

객체지향 프로그램 – Java

(1일 자바란?)

kimth4110@gmail.com

김 태 환

일정

교육일정	교육내용
20일	<ul style="list-style-type: none">• java 언어의 특징• java + eclipse 설치 및 사용법• java 예제 프로그램
21일	<ul style="list-style-type: none">• 데이터형
22일	<ul style="list-style-type: none">• 제어문
23일	<ul style="list-style-type: none">• 반복문
24일	<ul style="list-style-type: none">• 배열
27일	<ul style="list-style-type: none">• OOP 개념
28일	<ul style="list-style-type: none">• 클래스 기능 및 속성 (메소드, 캡슐화)
29일	<ul style="list-style-type: none">• 상속
30일	<ul style="list-style-type: none">• 다형성과 추상 클래스, 인터페이스
31일	<ul style="list-style-type: none">• 패키지

수업 목표

■ 프로그래밍 언어 Java



프로그래밍 언어 Java

■ 자바의 개요

- 자바 언어 : 오크(Oak)라는 언어로부터 탄생
- 오크 언어 : 1991년 미국 선(Sun) 마이크로시스템사의 컴퓨터 과학자인 제임스 고슬링에 의해 개발
 - 가전제품의 기능을 프로그램으로 제공

프로그래밍 언어 Java

■ 자바 언어

- 오크 언어의 많은 개념들을 그대로 이어 받음
- C/C++에서 어렵게 사용되는 포인터나 메모리 조작등을 과감하게 제거
- 신뢰성을 증대시키기 위해 예외 처리 (exception-handling) 기능을 효율적으로 제공

프로그래밍 언어 Java

- 자바의 사용 : 3가지 종류의 플랫폼 제공
 - Java 2 ME(Mobile Edition) : PDA나 스마트폰 등 소형 기기를 위한 개발 환경
 - Java 2 SE(Standard Edition) : 클라이언트 중심의 일반적인 자바 응용 프로그램 개발 환경
 - Java 2EE(Enterprise Edition) : 서버 중심의 기업용 소프트웨어 개발 환경

프로그래밍 언어 Java

■ 현재까지 가장 많이 사용되는 언어

표 1-1 tiobe사에서 발표하는 프로그래밍 언어 사용 순위(2016년 10월)

Oct 2016	Oct 2015	Change	Programming Language	Ratings	Change
1	1		Java	18.799%	-0.74%
2	2		C	9.835%	-6.35%
3	3		C++	5.797%	+0.05%
4	4		C#	4.367%	-0.46%
5	5		Python	3.775%	-0.74%
6	8	↑	JavaScript	2.751%	+0.46%
7	6	↓	PHP	2.741%	+0.18%
8	7	↓	Visual Basic,.NET	2.660%	+0.20%
9	9		Perl	2.495%	+0.25%
10	14	↑↑	Objective-C	2.263%	+0.84%
11	12	↑	Assmby language	2.232%	+0.66%
12	15	↑	Swift	2.004%	+0.73%
13	10	↓	Ruby	2.001%	+0.18%
14	13	↓	Visual Basic	1.987%	+0.47%
15	11	↓↓	Delphi/Object Pascal	1.875%	+0.24%
16	65	↑↑	Go	1.809%	+1.67%

프로그래밍 언어 Java

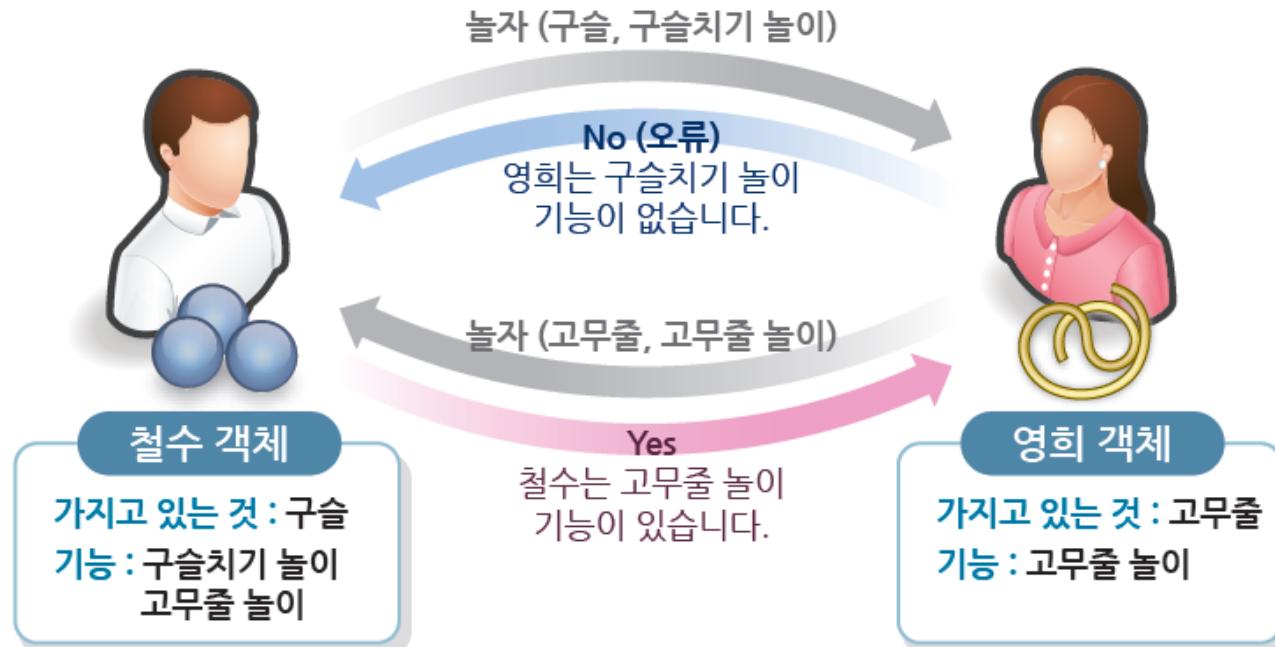
■ 자바 언어의 특징

- 객체지향언어 (Object Oriented Language)
 - 자바는 완전한 객체지향 언어
 - 객체지향의 특성인 클래스, 상속, 캡슐화, 다형성 등의 개념이 잘 적용된 언어
 - 객체지향 프로그래밍은 우리가 살아가는 실 세계와 동일한 사고 방식의 프로그램
 - 자바 언어는 가장 쉽게 이해할 수 있으며, 빨리 배울 수 있는 프로그래밍 언어

