

## 8주 3강

# 기본 자바 소개 1



# 이번 주차에는...

## 기본 자바 소개 1

- 실무에 사용되는 자바 개발 환경
- 자바 프로그램 작성

# 1. 실무에서 사용하는 JAVA 개발 환경(1)

- 실무에서 JAVA를 개발할 때는 전용 툴(tool)을 사용, 다양한 JAVA 개발 툴 중에서도 이클립스(Eclipse)가 가장 많이 사용
- 이클립스 : JAVA 프로그램을 작성하도록 도와주는 편리하고 강력한 기능 많이 가지고 있으며 무료. 사용법도 어렵지 않아서 대부분의 JAVA 프로그래머가 이용하는 개발 툴



## 2. 이클립스 배포판

### ■ 이클립스 배포판

- 이클립스는 다양한 언어의 프로그래밍을 지원하는 세계적으로 널리 사용되는 통합 개발 환경으로 JAVA 언어 및 안드로이드 개발에 필수 도구
- 이클립스는 오픈 소스로 공개되어 라이선스 비용을 지불하지 않고 무료 사용. 이클립스 4.9 부터 Java 11 지원

표 2-1 이클립스 배포판

배포판 이름	버전	배포 일자
Callisto	3.2	2006년 6월
Europa	3.3	2007년 6월
Ganymede	3.4	2008년 6월
Galileo	3.5	2009년 6월
Helios	3.6	2010년 6월
Indigo	3.7	2011년 6월
Juno	4.2	2012년 6월
Kepler	4.3	2013년 6월
Luna	4.4	2014년 6월
Mars	4.5	2015년 6월
Neon	4.6	2016년 6월
Oxygen	4.7	2017년 6월
Photon	4.8	2018년 6월
	4.9	2018년 9월

### 3. 실무에서 사용하는 JAVA 개발 환경(2)

- 이클립스 설치

- 이클립스 다운로드하고 설치

- <http://www.eclipse.org/downloads/packages/release/2018-09/r>에 접속, 'Eclipse IDE for Java Developers' 오른쪽의 [64bit] 클릭. 미리 사이트 열리면 다운로드 모양의 아이콘 클릭하여 다운로드

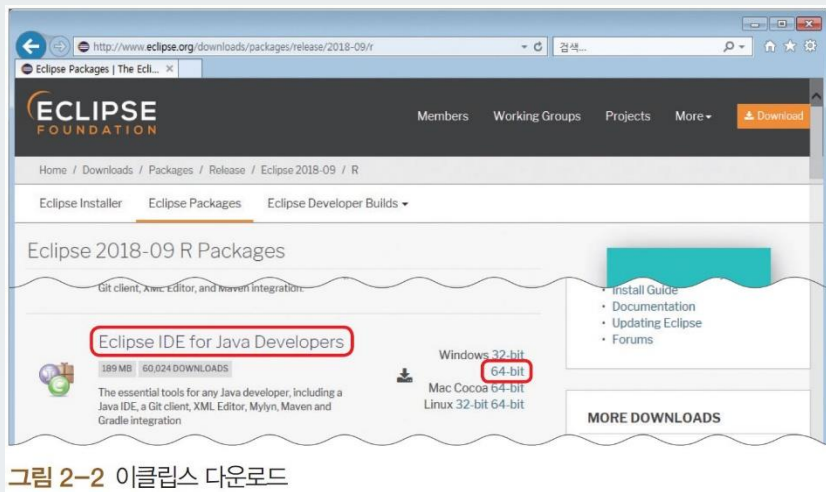


그림 2-2 이클립스 다운로드

- 다운로드한 파일 이름은 eclipse-java-2018-09-win32-x86\_64.zip

## 4. 실무에서 사용하는 JAVA 개발 환경(3)

- 파일 탐색기 실행하여 다운로드한 파일 선택, 마우스 오른쪽 버튼 클릭, [압축 풀기] 선택

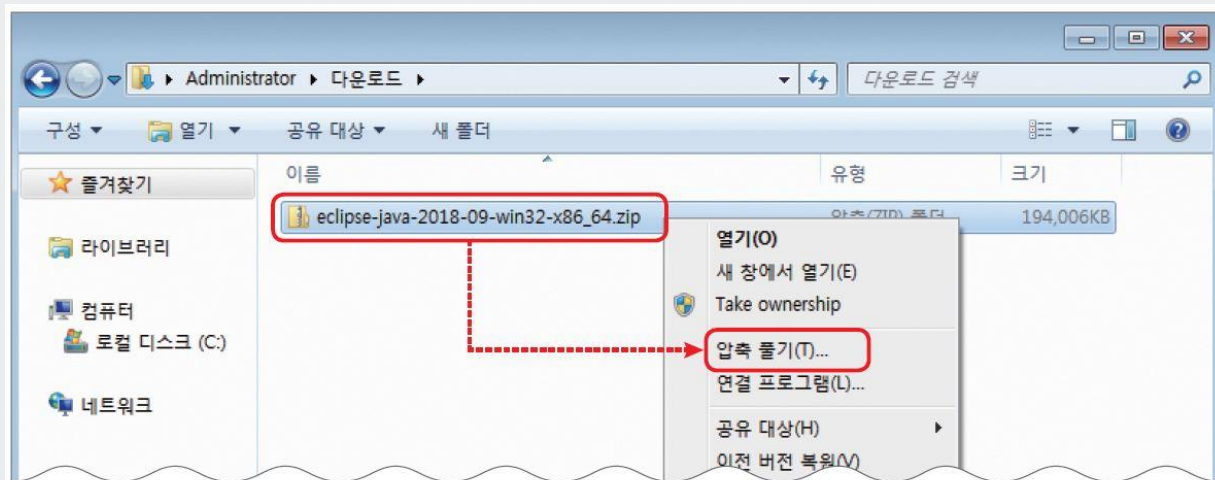


그림 2-3 이클립스 설치 1

## 5. 실무에서 사용하는 JAVA 개발 환경(4)

- 압축이 풀린 폴더 안의 'eclipse' 폴더를 통째로 C:\w 폴더로 옮기면 C:\w\eclipse\workspace.exe 파일 보임(이것이 이클립스의 실행 파일). 이 파일을 바탕화면에 바로가기 아이콘으로 만들기

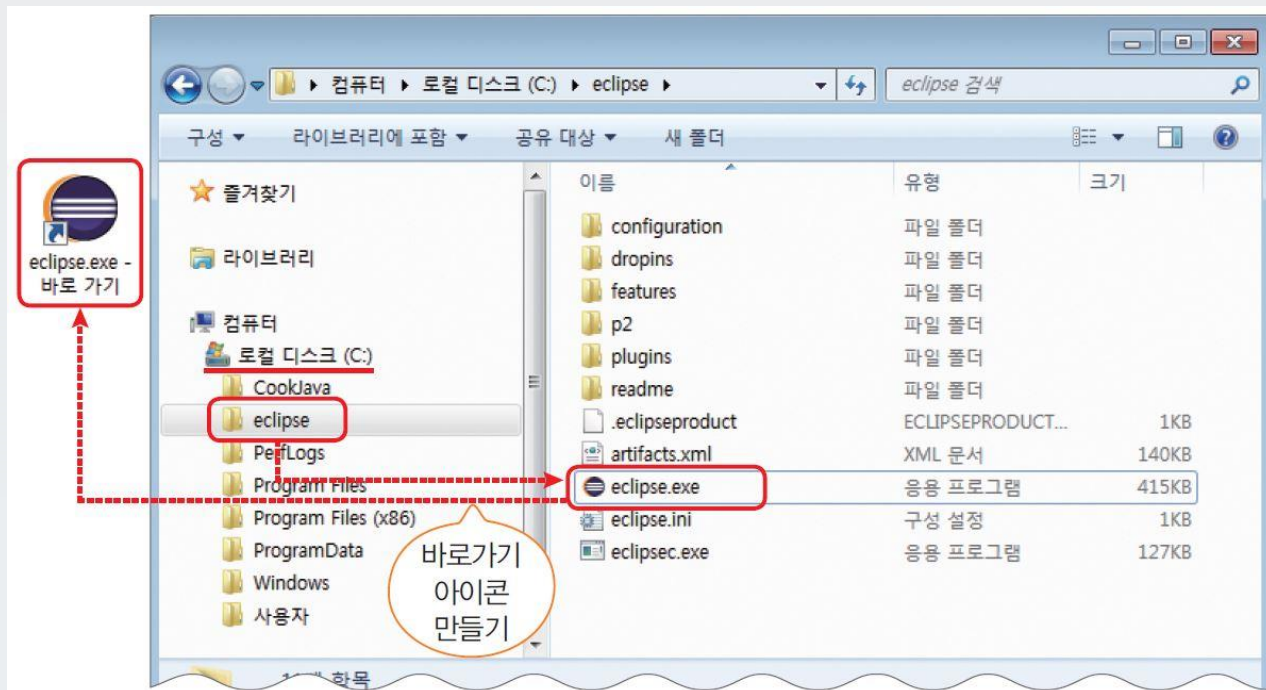
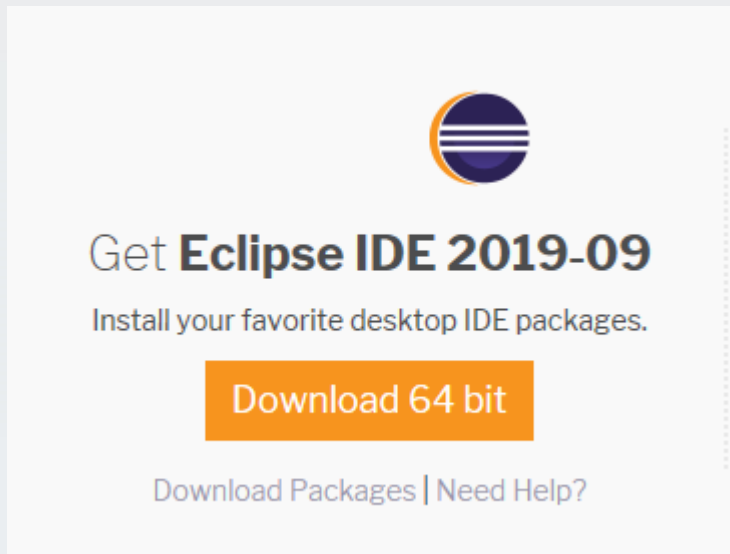


그림 2-4 이클립스 설치 2

## 6. 실무에서 사용하는 JAVA 개발 환경(5)

- 이클립스 실행 위해 바탕화면의 바로가기 아이콘 더블클릭, 이클립스 로고 화면 나옴
- 이클립스 다운로드 : <https://www.eclipse.org/>





