

SOFTWARE

2015.03.20

VERIFICATION

TEAM 4

[JUnit & Eclipse 및 빌드 환경]

컴퓨터 공학부

200911393 박현규

201010768 최정한

201111339 김민우

Presenter → 201211389 함진아

CONTENTS

●-----	PART 1	Purpose & CI	3
●-----	PART 2	Eclipse	5
●-----	PART 3	JUnit	32
●-----	PART 4	Build Environment	55
●-----	✘	References	86

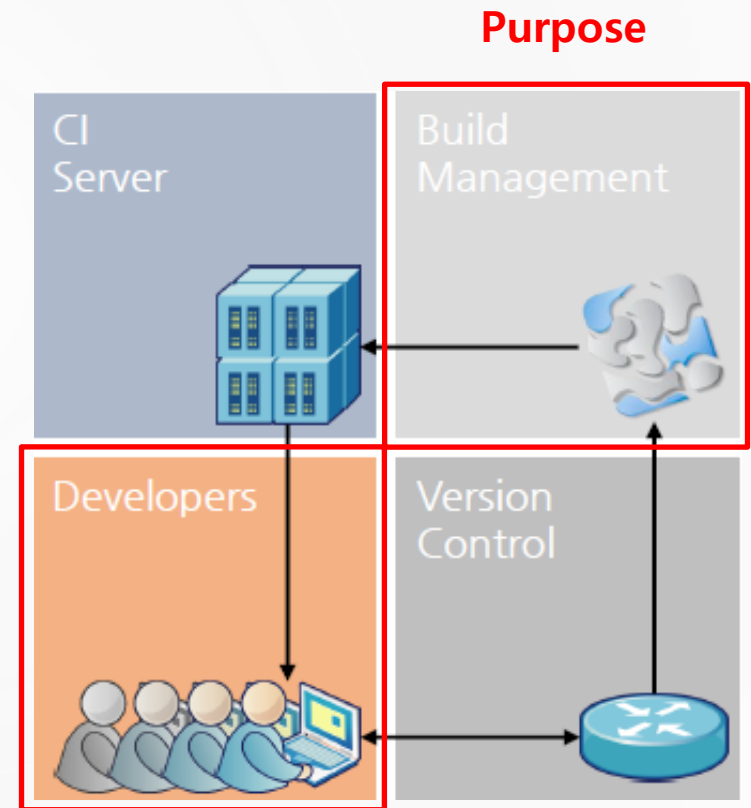
PART 1

Purpose & CI

PART 1 Purpose & CI

What is CI ?

- Continuous integration
- 프로젝트에 참여중인 개발자들의 결과물을 지속적으로 통합하고, build 함으로써 각자의 저작물을 통합할 때 생기는 문제를 예방하기 위해 도입됐다.
- 개발자들은 문제점을 빠른 시점에 알고 고칠 수 있고, test code를 한번 작성해 두면 개발 중 해당 부분에 대한 테스트는 자동적으로 진행 된다.
- 자동화된 단위 테스트, 서버 환경에서 주기적으로 혹은 commit 시 build, 정적 or 동적 테스트 성능 측정, 소스코드로부터 자동 문서화 작업 등을 거친다.



PART 2

Eclipse

PART 2 Eclipse

What is Eclipse ?



- Eclipse is an **integrated development environment (IDE)**.
- It contains a base **workspace** and an extensible **plug-in system** for customizing the environment.
- Written mostly in Java, Eclipse can be used to **develop applications**.
- Eclipse는 통합 개발 환경으로 코딩, 디버그, 컴파일 등 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공한다.
- 이번 프로젝트에 대한 설치 및 사용 가이드는 2014년에 나온 Luna로 가장 최신 버전인 **4.4.1**과 **4.4.2**를 기준으로 작성했다.

PART 2 Eclipse

How to install Eclipse

1. JDK(Java Development Kit) 설치

Oracle 홈페이지 접속 : <http://www.oracle.com/index.html>

Downloads → Java for Developers 클릭

ORACLE

Sign In/Register Help Country ▾ Communities ▾ I am a...

Products Solutions **Downloads** Store S

Popular Downloads

- Java for Developers**
- Java for Your Computer
- Java SE
- Solaris
- MySQL
- Oracle Fusion Middleware 11g
- Oracle Database
- Prebuilt Developer VMs

Database

- Oracle Database
- Oracle Database 11g Express Edition
- MySQL
- Oracle Berkeley DB
- Oracle Instant Client
- Oracle Application Express
- See All...

Middleware

- Oracle Fusion Middleware (including Oracle WebLogic Server)
- Oracle JRockit
- Oracle SOA Suite
- See All...

Enterprise Management

- Oracle Enterprise Manager
- Oracle Application Testing Suite
- See All...

Server and Storage Systems

- Solaris
- Linux and Oracle VM
- Firmware
- See All...

PART 2 Eclipse

How to install Eclipse

1. JDK(Java Development Kit) 설치

JDK 다운로드

Overview Downloads Documentation Community Technologies Training

Java SE Downloads


DOWNLOAD ↓
Java Platform (JDK) 8u40


DOWNLOAD ↓
NetBeans with JDK 8

Java Platform, Standard Edition

Java SE 8u40

This latest release of Oracle's implementation of Java SE, JDK 8u40, brings improvements to performance, scalability and administration, making it easier for Java developers, partners and IT decision makers to innovate faster in a simple, easy manner and improve application services. The release also includes new updates to JavaFX. Full release notes can be found [here](#).
[Learn more](#) ▶

- Installation Instructions

JDK
DOWNLOAD ↓

PART 2 Eclipse

How to install Eclipse

1. JDK(Java Development Kit) 설치

JDK 다운로드

위의 차트 (Just JDK)

데모가 없는 버전.
사용자의 운영체제 환경에
맞춰서 다운 받으면 된다.
이번 프로젝트에 대한 설치 및 사
용 가이드는 맥과 윈도우
두 가지를 기준으로 작성했다.

Java SE Development Kit 8u40

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux x86	146.84 MB	jdk-8u40-linux-i586.rpm
Linux x86	166.85 MB	jdk-8u40-linux-i586.tar.gz
Linux x64	145.14 MB	jdk-8u40-linux-x64.rpm
Linux x64	165.17 MB	jdk-8u40-linux-x64.tar.gz
Mac OS X x64	221.9 MB	jdk-8u40-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.09 MB	jdk-8u40-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.68 MB	jdk-8u40-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	130.57 MB	jdk-8u40-solaris-x64.tar.Z
Solaris x64	89.91 MB	jdk-8u40-solaris-x64.tar.gz
Windows x86	175.71 MB	jdk-8u40-windows-i586.exe
Windows x64	180.19 MB	jdk-8u40-windows-x64.exe

아래 차트 (JDK+DEMO)

Java SE Development Kit 8u40 Demos and Samples Downloads

You must accept the Oracle BSD License. to download this software.

Accept License Agreement Decline License Agreement

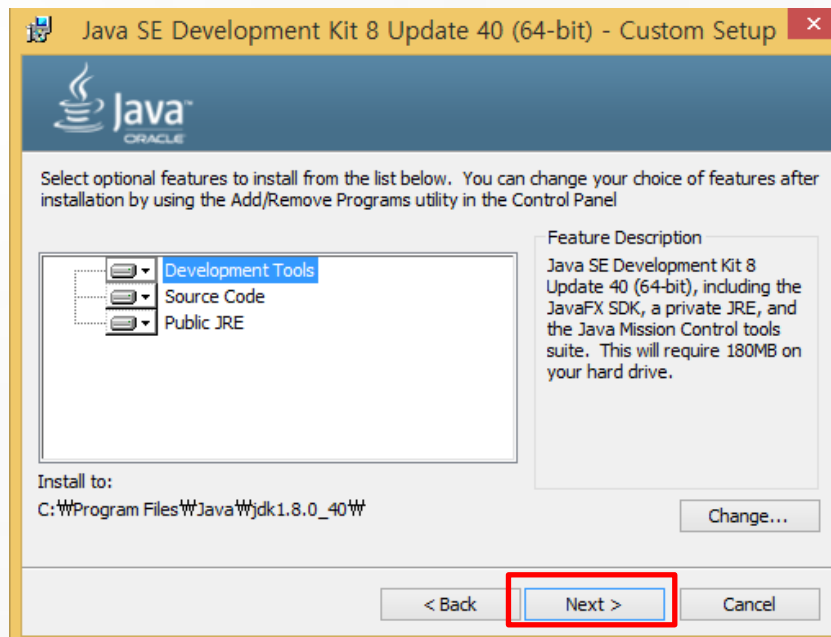
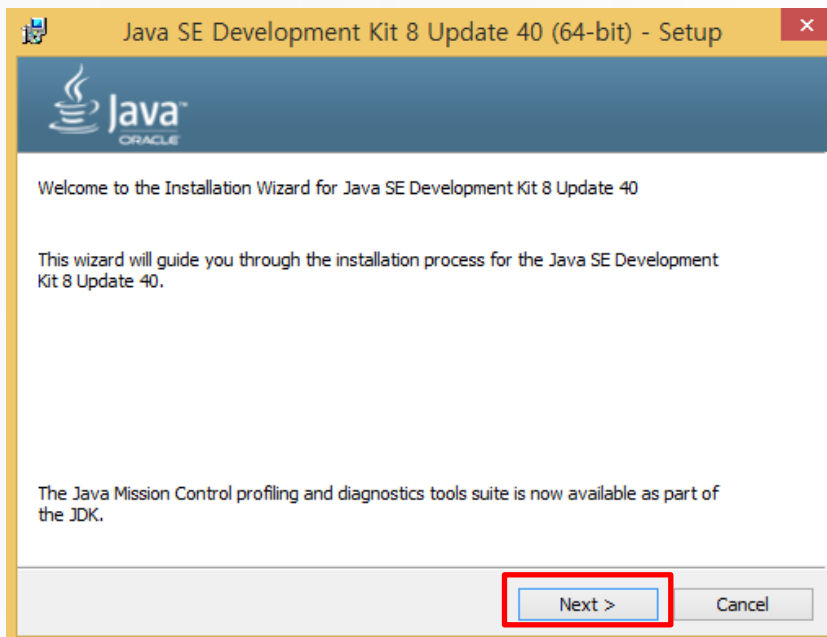
Product / File Description	File Size	Download
Linux x86	52.4 MB	jdk-8u40-linux-i586-demos.rpm

PART 2 Eclipse

How to install Eclipse

1. JDK(Java Development Kit) 설치

JDK 설치 진행

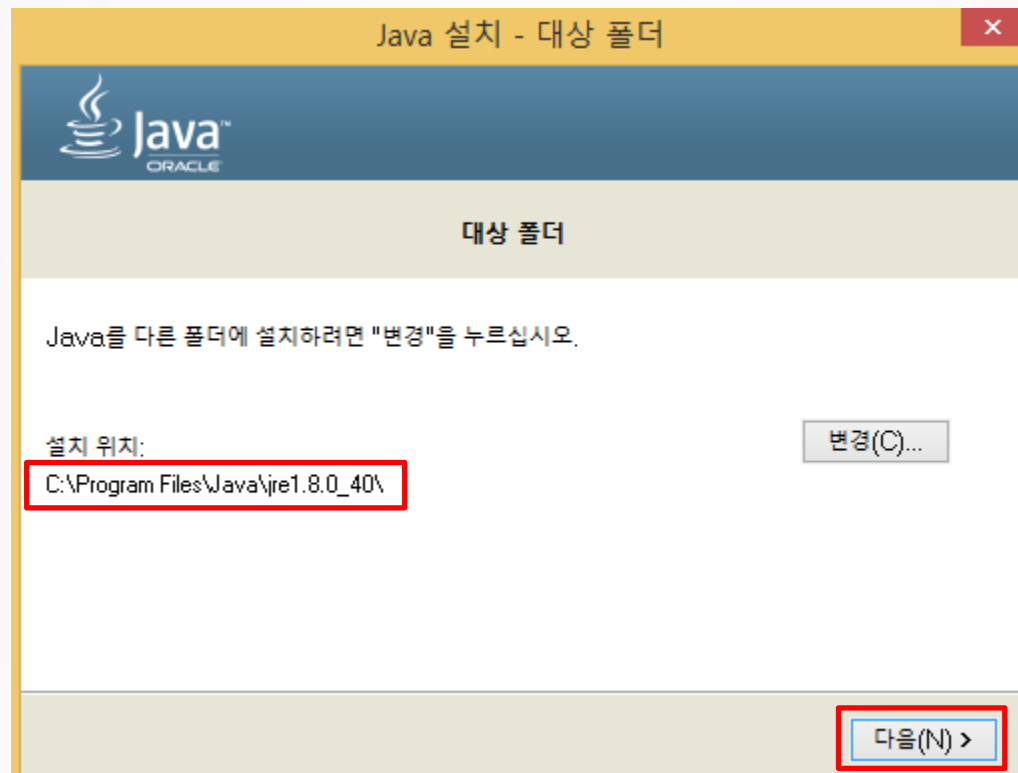


PART 2 Eclipse

How to install Eclipse

1. JDK(Java Development Kit) 설치

Java 설치 경로 확인 후 복사 (환경 설정에서 경로 지정 시 필요하다.)



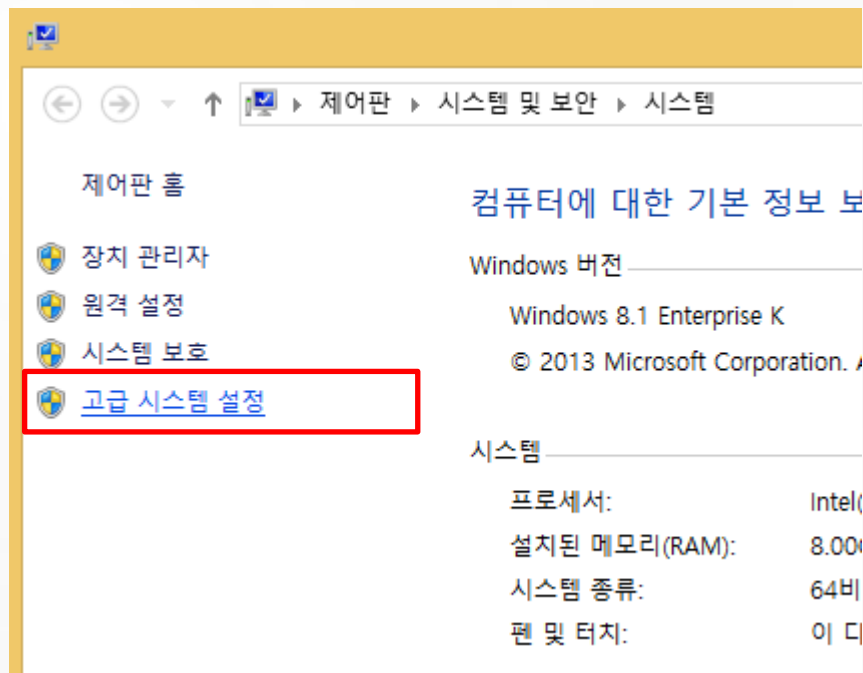
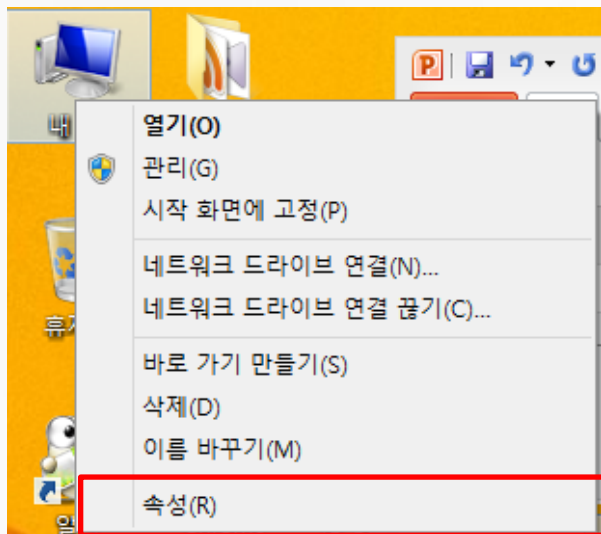
PART 2 Eclipse

