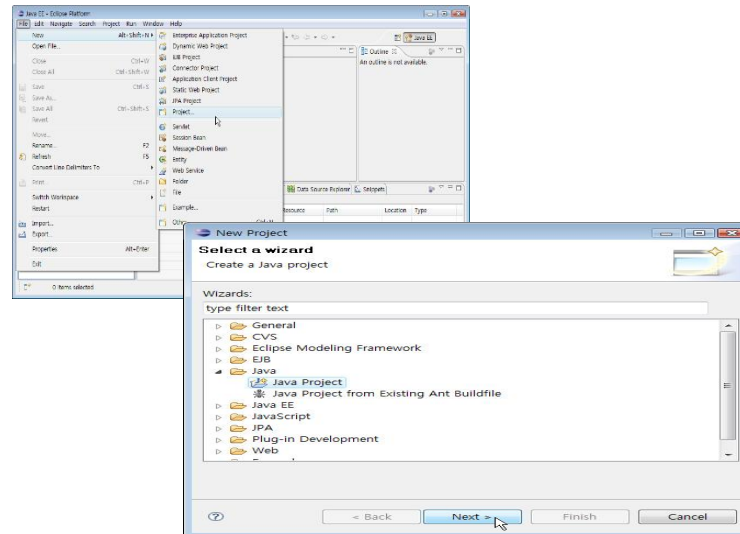


2. Java 개발환경 구축

12 간단한 자바 프로그램 작성하기

1

- [File]
-[New]
-[Project...]
메뉴를 선택함



2. Java 개발환경 구축

12 간단한 자바 프로그램 작성하기

2

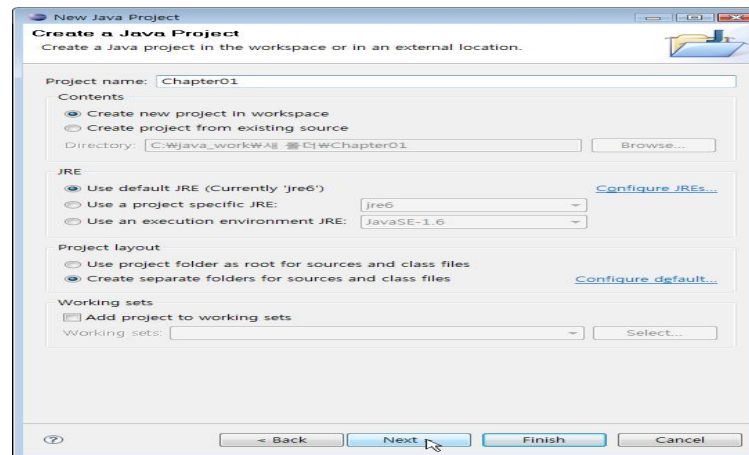
- [New Java Project] 창이 나타나면 가장 윗줄에 'Create a Java project'이라는 문장을 확인할 수 있음

2. Java 개발환경 구축

12 간단한 자바 프로그램 작성하기

2

- 다음과 같이 현재 화면의 내용을 입력 또는 선택한 후 [Next] 버튼을 클릭함



2. Java 개발환경 구축

12 간단한 자바 프로그램 작성하기

3

- [Project name] : 'ch01' 란 내용을 입력

4

- [Contents] : 기본값 사용

2. Java 개발환경 구축

12 간단한 자바 프로그램 작성하기

5

– [JRE] : 기본값 사용

- [Project name] : 작업 공간으로 지정한
폴더 하위 폴더에 이 프로젝트 이름과 같은 이름의 폴더가
생성되고 그곳에서 이 프로젝트에서 사용되는 파일들을
관리함