## 7. 실무에서 사용하는 JAVA 개발 환경(6)

- 'Select a directory as workspace' 창에서 C:\₩CookJava₩라고 직접 써서 변경 후 [Launch] 클릭

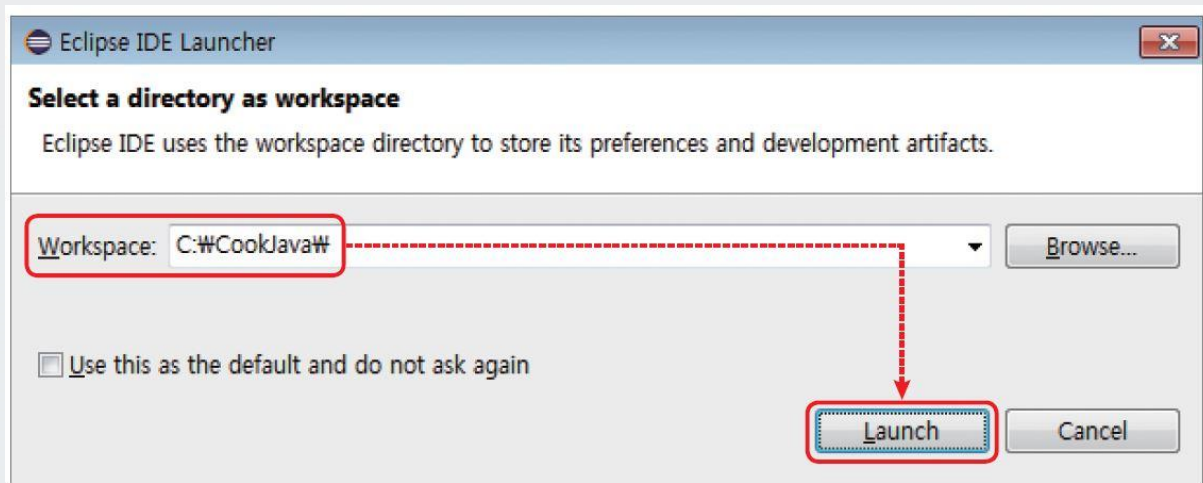


그림 2-6 이클립스 워크스페이스 지정

## 8. 실무에서 사용하는 JAVA 개발 환경(7)

- 이클립스 초기 화면 나옴

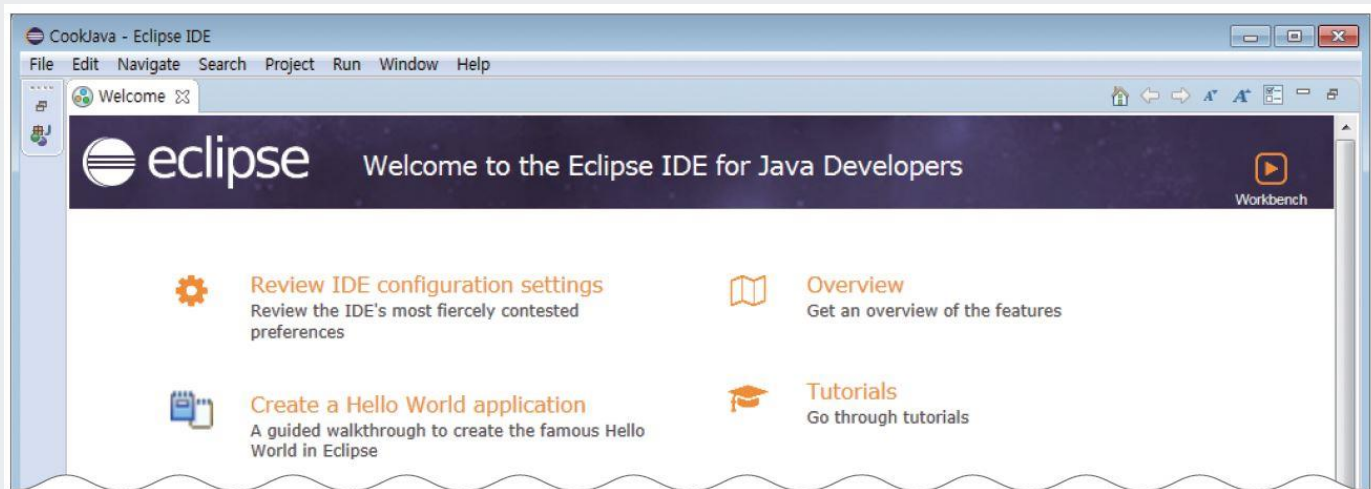


그림 2-7 이클립스 초기 화면

## 9. 실무에서 사용하는 JAVA 개발 환경(8)

- 개발이 편리하도록 이클립스 설정 변경하기
  - 행 번호 보이게 하기 : 앞으로 코딩할 소스의 행 번호가 보이도록 설정. Eclipse 메뉴의 [Window]-[Preferences]를 선택, 왼쪽 트리에서 [General]-[Editors]-[Text Editors] 선택, 오른쪽의 'Show line numbers'가 체크되어 있는지 확인. 체크되어 있지 않으면 체크

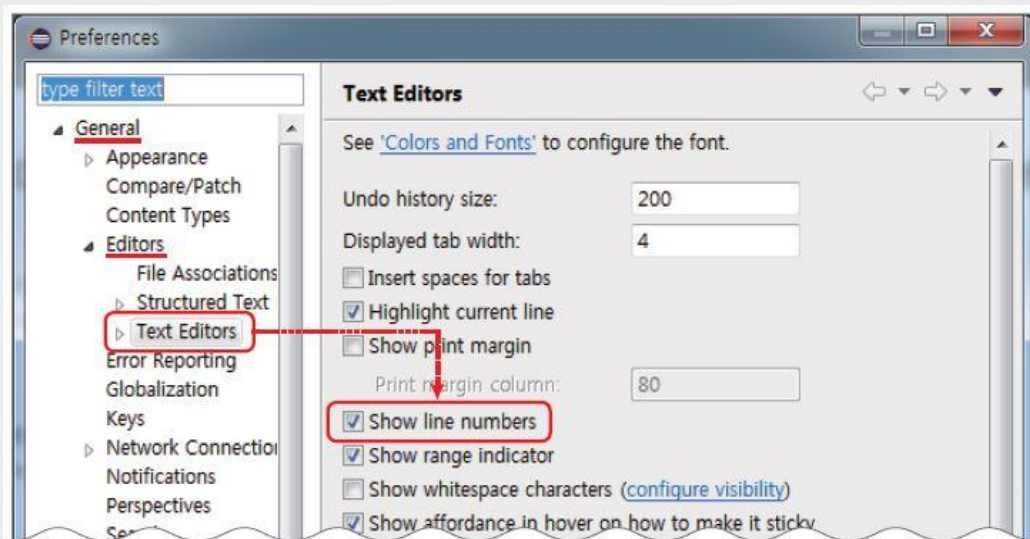


그림 2-8 이클립스 설정 : 행 번호 보이게 하기

# 10. 실무에서 사용하는 JAVA 개발 환경(9)

- 폰트 변경하기 : 왼쪽 트리의 [General]-[Appearance]-[Colors and Fonts] 선택. 오른쪽 트리의 [Basic]-[Text Font] 선택, [Edit] 클릭, 원하는 글꼴, 크기 등 선택, [확인] 클릭. [OK]를 클릭하여 'Preferences' 창 종료

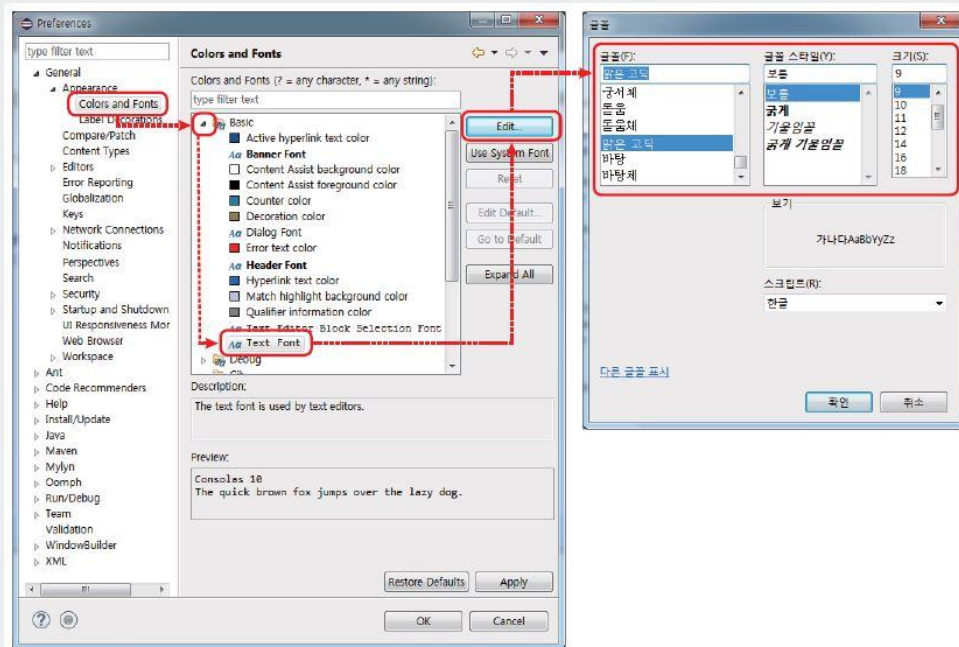


그림 2-9 이클립스 설정 : 폰트 변경하기

# 11. 실무에서 사용하는 JAVA 개발 환경(10)

- 자동 빌드 기능 끄기 : 이클립스는 기본적으로 코드 입력 후 자동으로 빌드(컴파일)되도록 설정되어 있음. 메뉴에서 [Project]-[Build Automatically] 선택, 체크되어 있는 것 해제

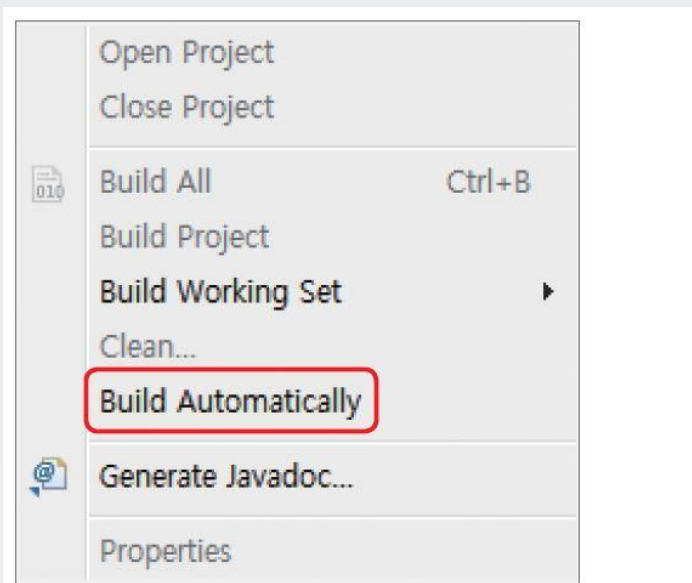


그림 2-10 이클립스 설정 : 자동 빌드 기능 끄기

## 12. 실무에서 사용하는 JAVA 개발 환경(11)

- 초기 Welcome 화면 필요 없으니 오른쪽 아래의 'Always show Welcome at start up'의 체크를 끄고 'Welcome' 옆의 [x] 클릭하여 창 닫기



그림 2-11 Welcome 화면 닫기

# 13. 실무에서 사용하는 JAVA 개발 환경(12)

- 다음과 같은 프로젝트 창이 나옴. 주로 이 화면에서 JAVA 프로그래밍을 할 것. 일단 이클립스 종료

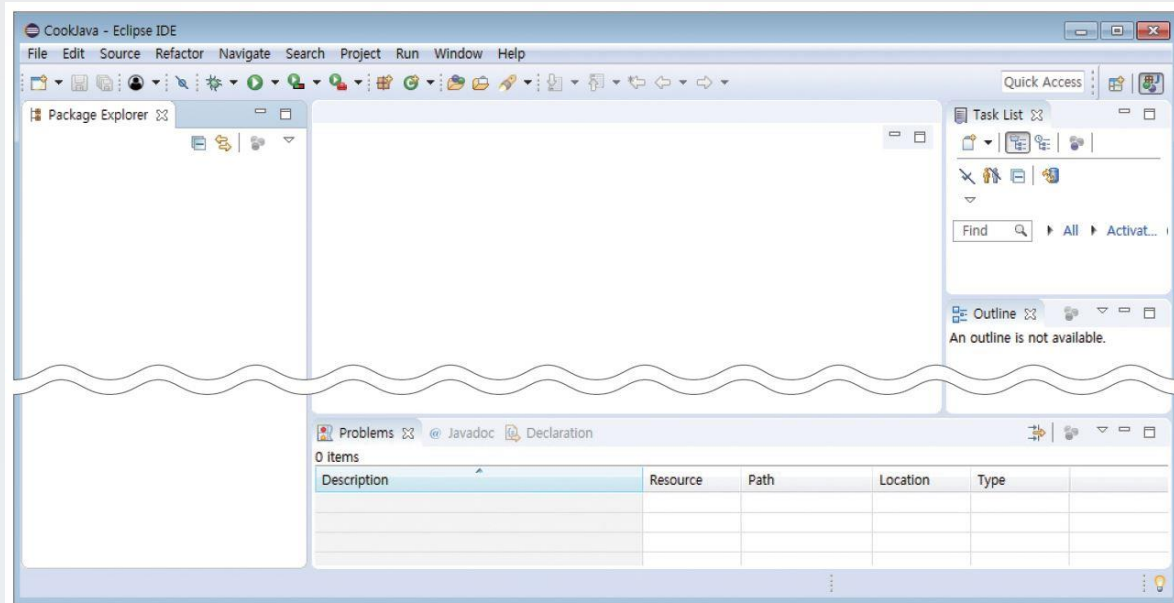


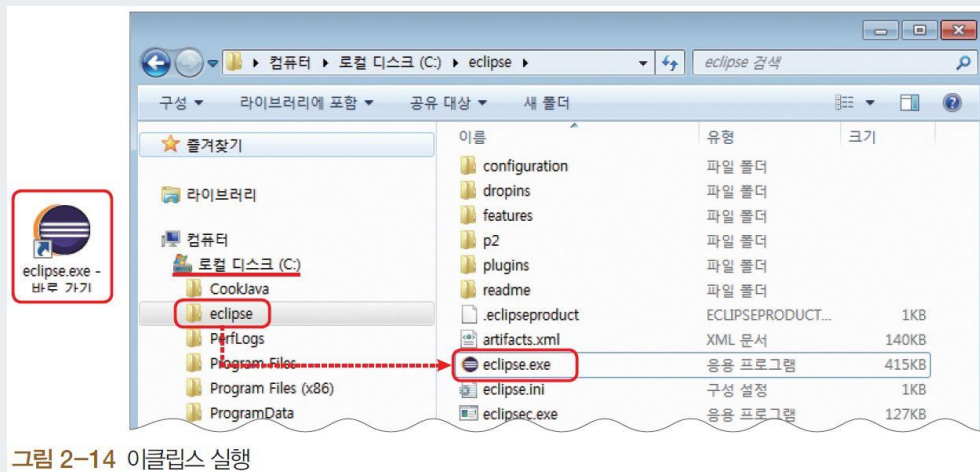
그림 2-12 초기 이클립스 화면

# 14. 자바 프로그램 작성(1)

- 프로그램 문법은 신경 쓰지 말고 전반적으로 프로그램 작성 순서에 집중해서 실습



- 프로젝트 생성(프로젝트 이름 : 'Chapter02')
  - 바탕화면의 이클립스 바로가기 아이콘 더블클릭, 또는 C:\WeclipseWeclipse.exe 파일 더블클릭