프로그래밍 언어 Java

■ 자바 언어의 특징

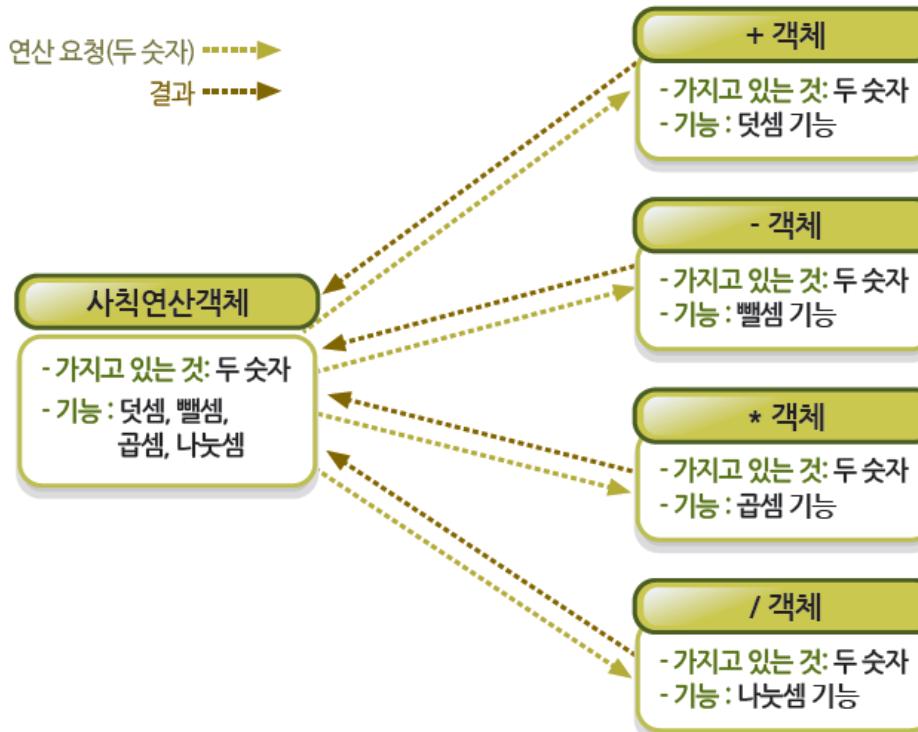
■ 객체지향언어 : 실 세계 객체 지향의 예



프로그래밍 언어 Java

■ 자바 언어의 특징

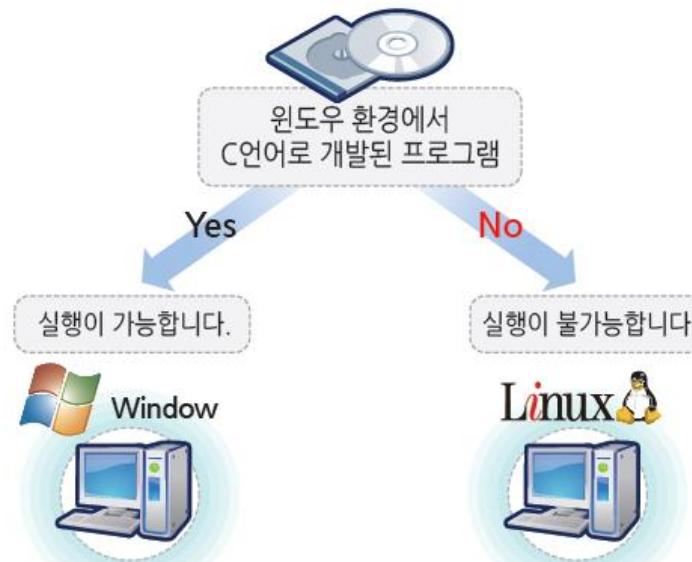
■ 객체지향언어 : 자바의 객체 지향의 예



프로그래밍 언어 Java

■ 자바 언어의 특징

- 운영 체제에 독립적 : 소프트웨어와 운영체제
 - 일반적으로 컴퓨터에서 실행되는 소프트웨어
 - 운영체제에 종속적



프로그래밍 언어 Java

■ 자바 언어의 특징

- 운영 체제에 독립적 : JVM이 구축된 컴퓨터에서 실행 가능



프로그래밍 언어 Java

■ 자바 언어의 특징

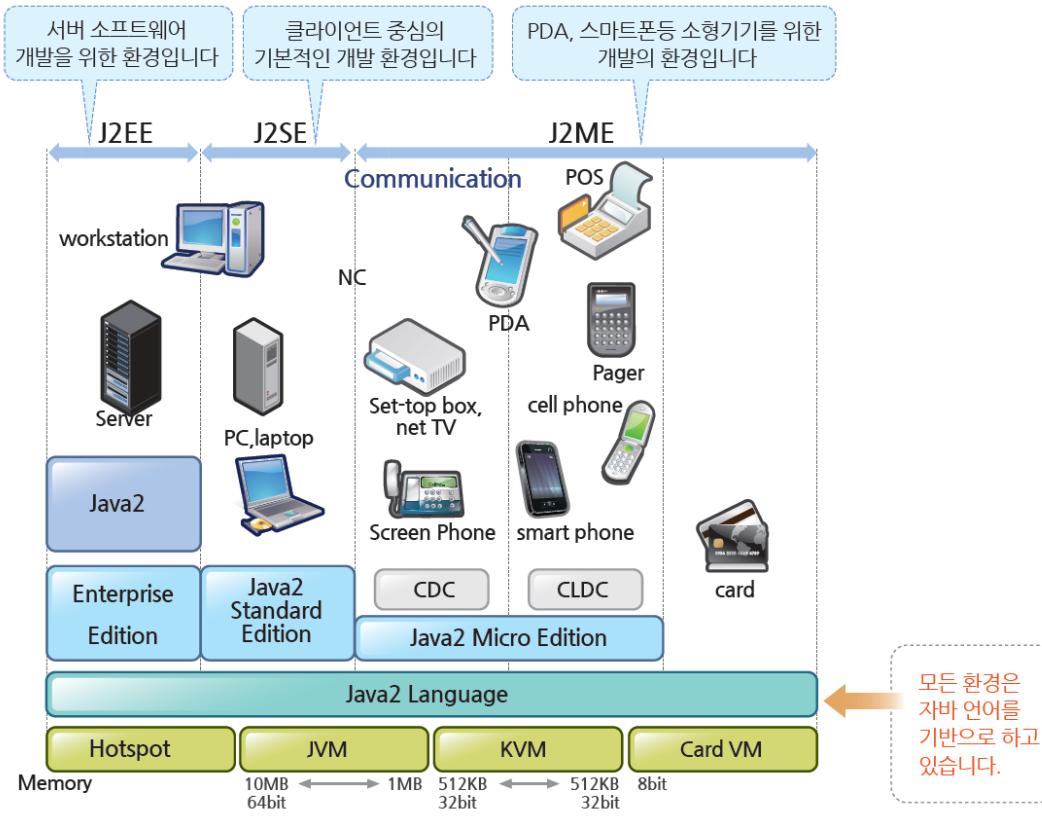
■ 무료 개발 환경

- 자바의 무료 개발 환경 : 개방된 형태를 취하고 있음
- 자바의 모든 개발 환경을 무료로 제공
- 지속적으로 최신 버전을 제공

프로그래밍 언어 Java

■ 자바 언어의 특징

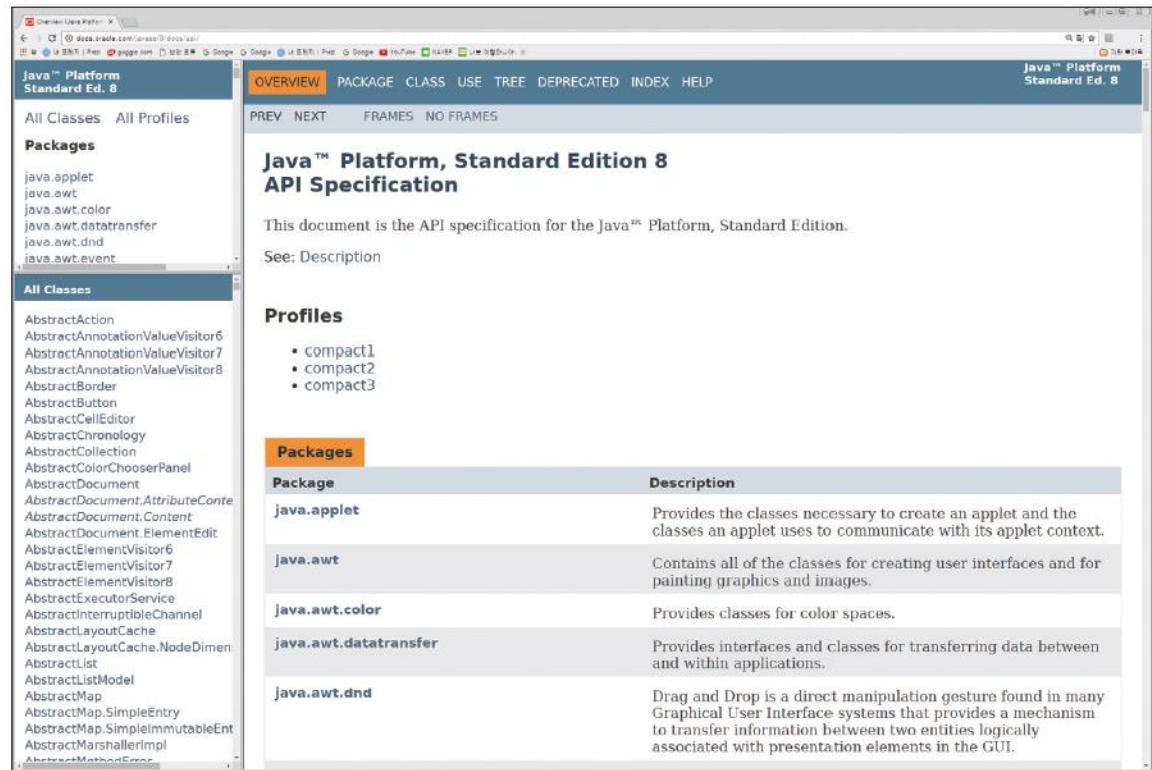
■ 무료 개발 환경 : 선사에서 제공하는 플랫폼



프로그래밍 언어 Java

■ 자바 언어의 특징

- 무수한 라이브러리 : 대규모의 라이브러리 클래스 제공

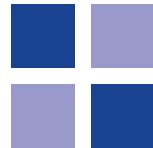


프로그래밍 언어 Java

■ 자바 언어의 특징

■ 자바의 버전

버전명	연도	특징	라이브러리 클래스	버전명	연도	특징	라이브러리 클래스
JDK 1.0 ~ JDK 1.4	1996 ~ 2002	<ul style="list-style-type: none">Java Applet, RMI, JDBC, reflection, JT, Inner Class 개념 도입1.2 : 자바를 세가지 버전으로 발표(Java SE 1.2, ME 1.2, EE 1.2)Swing, Corba IDL, Collection Framework 포함HotSpot JVM 포함JNDI(Java Naming and Directory Interface) 포함1.4 : JOP에 의해서 오픈소스 정책으로 자바가 관리되기 시작한 버전1.4 : Java 2 Security 모델의 확립(Sandbox)Language: assert 도입	약 200여개로 시작되어 1.4에서는 약 2,500여개	Java 8	2014	<ul style="list-style-type: none">오라클로 인수된 후 첫 번째 버전Java5 이후 가장 큰 언어적 변화(Lambda 및 함수형 프로그래밍, default method 도입)Language : Lambda expression, Default Method Interface(다중상속지원) 등API : Nashorn(JS엔진), new Date and Time API, stream api, Collection에 대한 함수형화병렬처리에 접합한 구조로 진화	약 4,500여개
Java 5	2004	<ul style="list-style-type: none">기능적으로 가장 많은 변화, Java 5로 불림.Language : Generics , annotation, auto boxing, enum, vararg, foreach, static imports 도입API : java.util.concurrent API, Scanner class	약 3,000여개	Java 9	2016~ (예정)	<ul style="list-style-type: none">Modular System (Jigsaw)지원예정Money API지원예정Java Shell지원예정변수에 대한 타입 추론 지원예정(var,va)OpenCL이용한 자동화된 병렬 프로그래밍 지원예정value 타입 지원예정	
Java 6	2006	<ul style="list-style-type: none">JVM/Swing에 있어 많은 Performance 향상 (synchronization, compiler, GC, start-up time)G1(Garbage First) GC도입.	약 3,500여개				
Java 7	2011	<ul style="list-style-type: none">JVM : Dynamic Language support (invokedynamic – new byte operation)Language : Switch에서 String, try–resource, generics에서 타입추론, 숫자에서 underscore 사용API : Concurrency, NIO, sort, crypto, GPU 강화JavaFX가 기본으로 포함	약 4,000여개				



(1일 개발환경 설정 및 Java 예제 프로그램)

kimth4110@gmail.com

김태환

수업 목표

- 자바 프로그램의 형태
- 자바 프로그램의 실행 환경
- 자바 학습을 위한 준비



자바 프로그램의 실행환경

■ 자바 개발 환경

- 선(sun)사에서 제공되는 개발 환경 설정
- 프로그램을 작성할 수 있게 해주는 이클립스
소프트 웨어 사용

⇒인터넷에서 무료로 공개되는 소프트 웨어

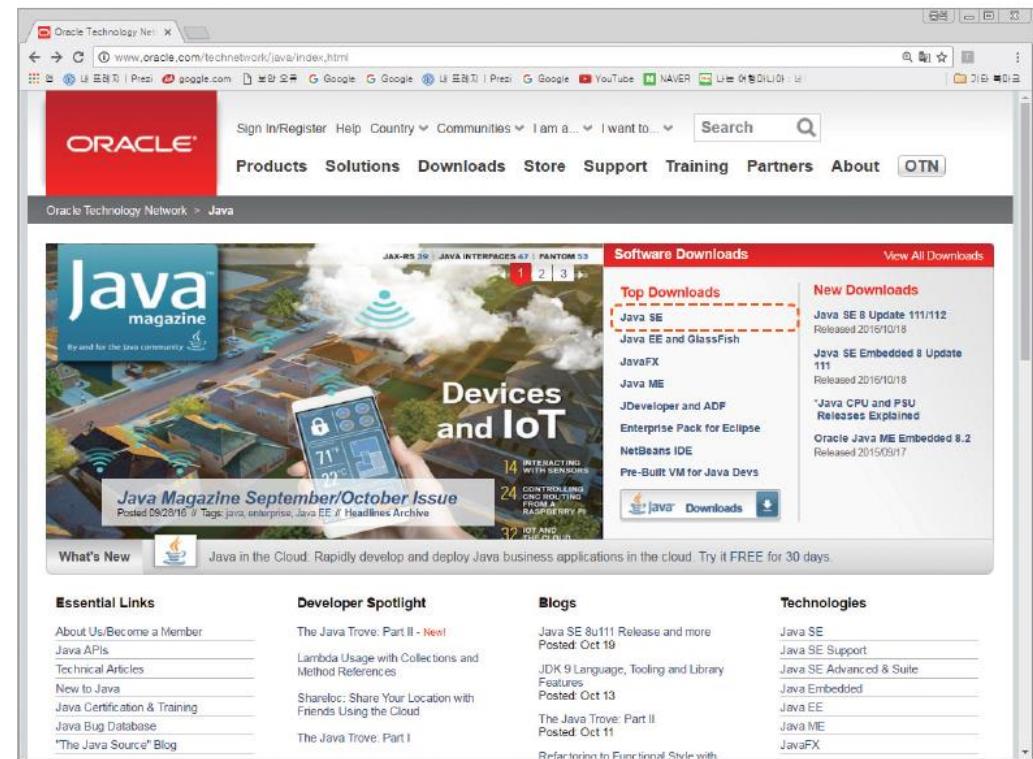
자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 설치

■ 오라클사 : <http://www.oracle.com/technetwork/java/index.html>

■ Download의

“Java SE”를 선택



자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 설치

■ Step 1 : 선사의 홈페이지에서 JDK 다운

- Java SE 8u181 Java Platform(JDK)를 선택

The screenshot shows the Java SE 8u181 download page. At the top, it says "Java SE 8u181" and includes a note: "Java SE 8u181 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release." Below this is a "Learn more" link. On the right, there are two large download buttons: "JDK DOWNLOAD" and "Server JRE DOWNLOAD". The left sidebar contains links to "Installation Instructions", "Release Notes", "Oracle License", "Java SE Licensing Information User Manual" (which includes "Includes Third Party Licenses"), "Certified System Configurations", and "Readme Files".

Java SE 8u181
Java SE 8u181 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.
[Learn more](#)

JDK
[DOWNLOAD](#)

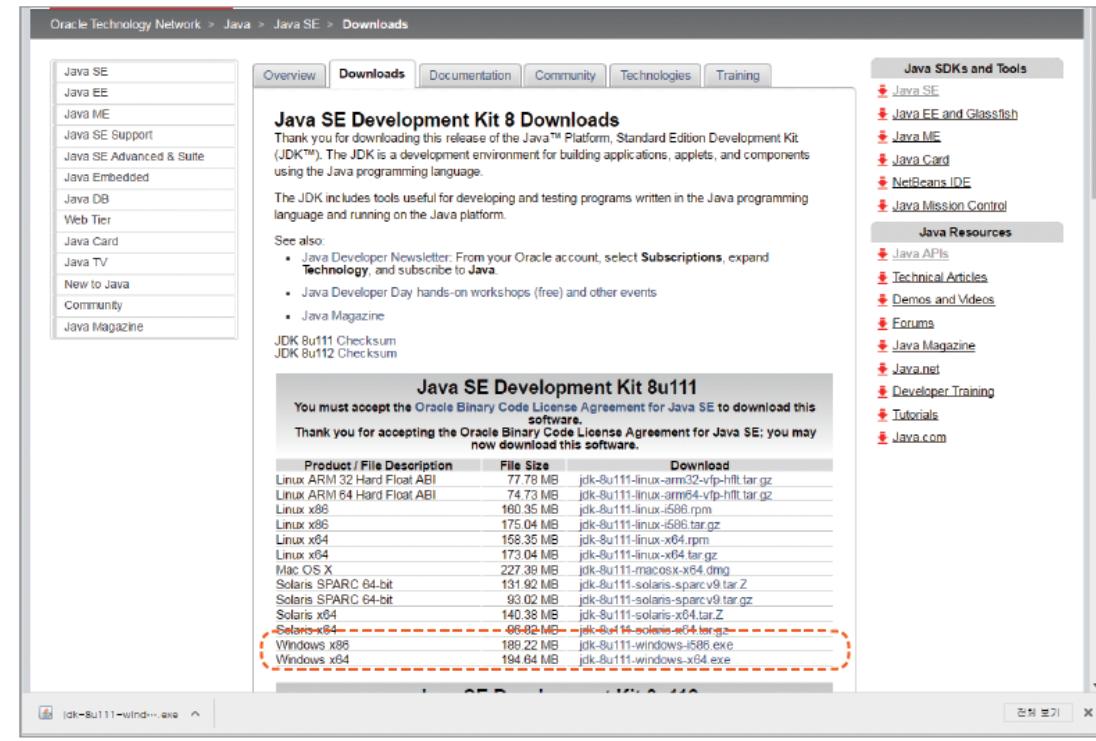
Server JRE
[DOWNLOAD](#)

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Licensing Information User Manual](#)
 - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme Files](#)

자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 설치

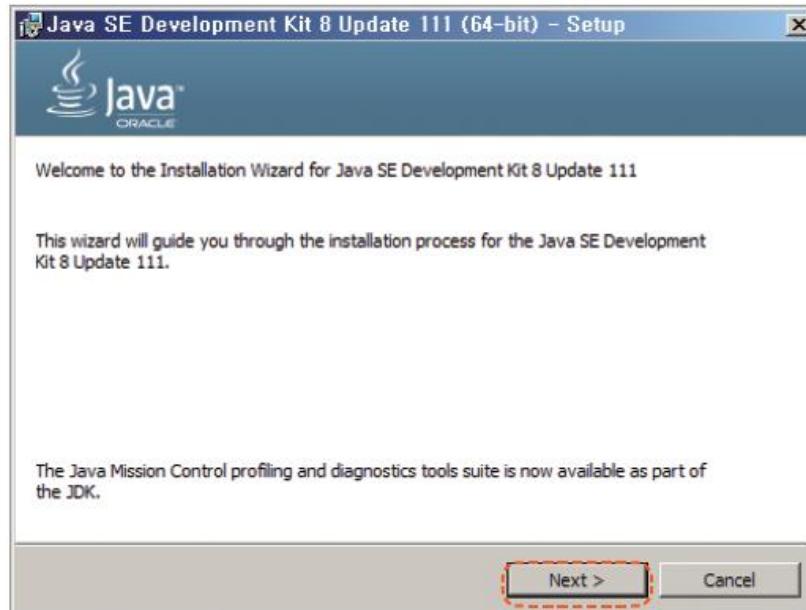
- Step 1 : 선사의 홈페이지에서 JDK 다운
 - 컴퓨터에 적합한 운영체제를 선택



자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 설치

- Step 2 : 다운받은 소프트웨어를 실행한다.
 - 소프트에어를 더블 클릭하여 설치를 완료

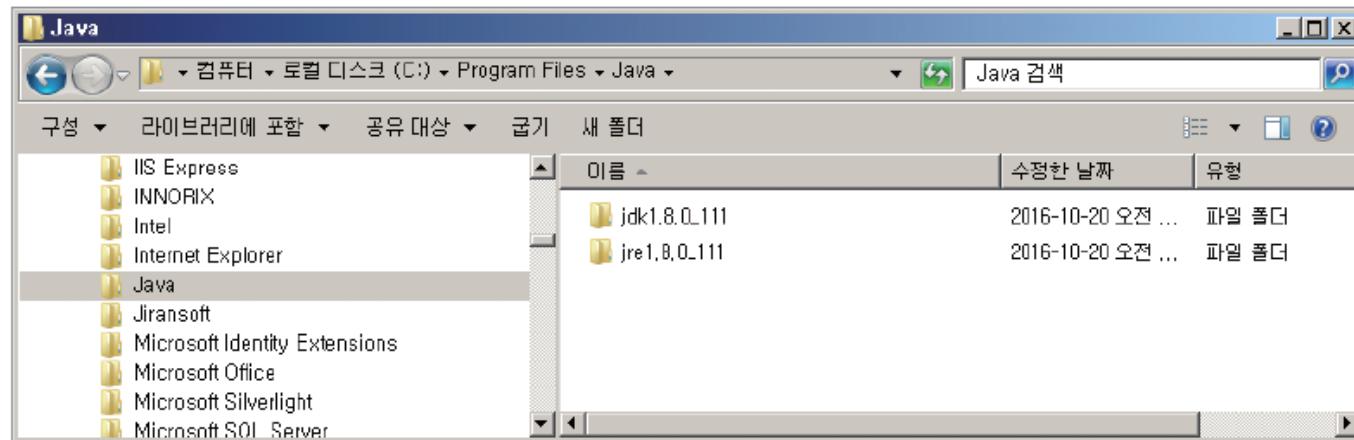


자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 설치

■ Step 2 : 다운받은 소프트웨어를 실행한다.