2. Java 개발환경 구축

12 간단한 자바 프로그램 작성하기

5

– [JRE] : 기본값 사용

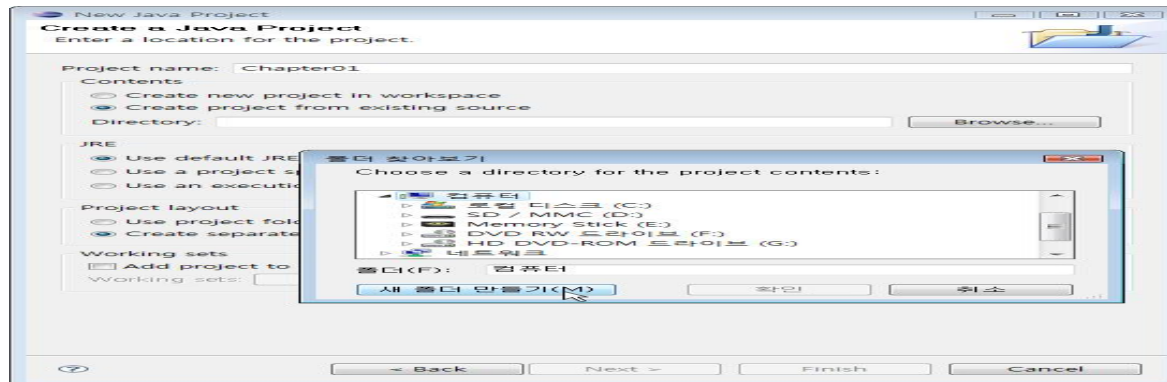
- [Contents] : 작성할 프로젝트의 배치(저장) 장소를 지정하는 것으로 [Create new Project in workspace] 항목을 선택하면 작업 공간 내의 프로젝트 폴더가 생성됨
- [Create Project from existing source] 항목을 선택하면 그 아래의 [Directory:] 항목의 입력이 가능하며, 오른쪽 [Browse] 단추를 클릭하여 프로젝트를 작성할 위치를 지정하거나 새로 작성할 수 있음

2. Java 개발환경 구축

12 간단한 자바 프로그램 작성하기

5

– [JRE] : 기본값 사용



2. Java 개발환경 구축

13 자바 소스 파일 작성하기

1

- 프로젝트를 작성하는 것만으로는 프로그램을 개발할 수 없음
- 프로젝트에 소스 파일을 추가하여 그 파일에 로직을 기술해야 함
- 실행 가능한 자바 소스 파일을 작성하려면 `main()` 메소드를 갖는 자바 소스 파일을 작성해야 함