How to install Eclipse

2. 환경 변수 설정

JDK 설치 후 환경변수 설정을 위해 내 컴퓨터에서 오른쪽 마우스

→ 속성 → 고급 시스템 설정



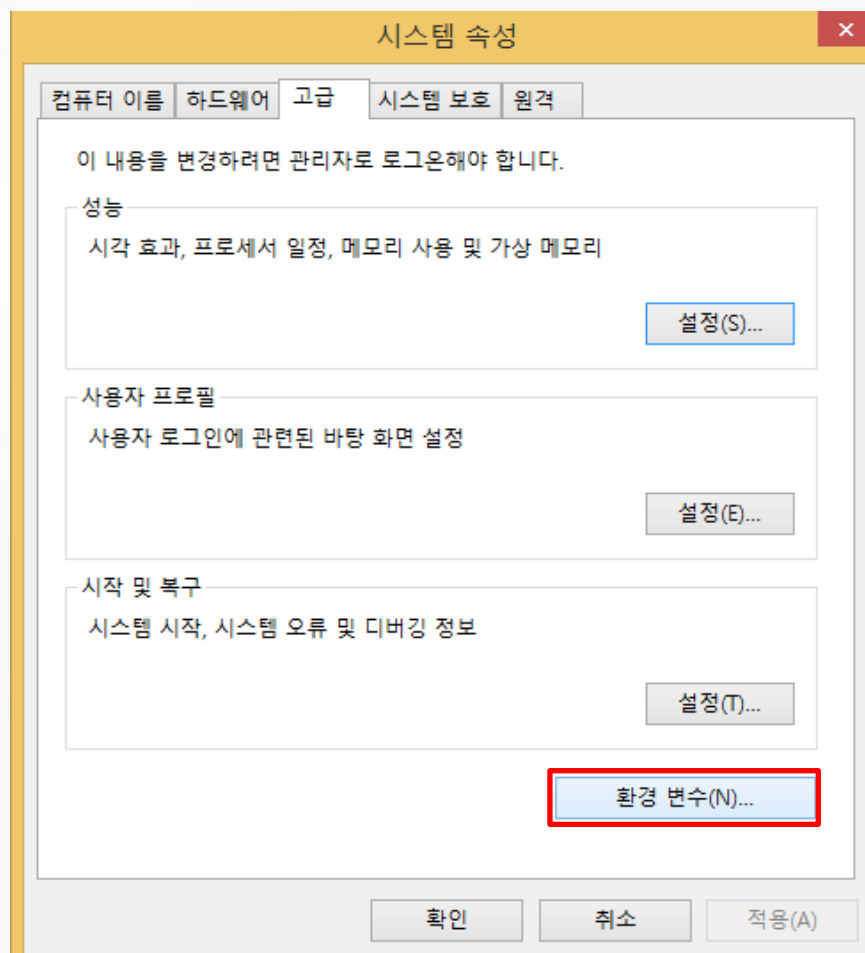
PART 2 Eclipse

How to install Eclipse

2. 환경 변수 설정

시스템 속성에서 고급 탭

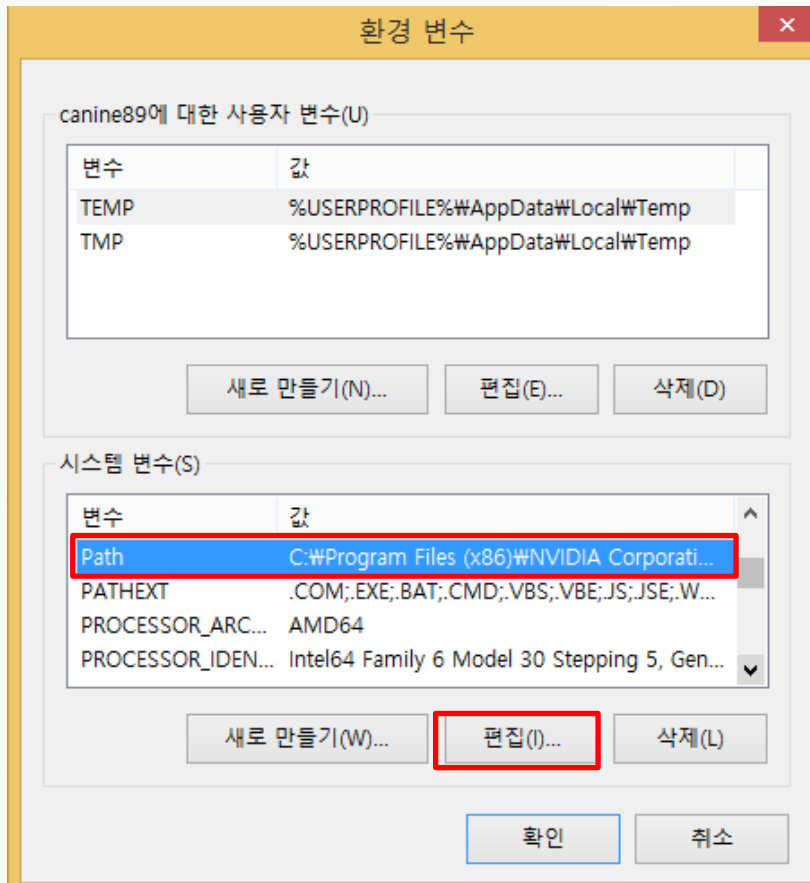
→ 환경 변수



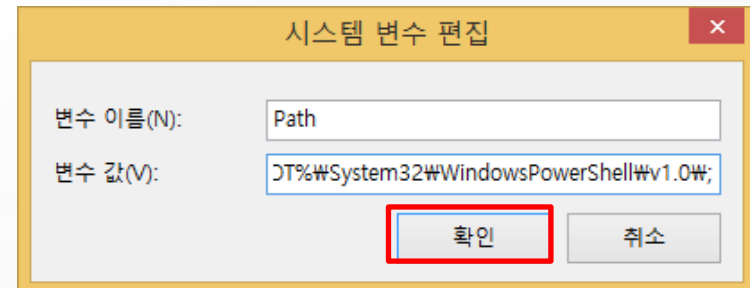
PART 2 Eclipse

How to install Eclipse

2. 환경 변수 설정



환경 변수 창이 뜨면, 시스템 변수에서 Path를 찾아 클릭 → 편집



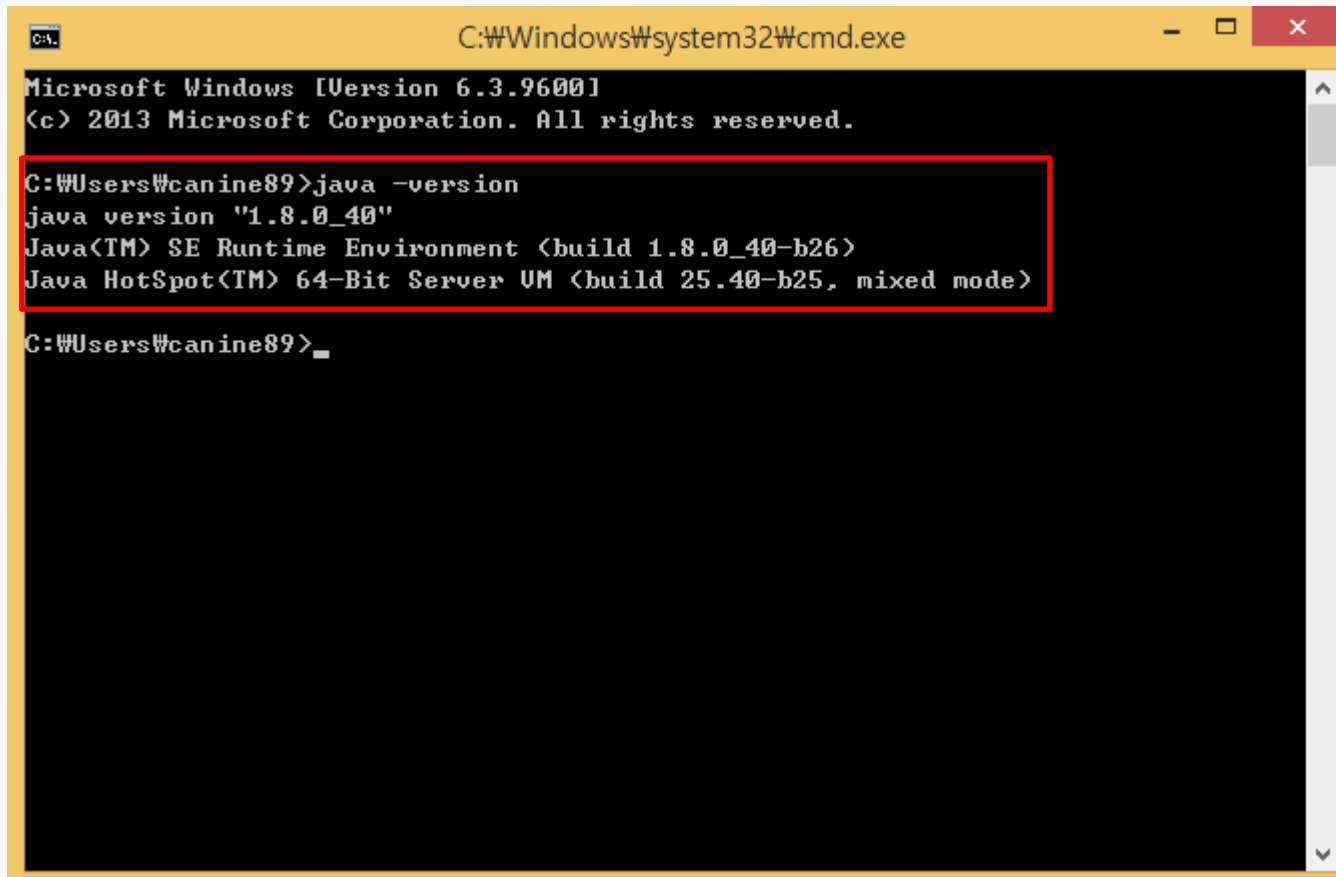
시스템 변수 편집 창이 뜨면,
변수 값의 맨 뒤에 아까 복사해둔 자바 설치 경로 붙여넣기

※ “ ” 뒤에 이어 붙인 후 맨 뒤에 “ ” 삽입

PART 2 Eclipse

How to install Eclipse

3. Java 설치 확인



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\canine89>java -version
java version "1.8.0_40"
Java(TM) SE Runtime Environment (build 1.8.0_40-b26)
Java HotSpot(TM) 64-Bit Server VM (build 25.40-b25, mixed mode)

