능. 객체의 제거, 네트워크 종료, DB 연결 종료 등을 수행**
- **Class Under Test : Test 하려는 Case**

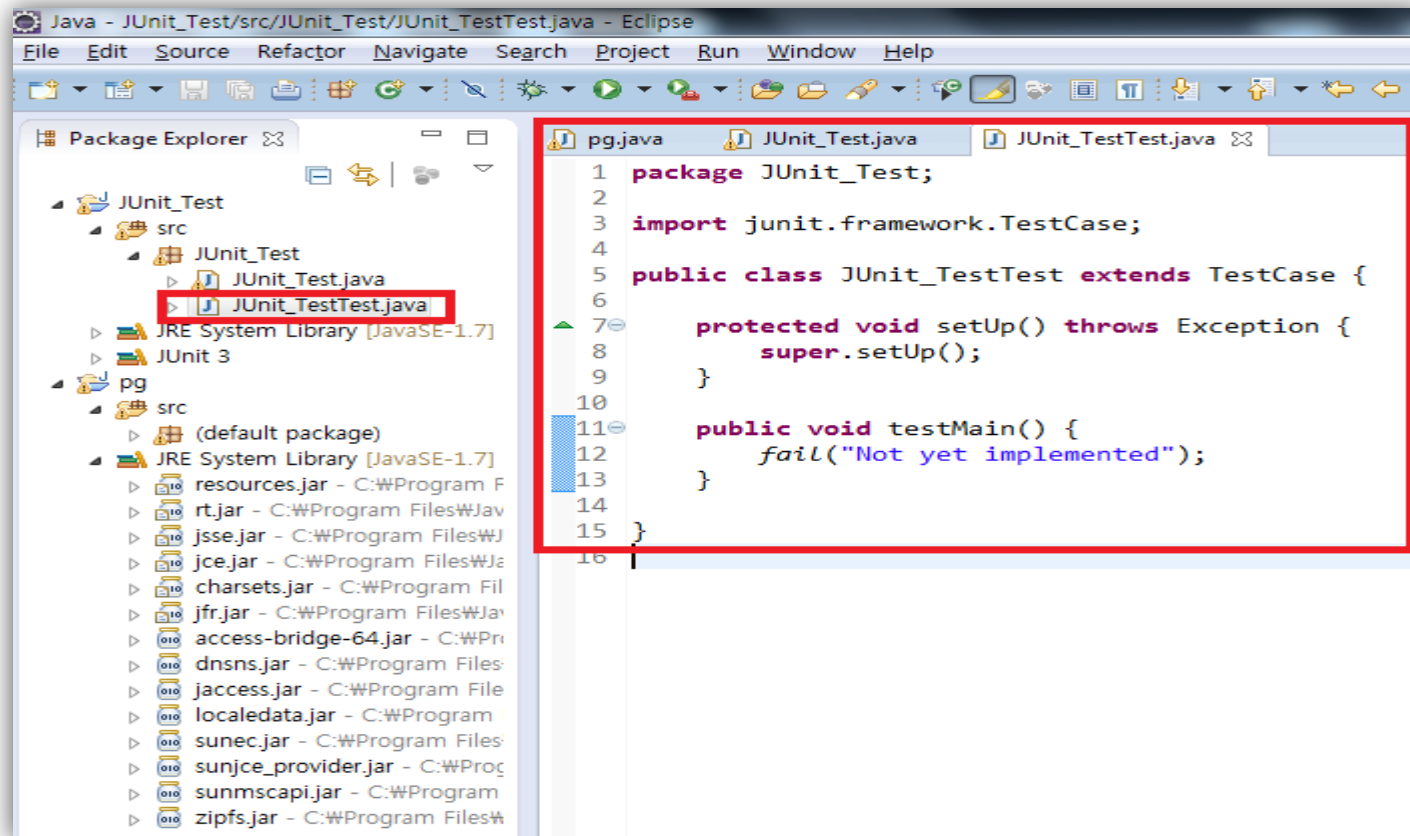
JUnit_How to Use?