2.Java개발환경구축

13 자바 소스 파일 작성하기

2

- [Package Explorer] 창에서 [Ch01] 프로젝트를 선택하고 [File]-[New]-[Class]를 선택함

2. Java 개발환경 구축

13 자바 소스 파일 작성하기

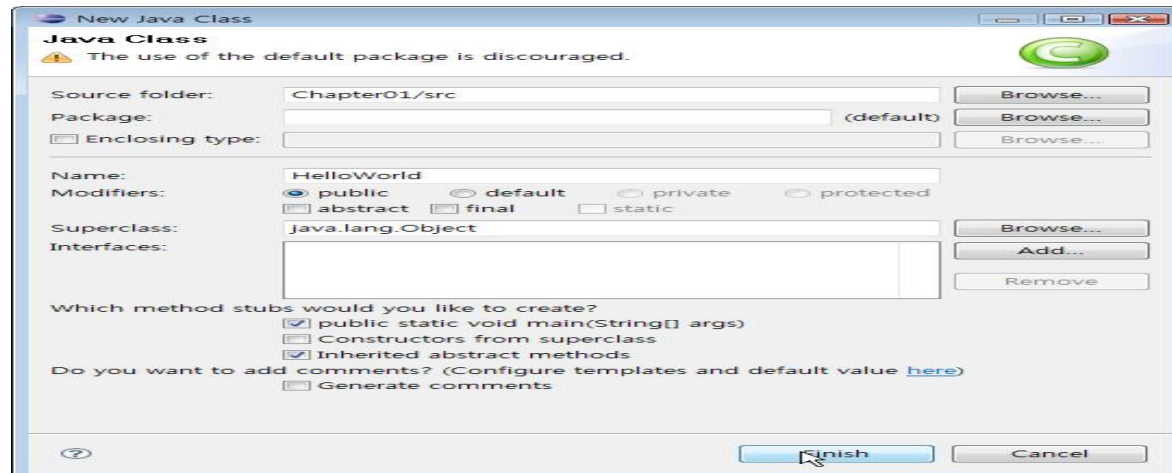
3

- [New Java Class] 창에서 [Name:] 입력란에 "HelloWorld"를 입력함
- [Which method stubs would you like to create?]에서 [public static void main(String[] args)] 항목을 체크한 후에 [Finish] 버튼을 클릭함

2. Java 개발환경 구축

13 자바 소스 파일 작성하기

3



2. Java 개발환경 구축

13 자바 소스 파일 작성하기

4

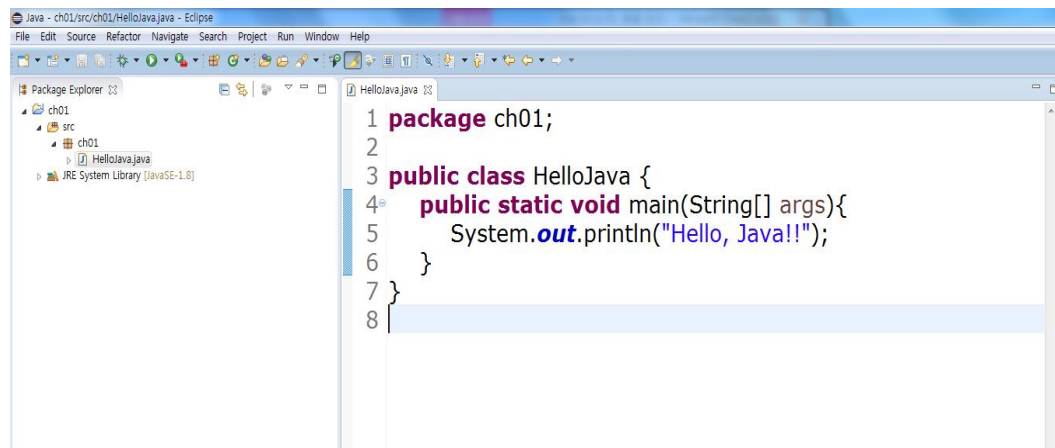
- [Package Explorer] 뷰를 살펴보면
"(default package)"가 자동 생성되어
이곳에 자바 소스 파일인 'HelloWorld.java'가
생성된 것을 확인할 수 있고 에디터 뷰에는
HelloWorld.java 파일이 열림

2. Java 개발환경 구축

13 자바 소스 파일 작성하기

4

– 여기에 아래와 같이 입력함



```
1 package ch01;
2
3 public class HelloJava {
4     public static void main(String[] args){
5         System.out.println("Hello, Java!!");
6     }
7 }
8
```

2. Java 개발환경 구축

13 자바 소스 파일 작성하기

5

- 소스 작성이 끝났다면 파일을 저장함
- 파일을 저장하면 파일명.java 앞에 붙은 * 표시가 사라지고 컴파일 오류가 발생되지 않으면
자바 소스 파일(HelloWorld.java)을 저장함과 동시에
컴파일이 되어 클래스 파일(HelloWorld.class)이
생성됨