- 완료되면 일반적으로 “C:\Program Files\Java” 디렉토리에 두 개의 디렉토리가 생성



자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 환경 설정

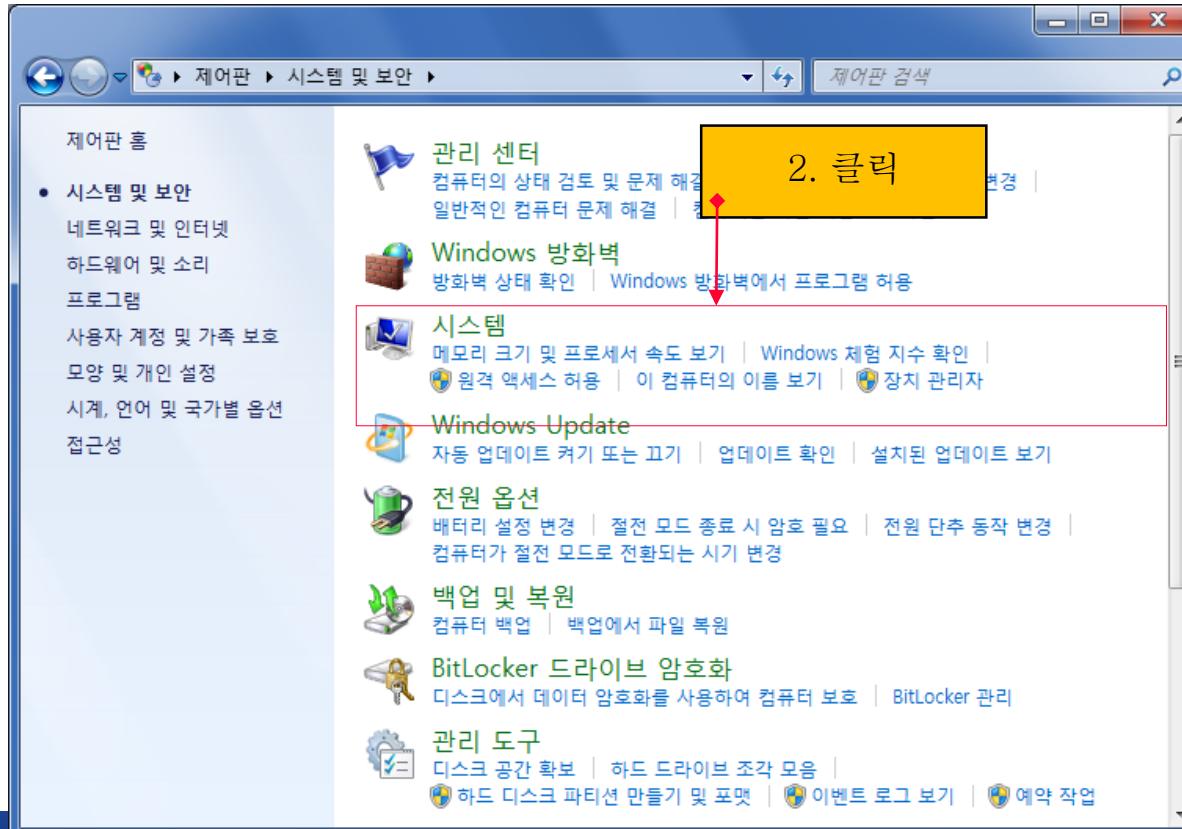
■ 시스템



자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 환경 설정

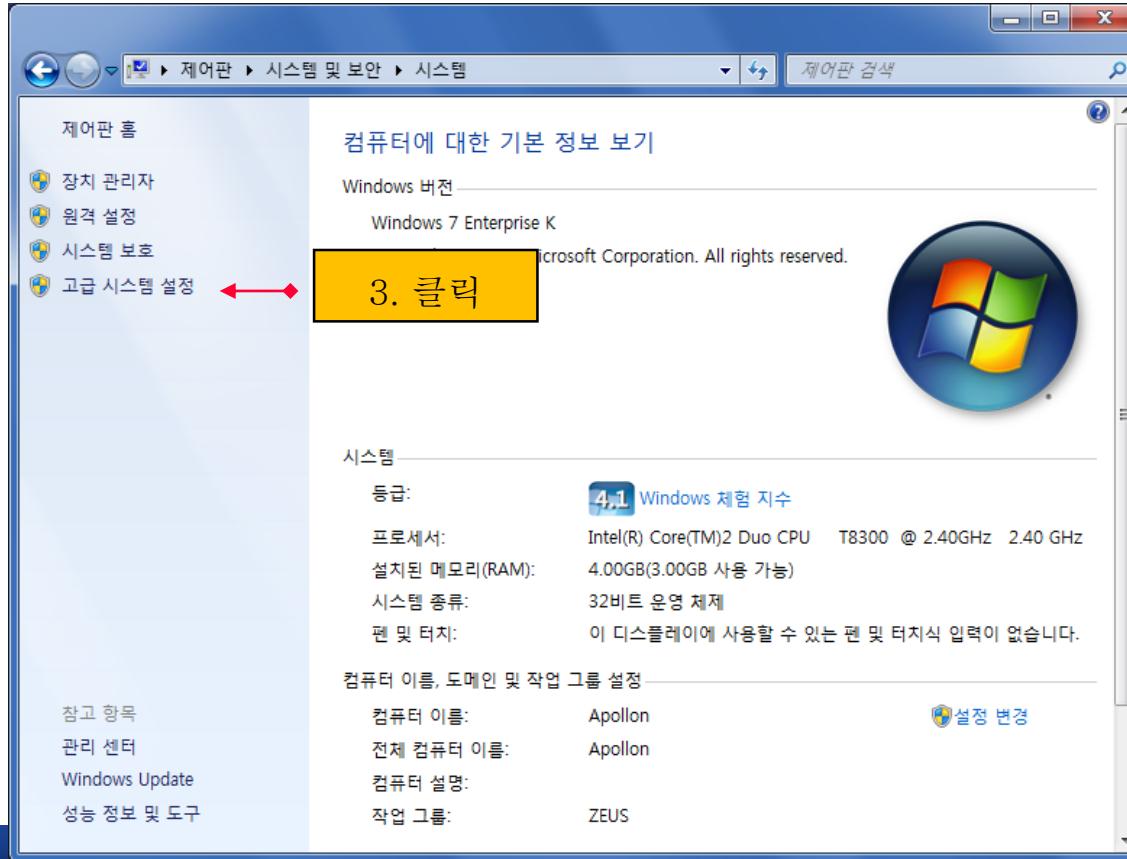
■ 시스템



자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 환경 설정

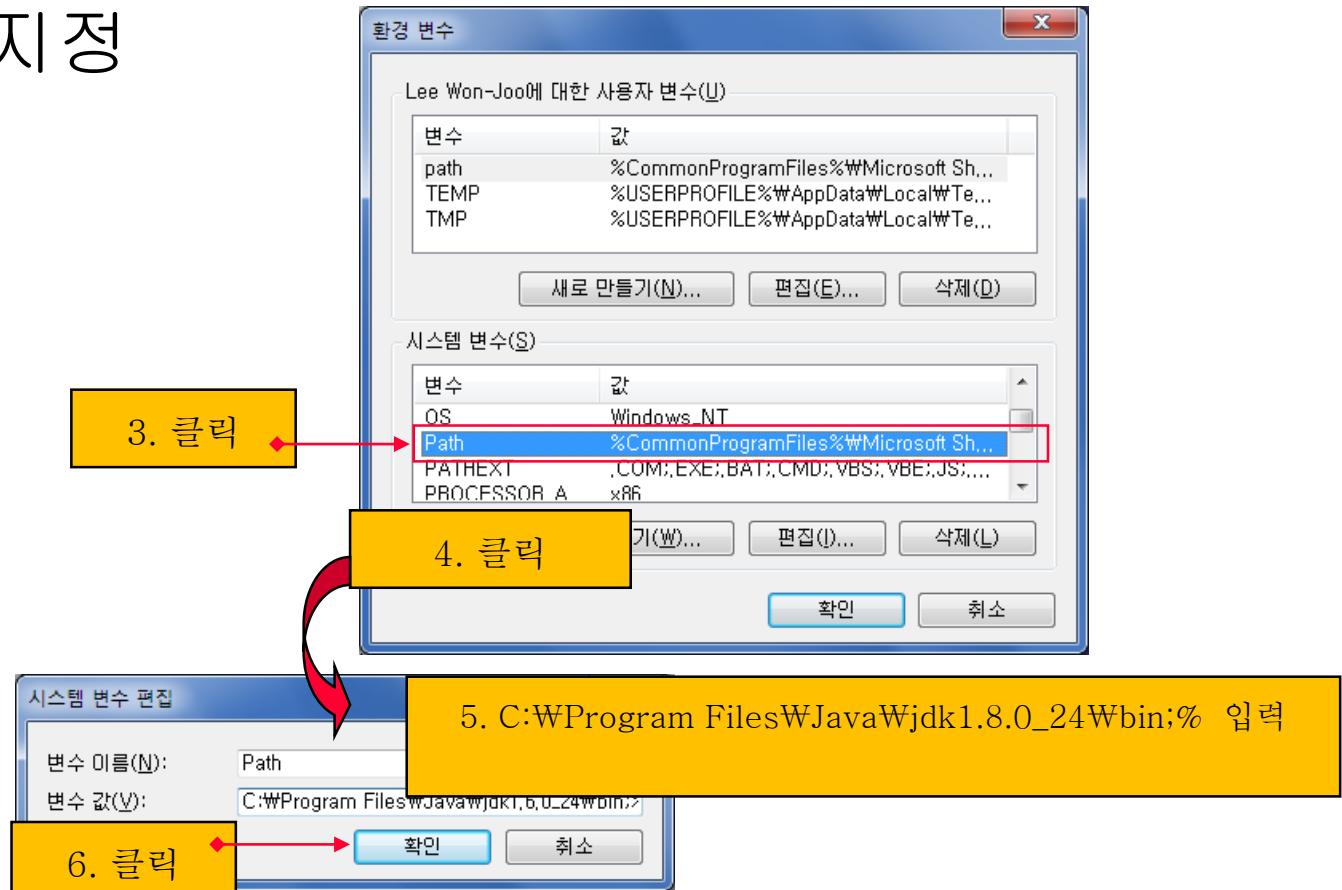
■ 시스템



자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 환경 설정

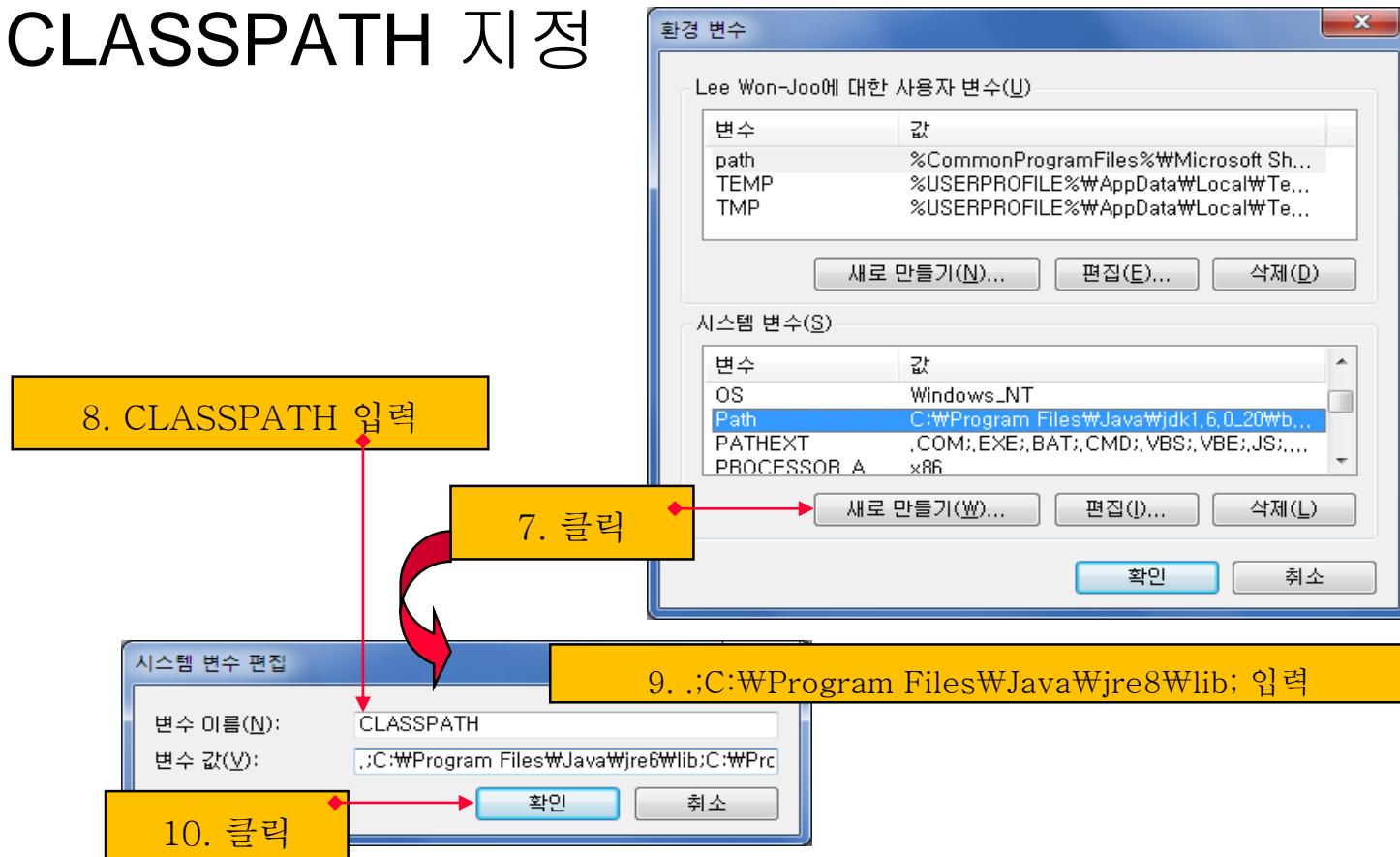
■ Path 지정



자바 프로그램의 실행환경

■ 자바 개발 환경 – JDK 환경 설정

■ CLASSPATH 지정

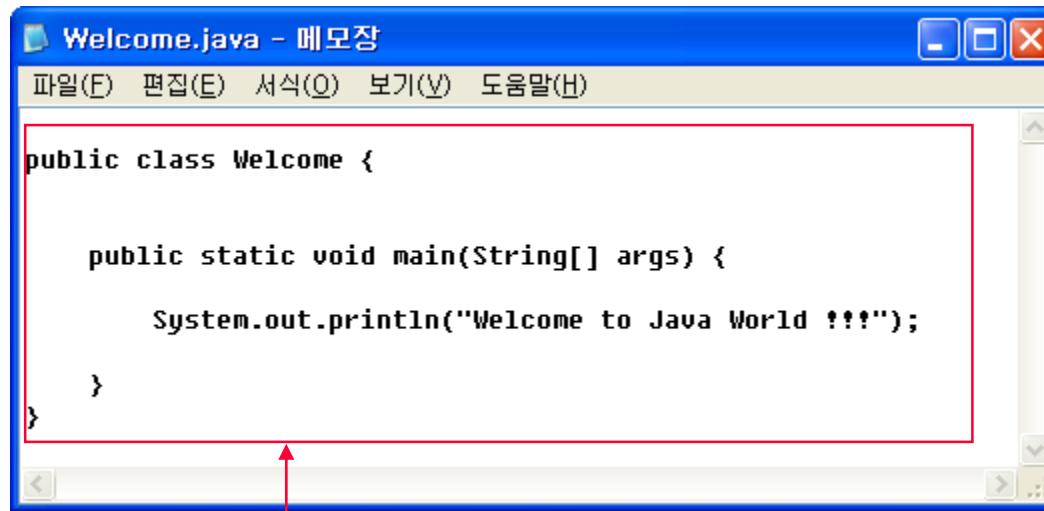


자바 프로그램의 실행환경

■ Java 파일 작성

■ 메모장 이용

- JavaSrc 폴더 생성
- Welcome.java 파일 생성



```
public class Welcome {  
  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java World !!!");  
    }  
}
```

18. Coding

자바 프로그램의 실행환경

■ Java 파일 실행

■ Compile

- javac Welcome.java
 - Welcome.class 생성

■ Execute

- java Welcome

```
C:\> C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\Won-Joo Lee>cd \JavaSrc

C:\JavaSrc>dir
  C 드라이브의 볼륨: SYSTEM
  볼륨 일련 번호: 40E0-FBD5

C:\JavaSrc 디렉터리
2008-03-02 오전 11:22 <DIR> .
2008-03-02 오전 11:22 <DIR> ..
2008-03-01 오후 04:18           162 Welcome.java
                           1개 파일           162 바이트
                           2개 디렉터리 107,707,097,088 바이트 남음

C:\JavaSrc>javac Welcome.java

C:\JavaSrc>dir
  C 드라이브의 볼륨: SYSTEM
  볼륨 일련 번호: 40E0-FBD5

C:\JavaSrc 디렉터리
2008-03-02 오전 11:30 <DIR> .
2008-03-02 오전 11:30 <DIR> ..
2008-03-02 오전 11:30           433 Welcome.class
2008-03-01 오후 04:18           162 Welcome.java
                           2개 파일           595 바이트
                           2개 디렉터리 107,707,039,744 바이트 남음

C:\JavaSrc>java Welcome
Welcome to Java World !!!
```

