



1.1 프로그래밍 언어와 자바

1.2 운영체제별 JDK 설치

1.3 윈도우 환경 변수 설정

1.4 맥OS 환경 변수 설정

1.5 바이트코드 파일과 자바 가상 머신

1.6 소스 작성부터 실행까지

1.7 이클립스 설치

1.8 이클립스 프로젝트 생성

1.9 이클립스 소스 작성부터 실행까지

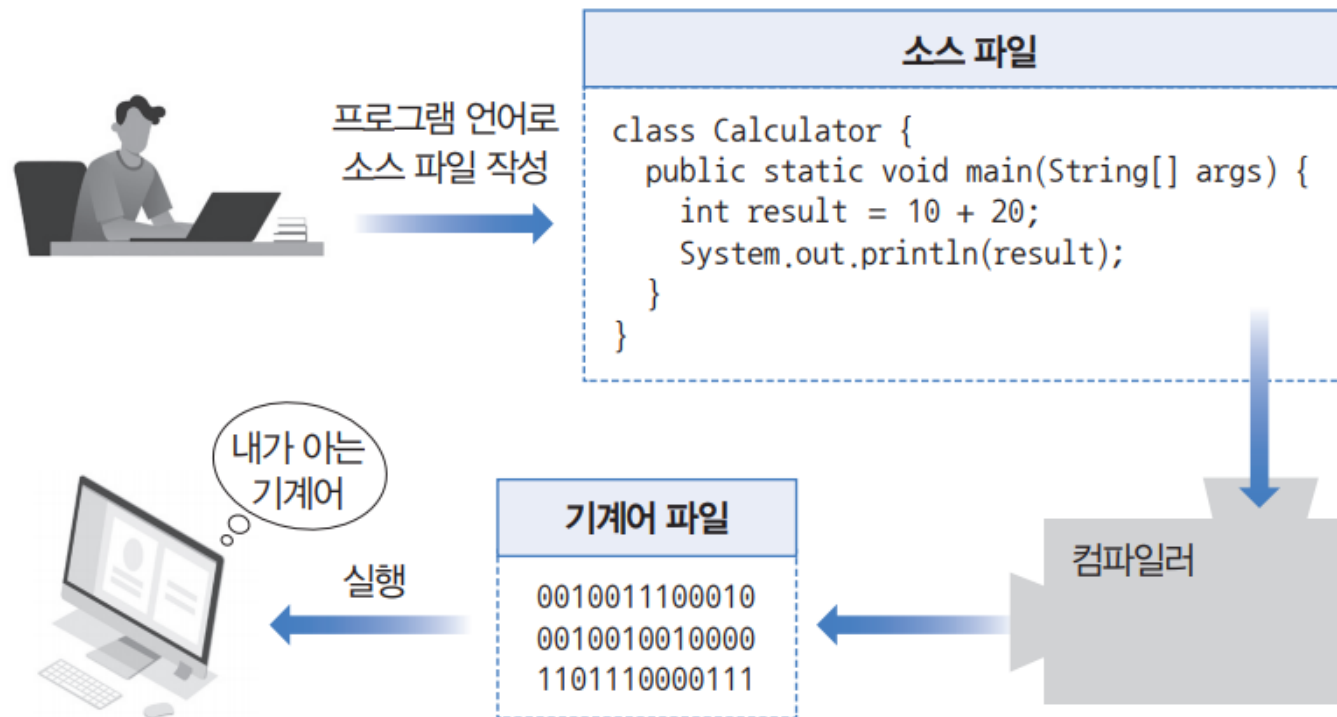
1.10 코드 용어 이해

1.11 코드 주석 달기

1.12 실행문과 세미콜론

### 프로그래밍 언어의 역할

- 사람과 컴퓨터의 대화 도움
- 사람의 언어와 기계어 사이에서 다리 역할 → 컴파일
- 고급 언어와 저급 언어로 구분



### 자바 소개

- 안드로이드 및 데스크톱 애플리케이션이나 웹사이트를 개발하는 핵심 언어
- 1995년 썬마이크로시스템즈(Sun Microsystems)에서 처음 발표
- 2010년 오라클에서 썬을 인수, 자바 개발 도구(JDK) 배포해 기술 지원



### 자바 특징

- 윈도우, 맥OS, 리눅스 등 모든 운영체제에서 실행 가능
- 먼저 객체(부품)를 만들고, 객체들을 서로 연결해서 더 큰 프로그램을 완성시키는 객체 지향 프로그래밍(OOP)에 최적화된 언어
- 메모리(RAM)를 자동 정리해 메모리 관리에 용이
- 무료로 다운로드해서 사용할 수 있는 오픈 소스 라이브러리(Open Source Library)가 풍부

### 자바 개발 도구(JDK) 설치

- JDK에는 Open JDK와 Oracle JDK 두 가지 있음

구분	Open JDK	Oracle JDK
라이선스	GNU GPL version 2	Oracle Technology Network License
사용료	무료	개발 및 학습용: 무료, 상업용: 유료
개발 소스 공개 의무	없음	없음