2. Java 개발환경 구축

13 자바 소스 파일 작성하기

6

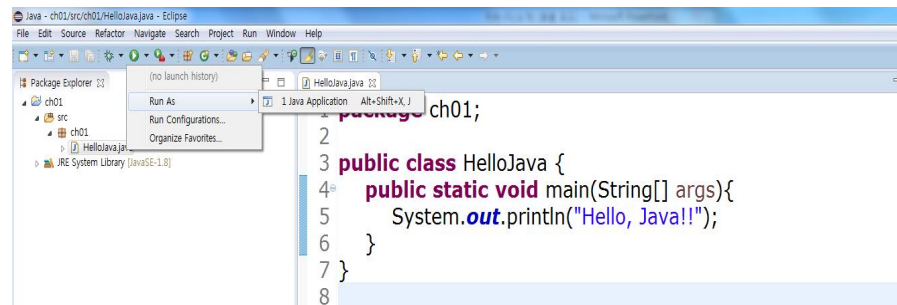
- 이렇게 생성된 클래스 파일을 실행해보자
-  아이콘에서 마크를 클릭해서 표시되는 메뉴에서 [Run As]-[Java Application] 메뉴를 선택함

2. Java 개발환경 구축

13 자바 소스 파일 작성하기

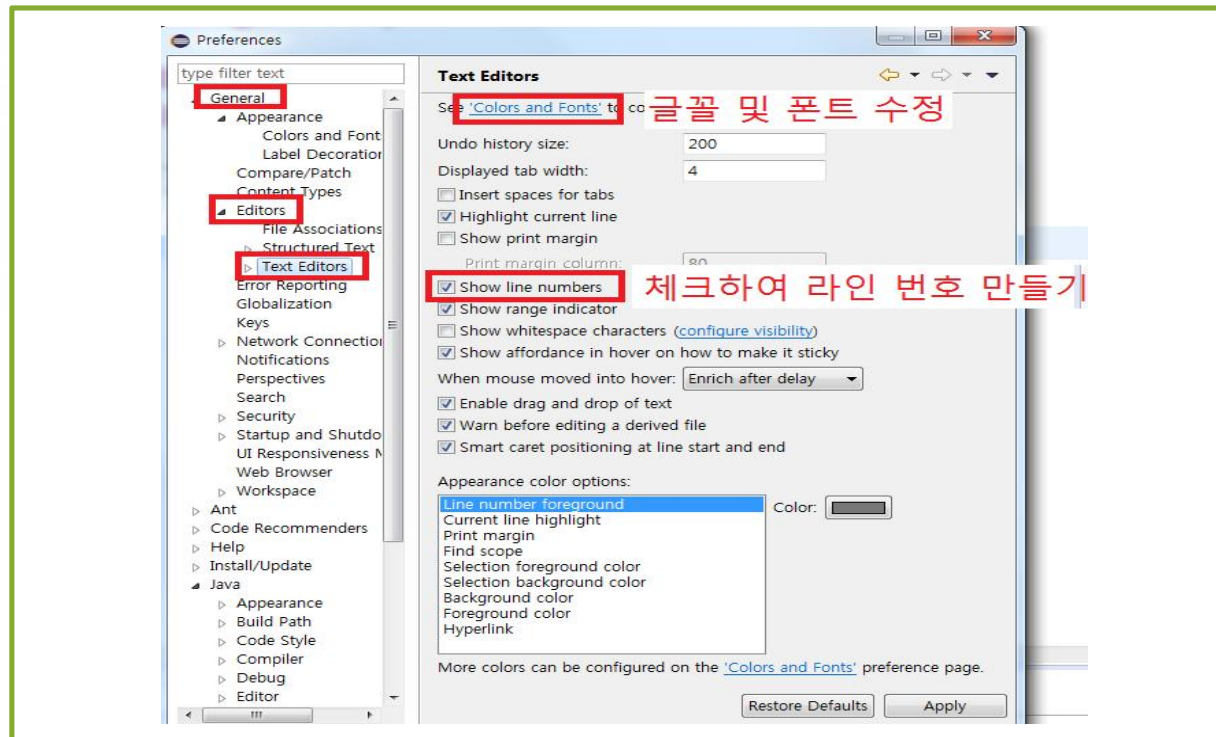
7

- 화면 아래쪽 [Console] 창에 "Hello World !"가 출력되었음을 확인할 수 있음