자바 프로그램의 실행환경

■ 자바 개발 환경 – 이클립스의 설치

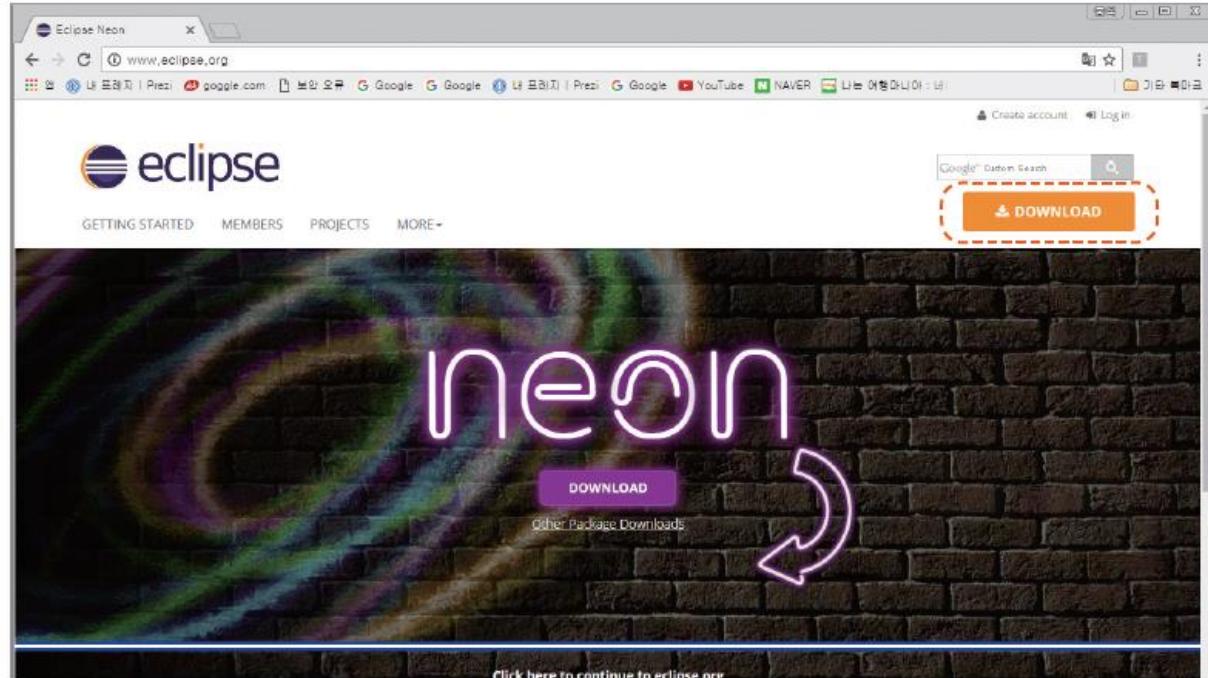
- 이클립스 (Eclipse) : 자바 프로그램을 개발하기 위한 통합 개발 환경
- 자바 프로그램을 개발하기 위한 다양한 도구들이 존재
- 무료로 제공되는 이클립스를 가장 많이 사용

자바 프로그램의 실행환경

■ 자바 개발 환경 – 이클립스의 설치

■ Step 1 : 이클립스 홈페이지(www.eclipse.org)

- 오른쪽 상단에 있는 “Download Eclipse”를 선택

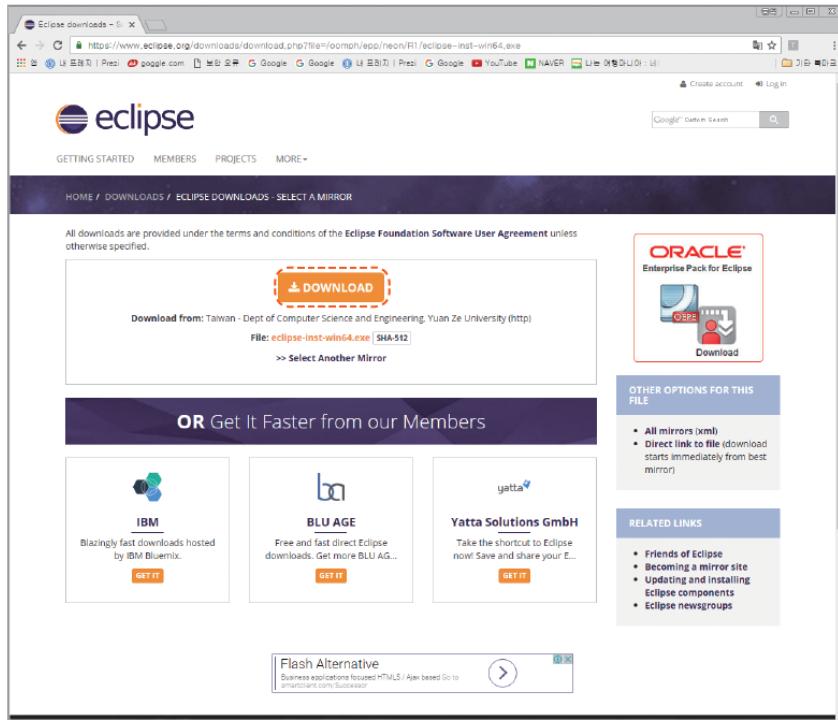


자바 프로그램의 실행환경

■ 자바 개발 환경 – 이클립스의 설치

■ Step 1 : 이클립스 홈페이지(www.eclipse.org)

- “Get Eclipse PHOTON”을 선택하여 다운로드

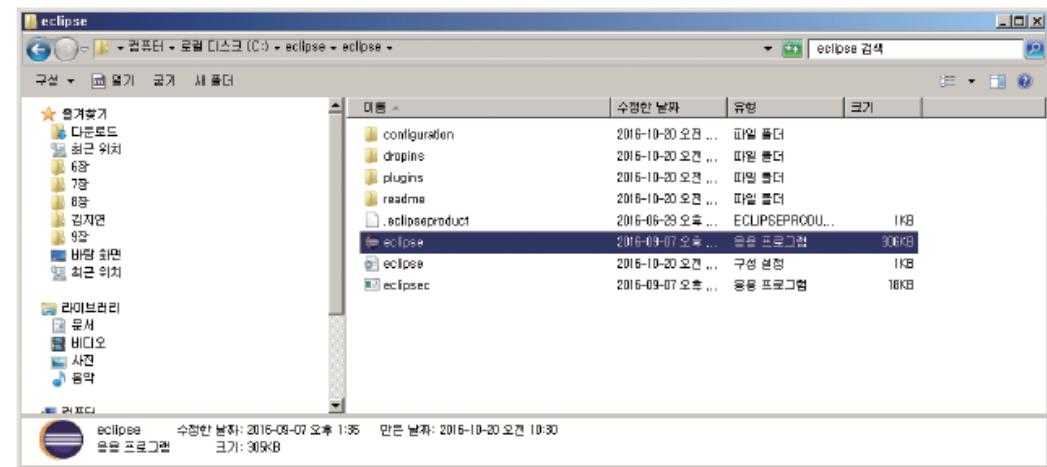


자바 프로그램의 실행환경

■ 자바 개발 환경 – 이클립스의 설치

■ Step 2 : 이클립스 설치 (31p.)

□ 적당한 위치에 압축을 풀면 설치 완료



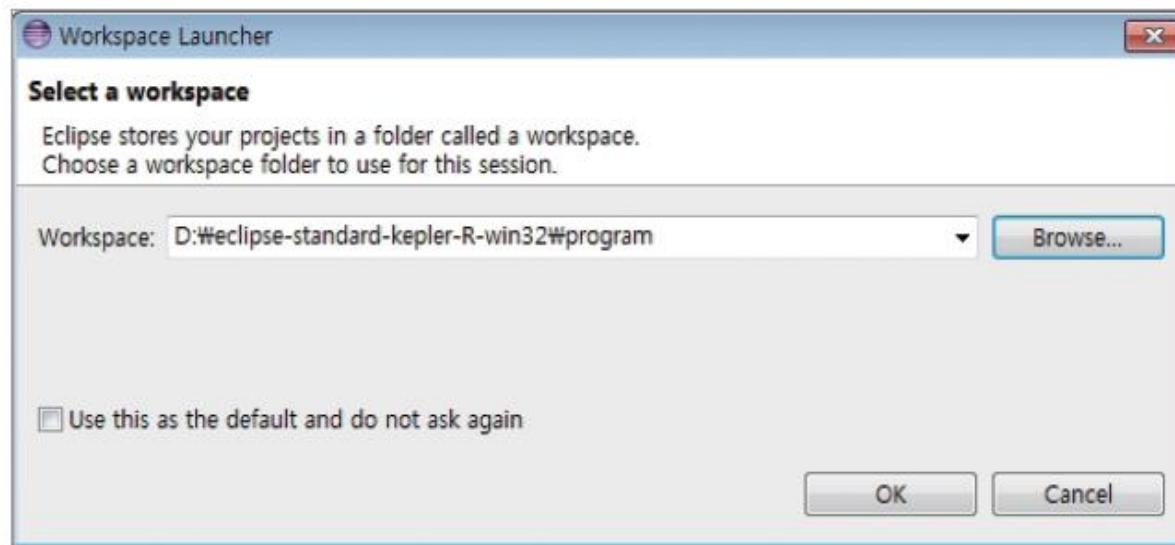
파일	수정한 날짜	유형
configuration	2016-10-03 오전 ...	파일 폴더
dropins	2016-10-20 오후 ...	파일 폴더
plugins	2016-10-20 오후 ...	파일 폴더
readme	2016-10-20 오후 ...	파일 폴더
.eclipseproduct	2016-09-29 오후 ...	ECLIPSEPRODU...
eclipse	2016-09-07 오후 ...	응용 프로그램
eclipse	2016-10-20 오후 ...	구성 설정
eclipsec	2016-09-07 오후 ...	응용 프로그램

자바 프로그램의 실행환경

■ 이클립스의 실행과 기본 구조

■ 이클립스의 실행

- Eclipse 폴더에 있는 `eclipse.exe` 파일을 실행
- `.workspace`를 묻는 화면에서 경로 선택

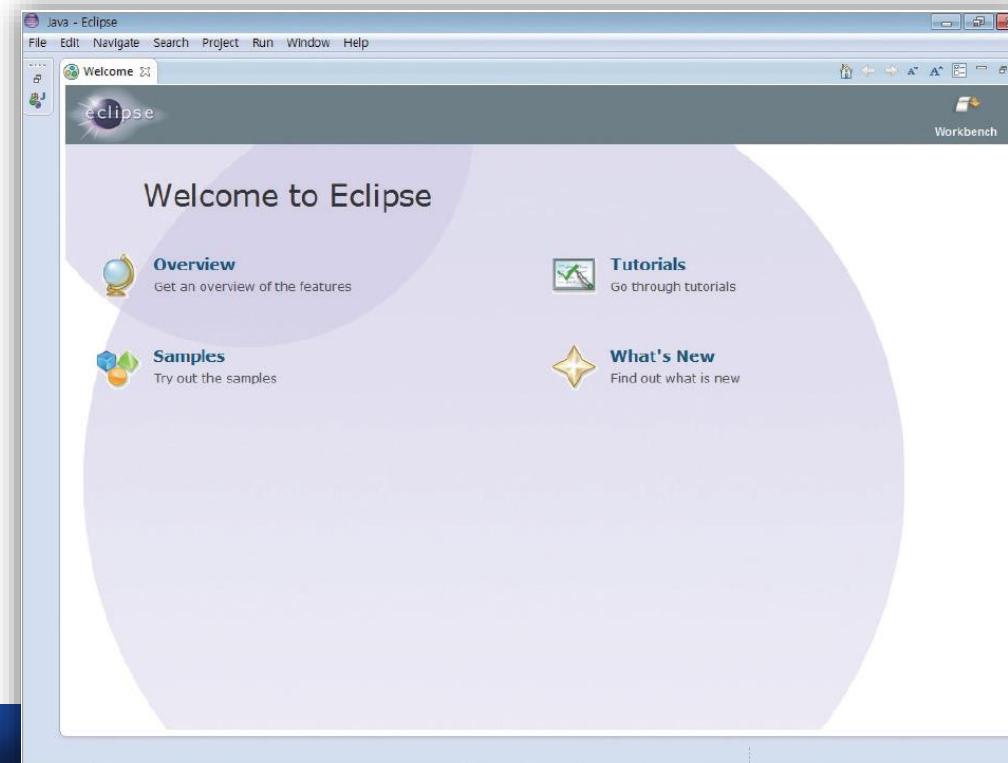


자바 프로그램의 실행환경

■ 이클립스의 실행과 기본 구조

■ 이클립스의 실행

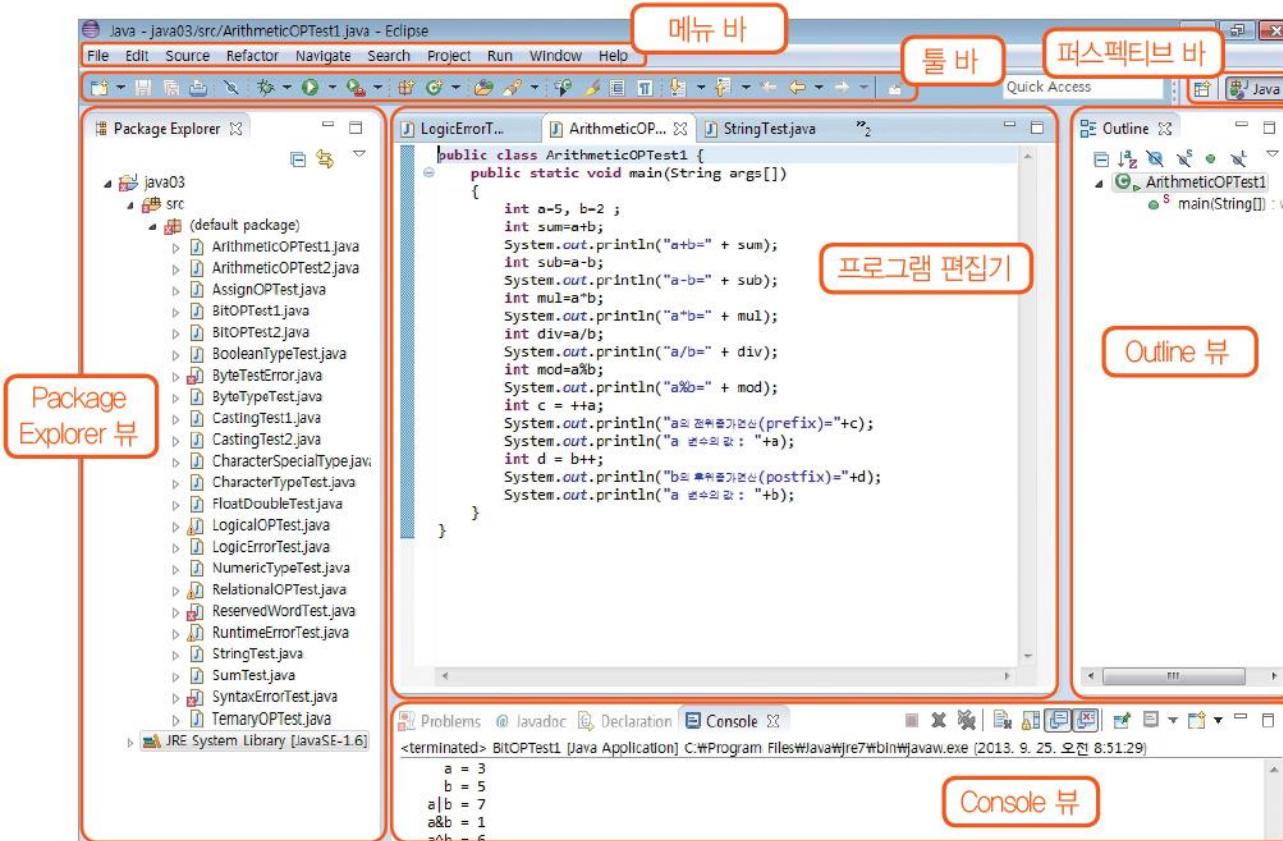
- Welcome 화면이 나타남



자바 프로그램의 실행환경

■ 이클립스의 실행과 기본 구조

■ 이클립스의 기본 구조



자바 프로그램의 실행환경

■ 이클립스의 실행과 기본 구조

- 메뉴바

- 이클립스의 모든 기능들을 사용

- 툴바

- 자주 쓰이는 기능들을 사용할 수 있도록 제공

- Package Explorer 뷰

- 패키지와 클래스 파일, 라이브러리를 관리하는 뷰

- 프로그램 편집기

- 자바 프로그램을 작성할 수 있는 편집기 창

자바 프로그램의 실행환경

■ 이클립스의 실행과 기본 구조

- Outline 뷰

- 편집되고 있는 프로그램 코드의 개요와 트리 구조

- Console 뷰

- 실행 결과나 오류가 있는 경우 오류를 나타내는 창

- 퍼스펙티브 바

- 이클립스를 사용할 때의 뷰나 창의 배열을 의미

자바 프로그램의 실행환경

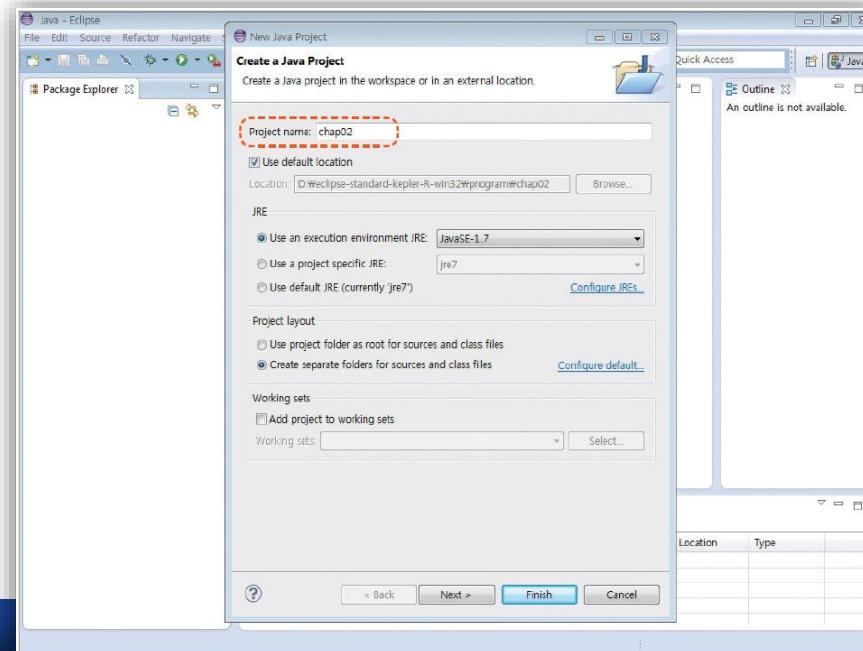
- 이클립스를 이용한 프로그램 작성 순서
 - Step 1 : 프로젝트 선정
 - Step 2 : 프로그램(클래스) 작성
 - Step 3 : 프로그램 실행