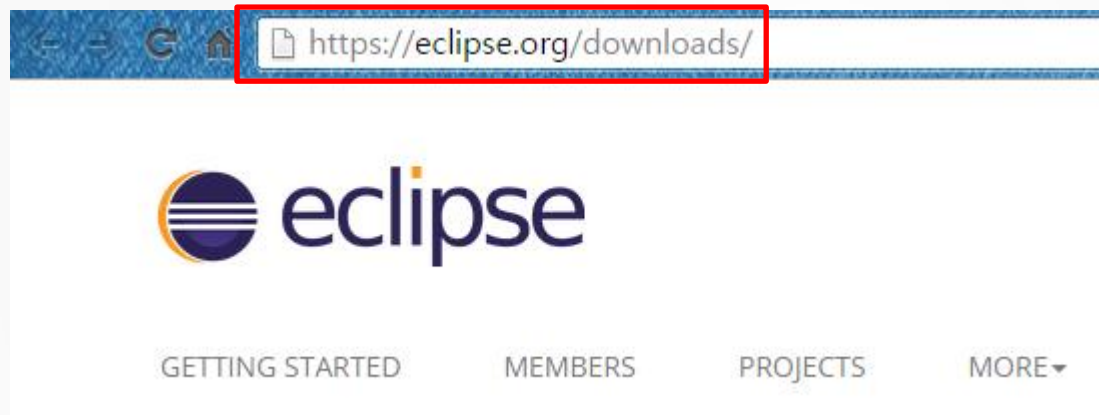
C:\Users\canine89>
```

PART 2 Eclipse

How to install Eclipse

4. Eclipse 설치

Eclipse 다운로드 사이트 접속 : <https://eclipse.org/downloads/>



※ 이번 프로젝트에 대한 설치 및 사용 가이드는 2014년에 나온 Luna로 가장 최신 버전인 4.4.1과 4.4.2를 기준으로 작성했다.

운영체제 선택

Eclipse Luna SR2 (4.4.2) Release for **Windows**



Eclipse IDE for Java Developers, 155 MB

Downloaded 732,542 Times

The essential tools for any Java developer, including a Java IDE, a CVS client,



Windows 32 Bit
Windows 64 Bit

PART 2 Eclipse

How to install Eclipse

4. Eclipse 설치

Eclipse downloads - mirror selection

All downloads are provided under the terms and conditions of the **Eclipse Foundation Software User Agreement** unless otherwise specified.

Download eclipse-java-luna-SR2-win32-x86_64.zip **from:**



[Korea, Republic Of] Daum Communications Corp. (<http>)

Checksums: [MD5] [SHA1] [SHA-512]

※ 다운로드 시간은 인터넷 환경에 따라 다르지만, 보통 5분 내외로 다운로드가 가능하다.

다운로드 완료!

Thank you for downloading Eclipse

If the download doesn't start in a few seconds, please [click here](#) to start the download.

PART 2 Eclipse

How to install Eclipse

5. Eclipse 실행

이름	수정한 날짜	유형	크기
eclipse-java-luna-SR2-win32-x86_64	2015-03-16 오후...	파일 폴더	
jdk-8u40-windows-x64.exe	2015-03-16 오후...	응용 프로그램	184,515KB

이름	수정한 날짜	유형
eclipse	2015-03-16 오후...	파일 폴더

이름	수정한 날짜
configuration	2015-03-16 오후...
dropins	2015-02-19 오전...
features	2015-03-16 오후...
p2	2015-03-16 오후...
plugins	2015-03-16 오후...
readme	2015-03-16 오후...
.eclipseproduct	2015-03-16 오후...
artifacts.xml	2015-03-16 오후...
eclipse.exe	2015-03-16 오후...
eclipse.ini	2015-03-16 오후...
eclipsesec.exe	2015-03-16 오후...
epl-v10.html	2015-03-16 오후...
notice.html	2015-03-16 오후...

만든 날짜: 2015-02-19 오전 3:26
크기: 313KB

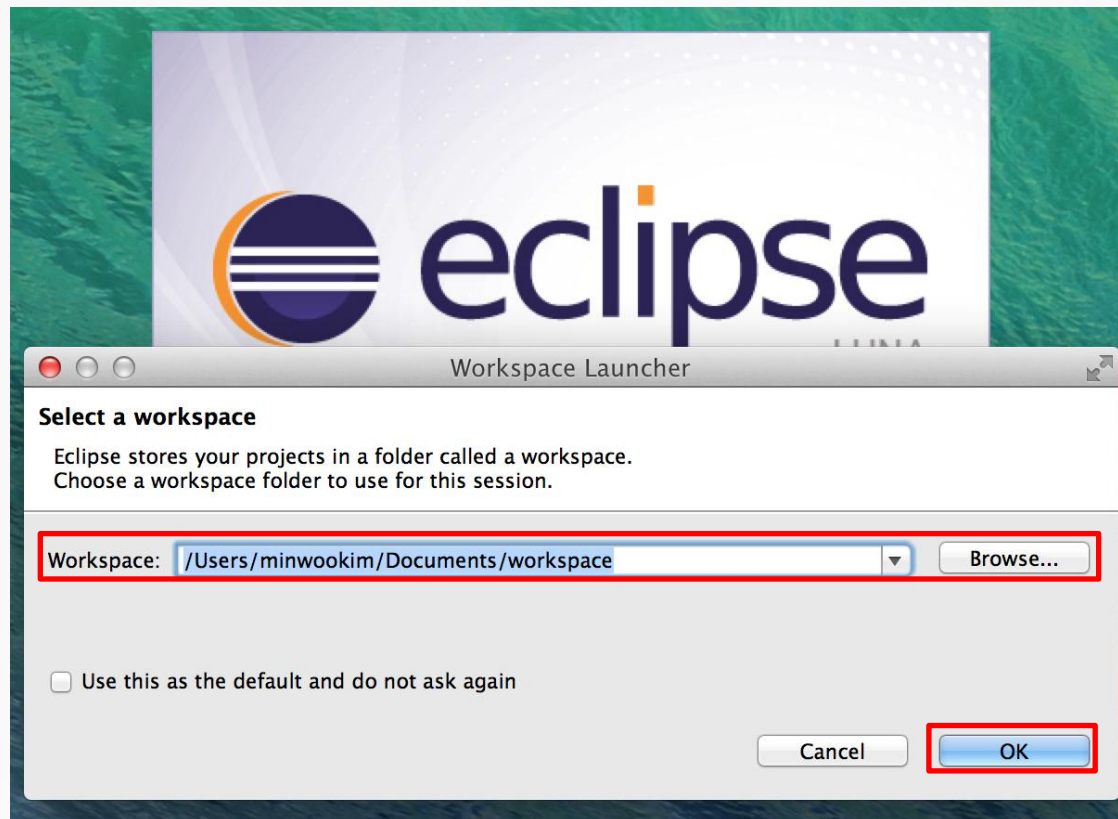
다운로드 한 Eclipse 폴더 압축 해제
→ eclipse → eclipse.exe

PART 2 Eclipse

How to use Eclipse

1. Eclipse 실행

Eclipse 시작 시 workspace 설정 (작성한 프로젝트가 저장될 경로 설정)



PART 2 Eclipse

How to use Eclipse

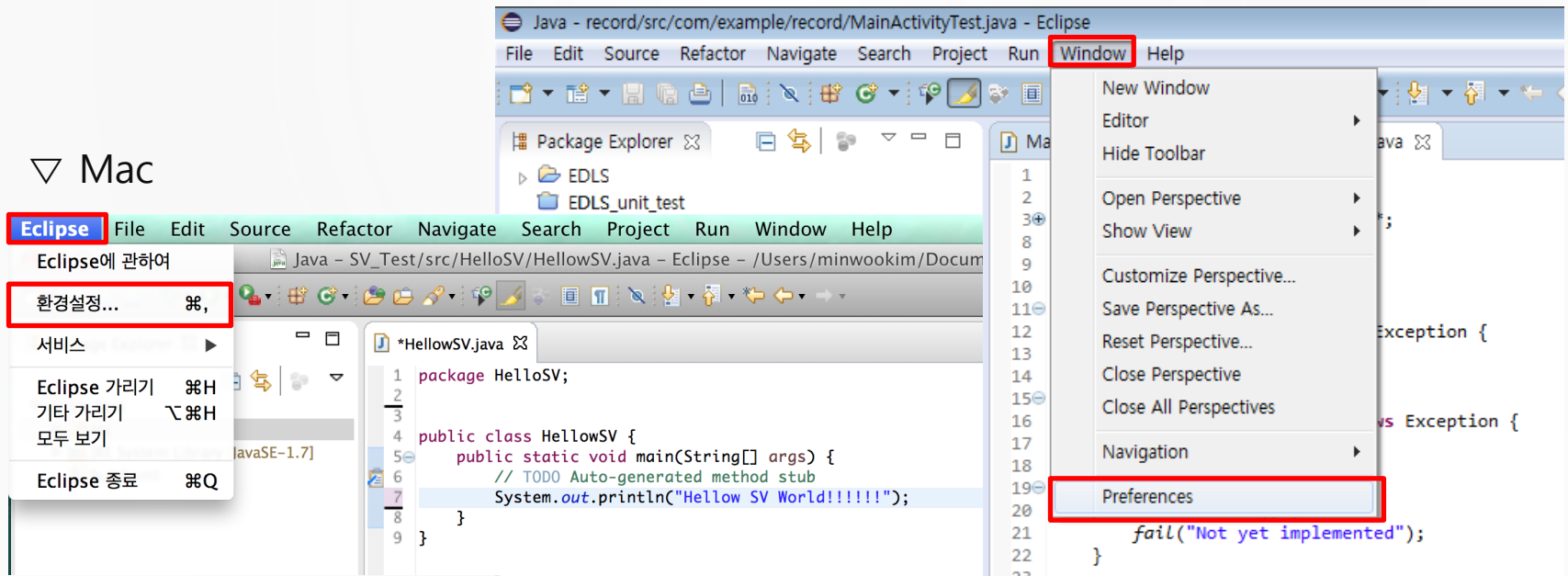
2. 팀 프로젝트를 위한 환경설정 Tip

Windows의 경우 Eclipse 실행 후 상단 바의 Window → Preferences

Mac의 경우 상단 바의 Eclipse → 환경설정

▽ Windows

▽ Mac

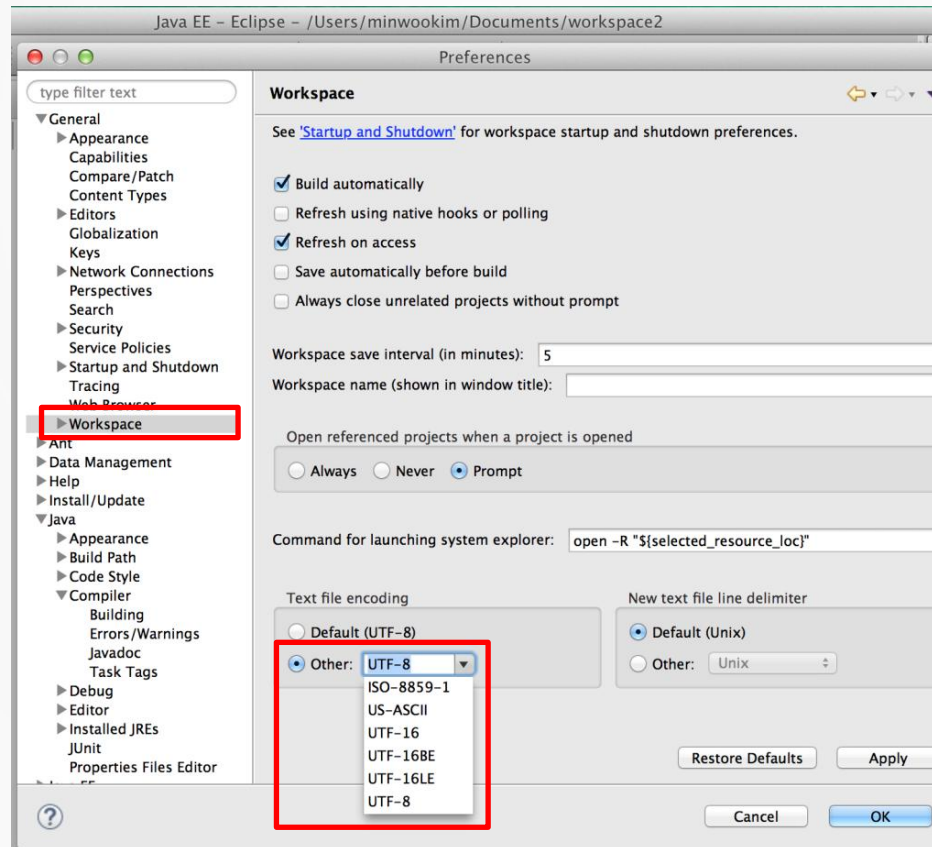


PART 2 Eclipse

How to use Eclipse

2. 팀 프로젝트를 위한 환경설정 Tip

1) Encoding 타입 통일 : 보통 utf-8

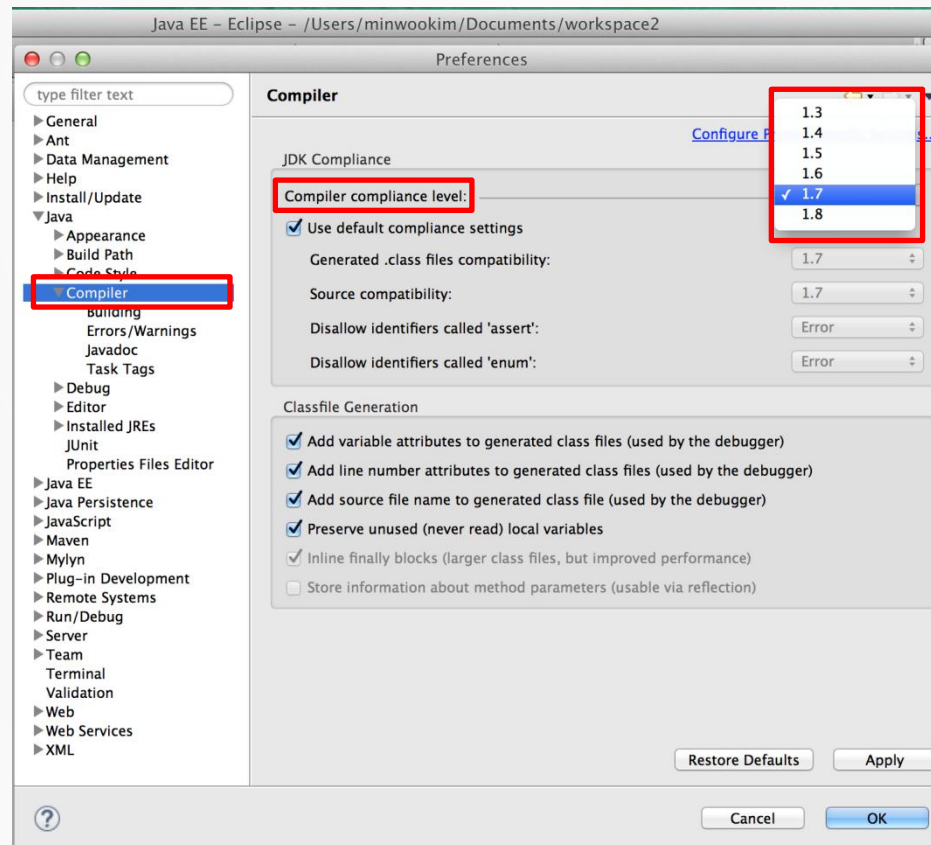


PART 2 Eclipse

How to use Eclipse

2. 팀 프로젝트를 위한 환경설정 Tip

2) JDK 컴파일러 버전 통일 : 2015/3/19 기준으로 1.7 / 1.8 (JDK 7 / JDK 8)

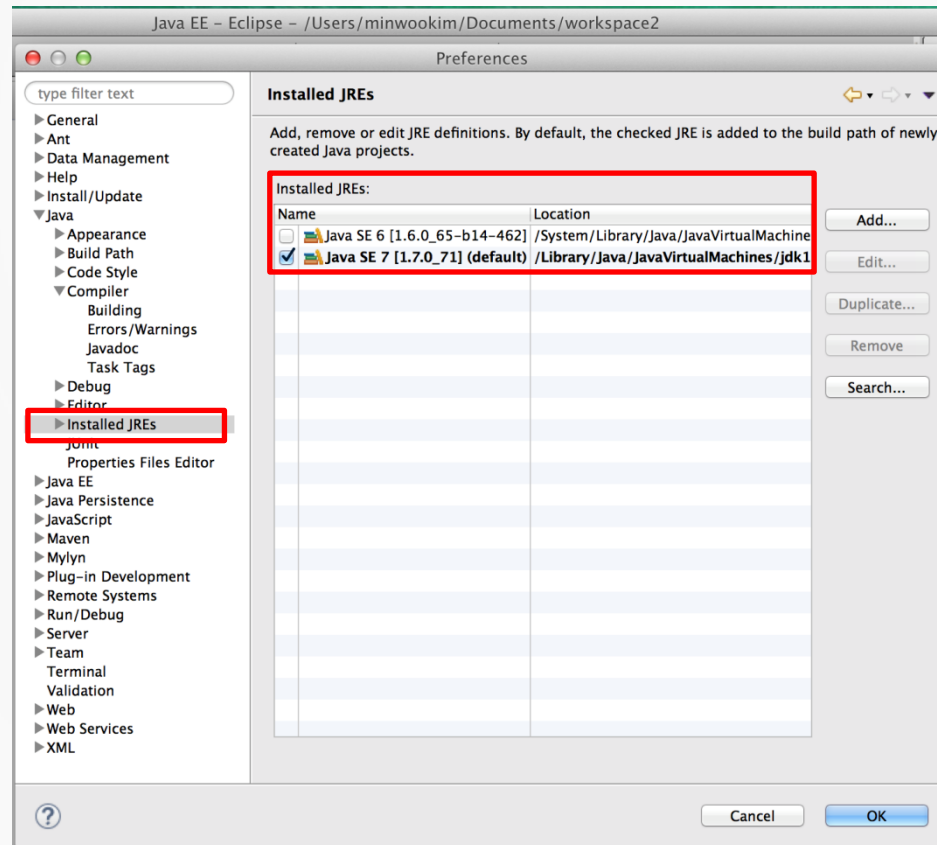


PART 2 Eclipse

How to use Eclipse

2. 팀 프로젝트를 위한 환경설정 Tip

3) Installed JREs 확인 : 2015/3/19 기준으로 1.7 / 1.8 (JDK 7 / JDK 8)

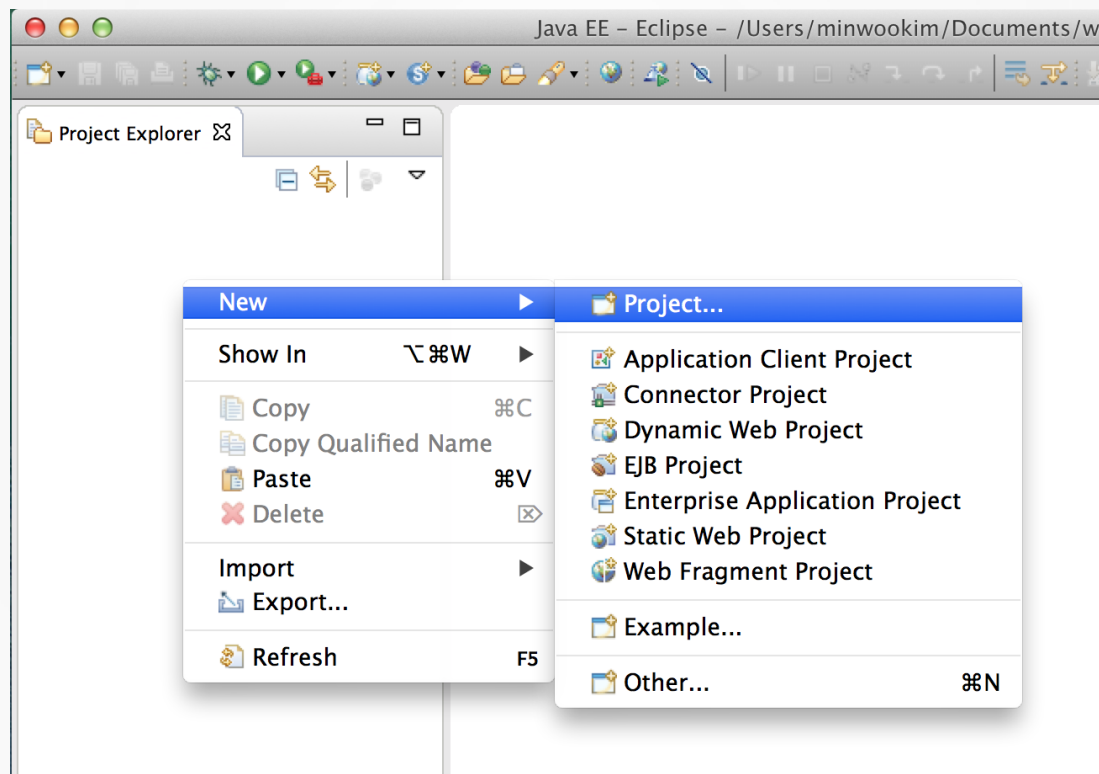


PART 2 Eclipse

How to use Eclipse

3. 프로젝트 만들기

Project Explorer에서 오른쪽 마우스 클릭 → new → project

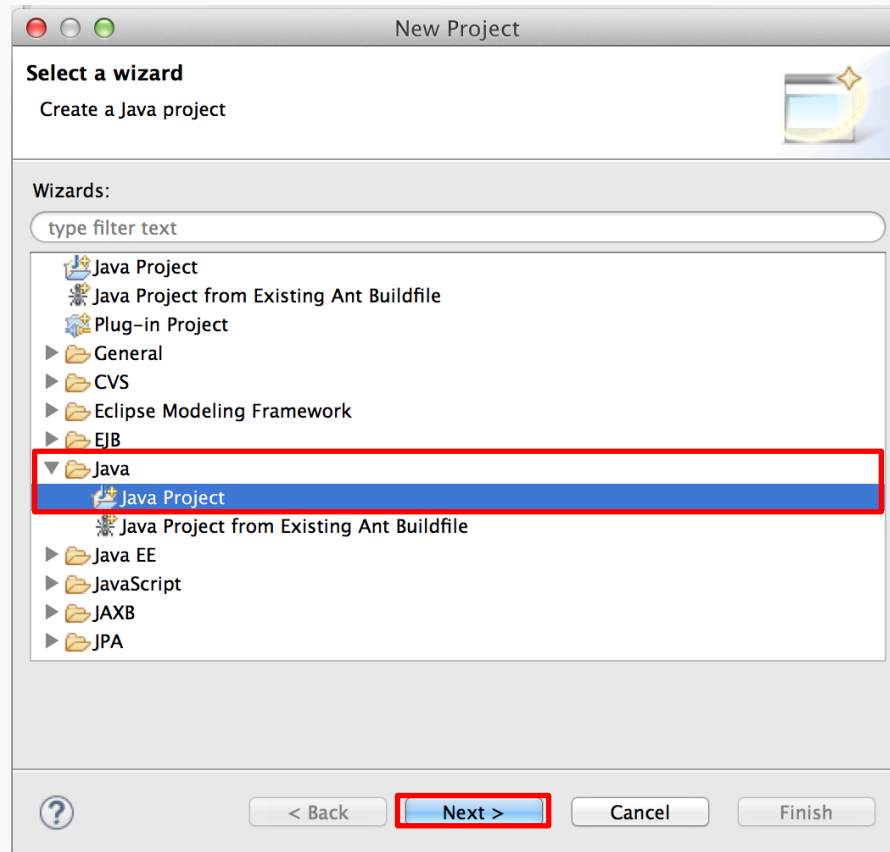


PART 2 Eclipse

How to use Eclipse

3. 프로젝트 만들기

Java → Java Project → Next



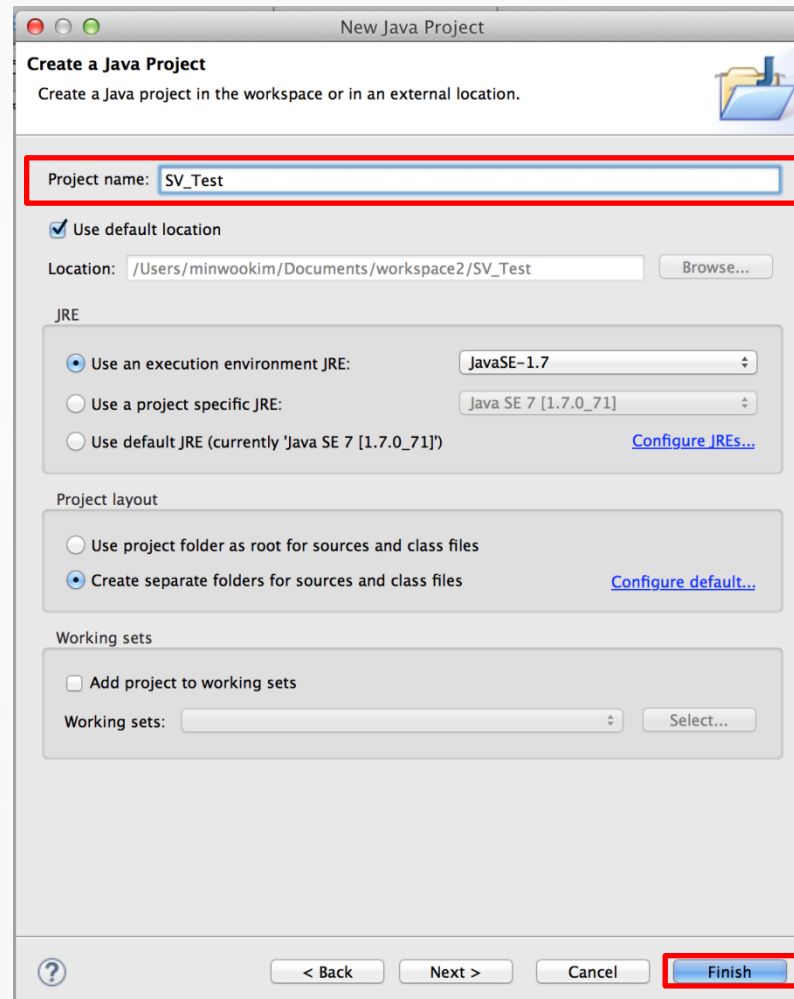
PART 2 Eclipse

How to use Eclipse

3. 프로젝트 만들기

프로젝트 이름 설정

→ Finish



PART 2 Eclipse

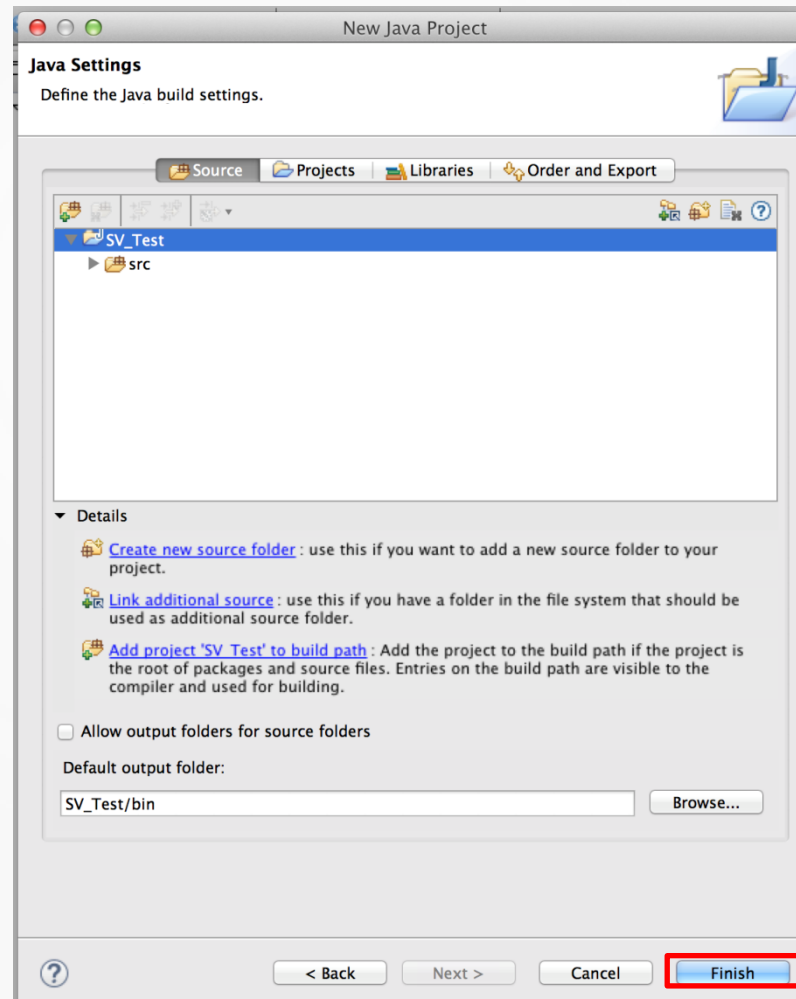
How to use Eclipse

3. 프로젝트 만들기

세부적인 셋팅

library 추가 등등

→ Finish



PART 2 Eclipse

How to use Eclipse

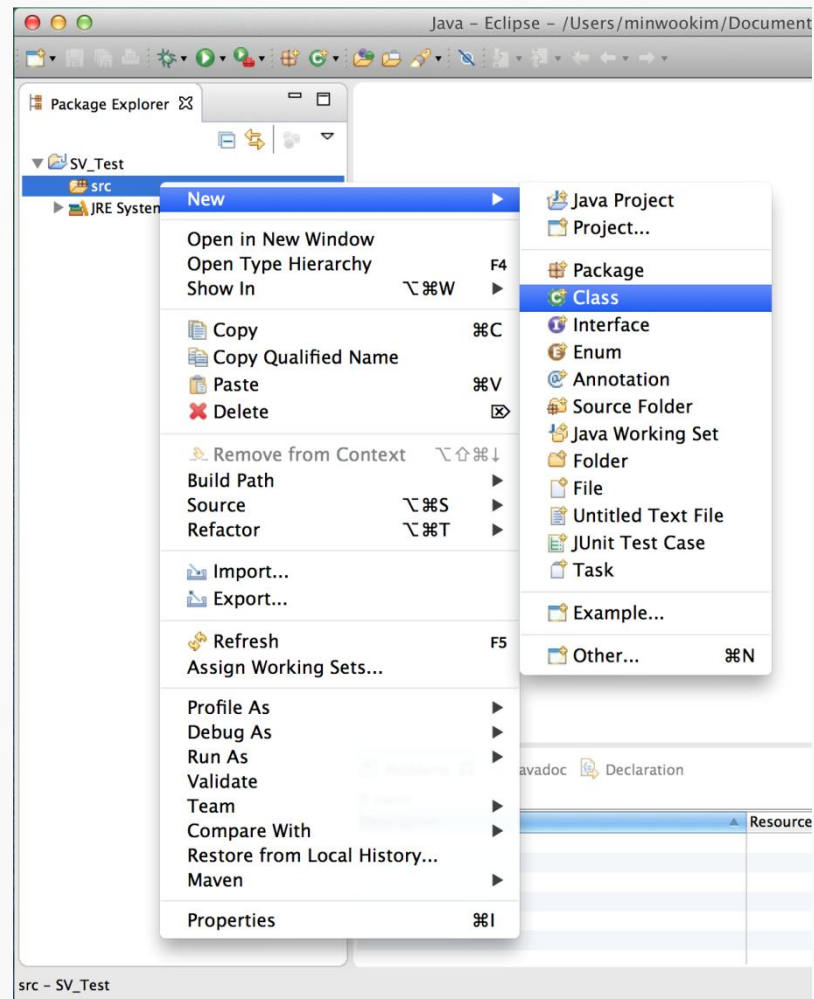
4. Class 만들기

프로젝트 src 폴더

→ 오른쪽 마우스 클릭

→ New

→ Class



PART 2 Eclipse

How to use Eclipse

4. Class 만들기

New Java Class 창

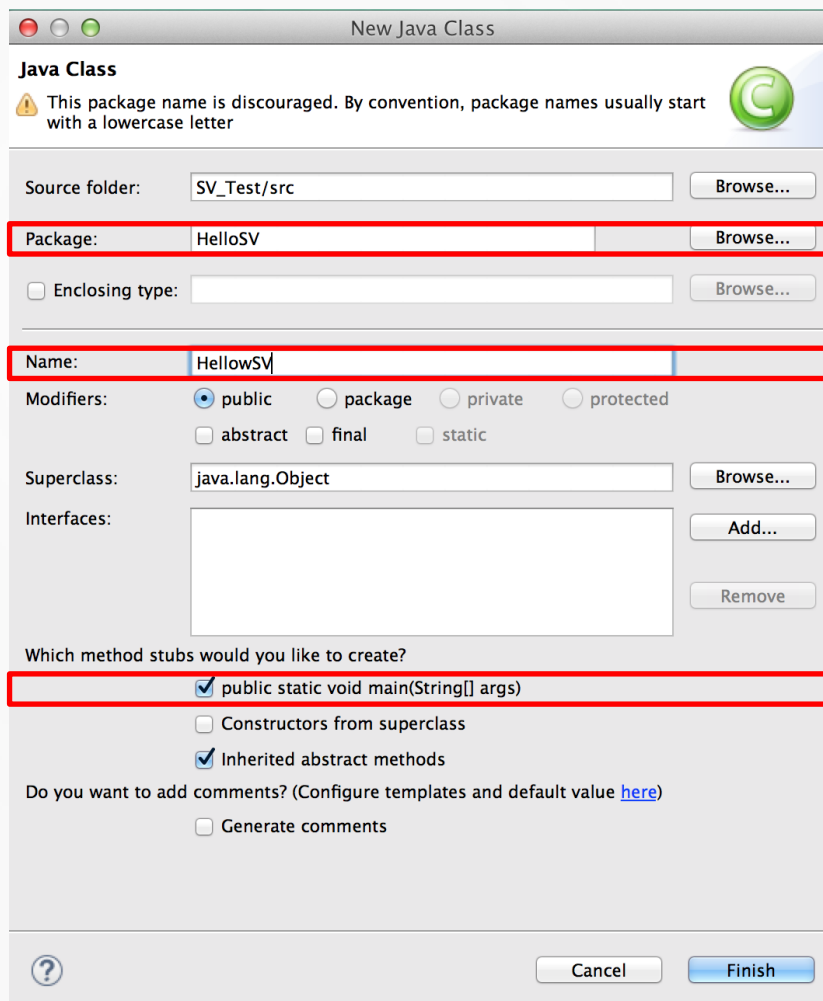
→ Package : 패키지 지정

(필요 시 작성)

→ Name : Class 이름 지정

→ main 있는 Class를 원할

경우, 체크 박스에 체크!

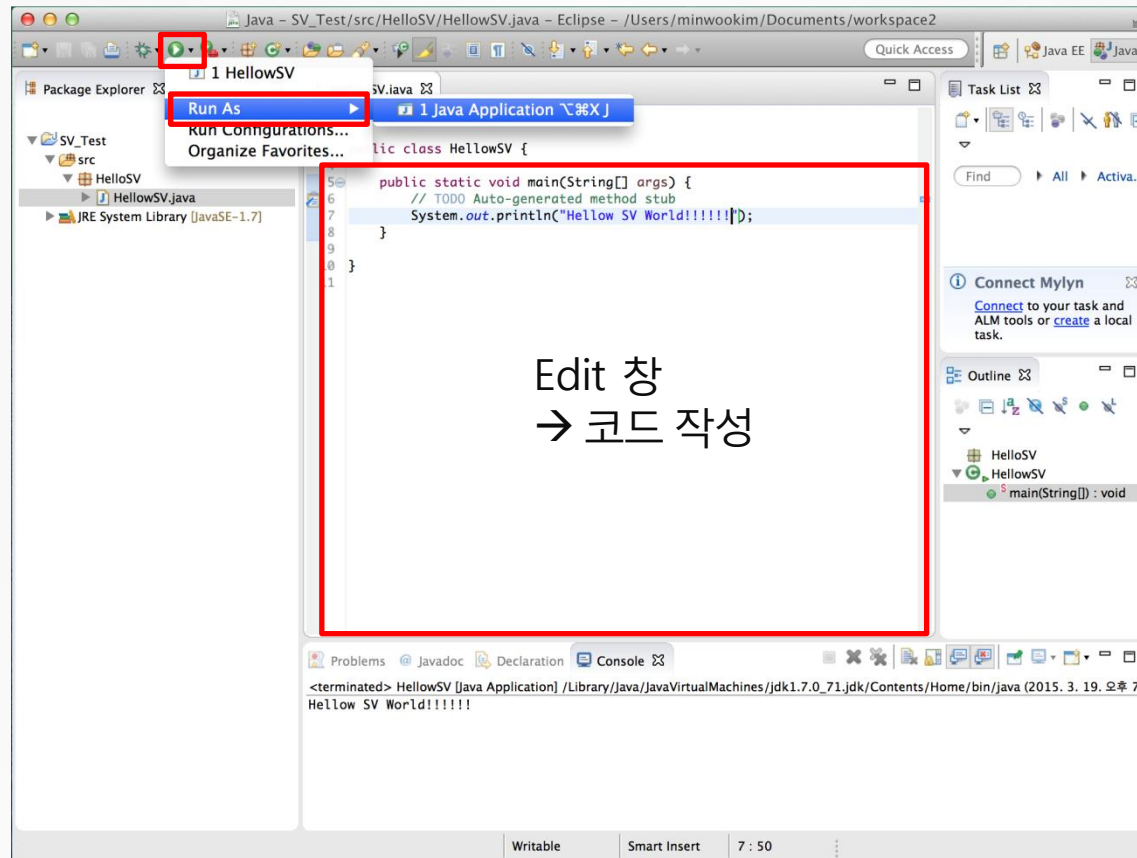


PART 2 Eclipse

How to use Eclipse

5. 프로젝트 작성 및 run

Edit 창에서 코드 작성 → 초록색 동그라미로 된 ▷ 표시 옆 ▼ 클릭 → Run As ...



PART 3

JUnit

PART 3 JUnit

What is JUnit ?



- A **unit testing framework** for the **Java** programming language.

A simple framework to write repeatable tests

An open source framework designed for the purpose of writing and running tests

반복적인 어떤 테스트를 작성하기 위한 간단한 unit testing framework으로, 특별히 자바 프로그래밍에 사용되며, 오픈 소스로 제공된다.

- JUnit has been important in the development **of test-driven development**, and is one of a family of unit testing frameworks which is collectively known as **xUnit** that originated with SUnit.

Smalltalk 기반의 unit testing의 frameworks의 파생된 형태. Test case의 자동화

- JUnit is **linked as a JAR at compile-time**.

- 이번 프로젝트에 대한 설치 및 사용 가이드는 JUnit 버전 4이상을 기준으로 작성했다.

PART 3 JUnit

What is Unit Testing ?

- Unit testing is a **software testing method** by which **individual units of source code**, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are **tested to determine whether they are fit for use**. Intuitively, one can view **a unit as the smallest testable part of an application**.

- Unit

Procedural programming	Entire module, individual function or procedure.
Object-oriented programming	Entire interface (e.g. class), individual method.

PART 3 JUnit

The goal of Unit Testing

- The goal of unit testing is to **isolate each part of the program** and **show that the individual parts are correct**. A unit test provides a strict, written contract that the piece of code must satisfy.

Unit 단위로 나눠서 testing하여 각 Unit을 검증하는 것이 목표.

Unit Testing limitations

- Testing will **not catch every error** in the program, since it cannot evaluate every execution path in any but the most trivial programs.

unit testing by definition only tests the functionality of the units themselves.

→ it will not catch integration errors or broader system-level errors.

모든 기능이나 예외를 다 test 해볼 수 없기 때문에, 100%의 에러를 찾아내지는 못한다.

PART 3 JUnit

Benefits of Unit Testing

- **Finds problems early**

코드 작업을 하면서 간단한 test를 통해, 오류가 난 지점과 내용을 쉽게 찾도록 도와준다.

- **Facilitates change**

각각의 기능과 method에 대한 test case를 나누어, 코드에 변화가 생겼을 때 이로 인한 오류 발생 여부와 지점을 빠르게 찾을 수 있다.

- **Simplifies integration**

bottom-up testing style에 적합하다.

작은 단위부터 점점 test 집합의 크기를 넓혀가며 다양한 test 범위에 대한 작업을 수월하게 해준다.

- Documentation

- Design

- ...

PART 3 JUnit

Benefits of JUnit

- 테스트 검증을 별도의 클래스에서 작성하며 이 테스트 클래스를 실제 소스와 함께 보관할 수 있어 테스트가 체계적으로 관리될 수 있도록 한다.
→ 테스트 결과를 Test클래스로 남김으로 개발자에게 테스트 방법 및 클래스의 History 제공
- 해당 함수에 대한 실행 시간을 표시해 줄 수 있다.

More about JUnit

- JUnit allows the developer to incrementally build test suites to measure progress and detect unintended side effects.
- Tests can be run continuously.
- Results are provided immediately.
- ...

PART 3 JUnit

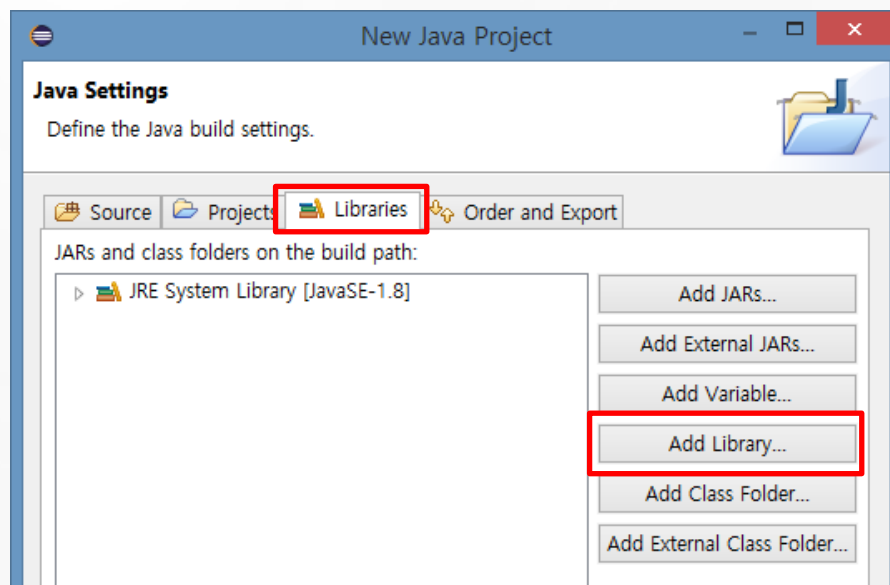
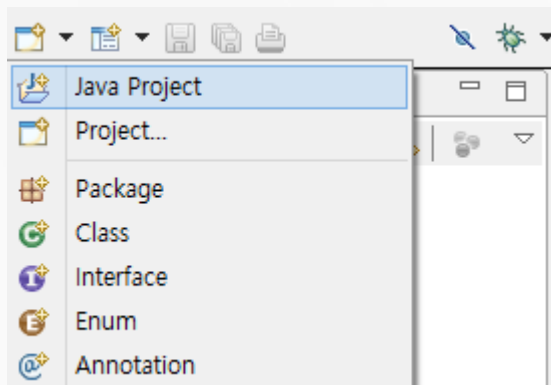
How to install JUnit

- JUnit은 Eclipse 내부에 이미 설치되어 있다. (따로 설치할 필요 없다.)
- 설치 대신, JUnit 라이브러리를 추가한다.

1. 프로젝트 생성 시 라이브러리 추가

새로운 프로젝트 생성 → Next

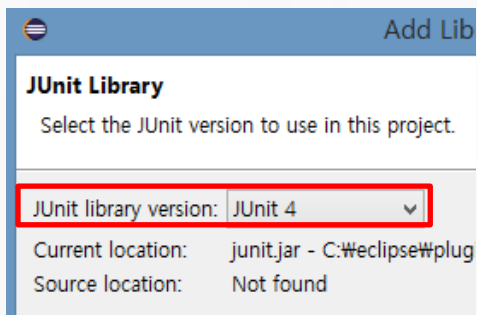
→ Libraries 탭 → Add Library



PART 3 JUnit

How to install JUnit

1. 프로젝트 생성 시 라이브러리 추가 (계속)



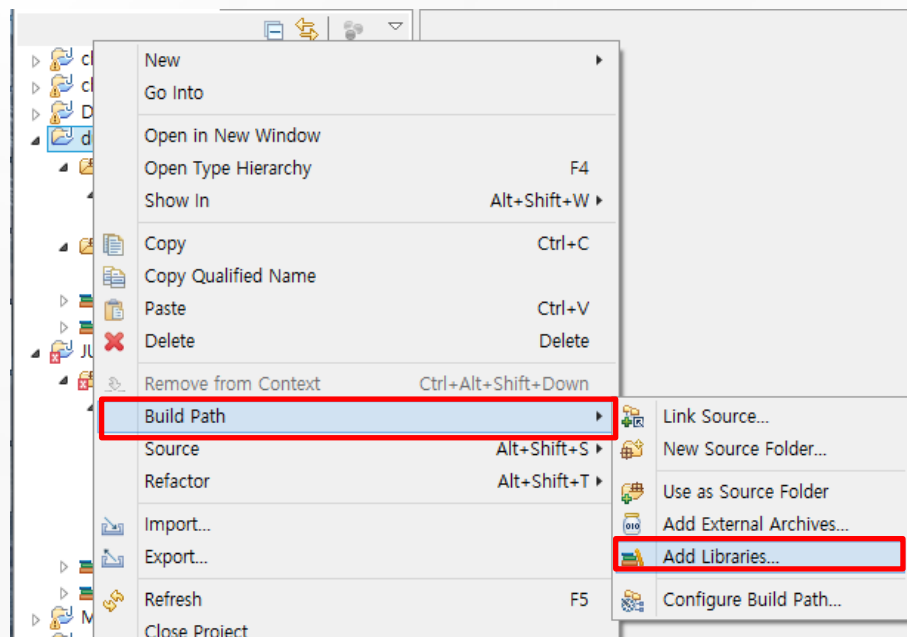
※JDK5.0 이상부터 문법이 바뀐 사항이 있어서 이 이상의 버전부터는 JUnit4를 선택해야 한다. Legacy program을 위해서 JUnit3가 남아있다.

2. 기존 프로젝트에 라이브러리 추가

프로젝트 위에서 오른쪽 마우스

→ Build Path

→ Add Libraries ...



PART 3 JUnit

How to use JUnit

1. 예제 코드