2. Java 개발환경 구축

14 글꼴 변경과 라인 번호 표시하기



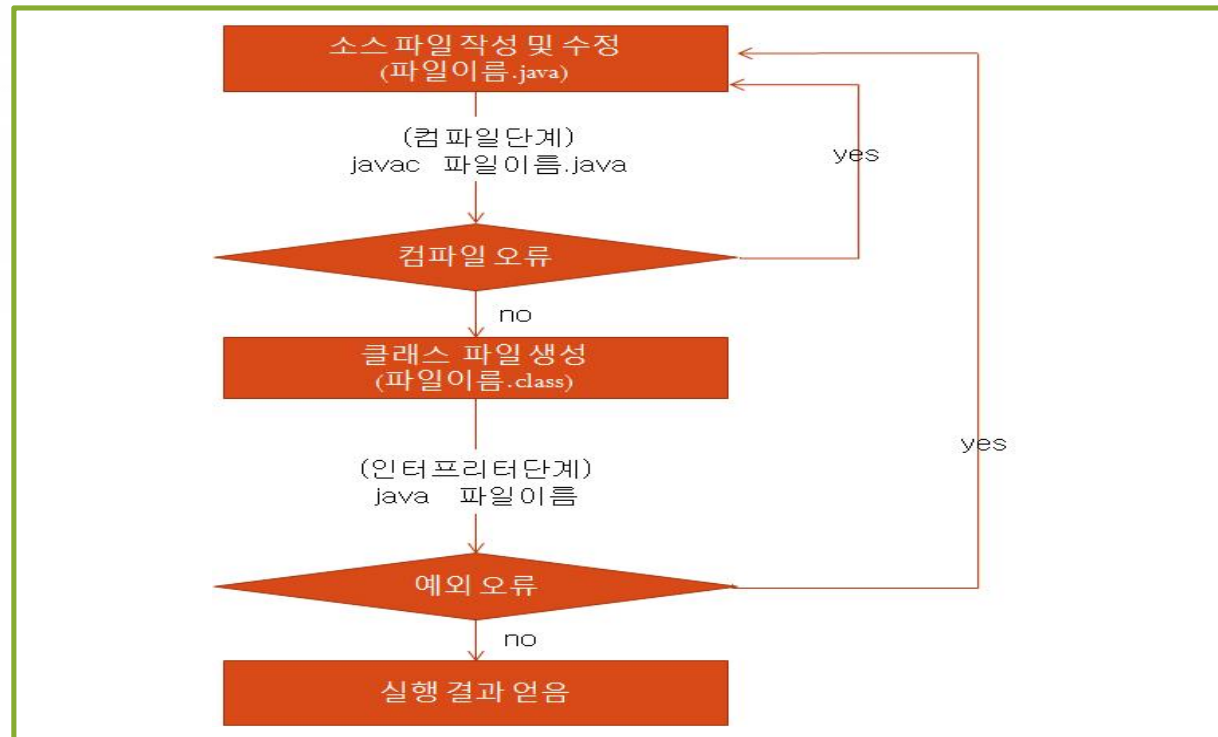
2. Java 개발환경 구축

15 자바 프로그래밍하기

- ✓ 우리가 보통 "자바를 한다!"하고 이야기할 때는 자바 프로그램을 입력하여 컴파일하고 디버깅하고, 필요에 따라서는 문서화도 시키고, 결과로 나온 실행파일(클래스)을 실행하고 관리하는 작업을 전부 말함
- ✓ 여러분이 본서를 통해서 학습할 내용은 프로그램을 개발과 자바를 이해하는 부분으로 나뉨