자바 프로그램의 실행환경

■ 이클립스를 이용한 프로그램 작성 순서

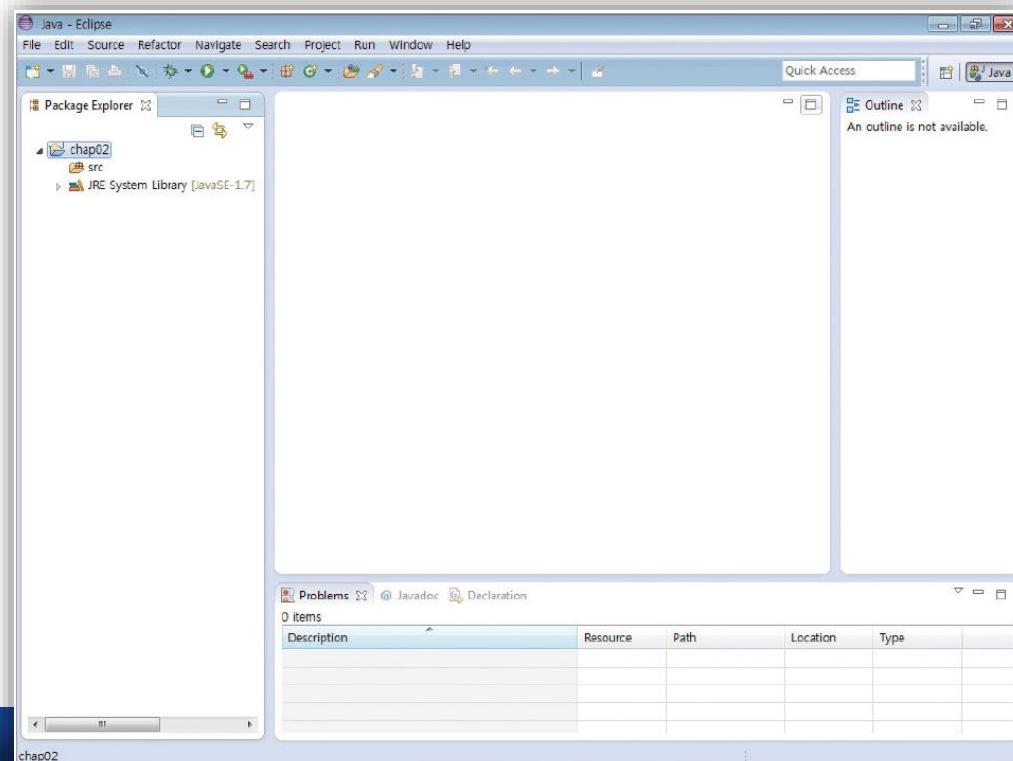
■ Step 1 : 프로젝트 만들기

- 메뉴에서 [File] → [New] → [Java Project]를 선택
- 프로젝트 chap02로 지정, 하단 “Finish” 버튼 선택



자바 프로그램의 실행환경

- 이클립스를 이용한 프로그램 작성 순서
 - Step 1 : 프로젝트 만들기
 - Package Explorer 뷰에 패키지가 표시

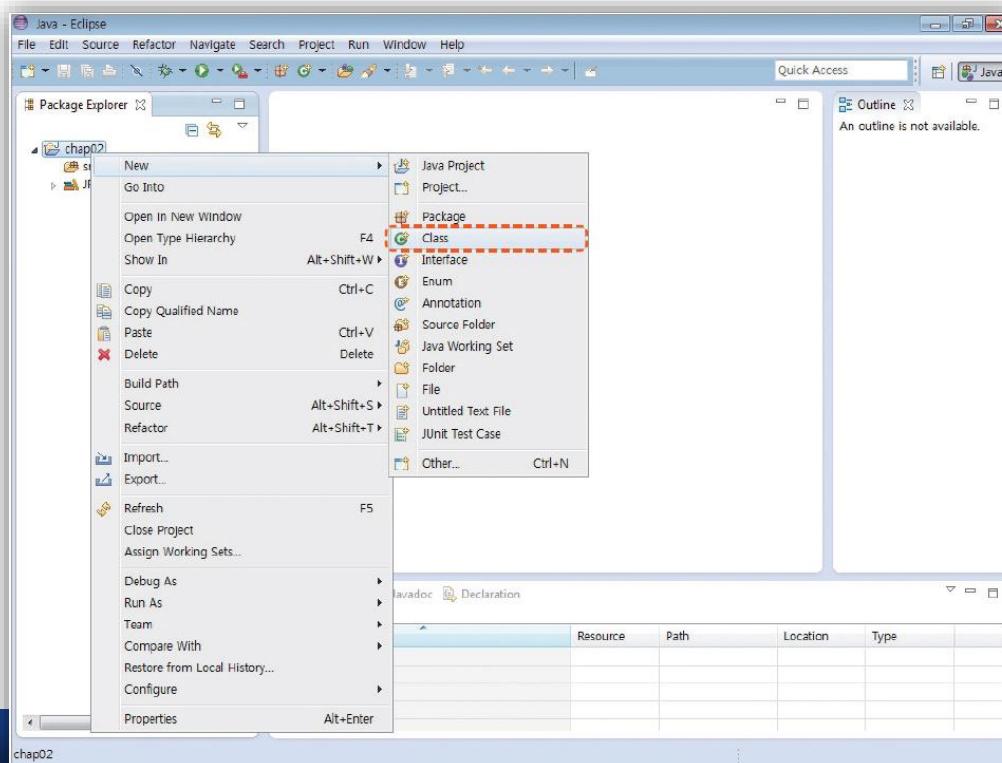


자바 프로그램의 실행환경

■ 이클립스를 이용한 프로그램 작성 순서

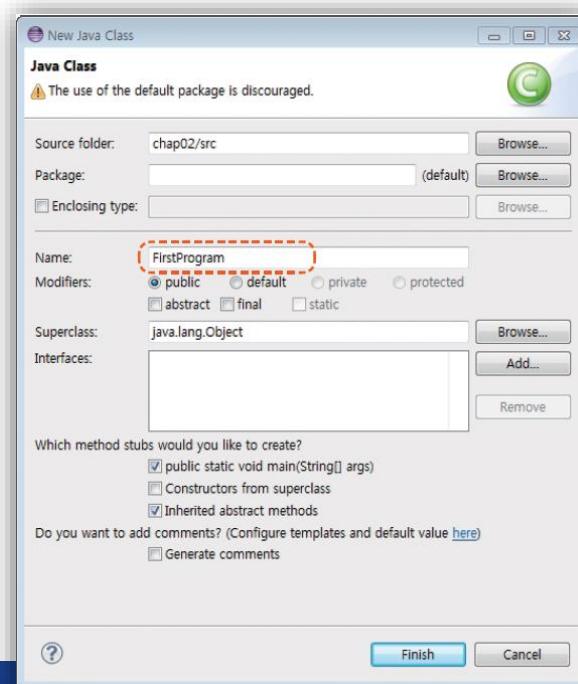
■ Step 2 : 프로그램(클래스) 작성

- 프로젝트 명을 마우스 우클릭 후 [new]-[class] 선택



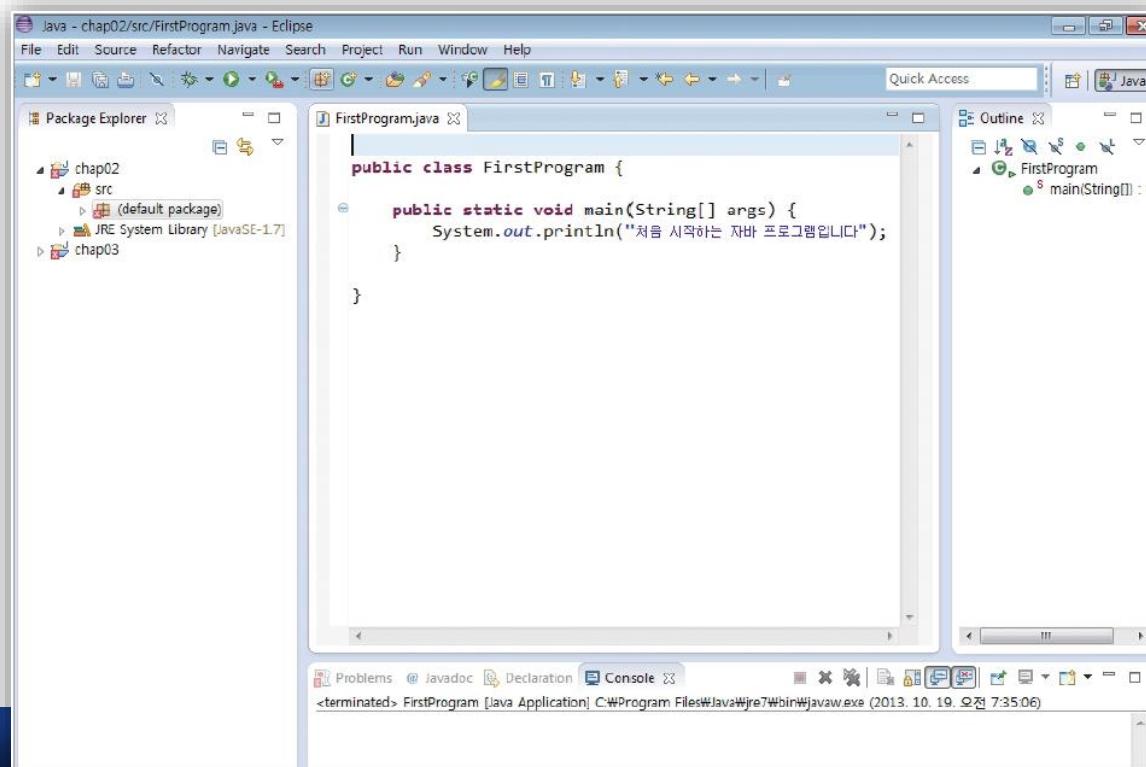
자바 프로그램의 실행환경

- 이클립스를 이용한 프로그램 작성 순서
 - Step 2 : 프로그램(클래스) 작성
 - 클래스 생성을 위한 팝업 창 생성
 - 아래 그림과 같이 작성



자바 프로그램의 실행환경

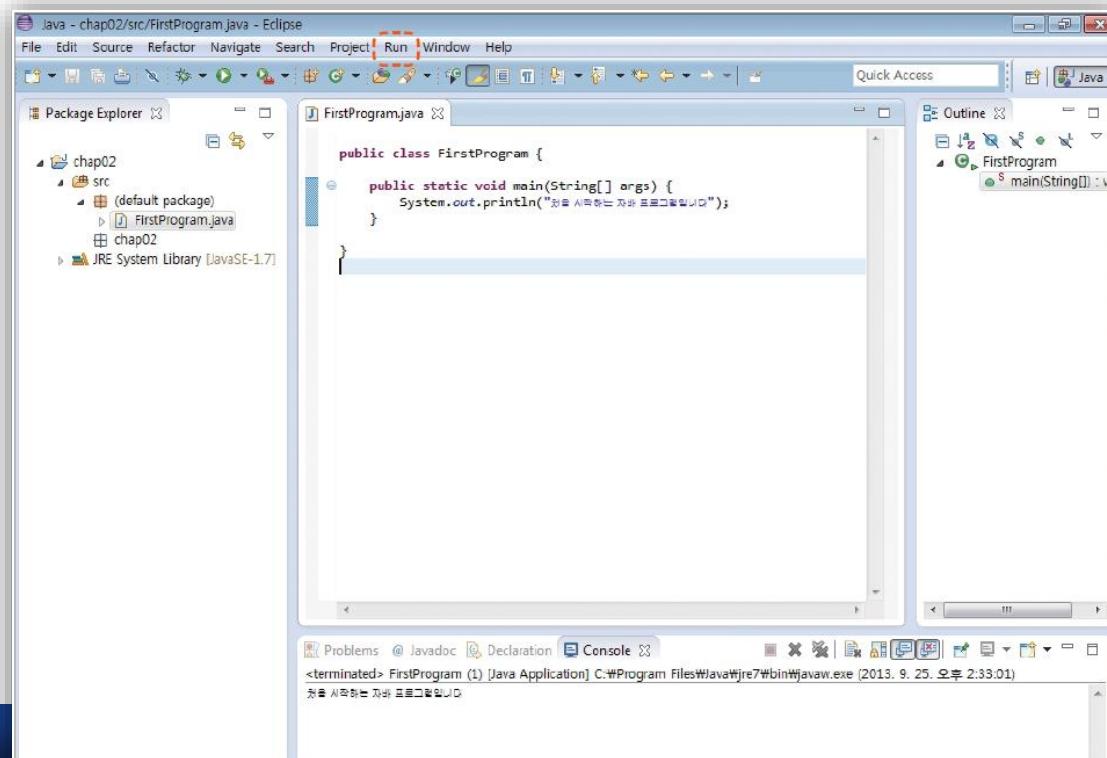
- 이클립스를 이용한 프로그램 작성 순서
 - Step 2 : 프로그램(클래스) 작성
 - 편집기 창에서 프로그램을 입력



자바 프로그램의 실행환경

■ 이클립스를 이용한 프로그램 작성 순서 ■ Step 3 : 프로그램 실행

- 메뉴에서 [Run] → [Run]을 실행



자바 프로그램의 실행환경

■ 이클립스 프로젝트 입출력

- 이클립스체서는 프로젝트 단위로 개발 환경 제공
- 프로젝트는 소프트웨어를 개발하는 단위
- Import-Export 기능을 이용하여 특정 컴퓨터에서 그대로 사용할 수 있는 편리한 개발환경 제공