```
/*
 * This class prints the given message on console.
 */
public class MessageUtil {

    private String message;

    //Constructor
    //@param message to be printed
    public MessageUtil(String message){
        this.message = message;
    }

    // prints the message
    public String printMessage(){
        System.out.println(message);
        return message;
    }

    // add "Hi!" to the message
    public String salutationMessage(){
        message = "Hi!" + message;
        System.out.println(message);
        return message;
    }
}
```

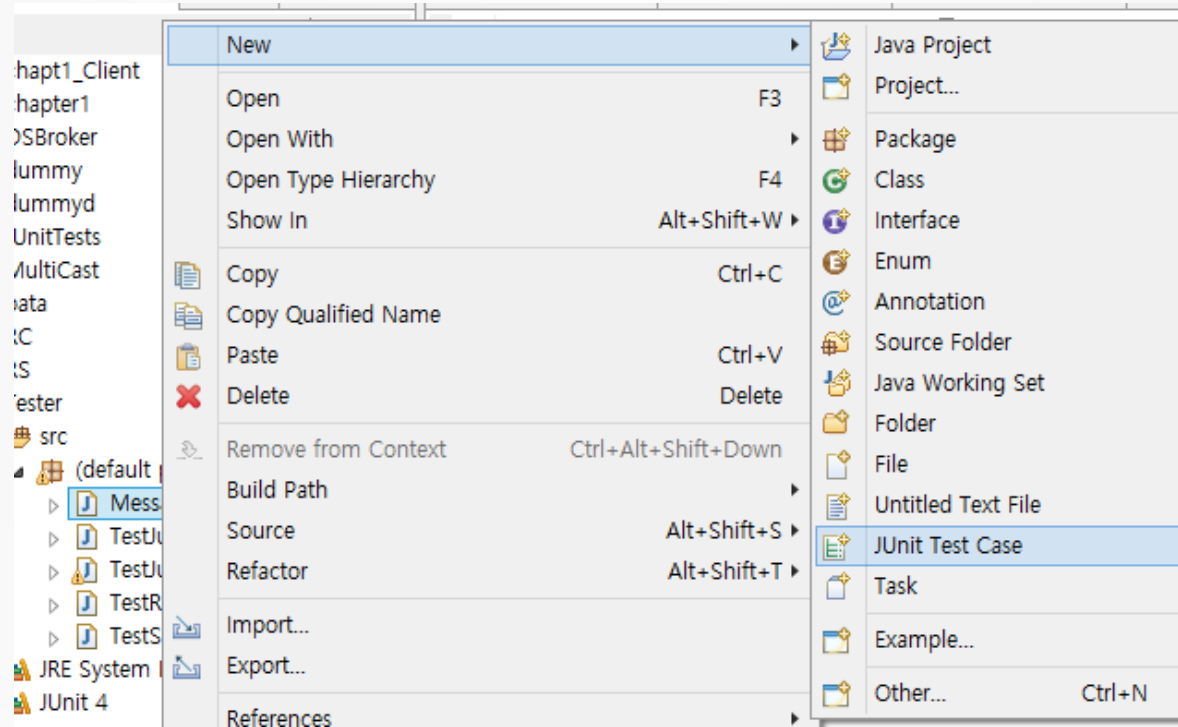
- 예제 코드를 가지고 진행할 Unit testing 과정
 - 1) printMessage()에 대한 test case 생성
 - 2) salutationMessage()에 대한 test case 생성
 - 3) 여러 테스트를 같이 진행하는 testSuite 생성
 - 4) testSuite를 구동하는 runner 실행

PART 3 JUnit

How to use JUnit

2. Test case 생성

Test 하고자 하는 파일에서 오른쪽 마우스 → New → JUnit Test Case

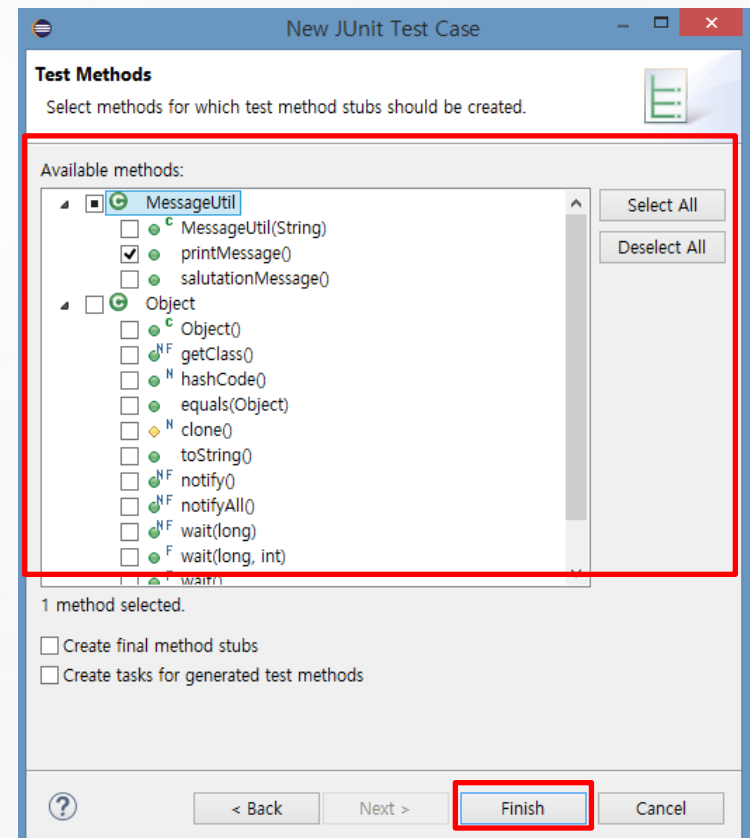
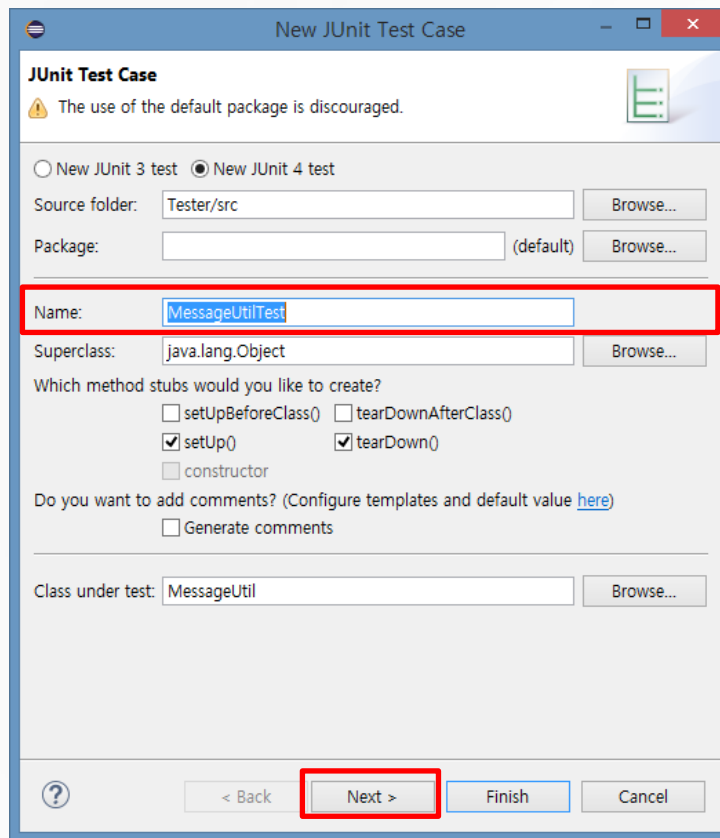


PART 3 JUnit

How to use JUnit

2. Test case 생성

클래스 이름 지정 → Next → Test 하고자 하는 method들을 선택 → Finish



PART 3 JUnit

How to use JUnit

3. Test case 예시

▽ printMessage()에 대한 Test code