## 15. 자바 프로그램 작성(2)

- 'Select a directory as workspace' 창에서 C:\₩CookJava₩ 확인하고 [Launch] 클릭

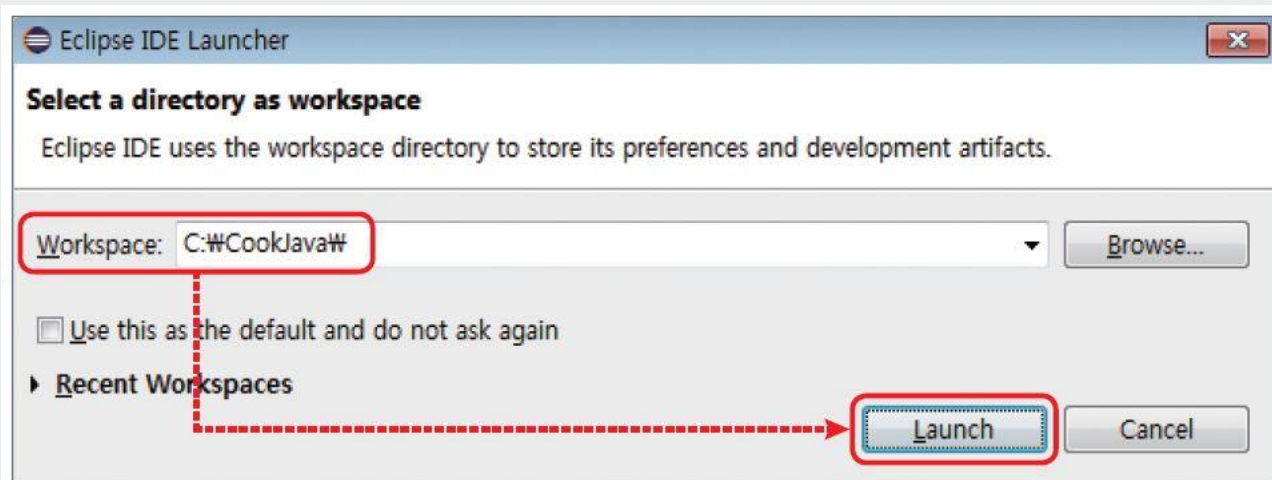


그림 2-15 워크스페이스 확인

# 16. 자바 프로그램 작성(3)

- 프로젝트 생성 위해 메뉴의 [File]-[New]-[Java Project] 선택

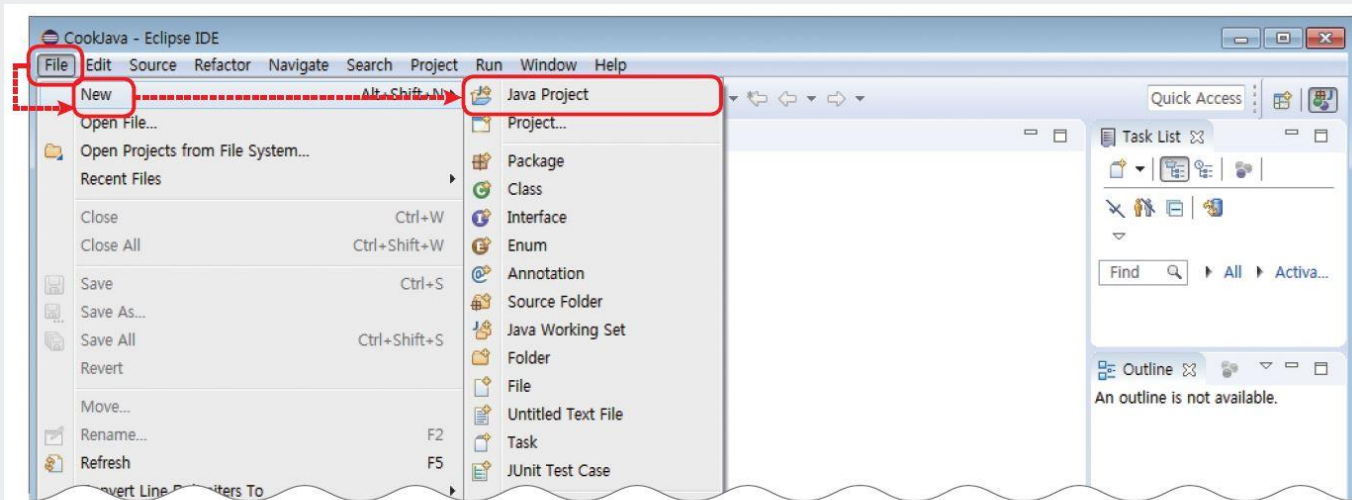


그림 2-16 새 프로젝트 생성

# 17. 자바 프로그램 작성(4)

- 'Create a Java Project' 창의 'Project name'에 프로젝트 이름(Chapter02) 입력, 하위 호환성 위해 'Use an execution environment JRE:'를 'JavaSE-1.8'로 선택한 후 [Next] 클릭

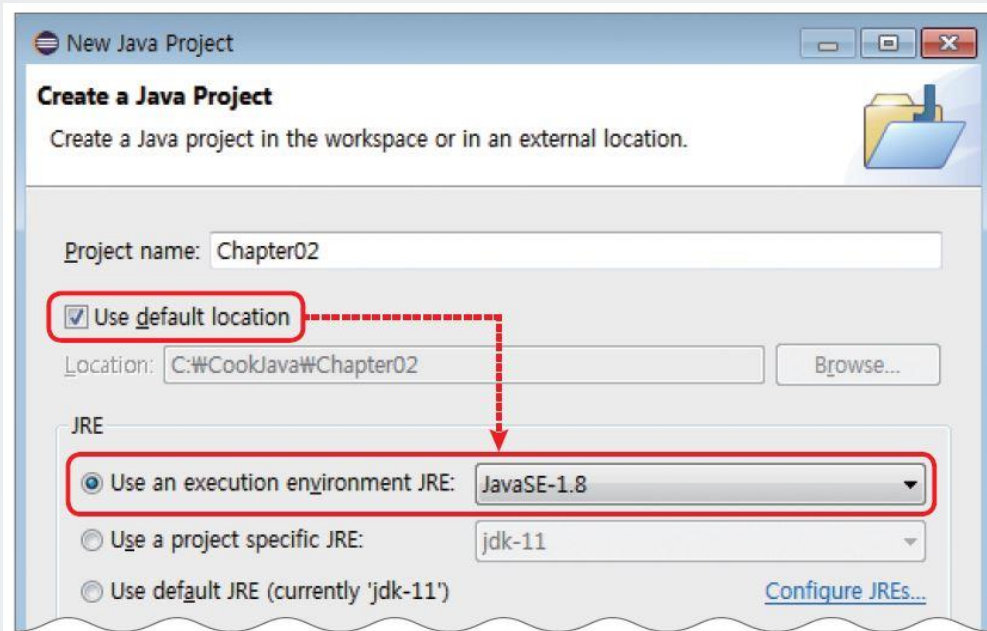


그림 2-17 프로젝트 이름 입력

## 18. 자바 프로그램 작성(5)

- 'Java Settings' 창에서는 특별한 변경 없음. [Finish] 클릭

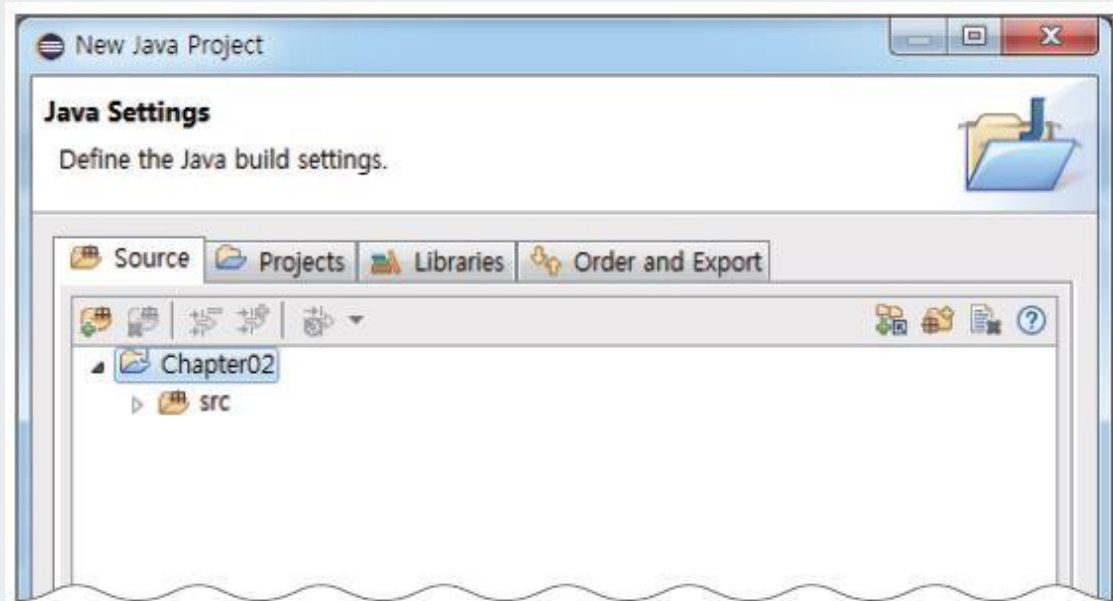


그림 2-18 JAVA 설정 창

# 19. 자바 프로그램 작성(6)

- 최종적으로 다음과 같이 왼쪽 'Package Explorer'에 Chapter02 프로젝트 생성 앞으로 프로젝트를 추가로 생성하면 계속 쌓임

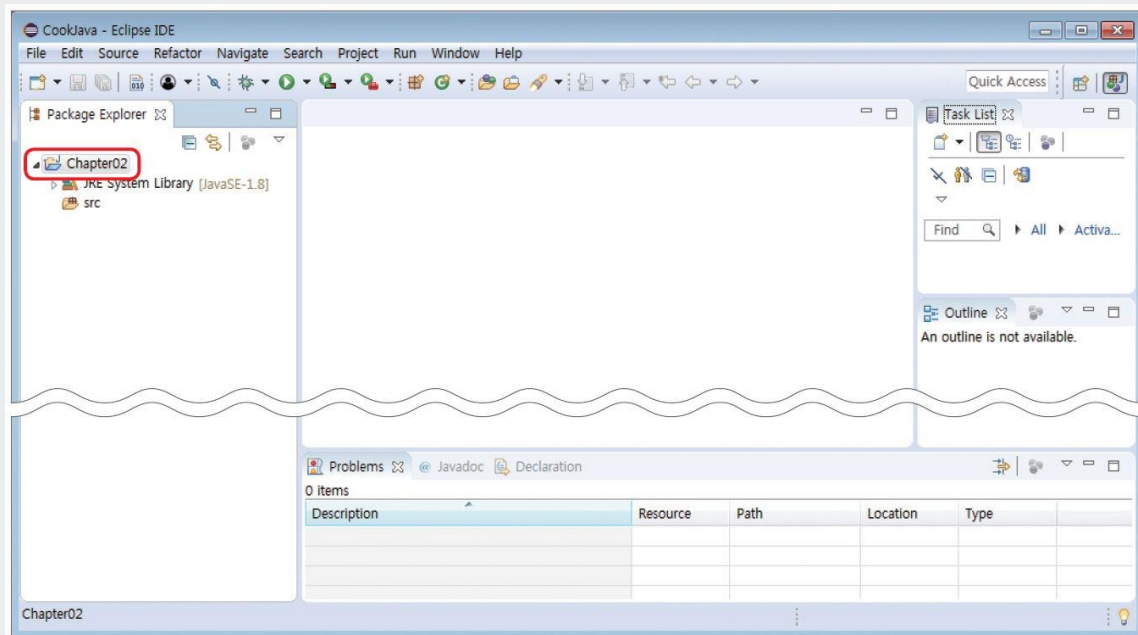


그림 2-19 생성된 프로젝트

## 20. 자바 프로그램 작성(7)

- 왼쪽 'Package Explorer'의 프로젝트 이름(Chapter02) 아래 'src' 폴더에서 마우스 오른쪽 버튼 클릭, [New]-[Class] 선택