자바 프로그램의 실행환경

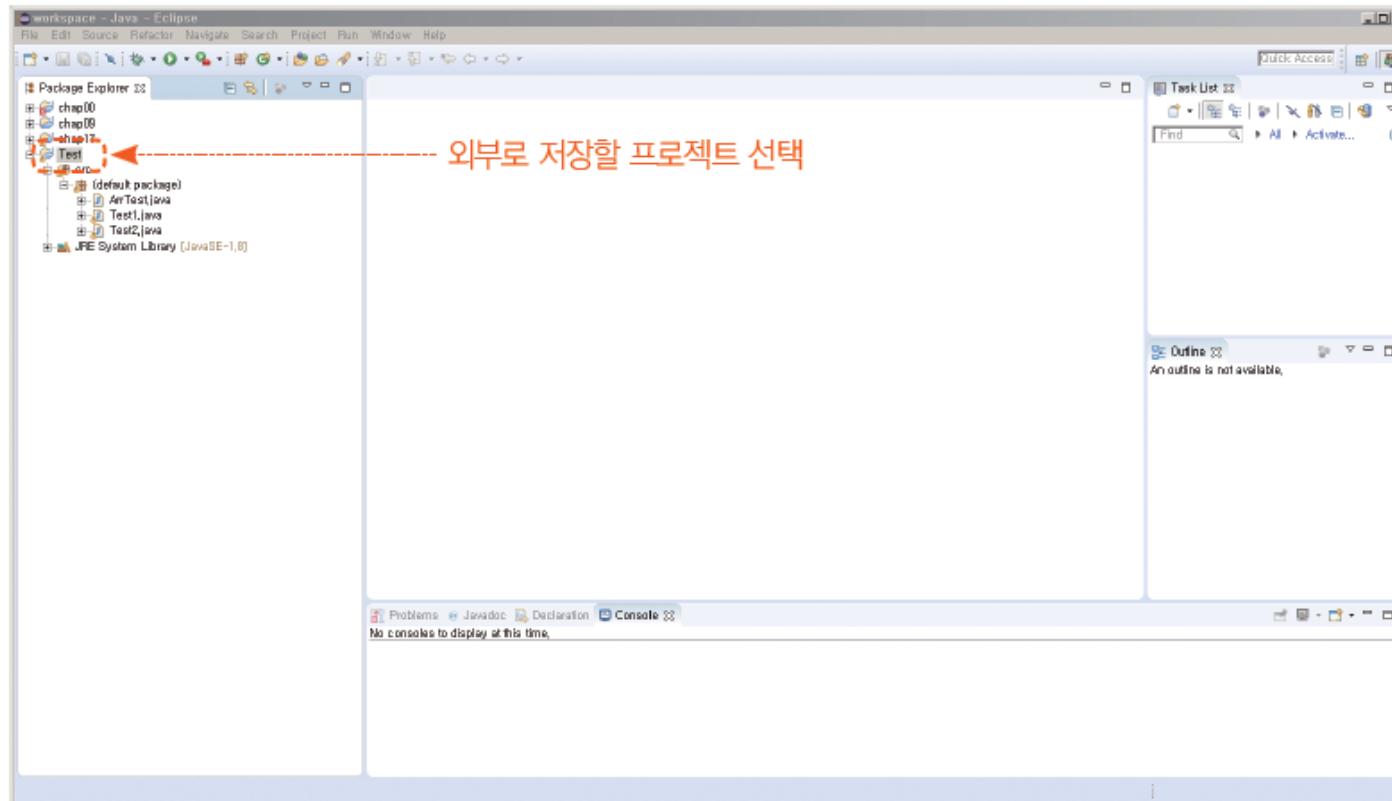
■ 이클립스 프로젝트 출력

- 개발된 프로젝트를 외부 컴퓨터에서 사용하기 위해 프로젝트 단위로 특정 폴더에 저장하는 기능

자바 프로그램의 실행환경

■ 이클립스 프로젝트 출력

■ Step 1 : 프로젝트를 선택

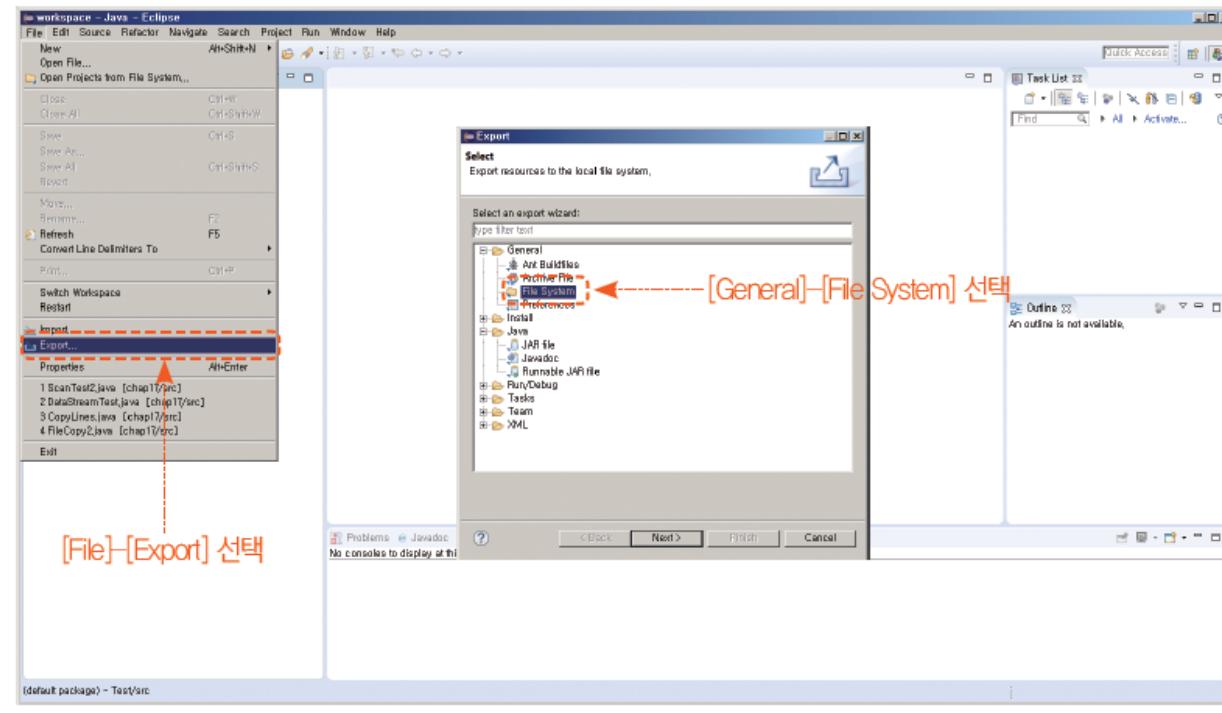


자바 프로그램의 실행환경

■ 이클립스 프로젝트 출력

■ Step 2 : [File]->[Export] 메뉴를 선택,

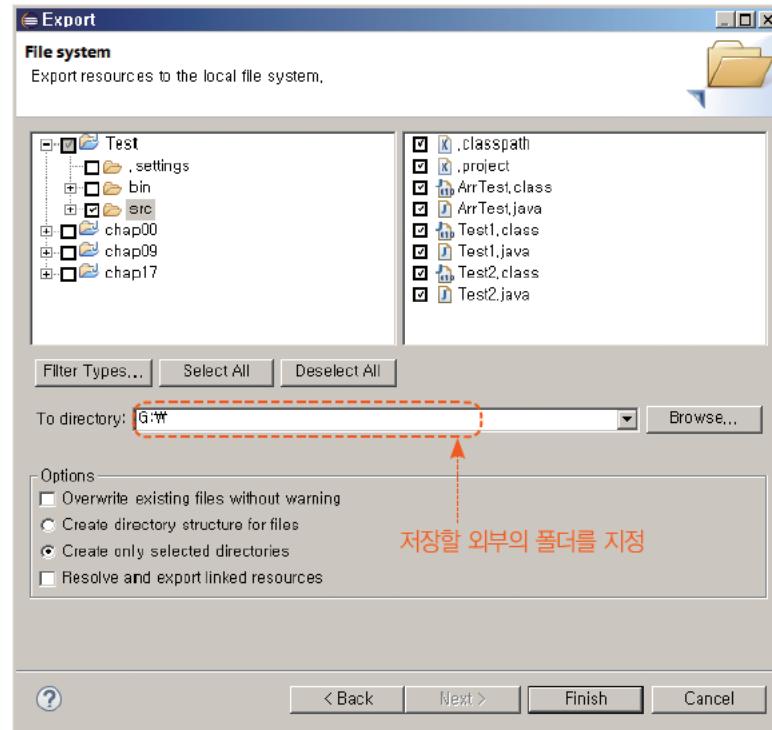
팝업창에서 [General]->[File System]을 선택



자바 프로그램의 실행환경

■ 이클립스 프로젝트 출력

- Step 3 : 나타난 팝업창에서 저장할 디렉토리를 지정하고 finish 버튼을 선택

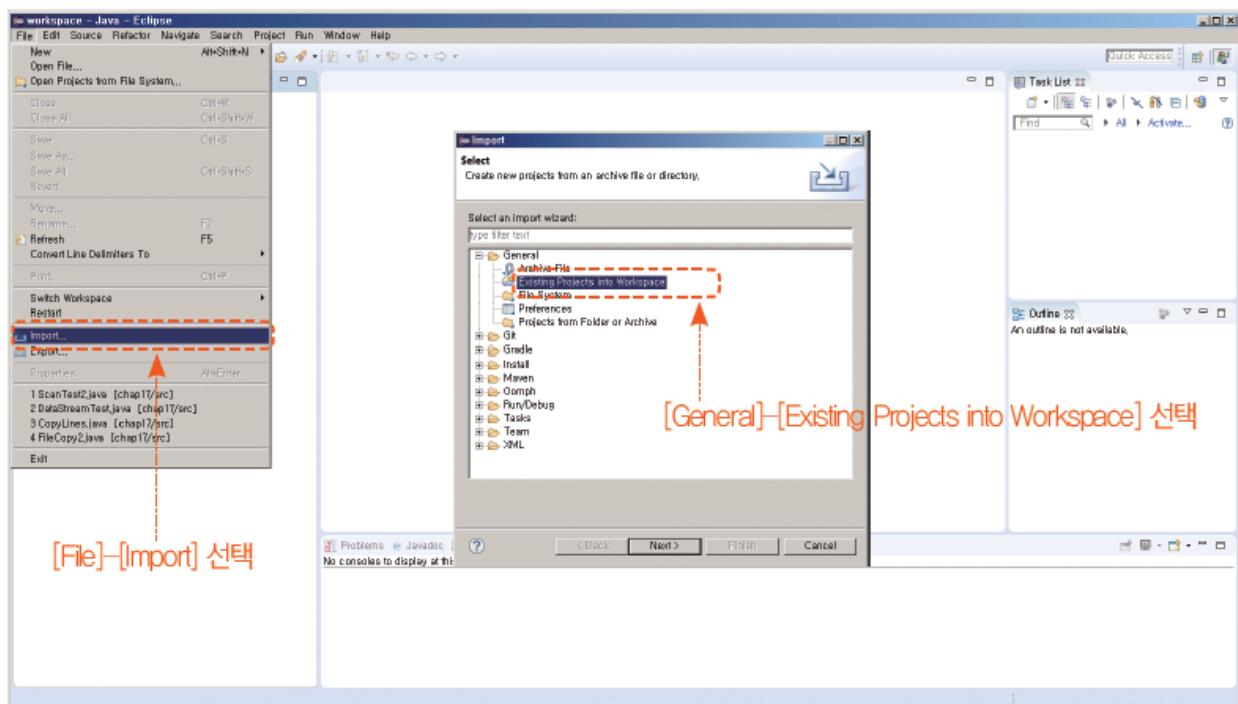


자바 프로그램의 실행환경

■ 이클립스 프로젝트 입력

- Step 1 : [file]-[import]를 선택하고 팝업창에서 [general]-[Existing Projects into Workspace]

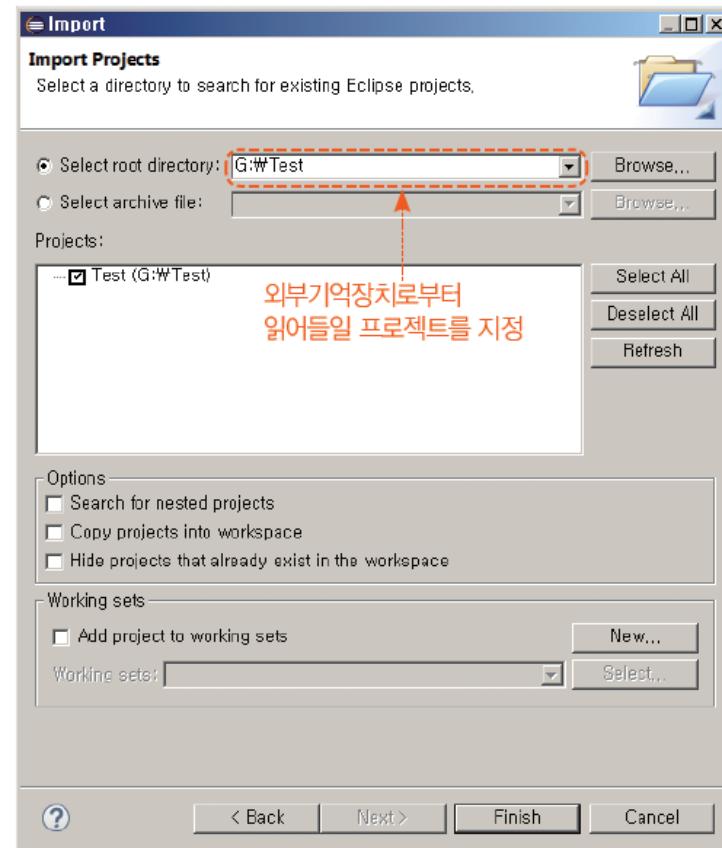
를 선택



자바 프로그램의 실행환경

■ 이클립스 프로젝트 입력

- Step 2 : 팝업창에서 외부장치로부터 읽어 들일 폴더를 선택



자바 프로그램의 실행환경

■ 자바 API 참조하기

- 자바에서 제공되는 클래스 라이브러리 (API)
- 자바 프로그램에서 클래스 라이브러리의 사용은 필수적
- 프로그래머는 라이브러리를 사용하기 위해서는 클래스의 속성과 기능을 숙지
- Java.sun.com에서 클래스 라이브러리의 속성과 기능 제공

자바 프로그램의 실행환경

■ 자바 API 참조하기

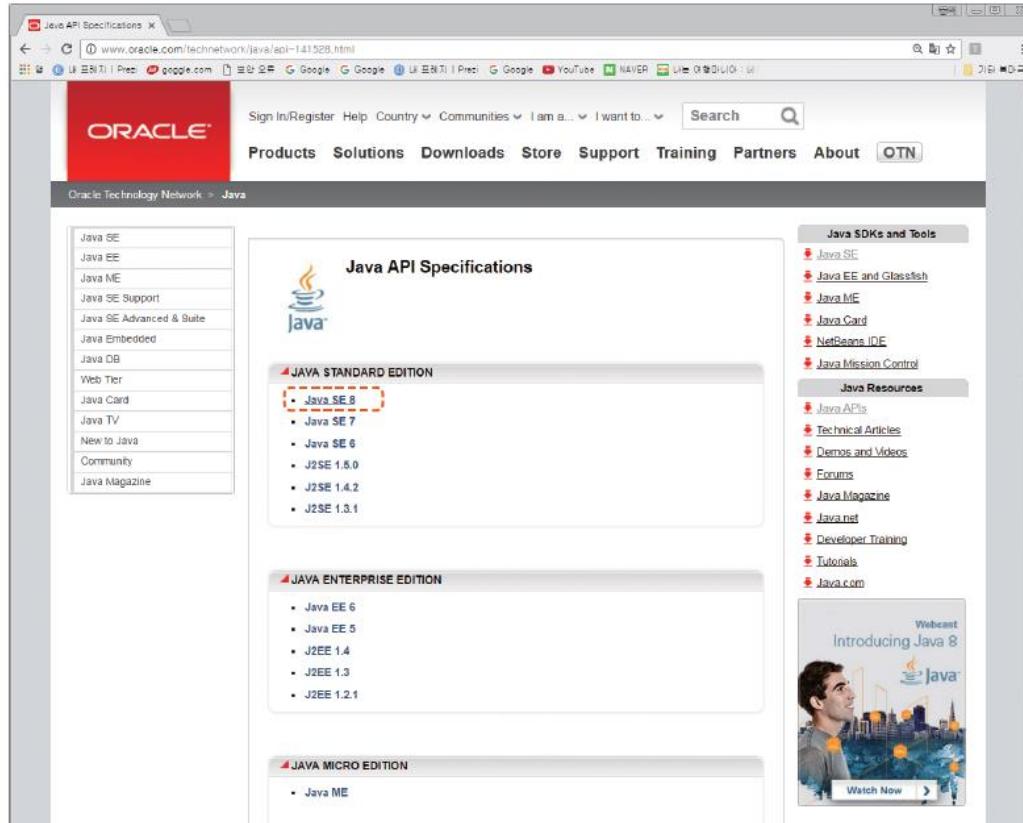
■ Step 1 : java.sun.com에 접근해 java APIs 선택