```

1 import org.junit.After;
2 import org.junit.Test;
3 import org.junit.Ignore;
4 import org.junit.Before;
5
6 import static org.junit.Assert.assertEquals;
7
8 public class TestJUnit1 {
9
10     String message = "Robert";
11     MessageUtil messageUtil = new MessageUtil(message);
12     @Before
13     public void setUp(){
14         System.out.println("Setup");
15     }
16     @After
17     public void tearDown(){
18         System.out.println("tearDown");
19     }
20     @Test
21     @Ignore
22     public void testPrintMessage2(){
23         System.out.println("이건 테스트만 해요.");
24     }
25     @Test
26     public void testPrintMessage() {
27         System.out.println("Inside testPrintMessage()");
28         assertEquals(message, messageUtil.printMessage());
29     }
30
31
32 }

```

```

1 import org.junit.Test;
2
3
4 public class TestJUnit2 {
5
6     String message = "Robert";
7     MessageUtil messageUtil = new MessageUtil(message);
8
9
10    @Test
11    public void testSalutationMessage() {
12        System.out.println("Inside testSalutationMessage()");
13        message = "Hi!" + "Robert";
14        assertEquals(message, messageUtil.salutationMessage());
15    }
16 }

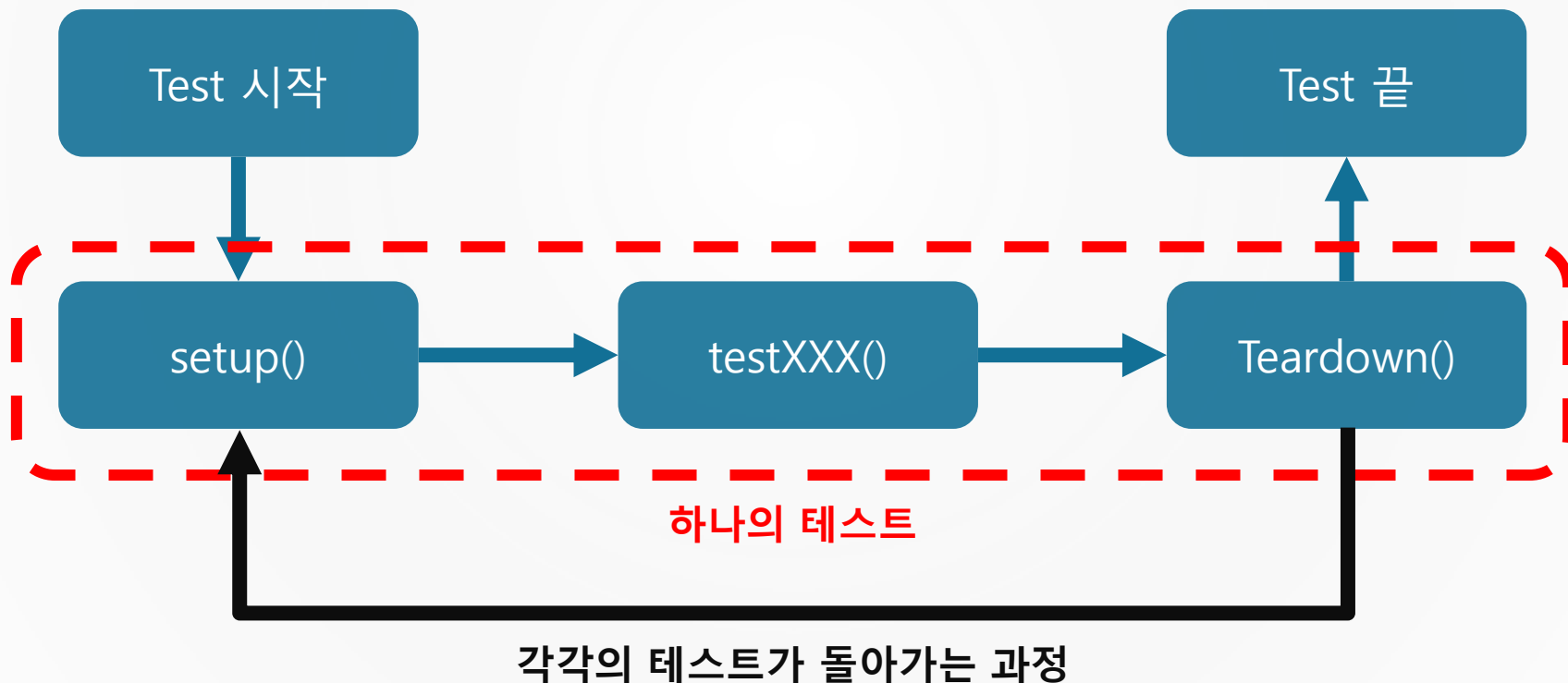
```

△ salutationMessage()에 대한 Test code

PART 3 JUnit

How to use JUnit

※ JUnit Test Case 실행 순서



PART 3 JUnit

How to use JUnit

4. Annotation(@)

JDK5.0 이상부터 적용되는 문법으로, 패키지부터 변수까지 @를 이용하여 메타 데이터 정보를 선언할 수 있는 기능이다.

Annotation	Description
@Test	Test 메소드임을 명시한다.
@Test (expected=Exception.class)	주어진 exception을 throw하지 않을 경우 fail 한다.
@Test(timeout=N)	메소드가 N ms(millisecons) 이상 걸릴 경우 fail 한다.
@Before	각각의 test가 실행되기 전에 먼저 실행되는 메소드이다. Test 환경을 준비하기 위해 사용된다. (예 - read input data, initialize the class)
@After	각각의 test가 실행된 후에 실행되는 메소드이다. Test 환경을 cleanup하기 위해 사용된다. (예 - 임시 데이터 삭제, default로 재정비)
@BeforeClass	전체 test가 시작되기 전에 한 번 실행되는 메소드이다. 이 annotation을 사용할 메소드는 static으로 선언되어야 한다.
@AfterClass	전체 test가 끝난 뒤에 한 번 실행되는 메소드이다. 이 annotation을 사용할 메소드는 static으로 선언되어야 한다.
@Ignore	해당 test 메소드를 무시한다. 코드 변경 후 test case를 아직 변경하지 않았을 때 활용할 수 있다.

PART 3 JUnit

How to use JUnit

4. Annotation(@)

```

@Test
@Ignore("Not working yet.")
public void testPrintMessage2(){
    System.out.println("이건 테스트안 해요.");
}
@Test
public void testPrintMessage() {
    System.out.println("Inside testPrintMessage()");
    assertEquals(message, messageUtil.printMessage());
}

```

@Test를 명시하여 test method를 선언한다.

@Ignore("blabla");
Test를 안 하고 지나간다.

```

public class TestJUnit2 {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test(timeout=500)
    public void testSalutationMessage() {
        System.out.println("Inside testSalutationMessage()");
        message = "Hi!" + "Robert";
        assertEquals(message, messageUtil.salutationMessage());
    }
}

```

@Test(timeout = 500)
해당 method가 500ms를 넘기면 fail

PART 3 JUnit

How to use JUnit

4. Annotation(@)

```
public class TestJUnit1 {
    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);
    @Before
    public void setUp(){
        System.out.println("Setup");
    }
    @After
    public void tearDown(){
        System.out.println("tearDown");
    }
}
```

@Before와 @After는 생성자와 소멸자라고 보면 된다. 생성자는 test에 필요한 클래스의 객체를 생성하거나, 네트워크 연결 등의 역할을 할 수 있다. 소멸자는 반대로 메모리 해제 등의 역할을 한다. 설정을 안 해주면 default로 된다. (아무것도 하지 않음)

Annotation	Description
@RunWith(value=class)	Test runner 임을 명시한다.
@SuiteClasses(value={classes})	Test suites에 포함될 수 있도록 정의된 test임을 명시한다.
@Parameters	다양한 parameters에 대한 multiple test를 제공한다.

PART 3 JUnit

How to use JUnit

5. Assert Method (AssertXXX)

Assert method로 예상 값, sample의 반환 값을 지정한다.

예상 값과 sample의 반환 값이 같으면, test를 통과한다.

```
@Test
public void testPrintMessage() {
    System.out.println("Inside testPrintMessage()");
    assertEquals(message, messageUtil.printMessage());
}
```

```
void org.junit.Assert.assertEquals(Object expected, Object actual)
```

assertEquals

```
public static void assertEquals(Object expected,
                                Object actual)
```

Asserts that two objects are equal. If they are not, an [AssertionError](#) without a message is thrown. If expected and actual are null, they are considered equal.

Parameters:

PART 3 JUnit

How to use JUnit

5. Assert Method (AssertXXX)

결과를 비교하는 여러 method를 제공한다.

```
import static org.junit.Assert.*;
```

Asset Method	Description
assertArrayEquals(a, b);	배열 A와 B가 일치함을 확인한다.
assertEquals(a, b);	객체 A와 B가 일치함을 확인한다.
assertNot Equals(a, b);	객체 A와 B가 불일치함을 확인한다.
assertSame(a, b);	객체 A와 B가 같은 객체임을 확인한다. assertEquals(a, b);는 두 객체의 값이 같은가를 검사하는 데에 반해 assertEquals(a, b);는 두 객체가 동일한가, 즉 하나의 객체인가를 확인한다. (== 연산자)
assertTrue(a);	조건 A가 참인가를 확인한다.
assertNotNull(a);	객체 A가 Null이 아님을 확인한다.

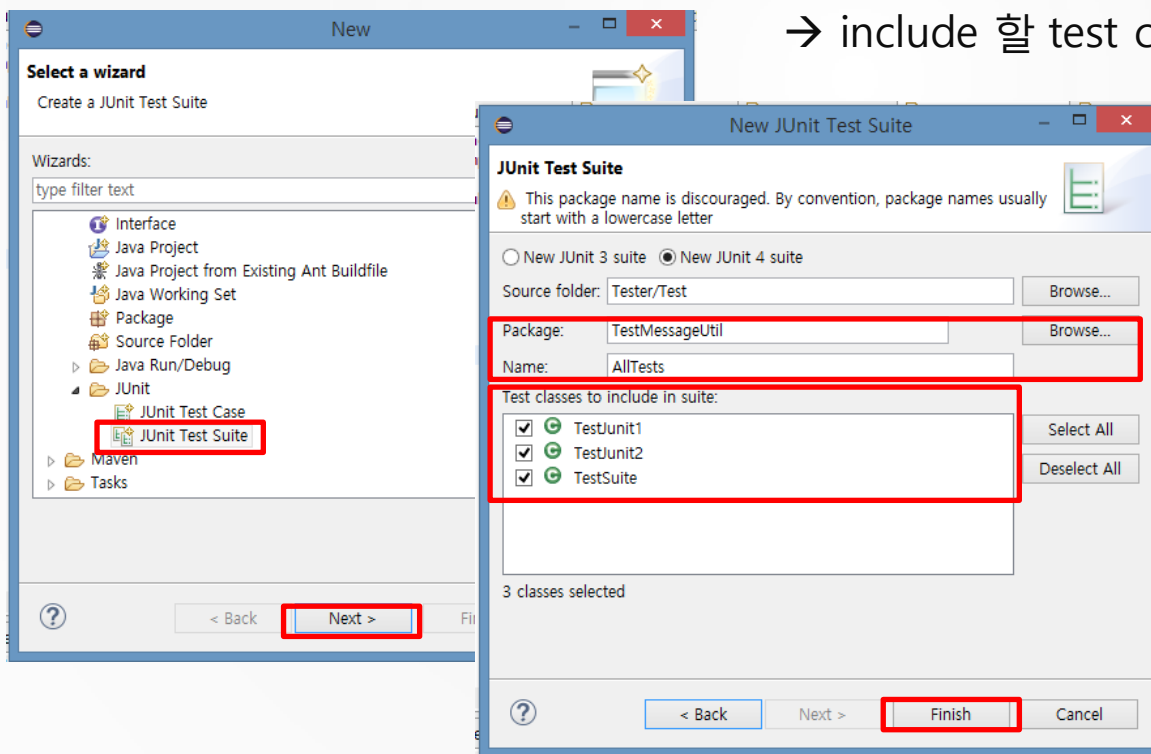
PART 3 JUnit

How to use JUnit

6. TestSuite

여러 단위 test method들을 모아서 한꺼번에 같이 돌려볼 수 있다.

File → New → Others → java → JUnit → JUnit Test Suite → Package, Name 설정
→ include 할 test class 선택 (체크)



```
1 import org.junit.runner.RunWith;
2 import org.junit.runners.Suite;
3 @RunWith(Suite.class)
4 @Suite.SuiteClasses({
5     TestJUnit1.class,
6     TestJUnit2.class
7 })
8 public class TestSuite {
9 }
```

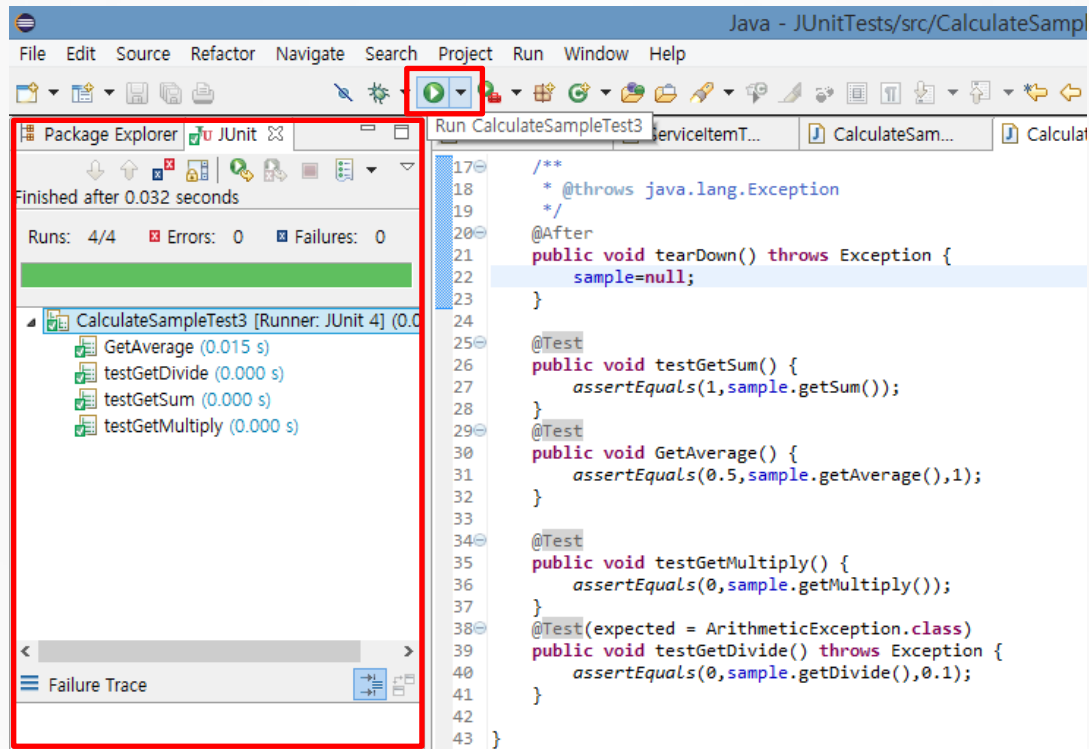
PART 3 JUnit

How to use JUnit

7. Test 실행 및 결과

1) test case 소스에서 run 버튼 클릭

→ JUnit 결과 창 확인 가능 (method 실행 시간, failure 여부 등)



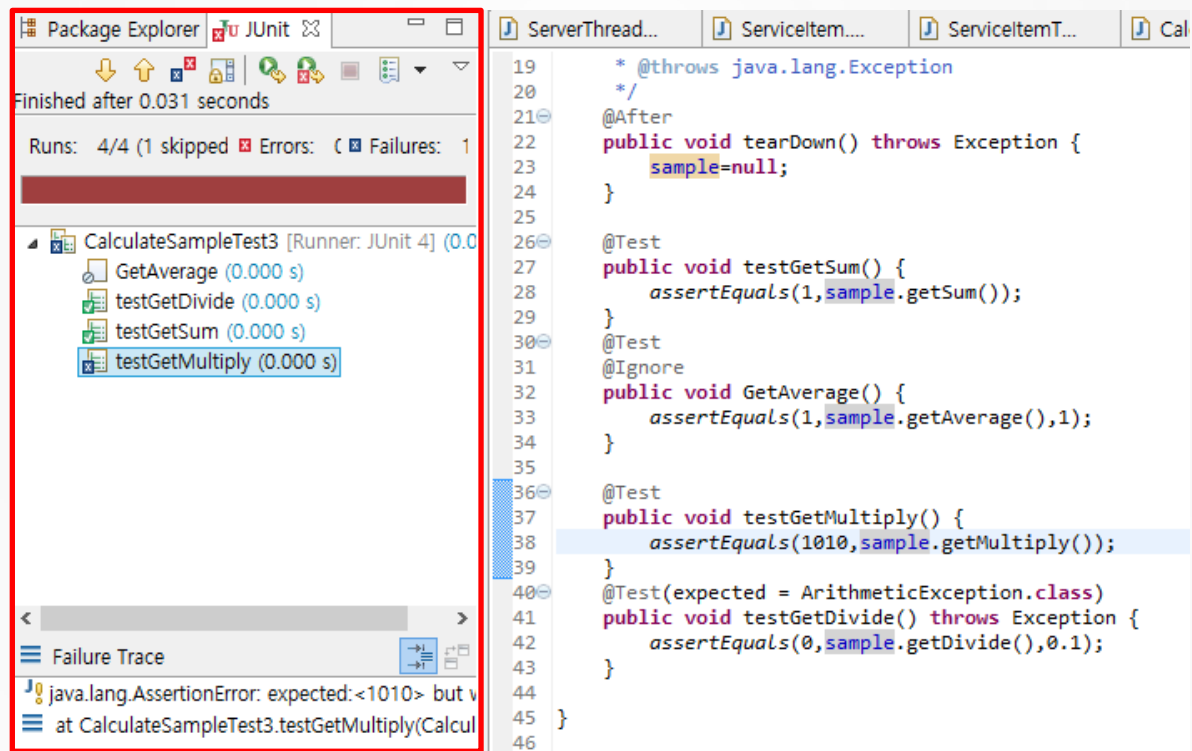
PART 3 JUnit

How to use JUnit

7. Test 실행 및 결과

1) test case 소스에서 run 버튼 클릭

→ fail이 발생할 경우, JUnit 결과 창에서 확인 가능하다.