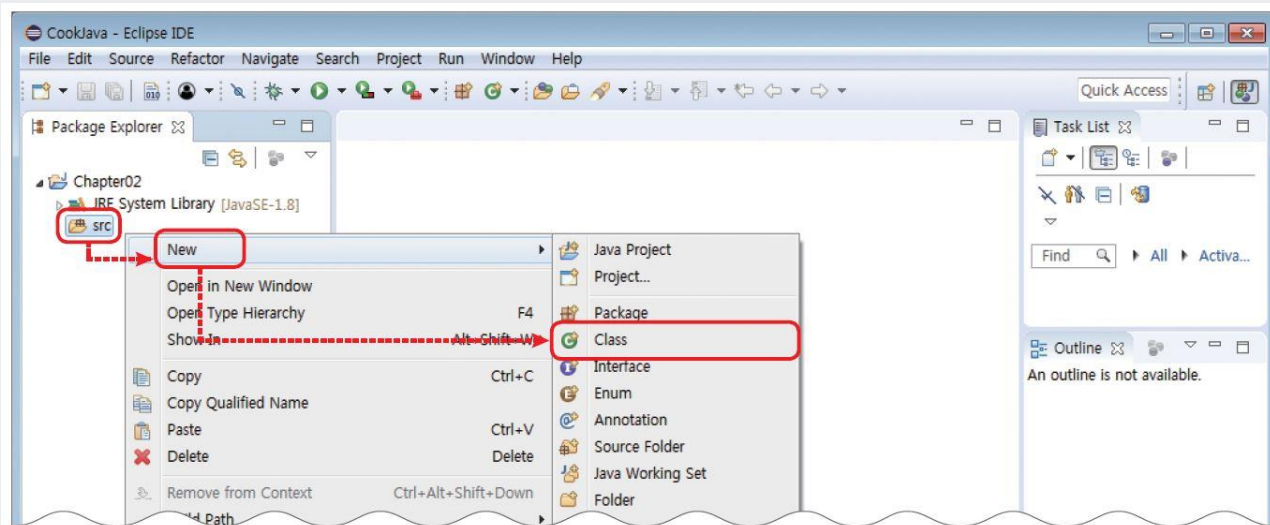


그림 2-20 JAVA 소스코드 추가 1

## 21. 자바 프로그램 작성(8)

- 'Java Class' 창에서 'Name'에 JAVA 소스코드 이름 입력('Ex02\_01')
- 아래쪽의 'public static void main(String[] args)' 체크, [Finish] 클릭

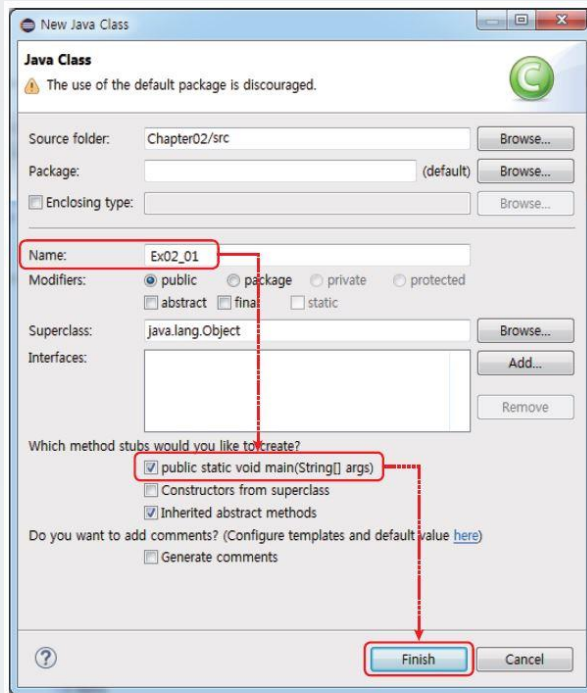


그림 2-21 JAVA 소스코드 추가 2

## 22. 자바 프로그램 작성(9)

- 오른쪽 코드 편집 창에 100과 50의 더하기, 빼기, 곱하기, 나누기 수행 프로그램 코딩

실습 2-1 두 번째 프로그램 작성

```
01 public class Ex02_01 {  
02  
03     public static void main(String[] args) {  
04         // TODO Auto-generated method stub  
05         int a, b;      — 정수형 변수 a와 b를 선언한다.  
06         int result;    — 정수형 변수 result를 선언한다.  
07  
08         a = 100;        — a에는 100을, b에는 50을 대입한다.  
09         b = 50;  
10  
11         result = a + b; — a와 b를 더한 결과를 result에 넣는다.  
12         System.out.println(a + "+" + b + "=" + result); — 화면에 a, b,  
13                                                         result를 출력한다.  
14  
14         result = a - b; — a와 b를 뺀 결과를 result에 넣는다.  
15         System.out.println(a + "-" + b + "=" + result); — 화면에 a, b,  
16                                                         result를 출력한다.  
17  
17         result = a * b; — a와 b를 곱한 결과를 result에 넣는다.  
18         System.out.println(a + "*" + b + "=" + result); — 화면에 a, b,  
19                                                         result를 출력한다.  
20  
20         result = a / b; — a를 b로 나눈 결과를 result에 넣는다.  
21         System.out.println(a + "/" + b + "=" + result); — 화면에 a, b,  
22                                                         result를 출력한다.  
23     }  
24 }
```

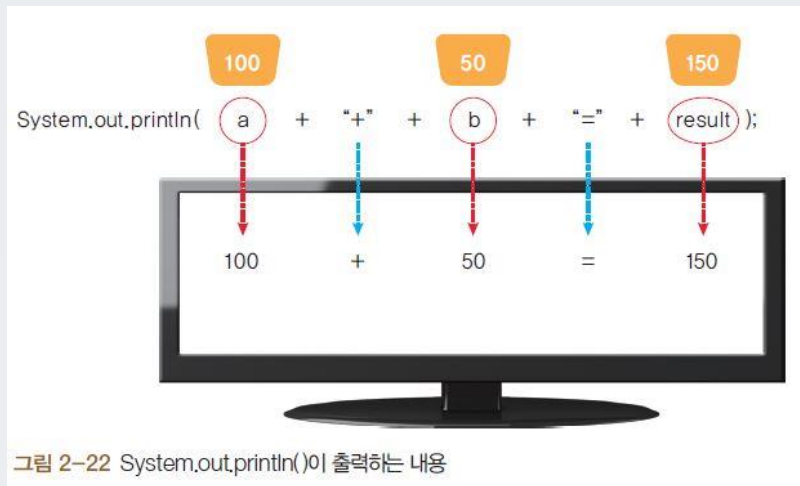
Ex02\_01 클래스와  
범위를 지정한다  
(자동 완성됨).

main 메소드이며  
실제로 실행하는  
부분이다(자동 완성됨).



## 23. 자바 프로그램 작성(10)

- 틀린 글자가 없는지 확인 후 메뉴의 [File]-[Save All] 선택, 입력한 내용 저장
- 변수 : '값을 저장하는 그릇 또는 방'이라고 볼 수 있음



## 24. 자바 프로그램 작성(11)

- 빌드(=컴파일+링크)
  - 작성한 프로그램 빌드. 메뉴에서 [Project]-[Build All] 선택

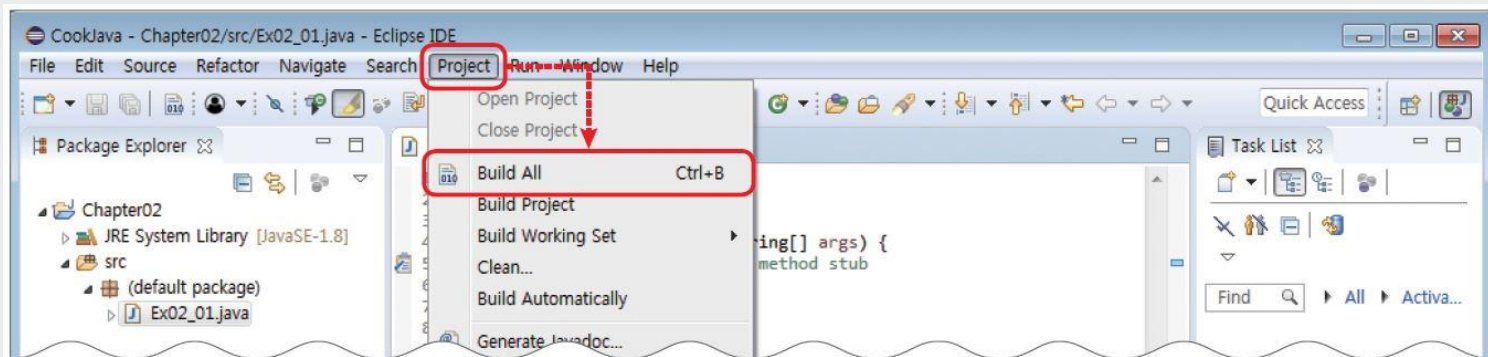


그림 2-23 프로젝트의 빌드

## 25. 자바 프로그램 작성(12)

- 아래쪽의 [Problems] 탭 클릭했을 때 특별한 오류 메시지가 없다면 코드에 오류가 없다는 의미

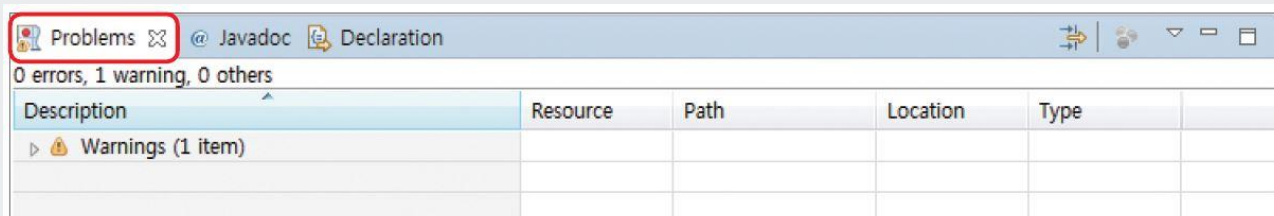


그림 2-24 프로젝트의 빌드 성공

- 만약 실패가 나오면 소스코드에서 틀린 부분을 수정하고 다시 빌드

## 26. 자바 프로그램 작성(13)

### ■ 실행

- [Ctrl]+[F11] 눌러 실행. 아래쪽의 [Console] 탭에 더하기, 빼기, 곱하기, 나누기의 결과 확인

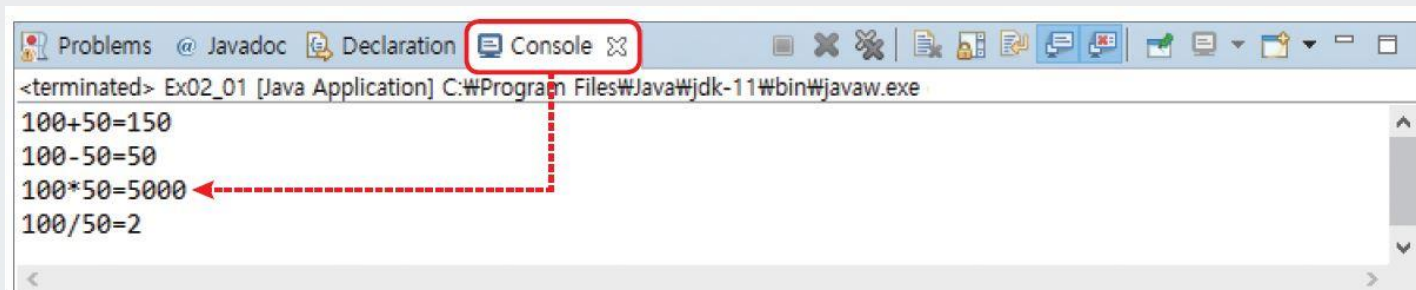
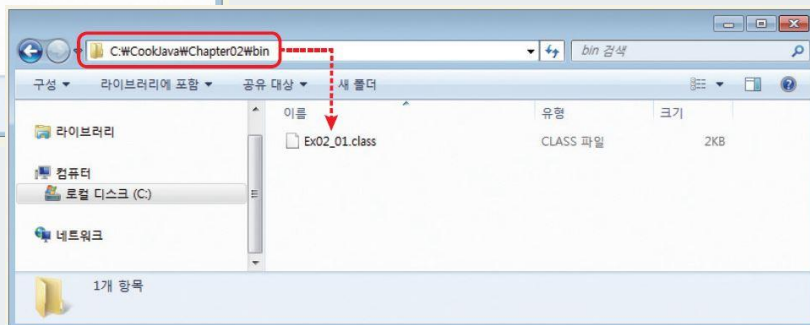
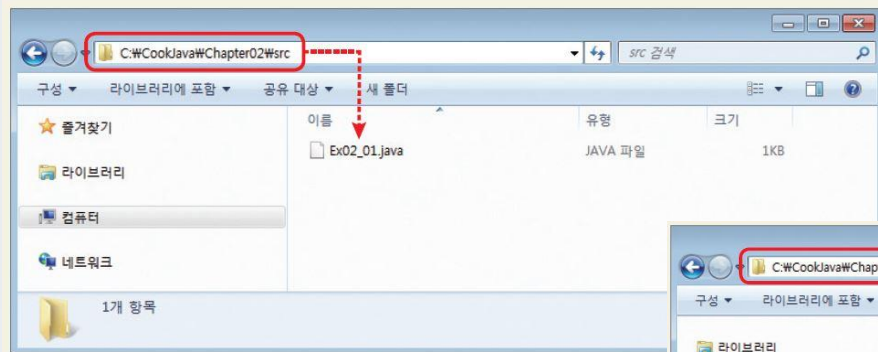