□ 테스트하고자 하는 메소드를 선택



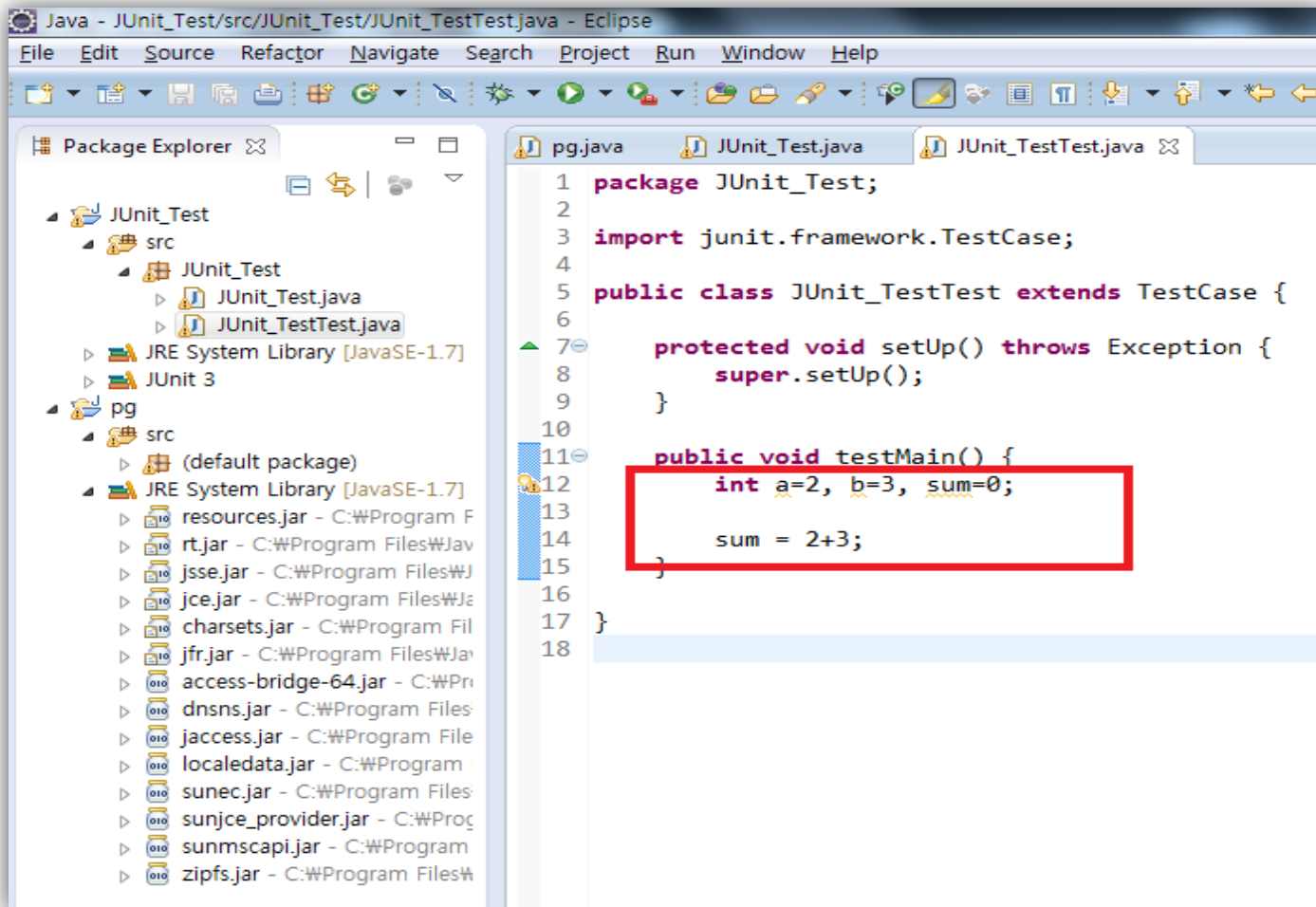
JUnit_How to Use?