자바 프로그램의 실행환경

■ 자바 API 참조하기

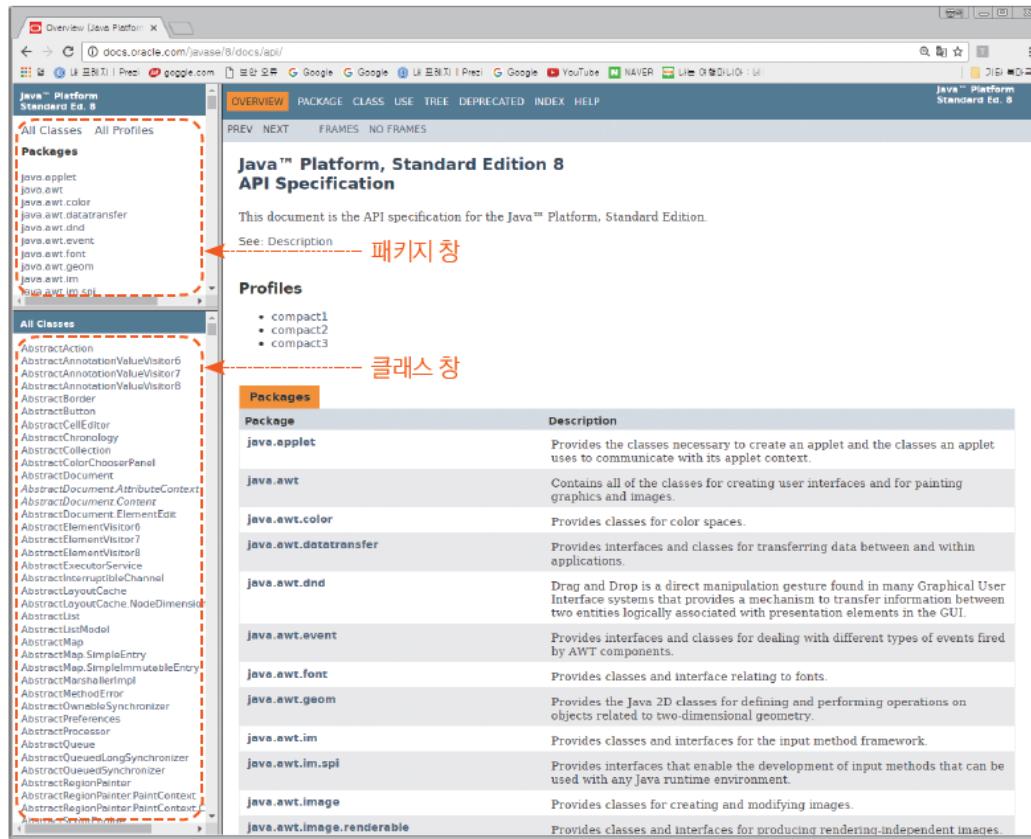
■ Step 2 : 가장 최근 버전을 선택



자바 프로그램의 실행환경

■ 자바 API 참조하기

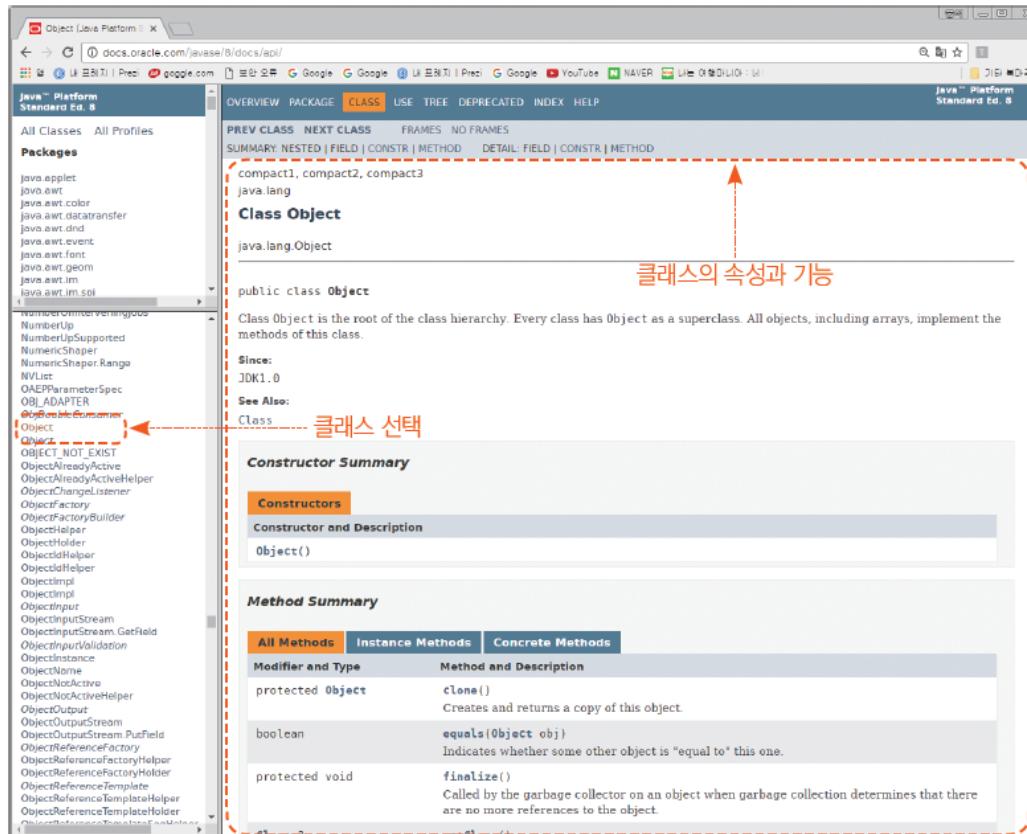
■ Step 2 : 가장 최근 버전을 선택



자바 프로그램의 실행환경

■ 자바 API 참조하기

■ Step 3 : 원하는 클래스를 선택



자바 학습을 위한 준비

■ 자바의 입출력 방법이나 오류 익히기

- 자바의 입출력은 라이브러리 클래스를 이용
- 입출력 방법을 상용 구문처럼 외워서 사용
- 프로그램을 설명하는 주석문에 대해 학습
- 프로그램 작성시 발생하는 오류에 대해 학습

자바 학습을 위한 준비

- 간단한 자바 프로그램과 문자열의 출력
- 실습 예제 2.1

예제 2.1 FirstProgram.java

```
01: public class FirstProgram {  
02:  
03:     public static void main(String[] args) {  
04:         System.out.println("처음 시작하는 자바 프로그램입니다");  
05:     }  
06: }
```

실행 결과

```
처음 시작하는 자바 프로그램입니다
```

자바 학습을 위한 준비

■ 문자열의 출력

- 자바 언어는 표준 출력문으로

System.out.println() 문장을 제공

```
System.out.println("처음 시작하는 자바 프로그램입니다"); ← 출력 후 줄을 바꿉니다
```

```
System.out.print("처음 시작하는 자바 프로그램입니다"); ← 출력 후 줄을 바꾸지 않습니다
```

자바 학습을 위한 준비

■ 간단한 자바 프로그램과 문자열의 출력

■ 실습 예제 2.2

예제 2.2 SecondProgram.java

```
01: public class SecondProgram {  
01:     public static void main(String[] args) {  
02:         System.out.print("두 번째로 작성해 보는 ");      ◀-내용을 출력한 다음 줄을 바꾸지 않는다  
03:         System.out.println("자바 프로그램입니다");  
04:         System.out.println("처음 시작하는 " + "자바 프로그램입니다");  
05:     }  
06: }
```

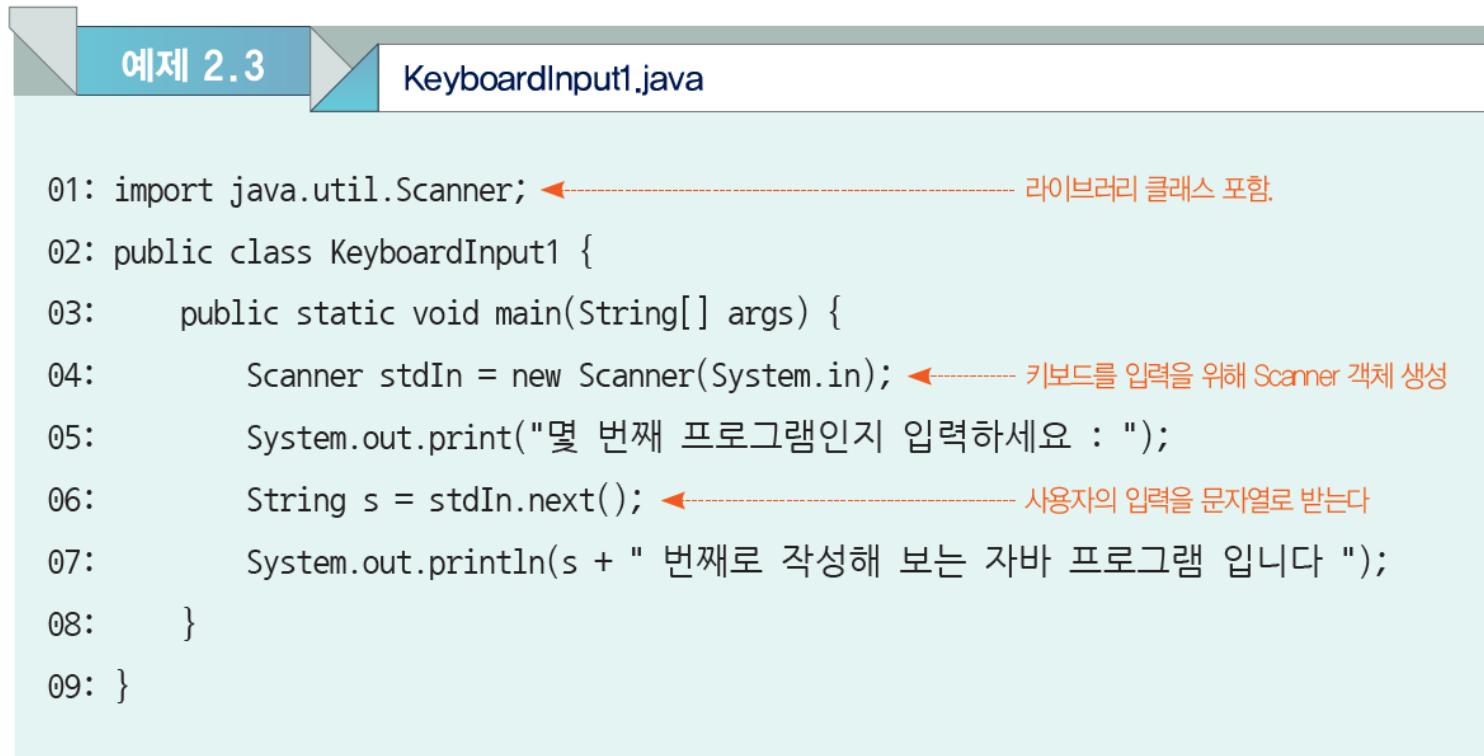
실행 결과

```
두 번째로 작성해 보는 자바 프로그램입니다  
처음 시작하는 자바 프로그램입니다
```

자바 학습을 위한 준비

■ 프로그램에 값을 입력하기 위한 방법

■ 실습 예제 2.3

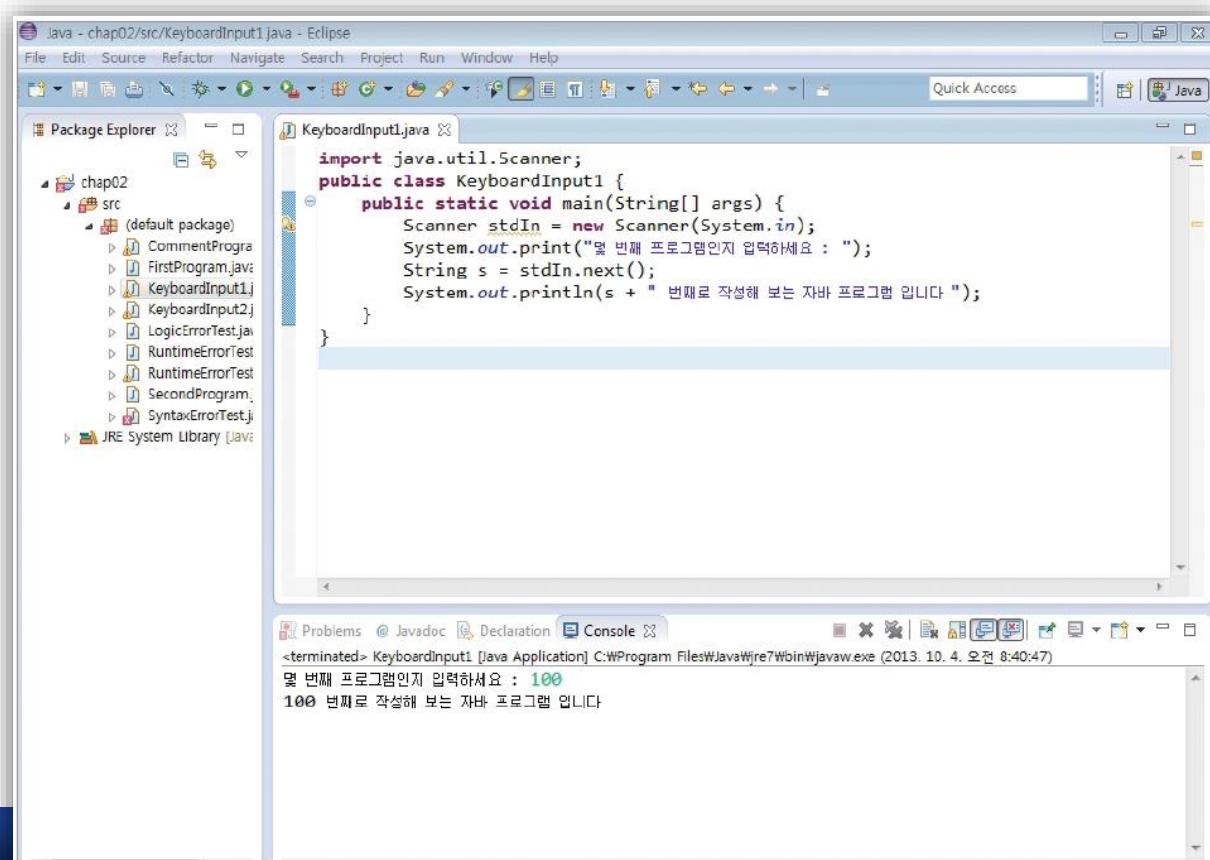


예제 2.3 KeyboardInput1.java

```
01: import java.util.Scanner;           ← 라이브러리 클래스 포함.  
02: public class KeyboardInput1 {  
03:     public static void main(String[] args) {  
04:         Scanner stdIn = new Scanner(System.in); ← 키보드를 입력을 위해 Scanner 객체 생성  
05:         System.out.print("몇 번째 프로그램인지 입력하세요 : ");  
06:         String s = stdIn.next();             ← 사용자의 입력을 문자열로 받는다  
07:         System.out.println(s + " 번째로 작성해 보는 자바 프로그램 입니다 ");  
08:     }  
09: }
```

자바 학습을 위한 준비

- 프로그램에 값을 입력하기 위한 방법
- 프로그램 실행 후 Consol창에 데이터 입력



자바 학습을 위한 준비

■ 프로그램에 값을 입력하기 위한 방법

■ 실습 예제 2.4

예제 2.4 KeyboardInput2.java

```
01: import java.util.Scanner;
02: public class KeyboardInput2 {
03:     public static void main(String[] args) {
04:         Scanner stdIn = new Scanner(System.in);
05:         System.out.print("이름과 나이, 몸무게를 공간(space)으로 구분하여 입력 : ");
06:         String name = stdIn.next(); // 키보드로부터 이름을 문자열로 입력
07:         int i = stdIn.nextInt(); // 키보드로부터 나이를 정수로 입력
08:         double d = stdIn.nextDouble(); // 키보드로부터 몸무게를 실수로 입력
09:         System.out.println(name + "씨의 나이는 " + i + "세이고");
10:         System.out.println("몸무게는 " + d + "Kg 입니다");
11:     }
12: }
```

실행 결과 입력 값으로 홍길동과 22, 62.5를 입력

홍길동 씨의 나이는 22세이고
몸무게는 62.5Kg 입니다

자바 학습을 위한 준비

- 프로그램에 값을 입력하기 위한 방법
- 실행 시 데이터 입력은 공간으로 구분

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "chap02/src". It includes files like CommentProgram, FirstProgram.java, KeyboardInput1.java, KeyboardInput2.java, LogicErrorTest.java, RuntimeErrorTest.java, RuntimeErrorTest.java, SecondProgram.java, and SyntaxErrorTest.java.
- Editor:** Displays the content of `KeyboardInput2.java`. The code uses `Scanner` to read input from `System.in`. It prompts for name and age separated by a space, then prints them out. The code is as follows:

```
import java.util.Scanner;
public class KeyboardInput2 {
    public static void main(String[] args) {
        Scanner stdIn = new Scanner(System.in);
        System.out.print("이름과 나이, 몸무게를 공란(space)으로 구분하여 입력 : ");
        String name = stdIn.next();
        int i = stdIn.nextInt();
        double d = stdIn.nextDouble();
        System.out.println(name + "씨의 나이는 " + i + "세이고");
        System.out.println("몸무게는 " + d + "Kg 입니다");
    }
}
```

- Console:** Shows the terminal output of the application. The user inputs "홍길동 22 62.5" and the program outputs:
<terminated> KeyboardInput2 [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (2013. 10. 4. 오전 8:46:45)
이름과 나이, 몸무게를 공란(space)으로 구분하여 입력 : 홍길동 22 62.5
홍길동씨의 나이는 22세이고
몸무게는 62.5Kg 입니다

자바 학습을 위한 준비

■ 주석문과 프로그램 오류

■ 주석문

- 프로그램의 구조와 동작 방법을 설명하기 위한 방법
- 프로그램의 실행에는 영향을 미치지 않음
- 다른 개발자가 프로그램을 수정하거나 보완 시 유용

자바의 주석문은 다음과 같이 3가지 형태로 많이 사용됩니다.