그림 2-25 실행 결과

## 27. 자바 프로그램 작성(14)

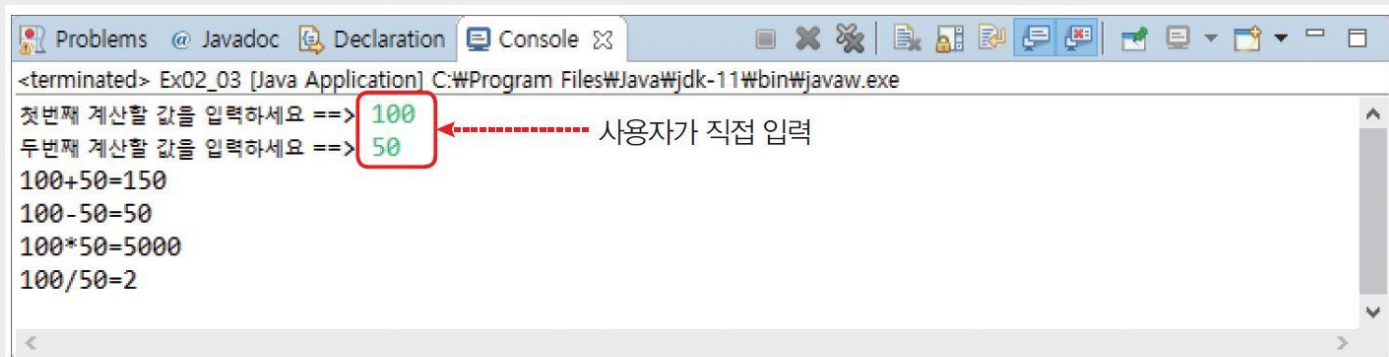
### ■ 프로젝트 폴더

- 이클립스로 생성한 프로젝트 폴더를 확인해보면 Chapter02 프로젝트는 C:\W\CookJava\WChapter02\W 폴더 안에 관련된 파일이 모두 들어감
- JAVA 코드는 프로젝트 폴더의 src 폴더에 들어가고, 컴파일된 결과인 바이트코드에 해당하는 \*.class 파일은 bin 폴더에 들어감.



## 28. 자바 프로그램 작성(15)

- 값을 입력받는 Scanner 클래스
  - 변수 내용을 매번 미리 입력 해두는 [실습 2-1]과 달리 실행할 때마다 키보드로 입력



The screenshot shows a Java application window titled "Ex02\_03 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe". The console output displays a series of prompts and calculations. The first prompt is "첫번째 계산할 값을 입력하세요 ==>". The user has entered "100", which is highlighted with a red box. A red dashed arrow points from the text "사용자가 직접 입력" to the "100". The second prompt is "두번째 계산할 값을 입력하세요 ==>". The user has entered "50", which is also highlighted with a red box. Below the prompts, the following calculations are shown: "100+50=150", "100-50=50", "100\*50=5000", and "100/50=2".

그림 2-26 사용자가 직접 값을 입력하는 프로그램

- Scanner 클래스를 활용하여 변수의 값을 입력받도록 프로그래밍 작성
- JAVA 소스코드 새로 추가한 다음 코드를 통해 확인
  - 왼쪽 'Package Explorer'의 프로젝트 이름(Chapter02) 아래 'src' 폴더에서 마우스 오른쪽 버튼 클릭, [New]-[Class] 선택

## 29. 자바 프로그램 작성(16)

- 'Java Class' 창에서 'Name'에 JAVA 소스코드의 이름 입력('Ex02\_02'), 아래쪽의 'public static void main(String[ ] args)'에 체크하고 [Finish] 클릭

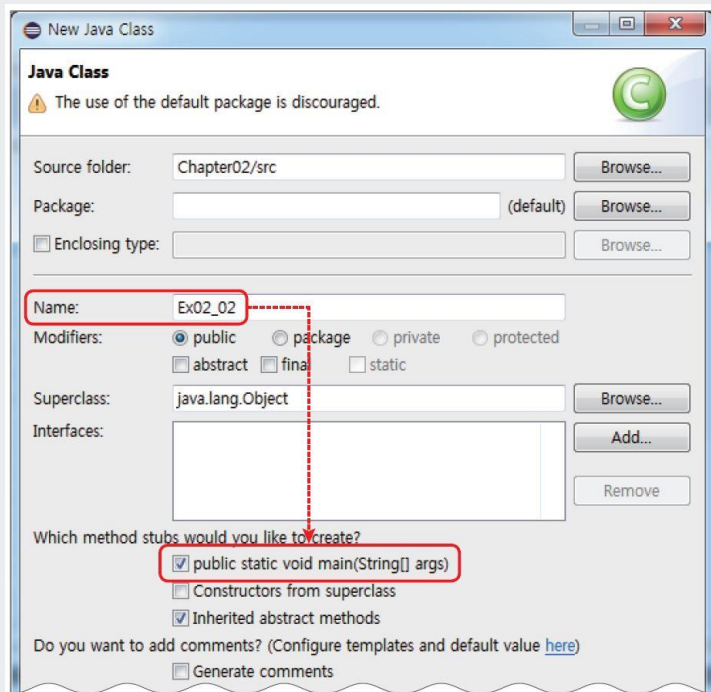


그림 2-27 JAVA 소스코드 추가

## 30. 자바 프로그램 작성(17)

- 값을 입력받는 Scanner 클래스 활용. [실습 2-1]과 동일하게 코딩 후 [실습 2-1]의 8, 9행을 [실습 2-2]의 9~11행으로 수정. Scanner에 빨간 줄이 생김. 이는 프로그램이 Scanner에 대해 이해하지 못하기 때문이므로 관련 패키지나 클래스를 임포트해야 한다. 1행을 직접 입력하거나 [Ctrl]+[Shift]+[O]를 누르면 자동으로 1행이 완성

실습 2-2 소스코드 수정하기(키보드로 값을 입력받음)

```
01 import java.util.Scanner;
02 public class Ex02_02{
03
04     public static void main(String[] args) {
05         // TODO Auto-generated method stub
06         int a, b;
07         int result;
08
09         Scanner s = new Scanner(System.in);
10         a = s.nextInt();
11         b = s.nextInt();
12
13         result = a + b;
14         (a + "+" + b + "=" + result);
15
16         ~~ 이하는 [실습 2-1]의 14~21행과 동일함 ~~
17         :
23     }
24     :
26 }
```

----- 문자나 숫자를 입력받기 위한 Scanner 클래스를  
사용한다(뒤에서 다룸).  
----- 키보드로 a에 들어갈 값을 입력받는다.  
----- 키보드로 b에 들어갈 값을 입력받는다.



## 31. 자바 프로그램 작성(18)

- [Ctrl]+[F11]을 눌러 동시에 빌드 및 실행, 아래쪽 [Console] 클릭, 커서만 깜박이는 것 확인

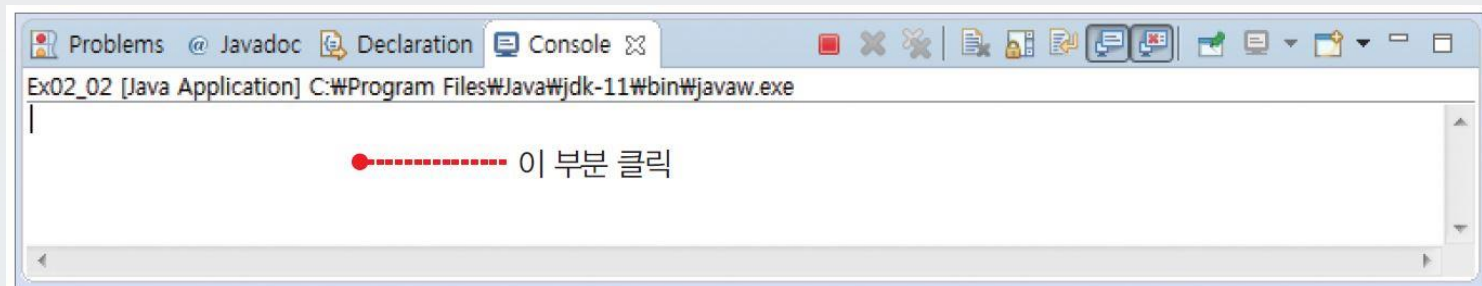
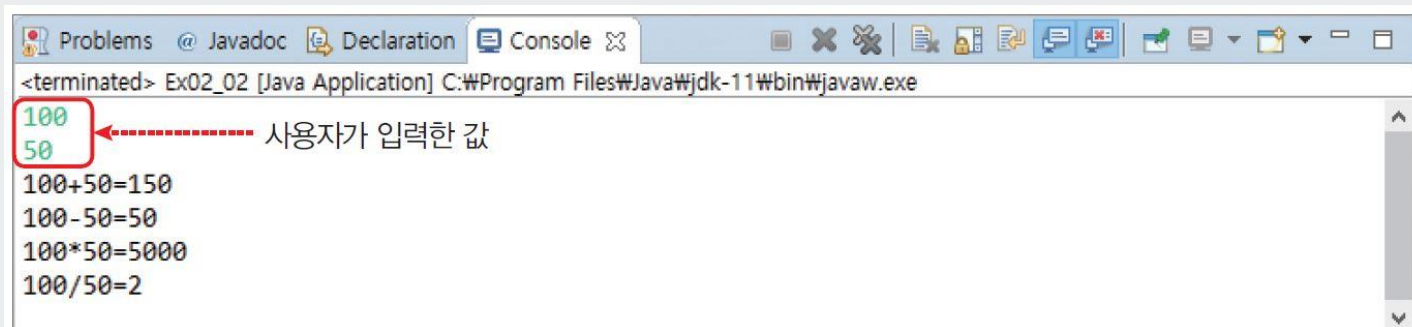


그림 2-28 실행 결과

## 32. 자바 프로그램 작성(19)

- 숫자 하나를 입력하고 [Enter] 누름, 다시 숫자 하나를 입력하고 [Enter] 누름



The screenshot shows a Java IDE's console window. The title bar indicates the application is 'Ex02\_02 [Java Application]' running at 'C:\Program Files\Java\jdk-11\bin\javaw.exe'. The console output shows the program has terminated. The user input '100' and '50' is highlighted with a red box, and a red dashed arrow points to it with the text '사용자가 입력한 값' (Value entered by the user). Below the input, the program outputs the results of arithmetic operations: '100+50=150', '100-50=50', '100\*50=5000', and '100/50=2'.

```
<terminated> Ex02_02 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe
100
50
100+50=150
100-50=50
100*50=5000
100/50=2
```

그림 2-29 숫자 입력과 실행 결과

## 33. 자바 프로그램 작성(20)

- 좀 더 쉬운 방법으로 클래스 추가하기. 앞의 Ex02\_02.java 복사하여 Ex02\_03.java 만든 다음 수정. 왼쪽 'Package Explorer'의 'Ex02\_02.java'에서 마우스 오른쪽 버튼 클릭, [Copy] 선택하여 복사