- Test Case가 생성된 모습
- 선택한 메소드에 test prefix가 붙은 소스가 생성되었음



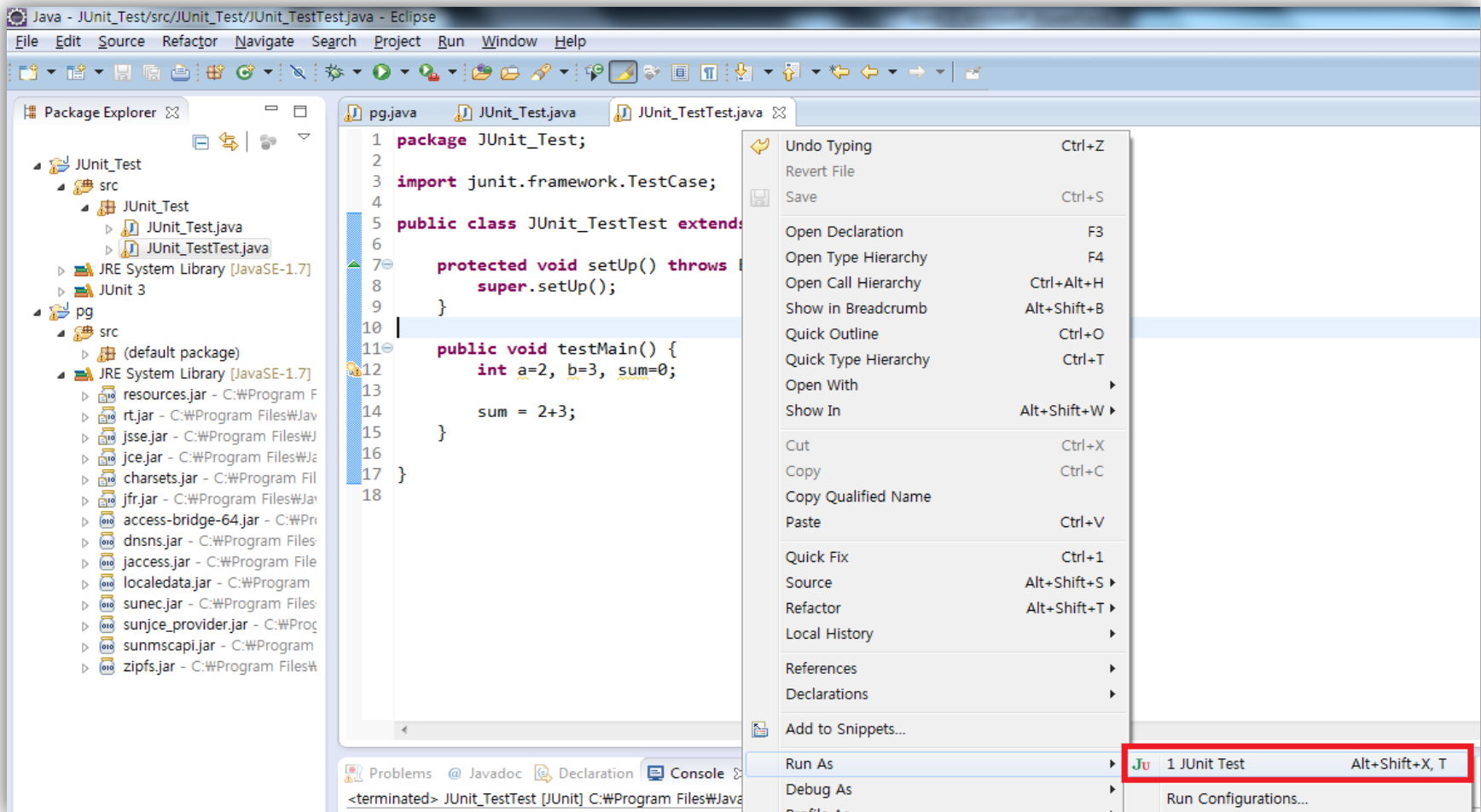
JUnit_How to Use?