```
19  * @throws java.lang.Exception
20  */
21  @After
22  public void tearDown() throws Exception {
23      sample=null;
24  }
25
26  @Test
27  public void testGetSum() {
28      assertEquals(1,sample.getSum());
29  }
30  @Test
31  @Ignore
32  public void GetAverage() {
33      assertEquals(1,sample.getAverage(),1);
34  }
35
36  @Test
37  public void testGetMultiply() {
38      assertEquals(1010,sample.getMultiply());
39  }
40  @Test(expected = ArithmeticException.class)
41  public void testGetDivide() throws Exception {
42      assertEquals(0,sample.getDivide(),0.1);
43  }
44
45  }
46
```

Package Explorer JUnit

Finished after 0.031 seconds

Runs: 4/4 (1 skipped Errors: Failures: 1)

CalculateSampleTest3 [Runner: JUnit 4] (0.031 s)

- GetAverage (0.000 s)
- testGetDivide (0.000 s)
- testGetSum (0.000 s)
- testGetMultiply (0.000 s)

Failure Trace

java.lang.AssertionError: expected:<1010> but v
at CalculateSampleTest3.testGetMultiply(Calcul

PART 3 JUnit

How to use JUnit

7. Test 실행 및 결과

2) TestRunner를 이용한 실행

Runner는 @로 된 annotation을 찾아서 test할 것을 찾는다.

```

1 import org.junit.runner.JUnitCore;
2 import org.junit.runner.Result;
3 import org.junit.runner.notification.Failure;
4
5 public class TestRunner {
6     public static void main(String[] args) {
7         Result result = JUnitCore.runClasses(TestSuite.class);
8         for (Failure failure : result.getFailures()) {
9             System.out.println(failure.toString());
10        }
11        System.out.println(result.wasSuccessful());
12    }
13 }

```

```

Setup
Inside testPrintMessage()
Robert
tearDown
Inside testSalutationMessage()
Hi!Robert
true

```

△ 모두 pass 된 경우

```

<terminated> TestRunner (1) [Java Application] C:#Program Files#Java#jre1.8.0_31#bin#javaw.exe (2015. 3. 19.
Inside testPrintMessage()
Robert
tearDown
Inside testSalutationMessage()
Hi!Robert|
testSalutationMessage(Test.TestJUnit2): expected:<Hi![;/]Robert> but was:<Hi![ ]Robert>
false

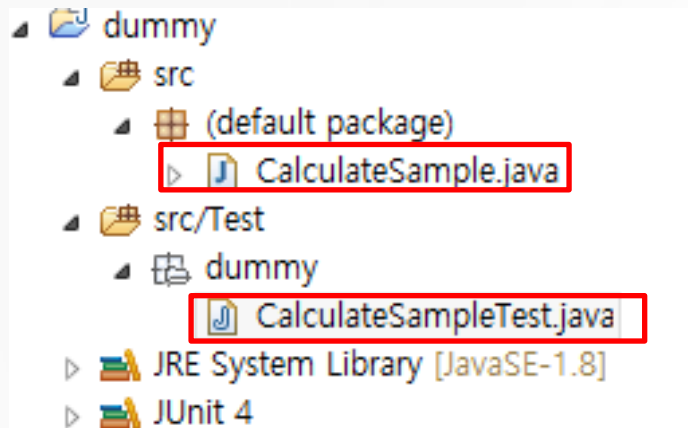
```

△ Fail이 발생한 경우

PART 3 JUnit

How to use JUnit

8. 코드 관리 Tip



컴포넌트 별로 package를 만들고,
해당되는 것들에 대한 test package를 따로 만들어
관리하는 것이 편하다.

※참조 : <http://www.androidpub.com/39606>

PART 4

Build Environment

PART 4 Build Environment

What is Build Environment ?

- 소프트웨어 빌드(software build)는 소스 코드 파일을 컴퓨터에서 실행할 수 있는 독립 소프트웨어 가공물로 변환하는 과정을 말하거나 그에 대한 결과물을 일컫는다.
- 소프트웨어 빌드에 있어 가장 중요한 단계들 가운데 하나는 소스 코드 파일이 실행 코드로 변환되는 컴파일 과정이다.
- 컴퓨터 프로그램을 빌드하는 과정은 보통 다른 프로그램을 제어하는 프로그램인 **빌드 도구**에 의해 관리된다.
컴파일 / 테스트 / 패키징 / 배포 / 문서화
- Ant, Gradle, Maven 등이 있다.

PART 4 Build Environment

Tools for Development

1. 웹 기반 프로젝트 관리 프로그램

- 프로젝트에서 진행되는 이슈사항들 관리, 소스관리, 일정관리 등 프로젝트의 전반적인 진행상황을 모니터링 하고 관리할 수 있는 프로그램이다.
- Redmine, Trac, Jira 등

2. 자동 빌드 프로그램

- 환경파일의 설정에 따라 전체 프로젝트를 쉽게 빌드할 수 있는 환경을 제공하는 툴이다.
- **Ant**, Maven, **Gradle** 등

3. 버전 관리 프로그램

- 하나의 프로그램이 여러 고객의 필요를 맞추려다 보면 다양한 버전의 프로그램으로 변화될 수 있다.
- 효과적으로 소스를 관리할 수 있게 도와주는 툴이 버전 관리 프로그램이다.
- SVN, CVS, GIT, Source Sfae 등

PART 4 Build Environment

What is Ant ?

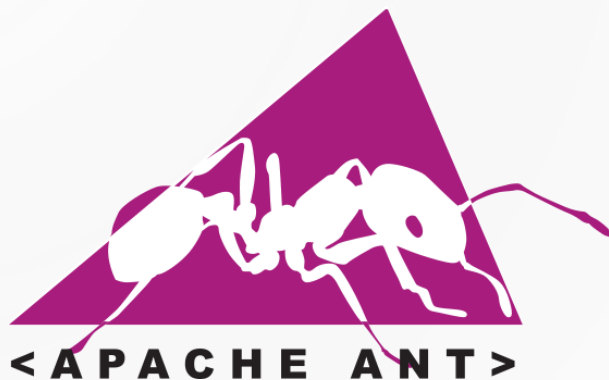
- 자바 기반의 빌드 자동화 도구

Ant는 창시자인 제임스 던컨 데이비슨에 의해 명명되었는데, 이것은 Another Neat Tool을 의미하면서, 영단어 뜻 그대로 개미의 형태와 비교되기도 한다.

- Ant는 그 자체가 Java 기반이면서 xml로 이루어져 있기 때문에 코드 작성과 적용이 쉽고 관리가 편리하다.

- Ant의 주요 기능

- 자바 소스 파일 컴파일
- jar, war, ear, zip 파일의 생성
- Javadoc을 실행하여 도움말 생성
- 파일이나 폴더의 이동 및 복사, 삭제
- 외부 프로그램의 실행
- ...



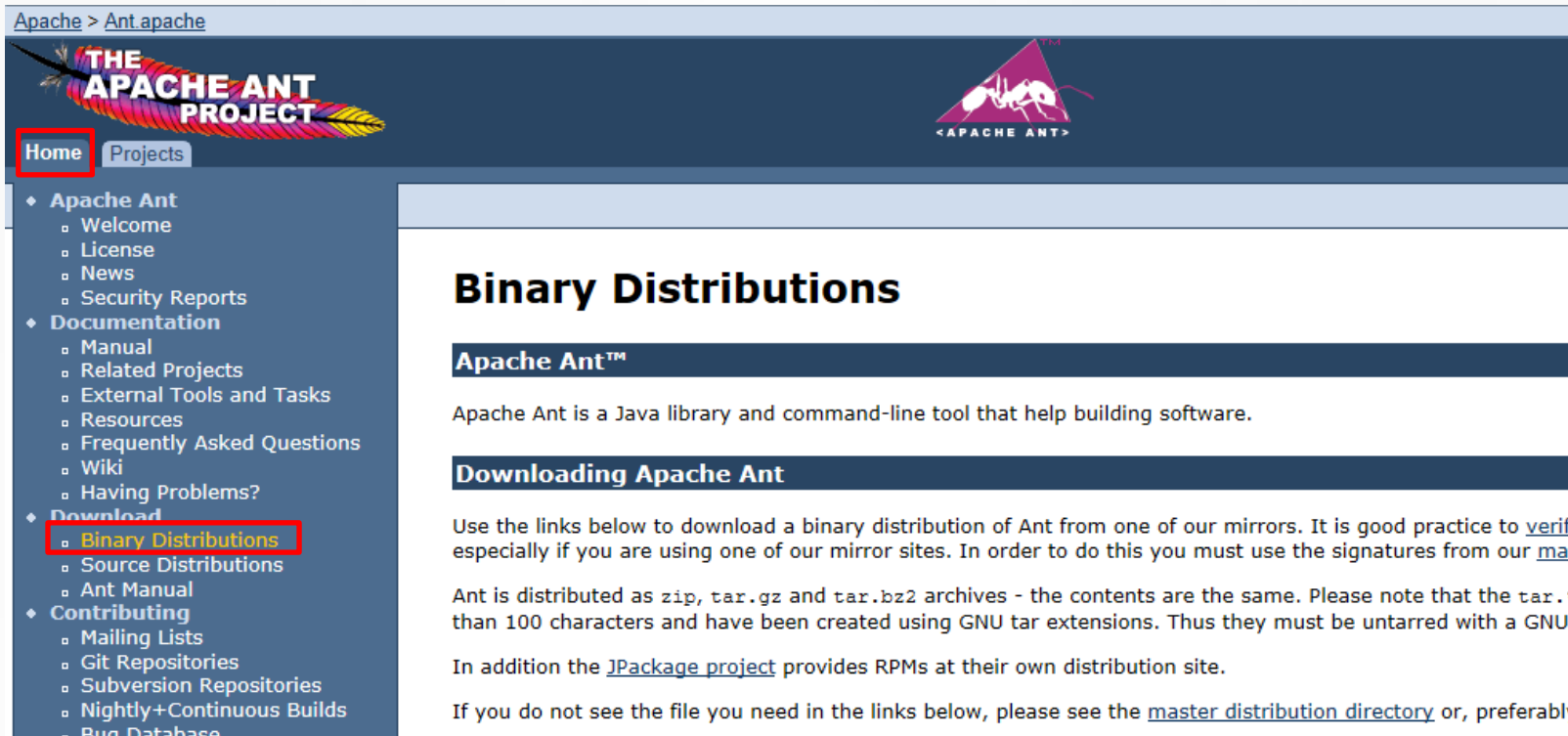
PART 4 Build Environment

How to install Ant

1. Ant 다운로드

Apache ant 홈페이지 접속 : <http://ant.apache.org/bindownload.cgi>

Home 탭 → Binary Distributions 클릭



Apache > Ant apache

THE APACHE ANT PROJECT

Home Projects

- ◆ Apache Ant
 - Welcome
 - License
 - News
 - Security Reports
- ◆ Documentation
 - Manual
 - Related Projects
 - External Tools and Tasks
 - Resources
 - Frequently Asked Questions
 - Wiki
 - Having Problems?
- ◆ Download
 - **Binary Distributions**
 - Source Distributions
 - Ant Manual
- ◆ Contributing
 - Mailing Lists
 - Git Repositories
 - Subversion Repositories
 - Nightly+Continuous Builds
 - Bug Database

Binary Distributions

Apache Ant™

Apache Ant is a Java library and command-line tool that help building software.

Downloading Apache Ant

Use the links below to download a binary distribution of Ant from one of our mirrors. It is good practice to [verify](#) especially if you are using one of our mirror sites. In order to do this you must use the signatures from our [ma](#)

Ant is distributed as zip, tar.gz and tar.bz2 archives - the contents are the same. Please note that the tar. than 100 characters and have been created using GNU tar extensions. Thus they must be untarred with a GNU

In addition the [JPackage project](#) provides RPMs at their own distribution site.

If you do not see the file you need in the links below, please see the [master distribution directory](#) or, preferabl

PART 4 Build Environment

How to install Ant

1. Ant 다운로드

Current Release of Ant

Currently, Apache Ant 1.9.4 is the best available version, see the [release notes](#).

Note

Ant 1.9.4 was released on 05-May-2014 and may not be available on all mirrors for a few days.

Tar files may require gnu tar to extract

Tar files in the distribution contain long file names, and may require gnu tar to do the extraction.

- .zip archive: [apache-ant-1.9.4-bin.zip](#) [PGP] [SHA1] [SHA512] [MD5]
- .tar.gz archive: [apache-ant-1.9.4-bin.tar.gz](#) [PGP] [SHA1] [SHA512] [MD5]
- .tar.bz2 archive: [apache-ant-1.9.4-bin.tar.bz2](#) [PGP] [SHA1] [SHA512] [MD5]

▽ /Users/{Username}/Documents에 압축해제 디렉토리를 가져다 놓으면 사실상 설치는 끝

▶	자구	2015. 3. 6. 오후 9:06
▶	apache-ant-1.9.4	오늘 오후 5:46
▶	Aptana Rubles	2015. 3. 12. 오후 3:27

PART 4 Build Environment

How to install Ant

2. 환경 변수 설정

1) Mac의 경우

.profile에 윈도우에 환경변수 설정하듯이 bin까지 경로를 미리 저장시킨다. 만일 처음 시도했다면 .profile 은 없으니, "vim .profile"로 만든다.

```
baghyeongyuui-MacBook-Air-2:~ canine$ ls -al
total 56
drwxr-xr-x+ 19 canine  staff    646  3 18 18:51 .
drwxr-xr-x   6 root    admin   204  3 18 09:17 ..
-rw-----   1 canine  staff     4  2 10 17:39 .CFUserTextEncoding
-rw-r--r--@  1 canine  staff  15364  3 18 18:56 .DS_Store
drwx-----   2 canine  staff    68  3 18 17:06 .Trash
-rw-----   1 canine  staff  1081  3 18 18:49 .bash_history
drwx-----   3 canine  staff   102  2 27 15:09 .cups
drwxr-xr-x   3 canine  staff   102  3 12 16:01 .eclipse
-rw-r--r--   1 canine  staff    93  3 18 18:51 .profile
```

▽.자세히 보면 window에서 환경변수 설정하는 것과 별반 다르지 않다.

```
export ANT_HOME=/Users/canine/Documents/apache-ant-1.9.4
export PATH=${PATH}:${ANT_HOME}/bin
~
```

PART 4 Build Environment

How to install Ant

2. 환경 변수 설정

1) Mac의 경우

```
baghyeongyuui-MacBook-Air-2:~ canine$ source ~/.profile
```

△ 위의 명령어로 환경변수 설정을 적용한다.

▽ 아래의 메시지를 확인하면 환경변수 설정 종료.