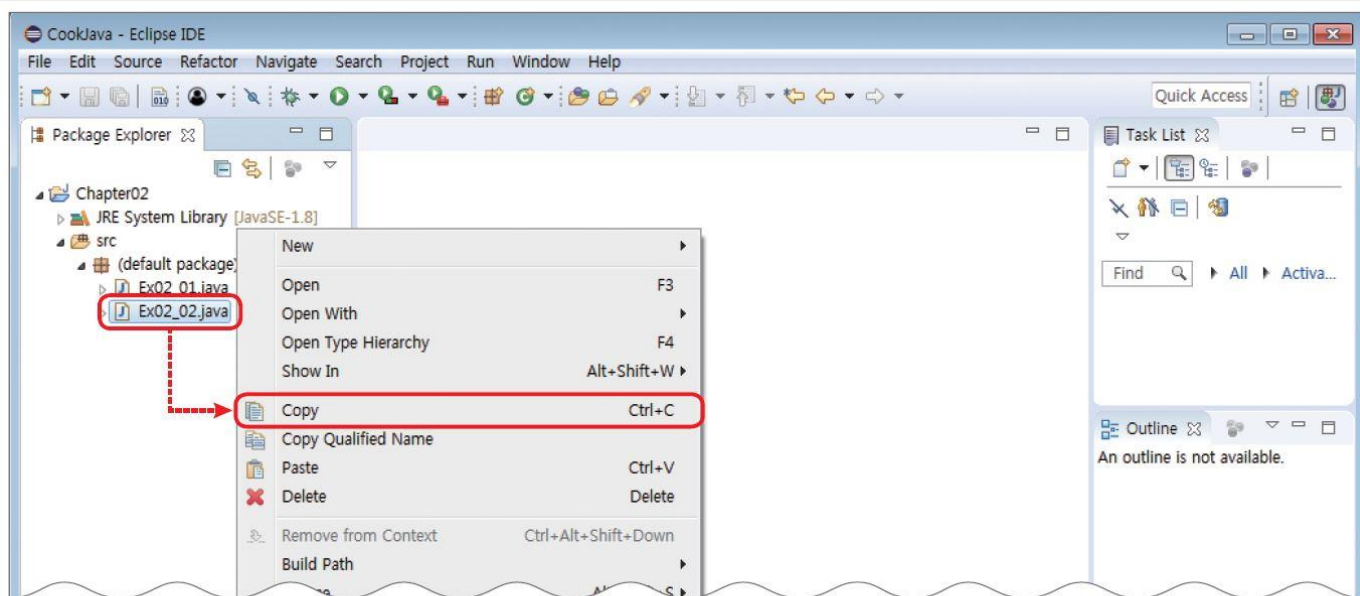


그림 2-30 JAVA 소스코드 복사 1

## 34. 자바 프로그램 작성(21)

- 'Package Explorer'의 'src'에서 마우스 오른쪽 버튼 클릭, [Paste] 선택.
- 새로운 이름으로 'Ex02\_03' 입력하고 [OK] 클릭

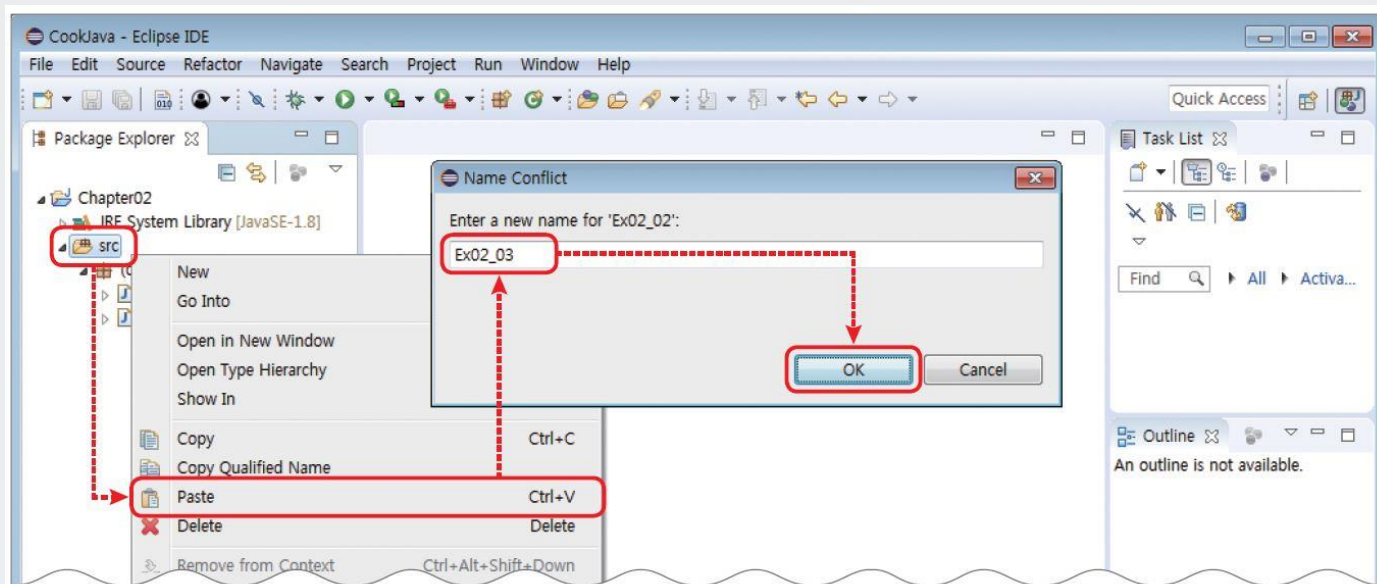


그림 2-31 JAVA 소스코드 복사 2

## 35. 자바 프로그램 작성(22)

- 'Package Explorer'에서 Ex02\_03.java 추가된 것 확인. 더블클릭하여 코드를 보면 클래스 이름도 Ex02\_03으로 변경

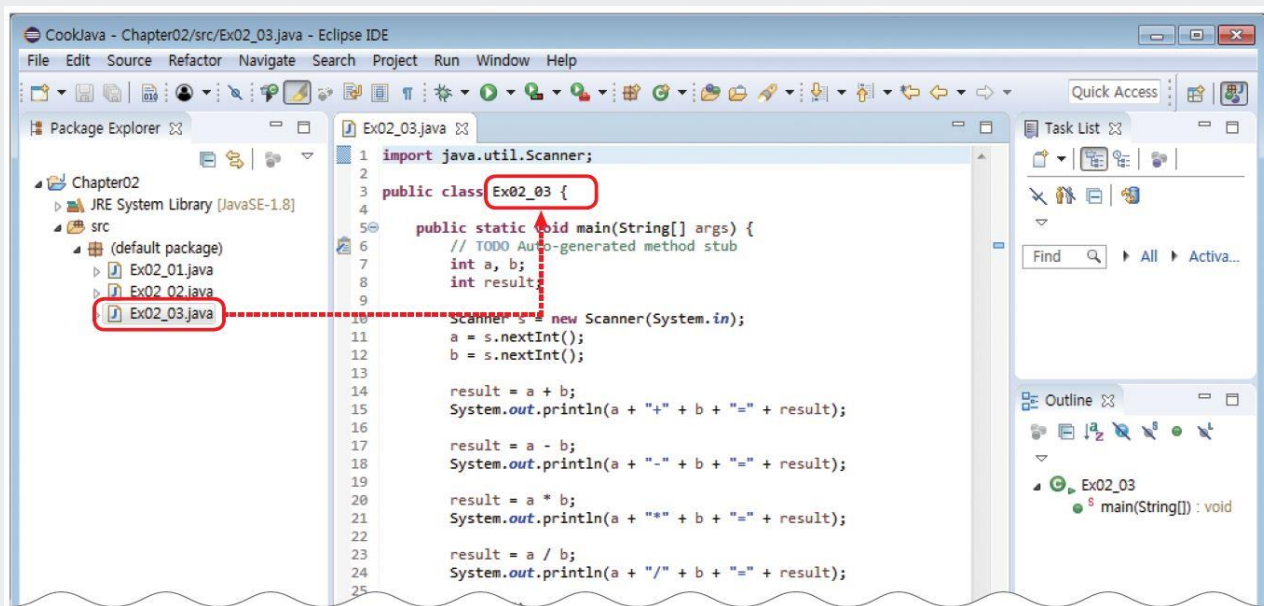


그림 2-32 JAVA 소스코드 복사3

## 36. 자바 프로그램 작성(23)

- 10행과 12행을 다음과 같이 수정

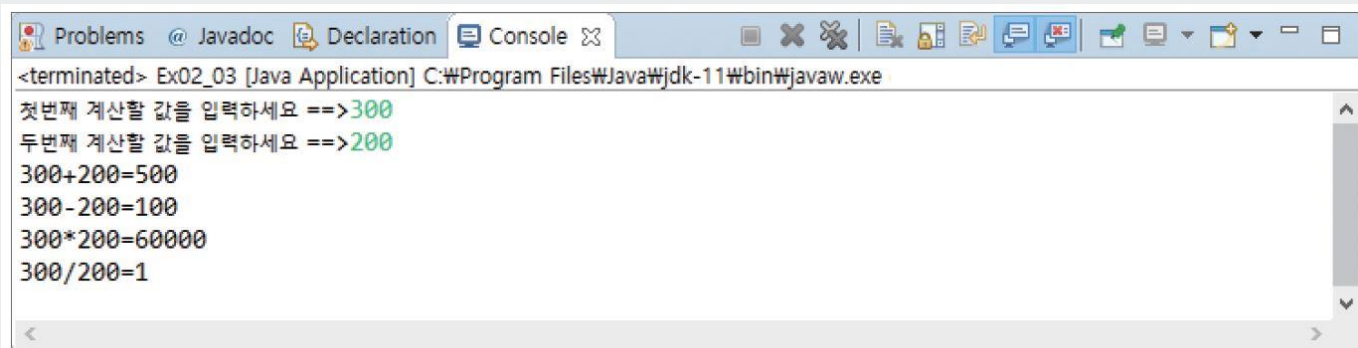
### 실습 2-3 소스코드 수정하기(도움말 출력)

```
01 import java.util.Scanner;
02 public class Ex02_03 {
03
04     public static void main(String[] args) {
05         // TODO Auto-generated method stub
06         int a, b;
07         int result;
08
09         Scanner s = new Scanner(System.in);
10         System.out.print("첫번째 계산할 값을 입력하세요 ==>");
11         a = s.();
12         System.out.print("두번째 계산할 값을 입력하세요 ==>");
13         b = s.();
14
15         result = a + b;
16         ~~~ 이하는 [실습 2-2]의 14~26행과 동일함 ~~~
17     }
18 }
```

도움말을 화면에 출력한다.

## 37. 자바 프로그램 작성(24)

- 다시 [Ctrl]+[F11] 눌러 동시에 빌드 및 실행. 화면에 계산할 값을 입력하라는 안내 문구가 뜨며, 원하는 값 입력하여 계산



```
<terminated> Ex02_03 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe
첫번째 계산할 값을 입력하세요 ==>300
두번째 계산할 값을 입력하세요 ==>200
300+200=500
300-200=100
300*200=60000
300/200=1
```