3. Test Case에 테스트 코드 입력



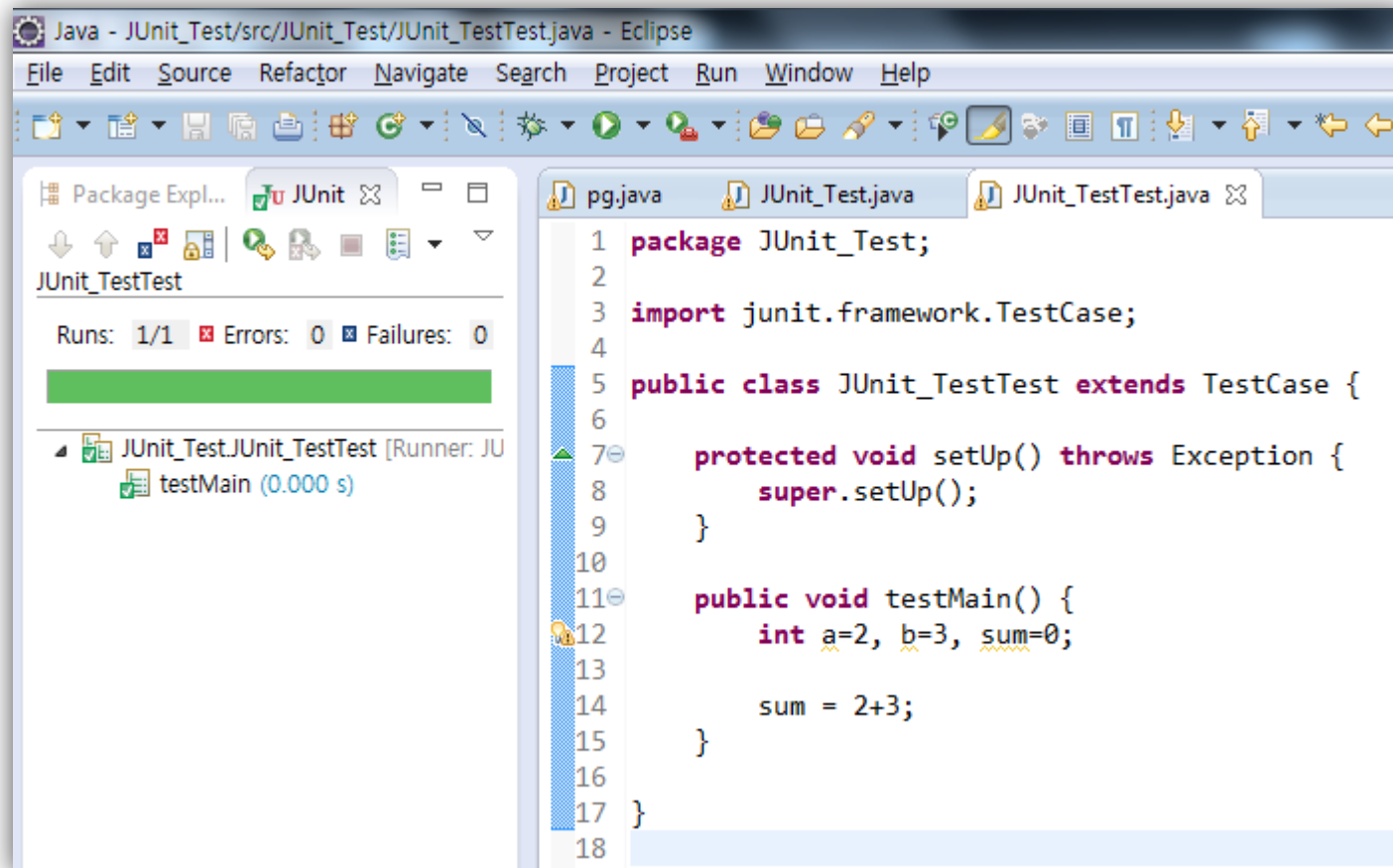
JUnit_How to Use?