```
Apache Ant(TM) version 1.9.4 compiled on April 29 2014  
baghyeongyuui-MacBook-Air-2:~ canine$
```

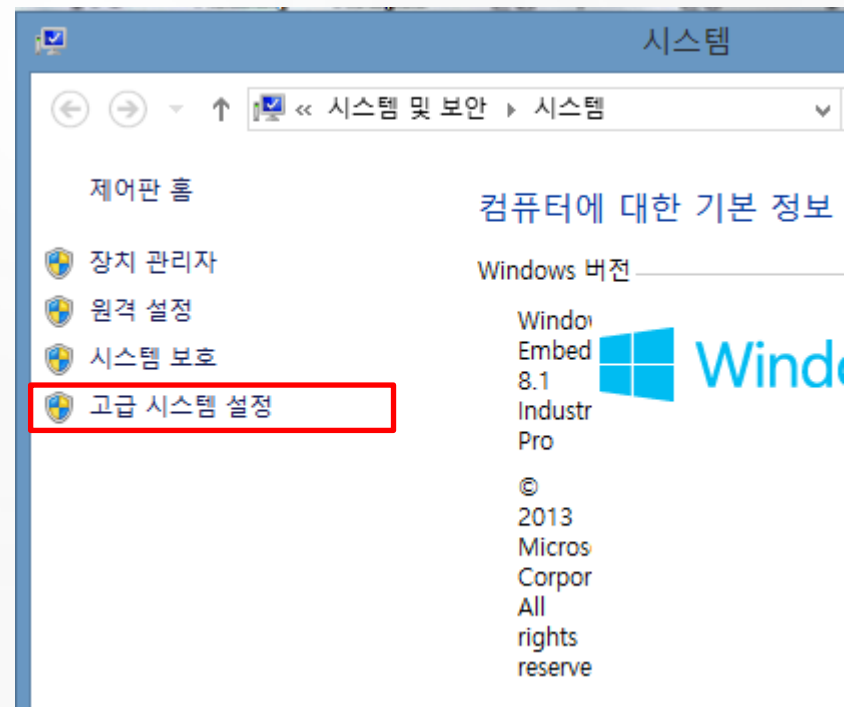
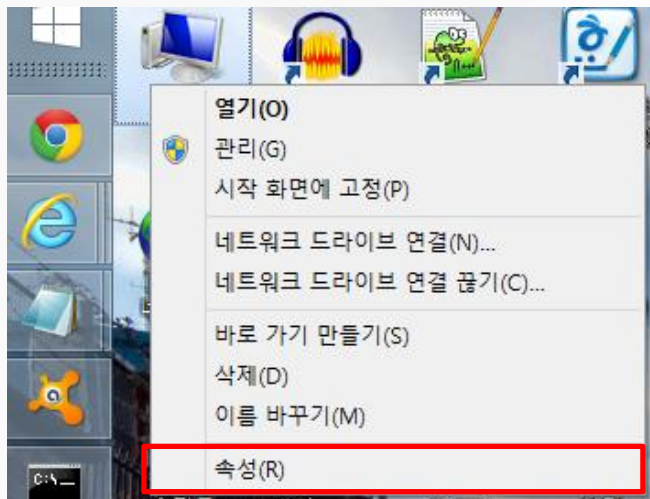
PART 4 Build Environment

How to install Ant

2. 환경 변수 설정

2) Windows의 경우

내컴퓨터 → 오른쪽 마우스 클릭 → 속성 → 고급 시스템 설정



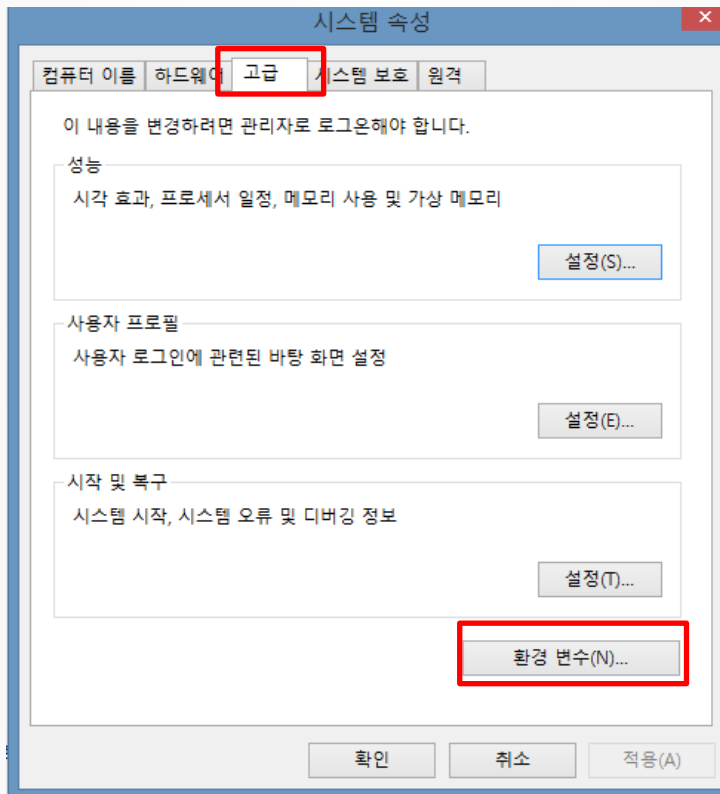
PART 4 Build Environment

How to install Ant

2. 환경 변수 설정

2) Windows의 경우

시스템 속성 → 고급 탭 → 환경 변수 클릭



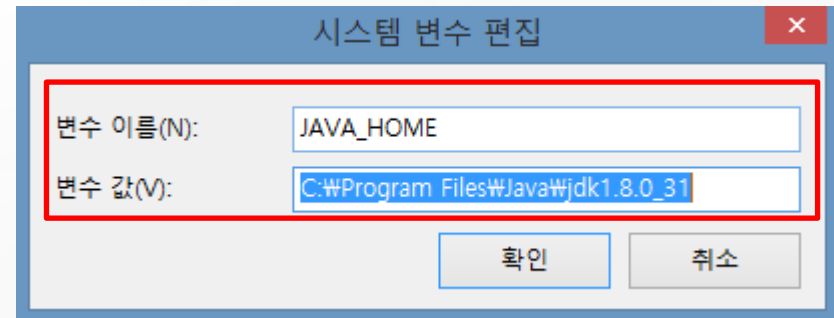
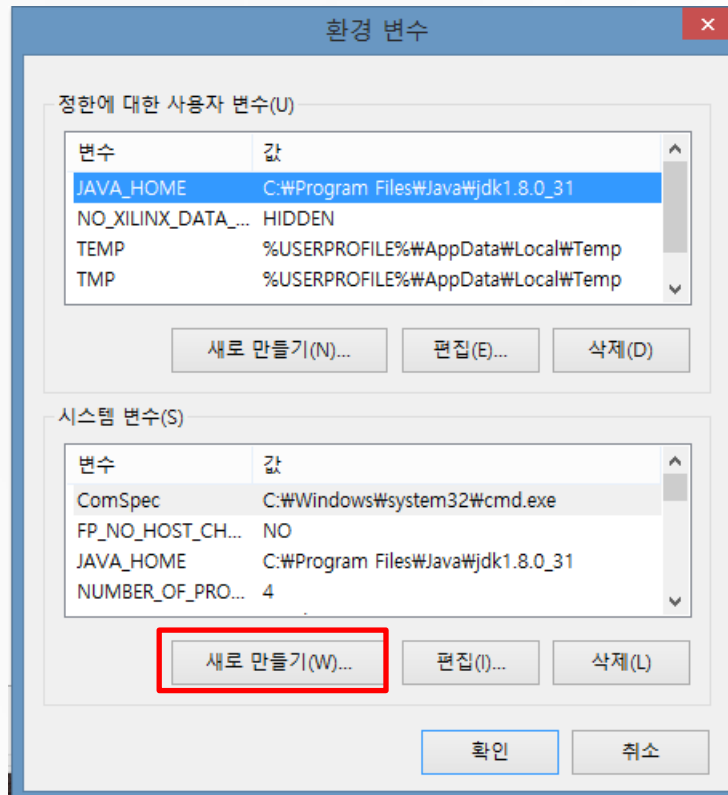
PART 4 Build Environment

How to install Ant

2. 환경 변수 설정

2) Windows의 경우

시스템 변수 새로 만들기 → JAVA_HOME (JDK 설치 directory) 생성



△ 현재 시스템에 설치된 JDK 추가.

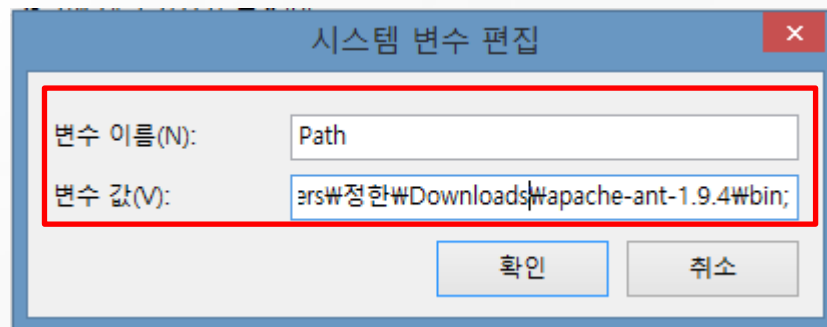
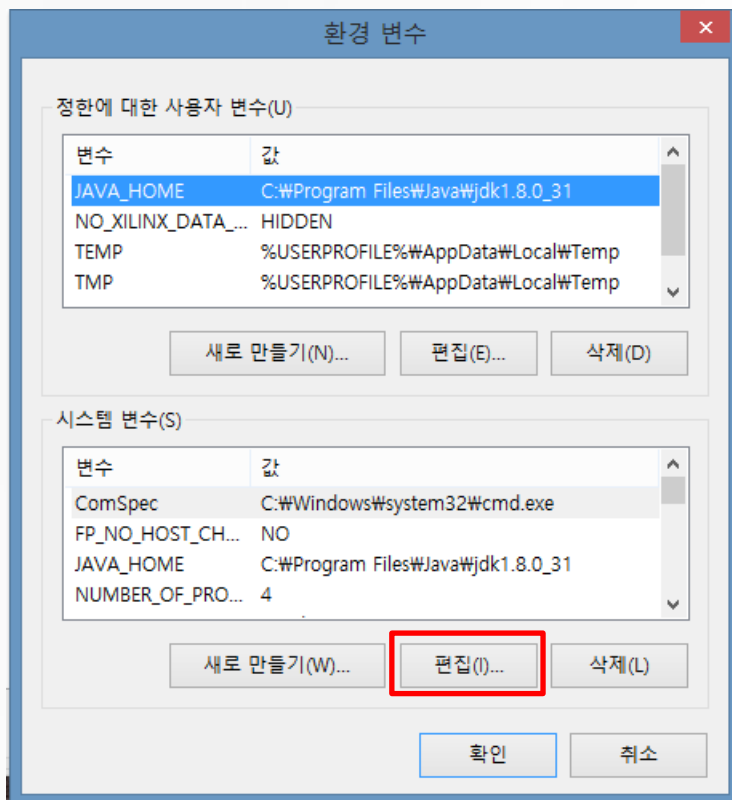
PART 4 Build Environment

How to install Ant

2. 환경 변수 설정

2) Windows의 경우

시스템 변수 편집 → Path에 Ant 설치 경로/bin 추가



△ Ant {설치 경로}/bin을 가장 뒤에 복사해서 붙인다.
각각의 path들은 ; (세미콜론)으로 구분된다.

※ 끝에 ';' 붙이기

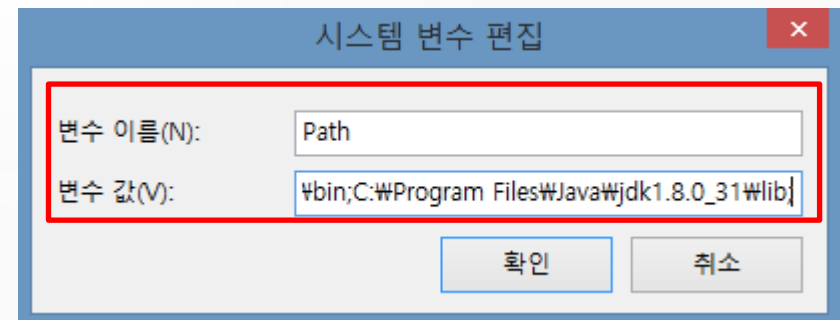
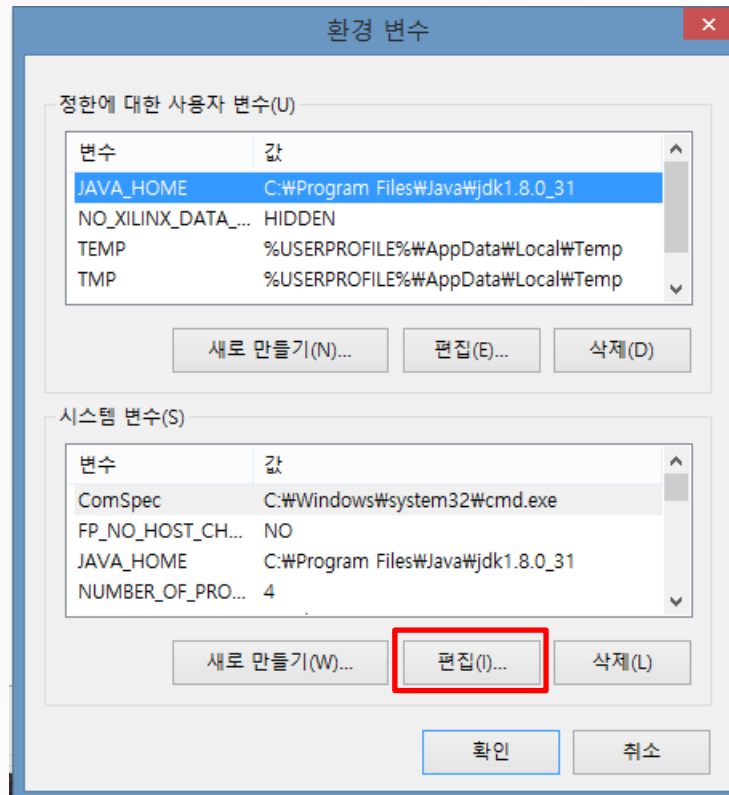
PART 4 Build Environment

How to install Ant

2. 환경 변수 설정

2) Windows의 경우

Tools.jar가 없다는 오류가 발생할 경우 → 시스템 변수 Path에 {JDK설치경로}/bin 추가

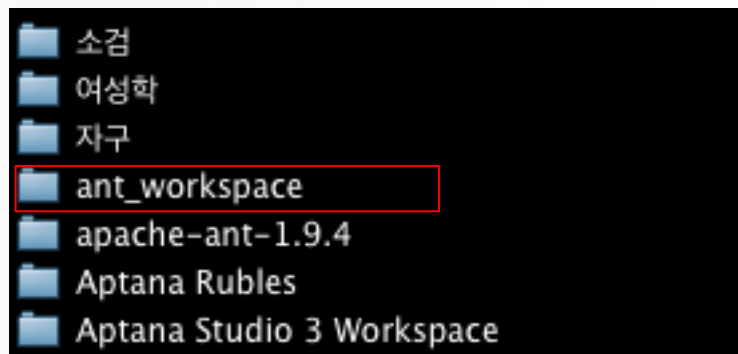


PART 4 Build Environment

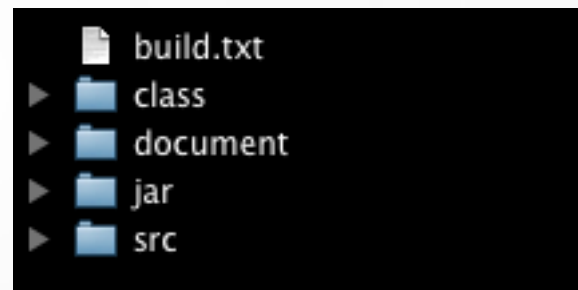
How to install Ant

3. 디렉토리 생성

임의의 폴더를 생성하여, 그 내부에 다음과 같은 파일 및 폴더들을 생성해 준다.



△ 임의의 폴더 생성



△ 파일 및 폴더 생성

Build.xml : build 설정 파일
(build 할 때마다 설정 변경 가능)

Src : 소스 폴더

Class : 컴파일한 클래스 파일 폴더

Jar : 배포판 폴더

Document: document 파일 폴더

PART 4 Build Environment

How to install Ant

4. Build.xml 파일 작성

붉은 박스 내부의 내용만 수정하면 된다.

```
<?xml version="1.0" encoding="UTF-8" ?>  
  
<project name="ant_test" default="compile" basedir="/Users/canine/Documents/ant" >  
  <property name="src.dir" value="${basedir}/src" />  
  <property name="classes.dir" value="${basedir}/class" />  
  <property name="jar.dir" value="${basedir}/jar" />  
  <property name="javadoc.dir" value="${basedir}/document" />  
</project>
```

1) Default="compile"

"default : 초기치로 설정되는 작업, 아무 것도 지정하지 않고 실행하면 이 작업을 수행"
해당 지정 타겟까지 작업을 수행해주는 것이다.

2) Basedir : "프로젝트의 경로"

3) \${basedir}/src, \${basedir}/class... : "자신이 직접 구성한 폴더들의 이름"

PART 4 Build Environment

How to install Ant

4. Build.xml 파일 작성

```

<target name="compile" >
  <javac srcdir="${src.dir}" destdir="${classes.dir}" />
</target>

<target name="jar" depends="compile" >
  <jar jarfile="${jar.dir}/hello.jar"
    basedir="${classes.dir}"
    includes="**/*.class"
  />
</target>

<target name="javadoc" depends="compile, jar" >
  <javadoc
    sourcepath="${src.dir}"
    sourcefiles="${src.dir}/ant_test.java"
    windowtitle="helloant"
    destdir="${javadoc.dir}"
  />
</target>

<target name="clean" >
  <delete dir="${classes.dir}*" />
  <delete dir="${jar.dir}*" />
  <delete dir="${javadoc.dir}*" />
</target>

```

<target>...</target> 내부는 해당 xml을 파싱했을때 하는 기능들이라고 생각하면 된다.

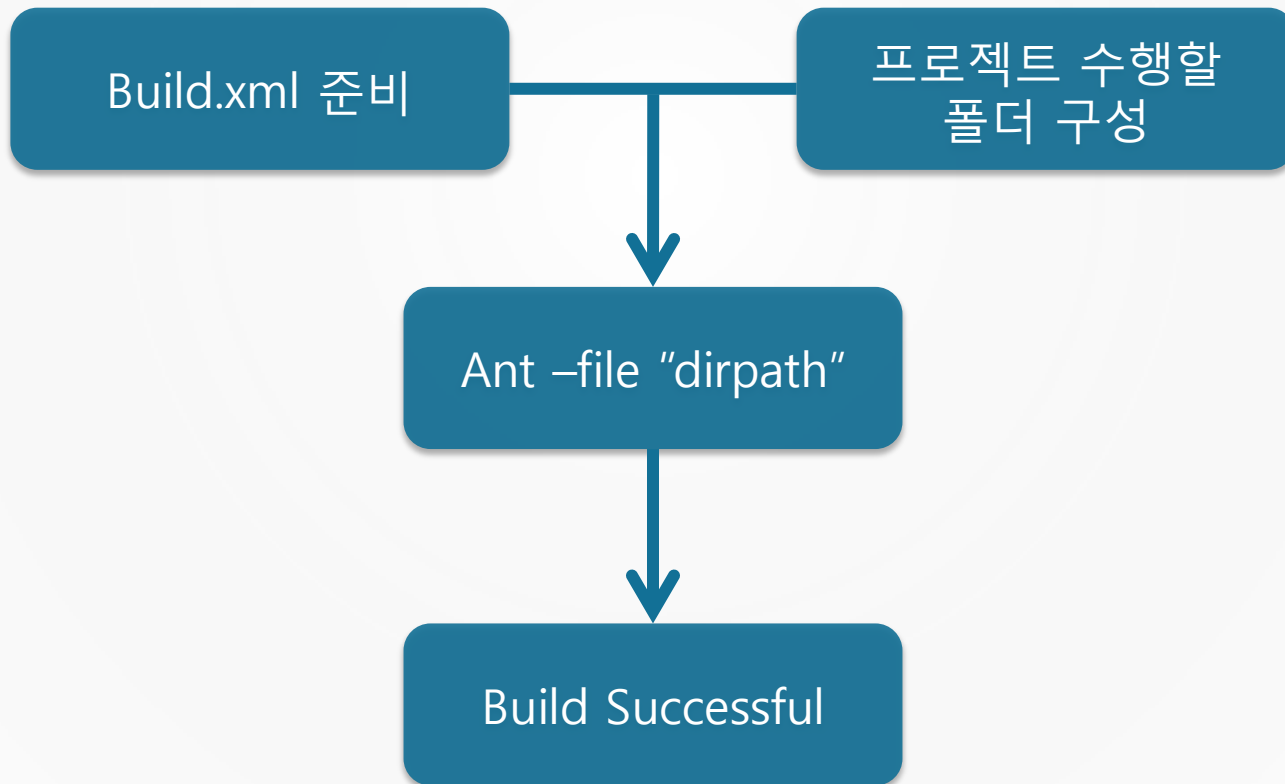
Jar부분에서 자신이 원하는 이름으로 jar파일을 생성할 수 있고,

Javadoc부분에서 자신이 원하는 java파일을 설정하여 컴파일 할 수 있다.

PART 4 Build Environment

How to use Ant

1. Command 창에서 Ant 실행하기



PART 4 Build Environment

How to use Ant

1. Command 창에서 Ant 실행하기

프로젝트 폴더로 이동 → build.xml 있는지 확인 후 → "ant" → .class 파일 생성 !