그림 2-33 최종 실행 결과

## 38. 자바 프로그램 작성(25)

- 이클립스 사용법
  - 열려 있는 JAVA 코드 창의 작은 [x] 눌러 소스 파일 닫기

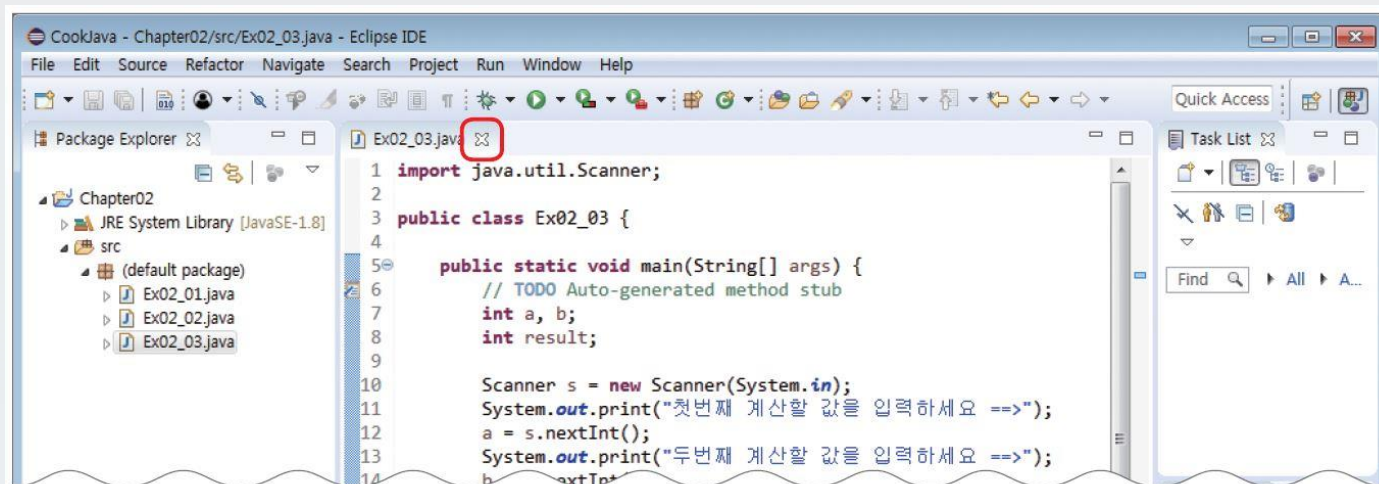


그림 2-34 열린 JAVA 소스 파일 닫기



## 39. 자바 프로그램 작성(26)

- 다시 JAVA 소스 파일을 열려면 'Package Explorer'에서 소스 파일 더블클릭

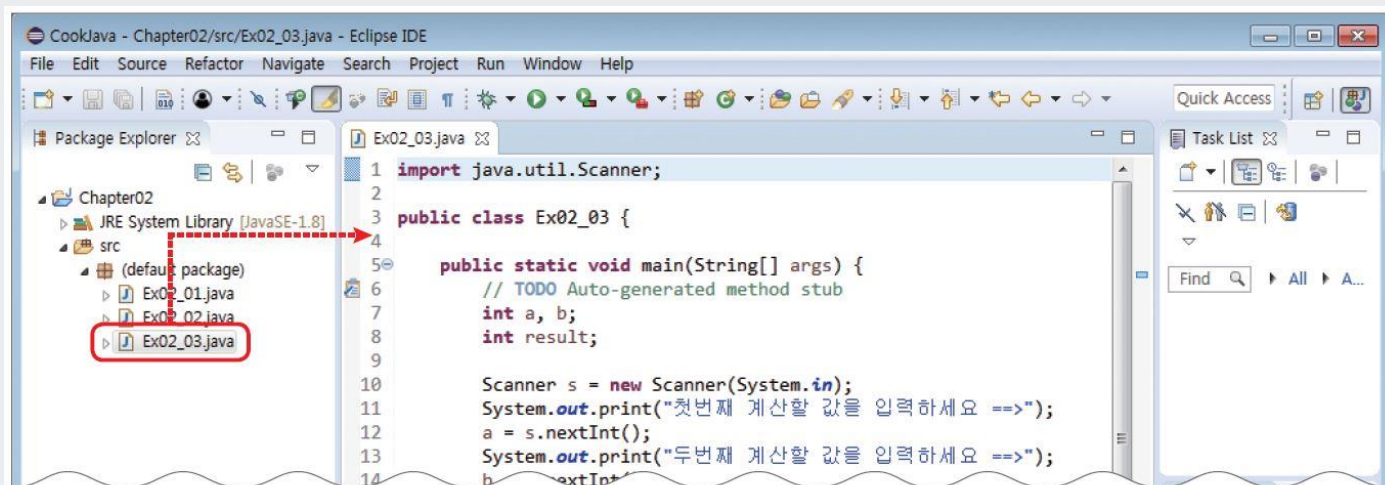


그림 2-35 JAVA 소스 파일 열기

## 40. 자바 프로그램 작성(27)

- 'Package Explorer'에서 다른 소스 파일 더블클릭하면 여러 개가 탭으로 구분되어 열림

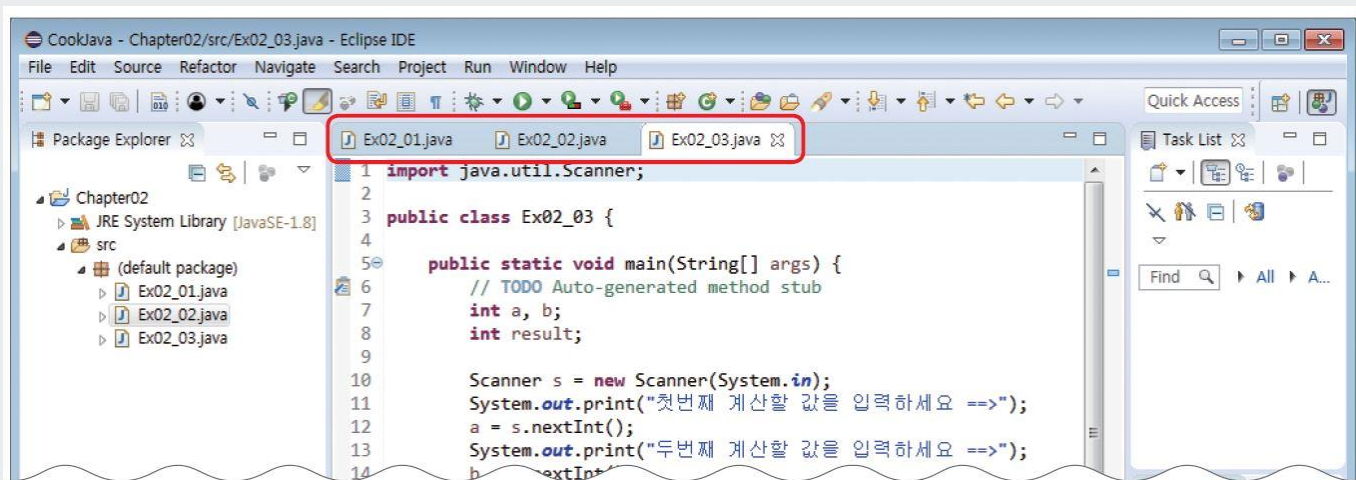


그림 2-36 여러 개의 소스 파일이 열린 상태

# 41. 자바 프로그램 작성(28)

- 열려 있는 소스 파일을 한꺼번에 닫으려면 메뉴에서 [File]-[Close All] 선택

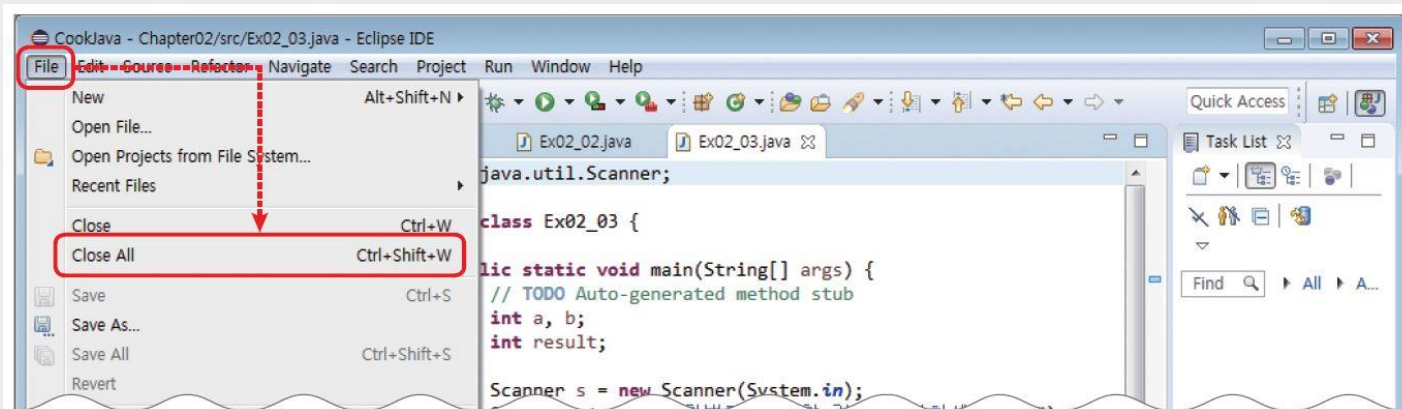


그림 2-37 열린 파일 모두 닫기

## 42. 자바 프로그램 작성(29)

- 프로젝트를 닫으려면 'Package Explorer'의 프로젝트(예 : Chapter02)에서 마우스 오른쪽 버튼을 클릭하여 [Close Project] 선택. 프로젝트 아이콘이 닫힌 모양으로 바뀐다

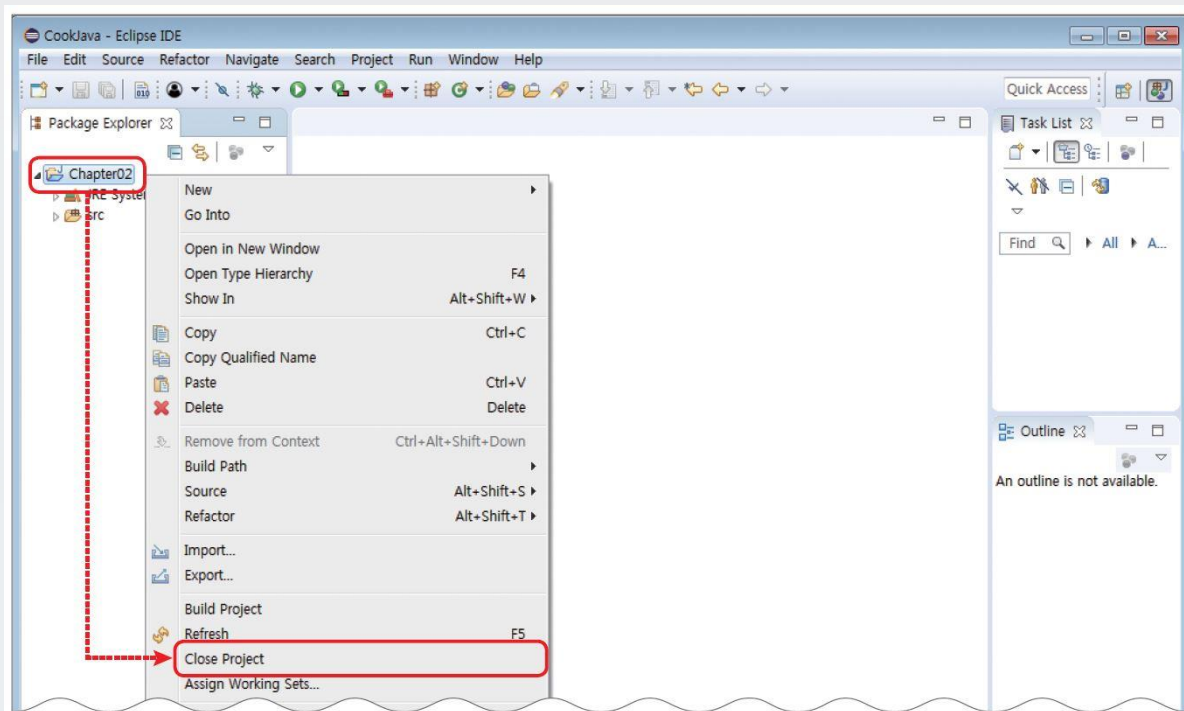


그림 2-38 프로젝트 닫기

## 43. 자바 프로그램 작성(30)

- 다시 프로젝트를 열려면 'Package Explorer'에서 닫힌 프로젝트를 더블클릭

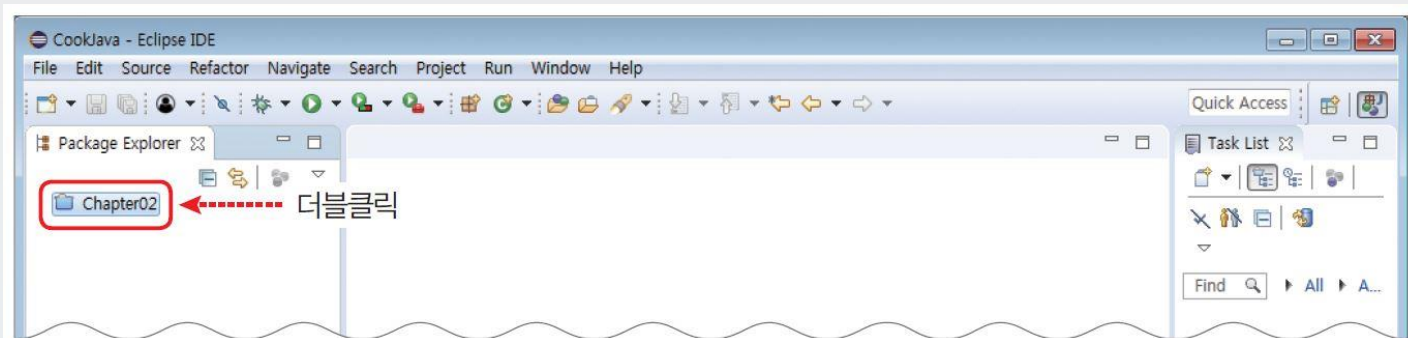


그림 2-39 프로젝트 다시 열기

## 44. 자바 프로그램 작성(31)

- 'Package Explorer'에서 프로젝트를 완전히 제거하려면 'Package Explorer'의 프로젝트(예 : Chapter02)에서 마우스 오른쪽 버튼을 클릭, [Delete] 선택, [Ok] 클릭. 이때 'Delete project contents on disk (cannot be undone)'에 체크하면 프로젝트와 소스 파일이 완전히 삭제되므로 특별한 경우가 아니라면 체크하지 않도록 주의