2. Java 개발환경 구축

15 자바 프로그래밍하기



2. Java 개발환경 구축

16 자바 프로그램의 기본 구조

클래스와 메소드 정의

클래스를 정의하는 문장

→ 자바로 프로그램을 작성한다는 것은
확장자가 “java”인 소스 파일을 하나 만들어서
자바 문법에 맞는 내용을 기술하는 것을 의미함

2. Java 개발환경 구축

16 자바 프로그램의 기본 구조

```
System.out.println("Hello World !");
```

“Hello World !”를 출력하기 위한 메소드를
호출하는 문장임

2. Java 개발환경 구축

16 자바 프로그램의 기본 구조

① `public class HelloWorld {`

클래스 정의

② `public static void main(String[] args) {`

메소드 정의

③ `System.out.println("Hello World !");`

`}`

`}`

2. Java 개발환경 구축

16 자바 프로그램의 기본 구조

I 주석문

1 /* ~ */ 주석문

- ✓ /* 로 시작해 */ 이 나올 때까지 모든 내용이 주석 처리가 됨
- ✓ 여러 줄에 걸쳐 블록 단위로 주석 처리할 경우 사용함

2. Java 개발환경 구축

16 자바 프로그램의 기본 구조

I 주석문

2 // 주석문

- ✓ // 뒤에 있는 한 줄만 주석처리 됨

3 /** ~ */ 주석문

- ✓ 여러 문장을 주석 처리할 수 있다는 면에서
/* ~ */ 와 유사한 기능을 가짐

2. Java 개발환경 구축

16 자바 프로그램의 기본 구조

가독성을 높이기 위한 들여쓰기

클래스나 메소드의 시작과 종료를 나타내는 {} 내에
내용을 기술할 때는 가독성을 높이기 위해
들여쓰기를 함

2. Java 개발환경 구축

16 자바 프로그램의 기본 구조

주석 예제

```
class Comment { // 프로그램 시작 부분
    public static void main(String[] args) {
        boolean A = (3 > 10);
        boolean B = (10 > 3);
        //System.out.println("A = " + A + " B = " + B );
    }
}
```

```
class Comment { // 프로그램 시작 부분
    public static void main(String[] args) {
        // boolean A = (3 > 10);
        boolean B = (10 > 3);
        System.out.println("A = " + A + " B = " + B );
    }
}
```

2. Java 개발환경 구축

16 자바 프로그램의 기본 구조

주석 예제

```
public class Comment2 {  
    /**  
        boolean 값 변수 A, B 처리 확인  
        프로그램 최종 수정일 : 2002/12/25  
    */  
    public static void main(String[] args) {  
        boolean A = /* A정의 */ ( 3 > 10 );  
        /* boolean B = ( 10 > 3 );  
        System.out.println("A = " + A + " B = " + B); */  
    }  
}
```

2. Java 개발환경 구축

17 사용자 정의 명칭

사용자가 정의한 이름이라는 뜻

➡ 폴더의 이름, 클래스의 이름, 메서드의 이름,
그리고 필드의 이름 등을 정의할때 사용

2. Java 개발환경 구축

17 사용자 정의 명칭

! 다음과 같은 규칙으로 이루어져 있음

1 첫글자는 \$,_,영문 대소문자여야 함

✓ 한글도 가능하지만 권장하지는 않음

2 글자수에는 제한이 없음

3 공백 문자는 포함할 수 없음

2. Java 개발환경 구축

17 사용자 정의 명칭

다음과 같은 규칙으로 이루어져 있음

4 특수문자(@, #, %, ^, &, !, ?)는 사용할 수 없음

5 숫자는 첫 글자가 아닐 경우 사용이 가능함

6 예약어는 사용할 수 없음

2. Java 개발환경 구축

17 사용자 정의 명칭

I 권장사항

1 Class

- ✓ 첫 글자는 \$, _, 대문자를 사용하자

2 Method(함수)

- ✓ 첫 글자는 \$, _, 소문자를 사용하자

2. Java 개발환경 구축

17 사용자 정의 명칭

I 권장사항

3 Field(변수)

- ✓ 일반적으로 전부 대문자인 경우는 상수를 정의하는 `static final` 필드이고, 그 외는 소문자임

4 합성어의 첫 글자는 대문자로 함

- ✓ 예 - helloJava