- ① /* */ : 여러 줄의 문장을 주석으로 처리할 수 있습니다.
- ② // : 한 줄의 문장을 주석으로 처리합니다.
- ③ /** */ : 여러 줄의 문장을 주석으로 처리하고, javadoc 도구에 의해 사양서를 만들 수 있는 주석입니다.

자바 학습을 위한 준비

■ 주석문과 프로그램 오류

■ 실습 예제 2.5 : 주석문을 사용한 프로그램

예제 2.5 CommentProgram.java

```
01: /**
02: * 이 클래스는 세 개의 값을 입력받아 첫 번째 값은 문자열값으로, 두 번째 값은
03: 정수값으로
04: * 세 번째 값은 실수값으로 출력하는 프로그램입니다.
05: * @author 김충석
06: *
07: import java.util.Scanner;      // Scanner 클래스를 포함
08: public class CommentProgram {
09:     public static void main(String[] args) {
10:         Scanner stdIn = new Scanner(System.in);
11:         // 다음 문장은 콘솔 창에서 입력을 요청하는 문장입니다
12:         System.out.print("이름과 나이, 몸무게를 공간(space)으로 구분하여 입력 : ");
13:         /* 표준 입력으로 문자열과 정수,
14:          * 실수를 입력받아 적합한 형의 변수에 값을 저장 */
15:         String name = stdIn.next();
16:         int i = stdIn.nextInt();
17:         double d = stdIn.nextDouble();
18:         System.out.println(name + "씨의 나이는 " + i + "세이고");
19:         System.out.println("몸무게는 " + d + "Kg 입니다"); // 저장된 값을 출력
20:     }
21: }
```

자바 학습을 위한 준비

■ 주석문과 프로그램 오류

■ 프로그래밍 오류

□ 능숙한 프로그래머도 오류를 피해갈 수는 없다.

■ 오류의 종류

□ 구문(Syntax) 오류 : 이클립스에 의해 자동 오류 표시

□ 실행 시간 (Runtime)오류 : JVM에 의해 오류 발생

□ 논리(Logic)오류 : 프로그래머에 의해 발견되어야 수 정 가능

자바 학습을 위한 준비

■ 주석문과 프로그램 오류

■ 실습 예제 2.6 : 구문 오류

예제 2.6 SyntaxErrorTest.java

```
01: public class SyntaxErrorTest {  
02:     public static void main(String args[]) {  
03:         i = 30 ; ← 변수를 선언하지 않고 사용  
04:         System.out.println( i ) ← 문자 종료 기호가 없음  
05:     }  
06: }
```

실행 결과 오류 메시지 출력

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

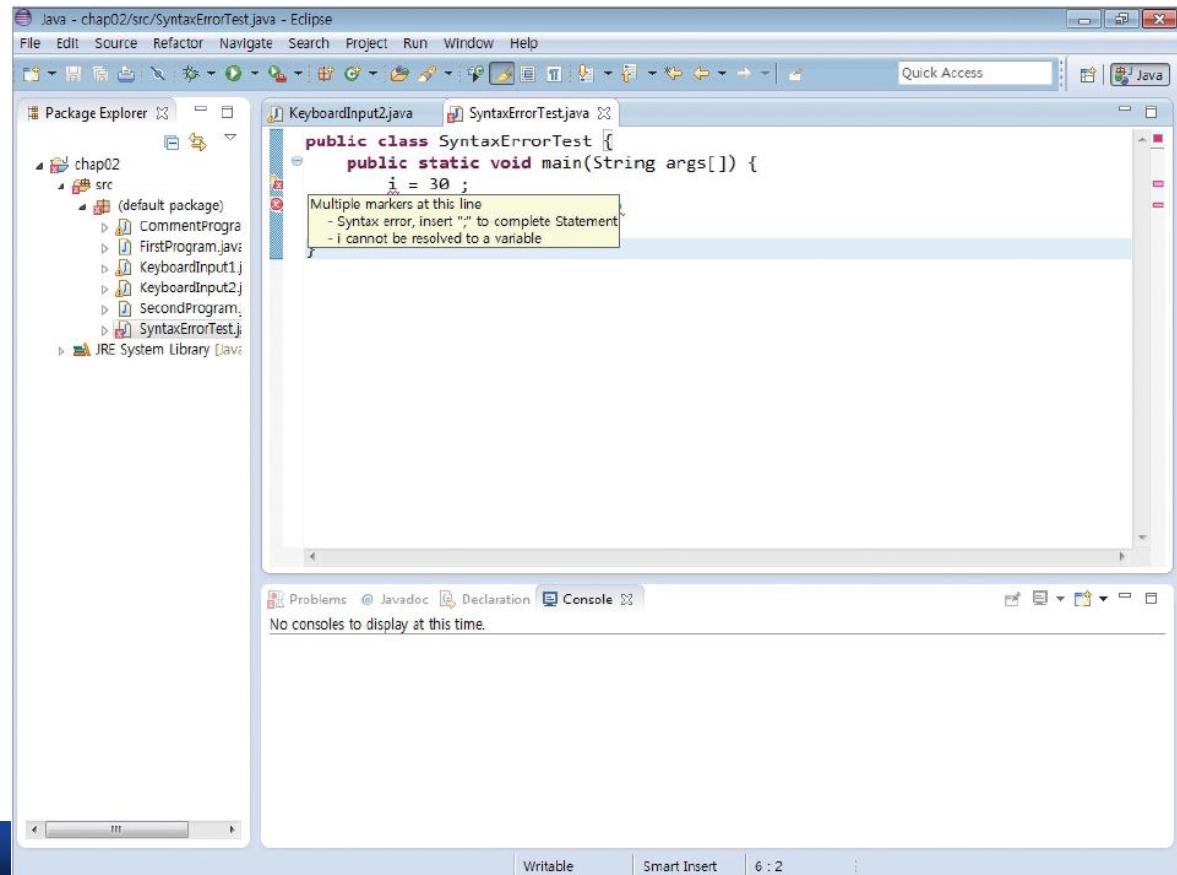
i cannot be resolved to a variable
i cannot be resolved to a variable
Syntax error, insert ";" to complete Statement

at SyntaxErrorTest.main(SyntaxErrorTest.java:3)

자바 학습을 위한 준비

■ 주석문과 프로그램 오류

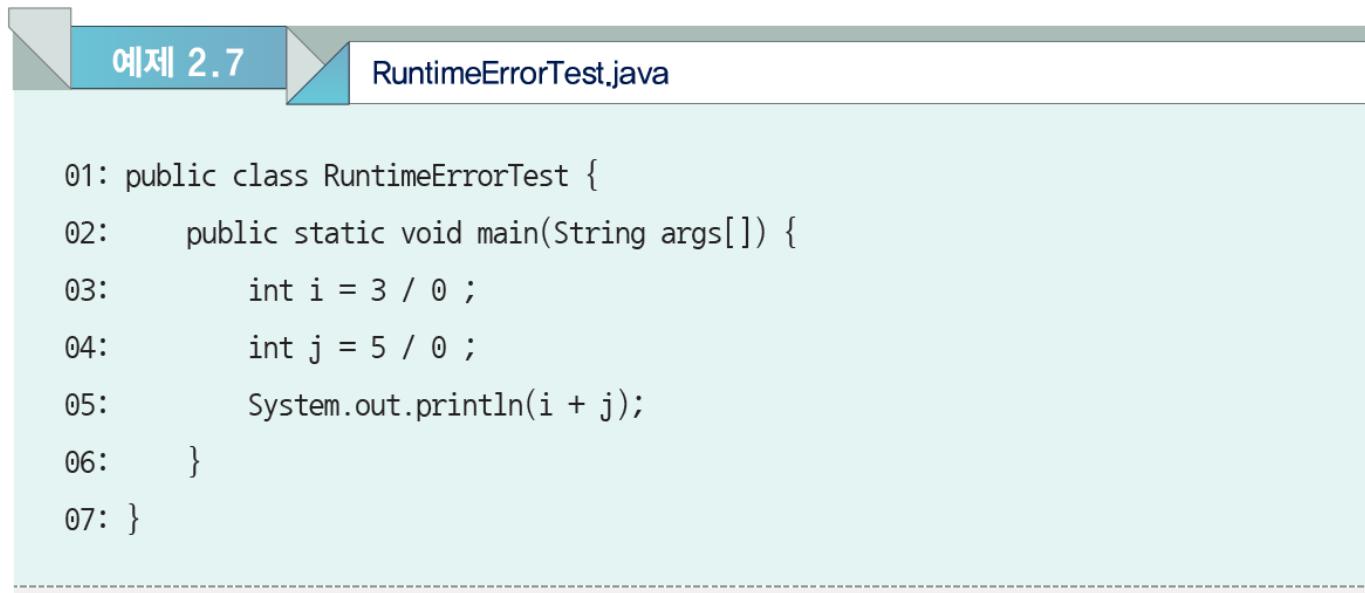
■ 실습 예제 2.6 : 구문 오류 (이클립스에서 적색 으로 표시)



자바 학습을 위한 준비

■ 주석문과 프로그램 오류

■ 실습 예제 2.7 : 프로그래밍 오류



예제 2.7 RuntimeErrorTest.java

```
01: public class RuntimeErrorTest {  
02:     public static void main(String args[]) {  
03:         int i = 3 / 0 ;  
04:         int j = 5 / 0 ;  
05:         System.out.println(i + j);  
06:     }  
07: }
```

자바 학습을 위한 준비

■ 주석문과 프로그램 오류

- 실행 시간 오류 메시지 출력
 - 3번 라인에서 실행시간 오류 발생하고 프로그램 종료

실행 결과 실행 시간 오류 메시지 출력

```
Exception in thread "main" java.lang.ArithmetricException: / by zero  
at RuntimeErrorTest.main(RuntimeErrorTest.java:3)
```

- 3번 라인을 수정한 다음 실행시키면 4번 라인에서 실행 시간 오류

실행 결과 3번 라인의 오류를 수정한 다음 실행

```
Exception in thread "main" java.lang.ArithmetricException: / by zero  
at RuntimeErrorTest.main(RuntimeErrorTest.java:4)
```

자바 학습을 위한 준비

■ 주석문과 프로그램 오류

■ 이클립스의 실행 시간 오류

The screenshot shows the Eclipse IDE interface. The top bar displays "Java - chap02/src/RuntimeErrorTest1.java - Eclipse". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The toolbar has various icons for file operations. The Package Explorer view on the left shows a project named "chap02" with several source files: CommentProgram, FirstProgram.java, KeyboardInput1.java, KeyboardInput2.java, RuntimeErrorTest, RuntimeErrorTest1.java, SecondProgram.java, and SyntaxErrorTest.java. The Java perspective is selected. In the center, the code editor shows the following Java code:

```
import java.util.Scanner;
public class RuntimeErrorTest1 {
    public static void main(String[] args) {
        Scanner stdIn = new Scanner(System.in);
        int a = stdIn.nextInt();
        int b = stdIn.nextInt();
        System.out.println("입력받은 두 수의 합 : " + (a+b));
    }
}
```

At the bottom, the Problems view shows an error message from the console:

```
<terminated> RuntimeErrorTest1 [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (2013. 10. 3. 오후 2:00:46)
kim 3
Exception in thread "main" java.util.InputMismatchException
at java.util.Scanner.throwFor(Unknown Source)
at java.util.Scanner.next(Unknown Source)
at java.util.Scanner.nextInt(Unknown Source)
at java.util.Scanner.nextInt(Unknown Source)
at RuntimeErrorTest1.main(RuntimeErrorTest1.java:5)
```

자바 학습을 위한 준비

■ 주석문과 프로그램 오류

■ 논리 오류

- 프로그램 작성자의 의도와 다른 결과를 나타내는 오류
- 논리 오류는 프로그램 작성자나 수정하는 사람에 의해 발견되어야 함.

자바 학습을 위한 준비

■ 주석문과 프로그램 오류

■ 실습 예제 2.8 : 논리 오류

예제 2.8 LogicError.java

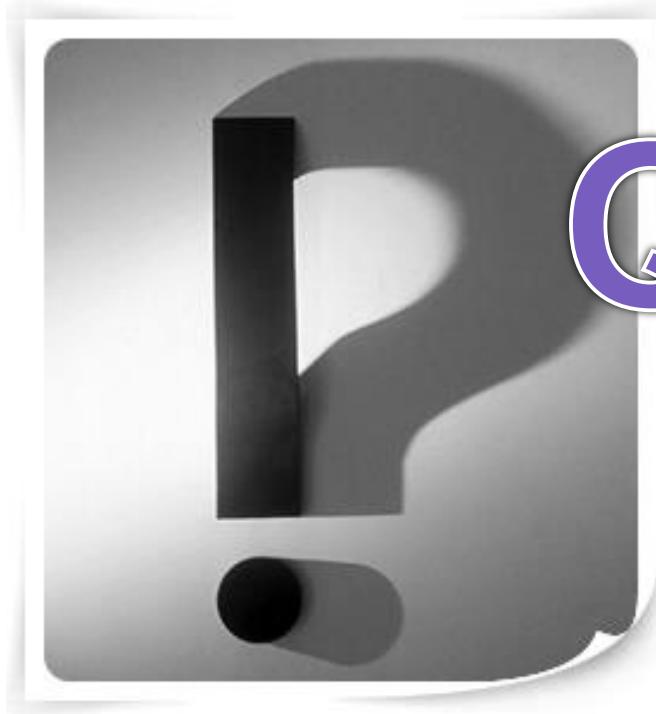
```
01: public class LogicErrorTest {  
02:     public static void main(String args[]) {  
03:         int i = 300 ;  
04:         int j = 500 ;  
05:         j += i + j ;  
06:         System.out.println("합계는 = " + j) ;  
07:     }  
08: }
```

» 설명

05 작성자의 의도는 $j = i + j$ 인데 잘못 기술됨

실행 결과

합계는 = 1300



Q & A

- Thank you for your attention