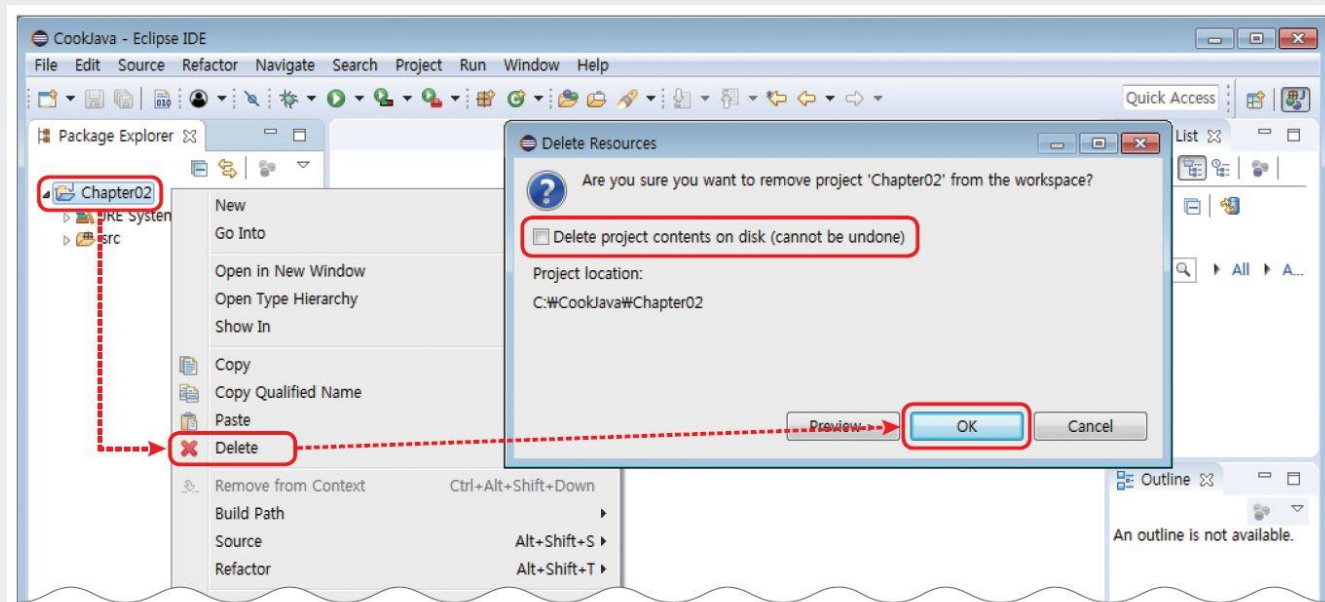


그림 2-40 'Package Explorer'에서 프로젝트 제거

## 45. 자바 프로그램 작성(32)

- 다른 프로젝트를 다시 가져오려면 'Package Explorer'의 빈 곳에서 마우스 오른쪽 버튼 클릭, [Import] 선택, 'Import' 창에서 [General]-[Existing Projects into Workspace] 선택, [Next] 클릭

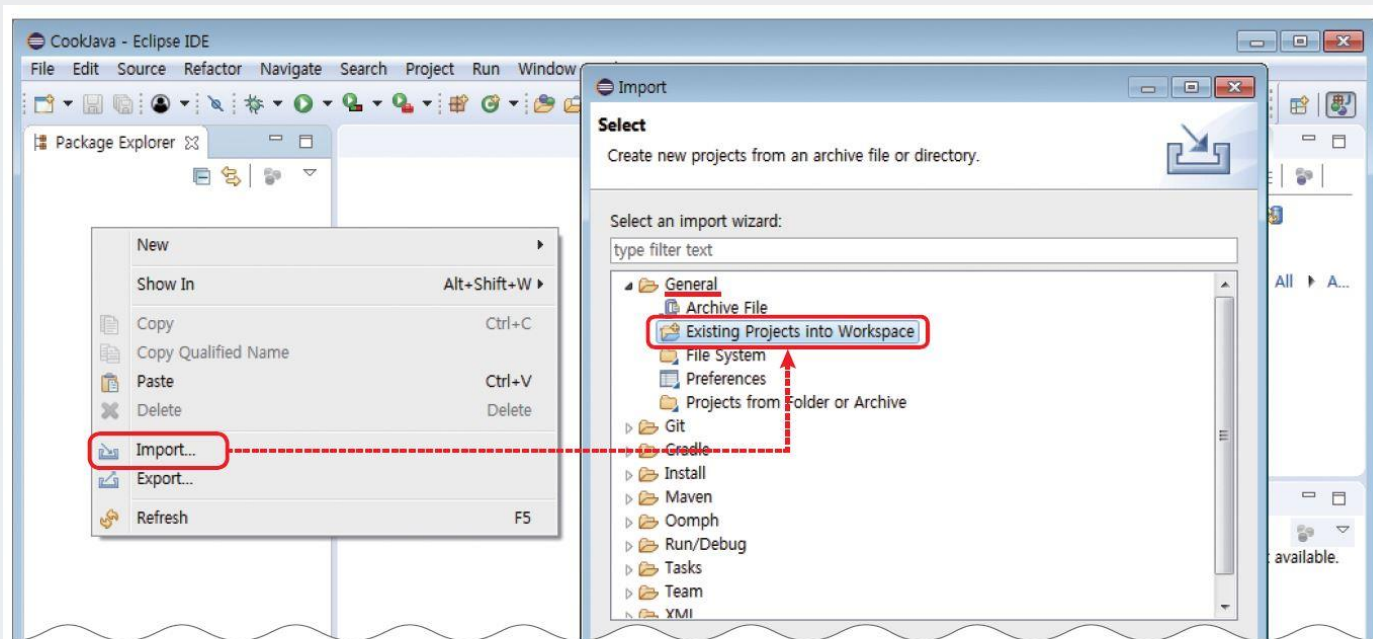


그림 2-41 'Package Explorer'에 프로젝트 다시 가져오기 1



## 46. 자바 프로그램 작성(33)

- 'Import Projects'에서 'Select root directory' 부분의 [Browse...] 클릭, 가져오고자 하는 프로젝트의 폴더(예 : C:\₩CookJava₩Chapter02) 선택, [Finish] 클릭. 선택한 프로젝트가 'Package Explorer'에 다시 나타남

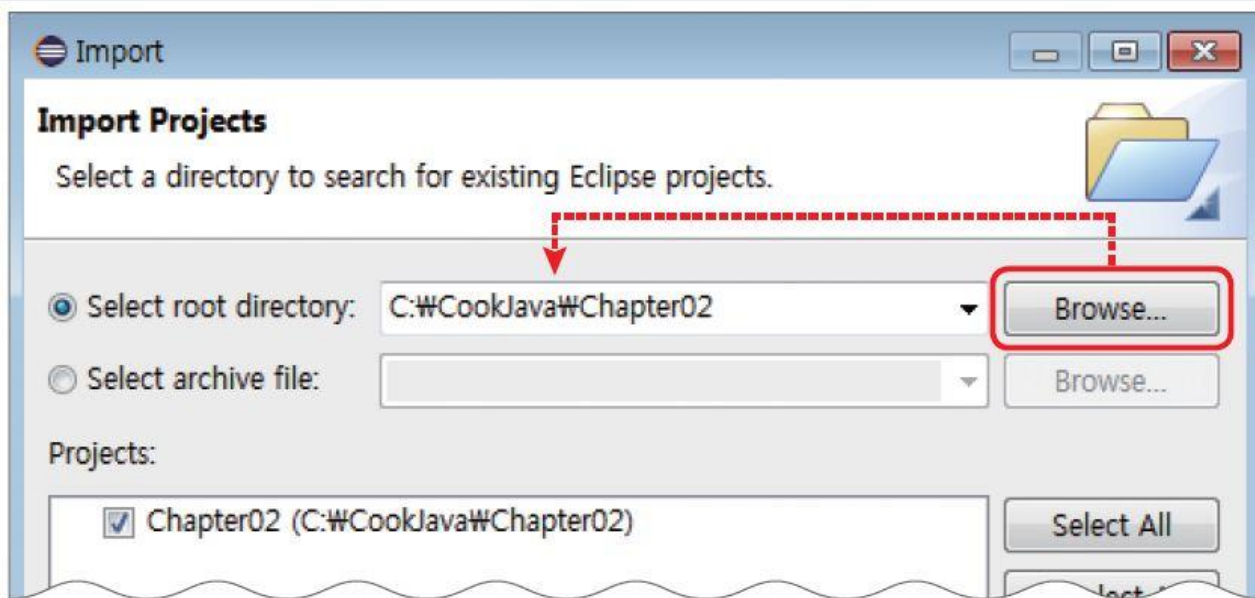


그림 2-42 'Package Explorer'에 프로젝트 다시 가져오기 2



## 47. 자바 프로그램 작성(34)

- 아래쪽 'Console' 창의 결과를 깨끗이 지우려면 'Remove Launch'나 'Remove All Terminated Launches' 아이콘 클릭

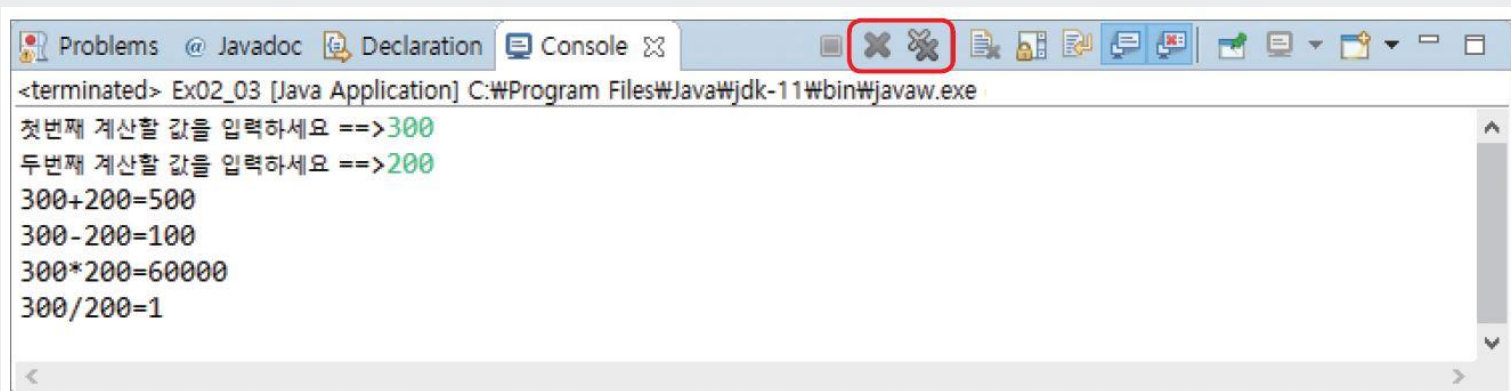


그림 2-43 콘솔 깨끗이 지우기

## 48. 자바 프로그램 작성(35)

- 'Package Explorer'에서는 여러 개의 프로젝트를 생성하거나 가져오기 하여 사용 가능

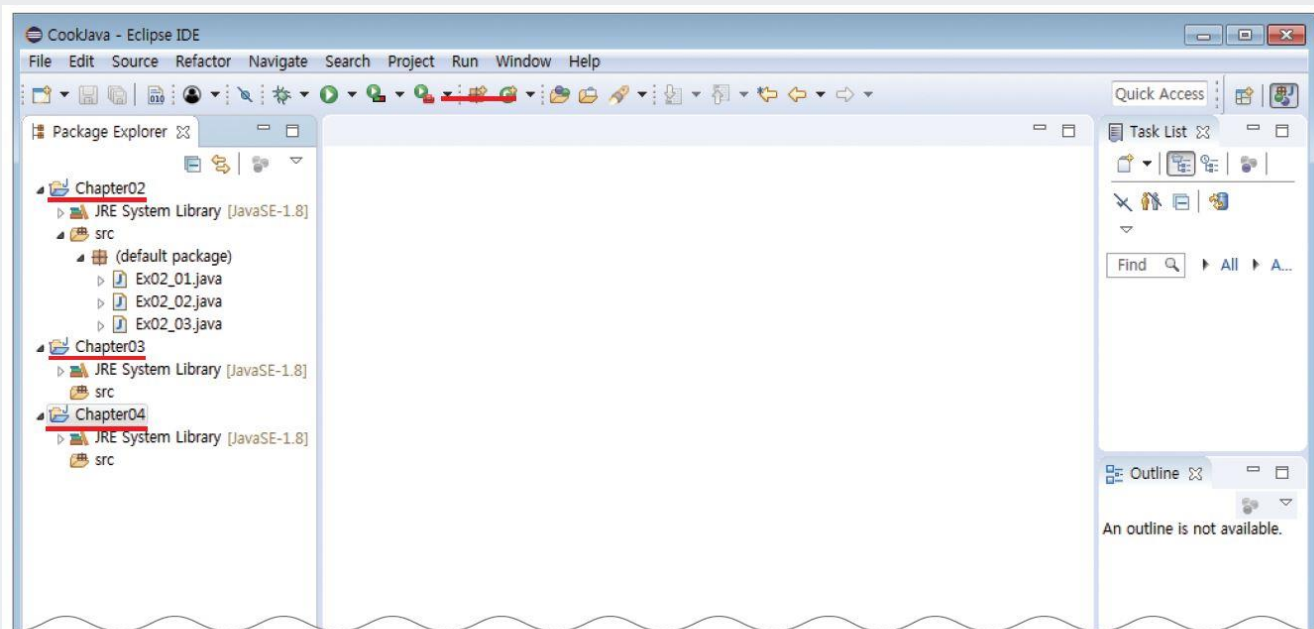


그림 2-44 여러 개의 프로젝트가 열린 상태

# 다음 시간 테스트



송실사이버대학교

송실사이버대학교의 강의콘텐츠는  
저작권법에 의하여 보호를 받는바, 무단  
전재, 배포, 전송, 대여 등을 금합니다.

\* 사용서체 : 나눔글꼴