4. 마우스 우클릭 > Run AS > JUnit Test



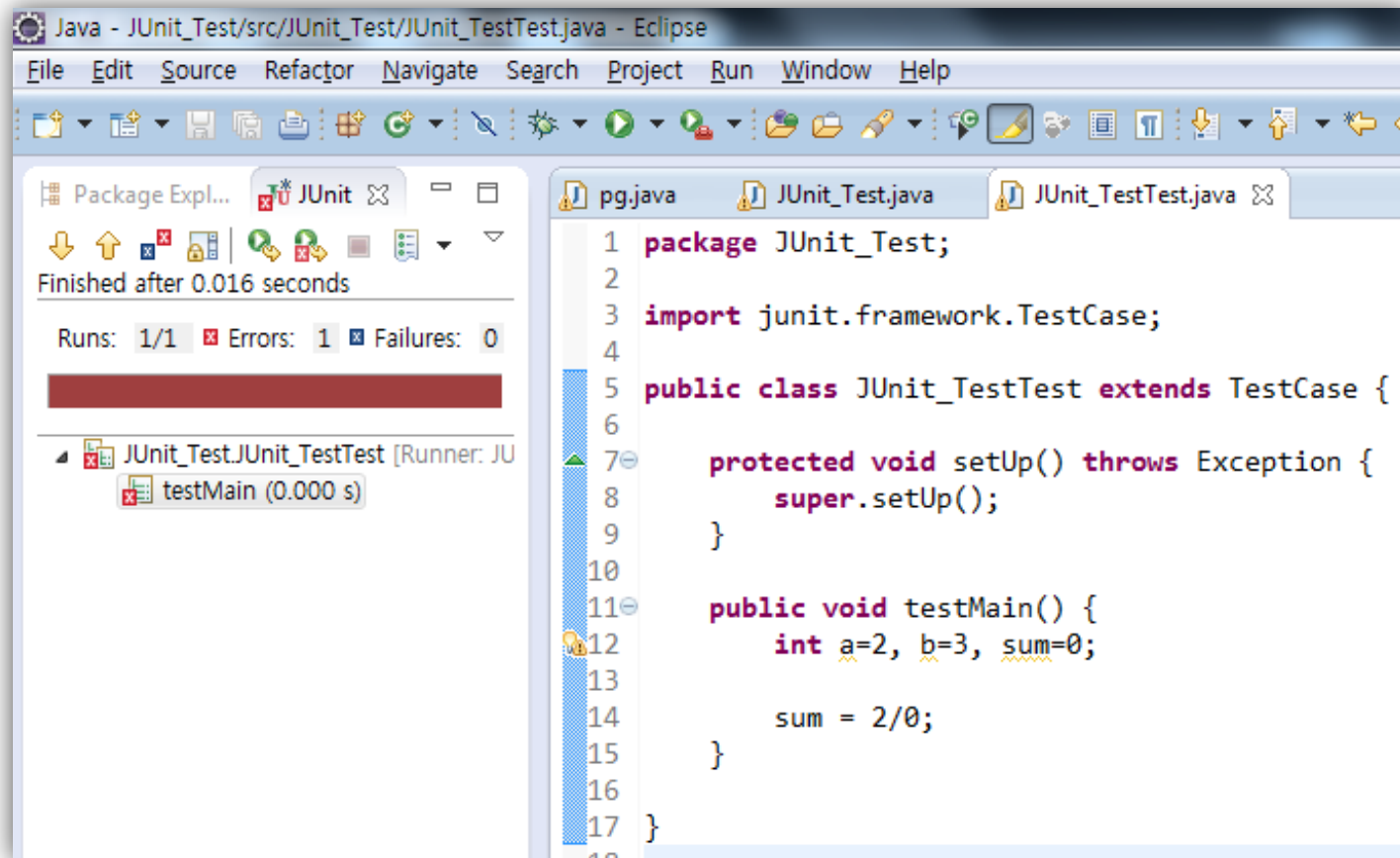
JUnit_How to Use?

- 테스트가 정상적으로 완료되면 아래와 같이 표시됩니다.



JUnit_How to Use?

- 만약 에러가 발생할 경우 아래와 같이 메시지가 표시됩니다.



```
Java - JUnit_Test/src/JUnit_Test/JUnit_TestTest.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

JUnit
Finished after 0.016 seconds
Runs: 1/1 Errors: 1 Failures: 0

JUnit_Test.JUnit_TestTest [Runner: JU]
testMain (0.000 s)

pg.java JUnit_Test.java JUnit_TestTest.java
1 package JUnit_Test;
2
3 import junit.framework.TestCase;
4
5 public class JUnit_TestTest extends TestCase {
6
7     protected void setUp() throws Exception {
8         super.setUp();
9     }
10
11     public void testMain() {
12         int a=2, b=3, sum=0;
13
14         sum = 2/0;
15     }
16
17 }
```

THANK YOU