```
gimmin-uu-MacBook-Pro:SV_Test minwookim$ ant
Buildfile: /Users/minwookim/Documents/workspace2/SV_Test/build.xml

build-subprojects:

init:

build-project:
    [echo] SV_Test: /Users/minwookim/Documents/workspace2/SV_Test/build.xml

build:

BUILD SUCCESSFUL
Total time: 0 seconds
```

△ ant -file {프로젝트 경로} 로도 실행이 가능하다.

PART 4 Build Environment

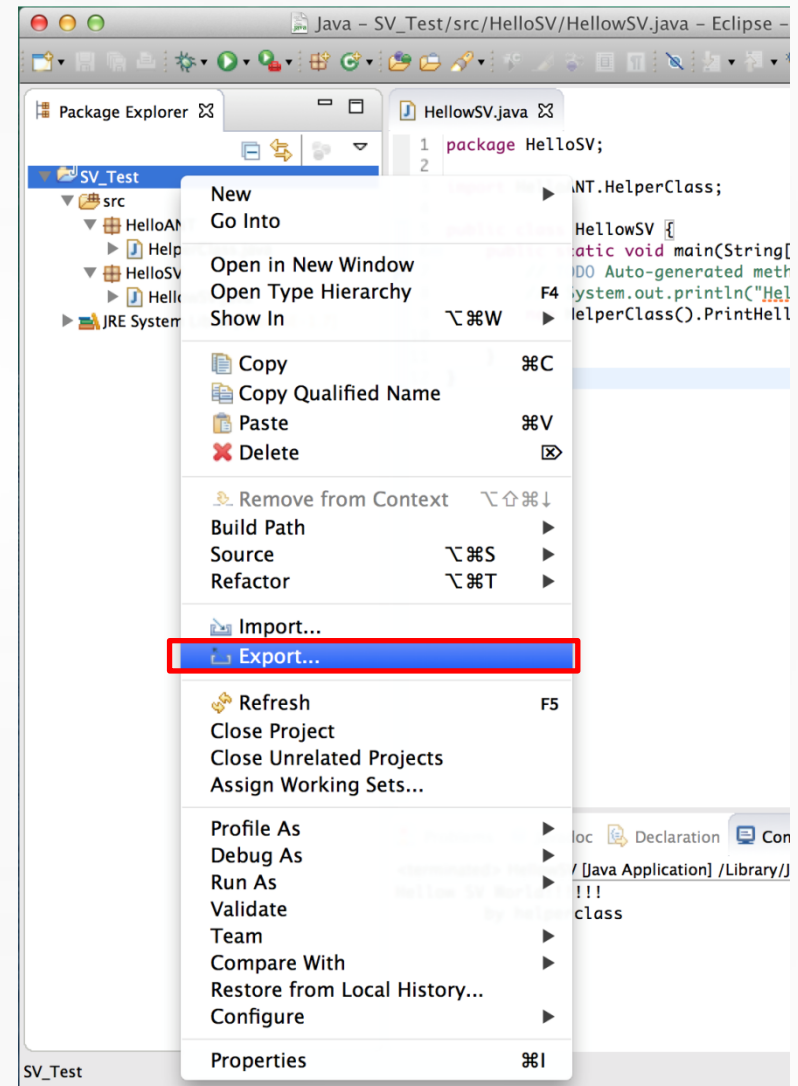
How to use Ant

2. Eclipse에서 Ant 실행하기

ant 빌드를 위한 build.xml 뽑아내기

프로젝트 위에서 오른쪽 마우스

→ Export

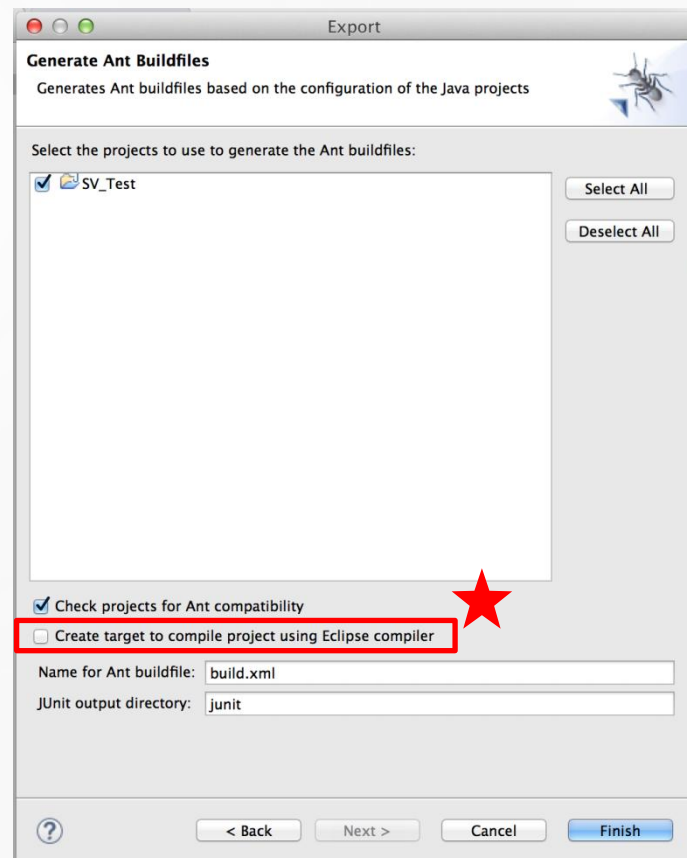
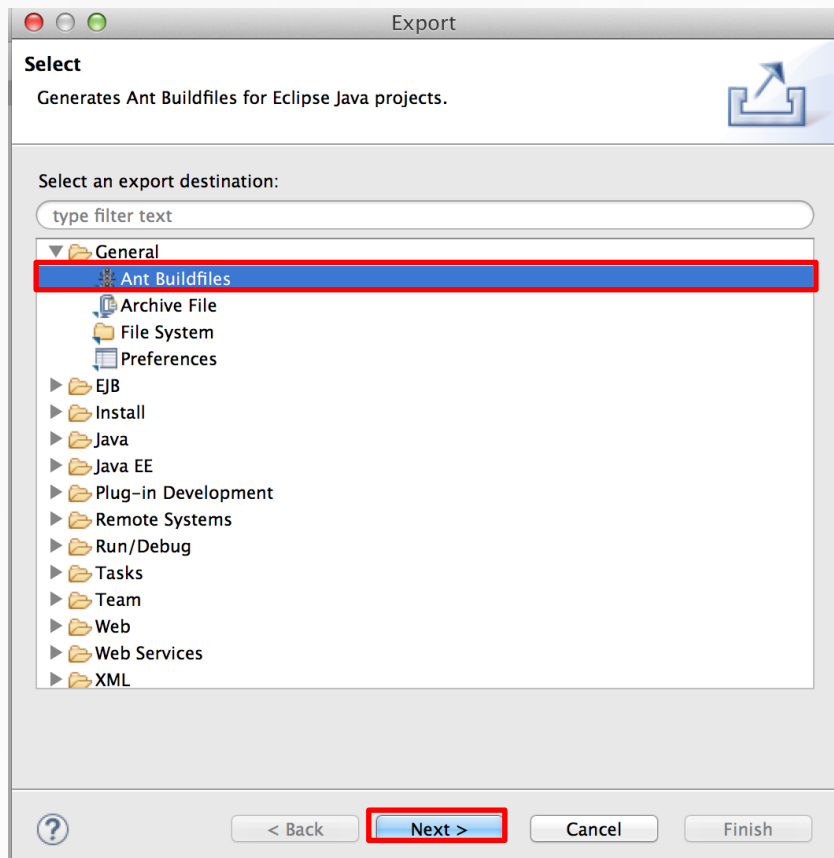


PART 4 Build Environment

How to use Ant

2. Eclipse에서 Ant 실행하기

General → Ant Buildfiles → Next → 아래 체크박스 해제 (Eclipse의 컴파일 사용 안함)

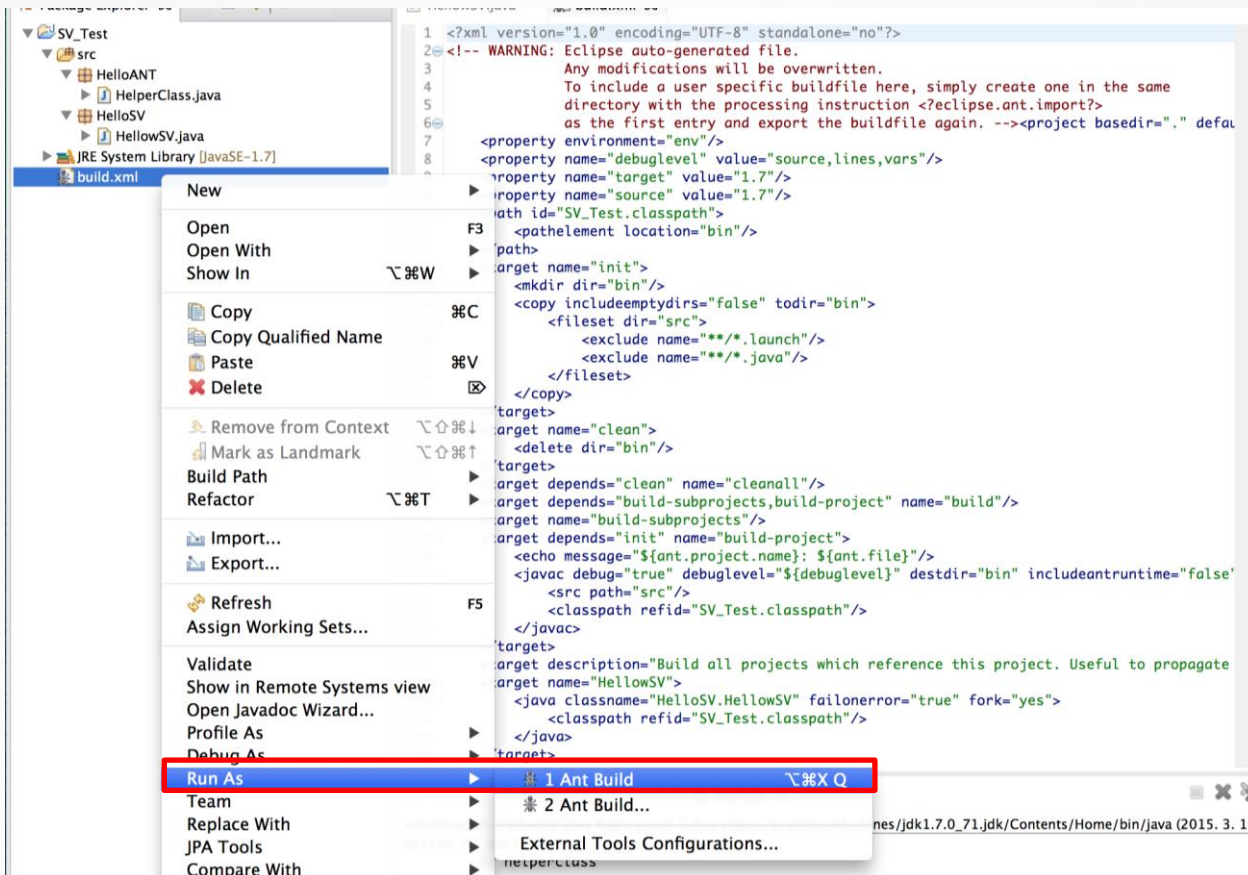


PART 4 Build Environment

How to use Ant

2. Eclipse에서 Ant 실행하기

build.xml → 오른쪽 마우스 → Run As → Ant Build

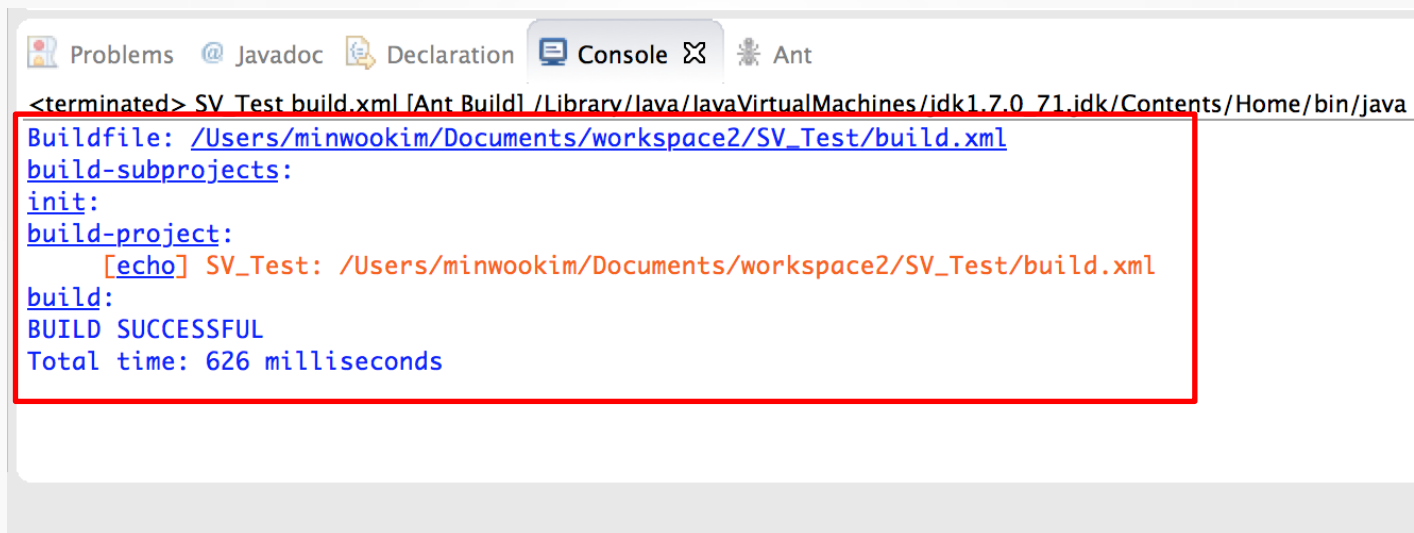


PART 4 Build Environment

How to use Ant

2. Eclipse에서 Ant 실행하기

build한 결과



The screenshot shows the Eclipse IDE's console window with the following output:

```
<terminated> SV_Test build.xml [Ant Build] /Library/Java/JavaVirtualMachines/jdk1.7.0_71.jdk/Contents/Home/bin/java
Buildfile: /Users/minwookim/Documents/workspace2/SV_Test/build.xml
build-subprojects:
init:
build-project:
  [echo] SV_Test: /Users/minwookim/Documents/workspace2/SV_Test/build.xml
build:
BUILD SUCCESSFUL
Total time: 626 milliseconds
```

PART 4 Build Environment

What is Gradle ?



- 프로젝트 구성, 빌드 도구

Ant와 Maven 이후에 나온 빌드 도구로, 차세대 빌드 툴로 각광받고 있다.

- Groovy 언어 기반

xml을 기반으로 하는 Ant나 Maven과는 달리, Groovy라는 언어를 기반으로 한다.

What is Maven ?



- Ant와 비슷하나 프로젝트 관리를 위한 컨셉으로 만들어진 프로젝트 관리툴

- Pom.xml에 프로젝트 구조와 동작 정의

- Maven의 단점

다중 프로젝트가 될 수록 상속관계로 설정해야 한다.

XML로 빌드파일을 작성하는데, 뼈대구조가 복잡하다.

사용자간의 플러그인이 다르면 일일이 다시 기술해야 한다.

PART 4 Build Environment

Gradle vs Ant(Maven)

- build 파일의 구조와 Gradle 비교

```
task hello {
    doLast {
        println 'Hello world!'
    }
}
```

```
<target name="jar" depends="compile" >
    <jar jarfile="${jar.dir}/hello.jar"
        basedir="${classes.dir}"
        includes="**/*.class"
    />
</target>
```

△ Build 파일을 열어서 확인해보면 task는 target과 대응이 되며, 하는 기능은 거의 비슷하다.

Task란, 미리 빌드 과정에서 해야 할 일들을 적어둔 것이다.

예를 들어, 자바 프로젝트를 빌드 한다면
Source Compile, 단위테스트, Jar생성 등을 수행해야 하는데,
이를 Task를 따라서 그대로 자동 빌드 할 수 있다는 것이다.
→ 빌드의 자동화

```
task hello << {
    println 'Hello Earth'
}
hello.doFirst {
    println 'Hello Venus'
}
hello.doLast {
    println 'Hello Mars'
}
hello << {
    println 'Hello Jupiter'
}
```

오른쪽 캡처는 Task의 예시 ▷

PART 4 Build Environment

Gradle vs Ant(Maven)

- build 파일의 구조와 Gradle 비교

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

VS

```
dependencies() {
    testCompile 'junit:junit:3.81'
}
```

dependency를 하나 선언할 때마다 xml 파일에서는 최소 **6줄 이상**을 작성해야 한다. 그러나 groovy에서는 단 **3줄**이면 dependency 선언이 가능하다.

추가하는 특징이 많아질 수록 복잡성의 차이는 더 커진다.

PART 4 Build Environment

How to install Gradle

1. Gradle 다운로드

Gradle 홈페이지 접속 : <http://www.gradle.org/downloads>

Complete distribution

DOWNLOAD GRADLE

The latest release of Gradle is version **2.3**,
released on *16th February 2015*.

Complete distribution (binaries, sources
and offline documentation)

Binary only distribution (no
documentation or source code)

Gradle source code (just the Gradle
source code; not a usable Gradle
installation)

PART 4 Build Environment

How to install Gradle

2. 압축 해제 후 환경 변수 추가

Mac 기준으로 작성했으나, 환경 변수 추가 이외에 과정은 Windows와 같다.

▶ Aptana Studio 3 Workspace	어제 오후 7:39
▶ eclipse	어제 오후 9:00
▶ gradle-2.3	2015. 2. 16. 오전 6:16
▶ Microsoft User Data	2015. 3. 10. 오후 7:30

△ 압축 해제

```
export GRADLE_HOME=/Users/canine/Documents/gradle-2.3
export PATH=${PATH}:${GRADLE_HOME}/bin
```

△Windows의 경우에는 시스템 환경 변수 Path에 Gradle 설치 경로 추가

```
baghyeongyuui-MacBook-Air-2:~ canine$ source ~/.profile
```

PART 4 Build Environment

How to install Gradle

3. 버전 체크

2015/03/19 기준으로 2.3버전이 최신이다.

```
gradle -v
```

```
baghyeongyuui-MacBook-Air-2:~ canine$ gradle -v
-----
Gradle 2.3
-----

Build time:   2015-02-16 05:09:33 UTC
Build number: none
Revision:     586be72bf6e3df1ee7676d1f2a3afd9157341274

Groovy:       2.3.9
Ant:          Apache Ant(TM) version 1.9.3 compiled on December 23 2013
JVM:          1.6.0_65 (Apple Inc. 20.65-b04-462)
OS:           Mac OS X 10.9.5 x86_64
```

PART 4 Build Environment

How to install Gradle

4. 프로젝트 initiate

```
gradle init --type groovy-library
```

```
baghyeongyuui-MacBook-Air-2:testgradle canine$ gradle init --type groovy-library
:wrapper
:init

BUILD SUCCESSFUL

Total time: 3.651 secs
baghyeongyuui-MacBook-Air-2:testgradle canine$ ls
build.gradle      gradlew           settings.gradle
gradle            gradlew.bat      src
```

PART 4 Build Environment

How to use Gradle

- Build.gradle

```
apply plugin: 'groovy' Plugin 사용 가능 : task 내부 수행 설정
```

```
// In this section you declare where to find the dependencies of your project
```

```
repositories {  
    // Use 'jcenter' for resolving your dependencies.  
    // You can declare any Maven/Ivy/file repository here.  
    jcenter()  
}
```

자바 프로젝트의 경우,
외부 jar 파일이 해당 프로
젝트 빌드에 필요한 경우
추가

```
// In this section you declare the dependencies for your production and test code
```

```
dependencies {  
    // We use the latest groovy 2.x version for building this library  
    compile 'org.codehaus.groovy:groovy-all:2.4.0'  
  
    // We use the awesome Spock testing and specification framework  
    testCompile 'org.spockframework:spock-core:0.7-groovy-2.0'  
    testCompile 'junit:junit:4.12'  
}
```

PART 4 Build Environment

How to use Gradle

- Build.gradle

```
dependencies {  
    compile group: 'org.hibernate', name: 'hibernate-core', version:  
}  
// 짧게 쓰면 "group:name:version"  
dependencies {  
    compile 'org.hibernate:hibernate-core:3.6.7.Final'  
}
```

△ 기초적인 dependency의 예

일반 파일로 저장된 외부 라이브러리도 지원한다. 빌드 스크립트에서 직접 의존성을 지정한다.

```
// In this section you declare the dependencies for your production and test code
```

```
dependencies {  
    // We use the latest groovy 2.x version for building this library  
    compile 'org.codehaus.groovy:groovy-all:2.4.0'  
  
    // We use the awesome Spock testing and specification framework  
    testCompile 'org.spockframework:spock-core:0.7-groovy-2.0'  
    testCompile 'junit:junit:4.12'  
}
```

✖ References

Downloads

- <http://www.oracle.com/index.html> : JDK 다운로드
- <https://eclipse.org/downloads/> : Eclipse 다운로드
- <http://ant.apache.org/bindownload.cgi> : Ant 다운로드
- <http://www.gradle.org/downloads> : Gradle 다운로드

References

- Introduction to CTIP-15SV.pdf
- "자바 개발자도 쉽고 즐겁게 배우는 테스트 이야기" / 이상민 지음 / 한빛미디어
- http://en.wikipedia.org/wiki/Continuous_integration
- http://en.wikipedia.org/wiki/Unit_testing
- <http://www.vogella.com/tutorials/JUnit/article.html>
- <http://www.androidpub.com/39606>
- <http://donzbox.tistory.com/384>
- <http://dynaticy.tistory.com/1>
- <http://blog.naver.com/kittenjun/10165921021>
- <http://dongsilove.tistory.com/entry/junit-test>
- ...

THANK YOU
ANY QUESTIONS?