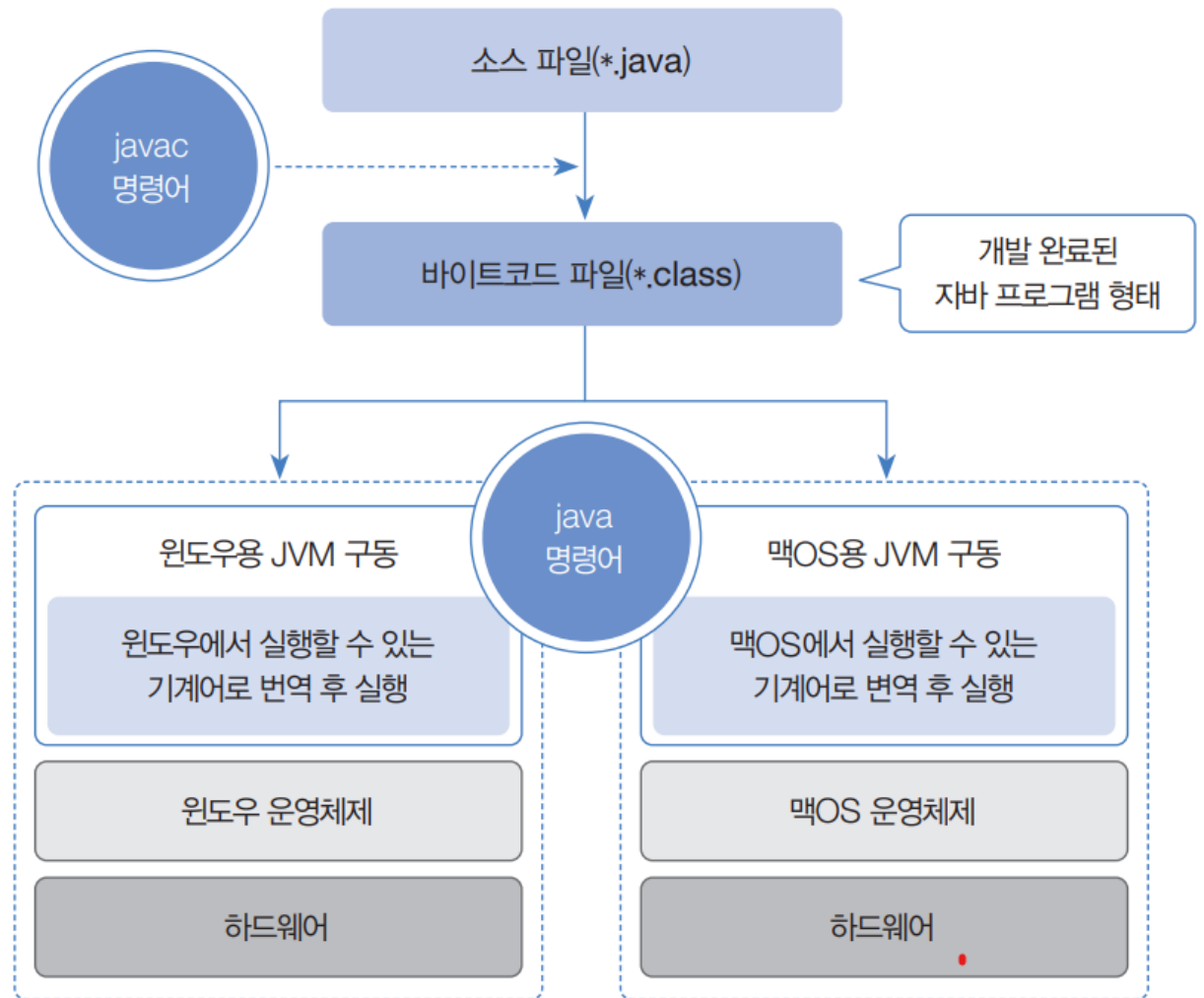
- Open JDK 다운로드: <https://jdk.java.net> 및 <https://adoptium.net>
- Oracle JDK 다운로드: <https://www.oracle.com/java/technologies/downloads>
- JDK LTS 버전 다운로드: <https://adoptium.net>

### 바이트코드 파일

- 소스 파일(.java)을 작성한 후 바이트코드 파일(.class)로 컴파일

### 자바 가상 머신

- java 명령어로 자바 가상 머신(JVM)을 구동시켜 바이트코드 파일(.class)을 기계어로 번역 및 실행
- 자바 가상 머신은 운영체제별로 다르게 설치됨



### 소스 파일 생성

- temp 디렉토리를 다음 구조로 생성

```
temp
|-- src
|   |-- ch01
|       |-- sec06
|           |-- Hello.java
|-- bin
```

← 소스 파일이 저장되는 디렉토리

← 패키지 디렉토리

← 소스 파일

← 바이트코드 파일이 저장되는 디렉토리

- Hello.java 소스 파일을 생성하고  
다음과 같이 코드 작성
- javac -d 및 java -cp 명령어로 컴파일

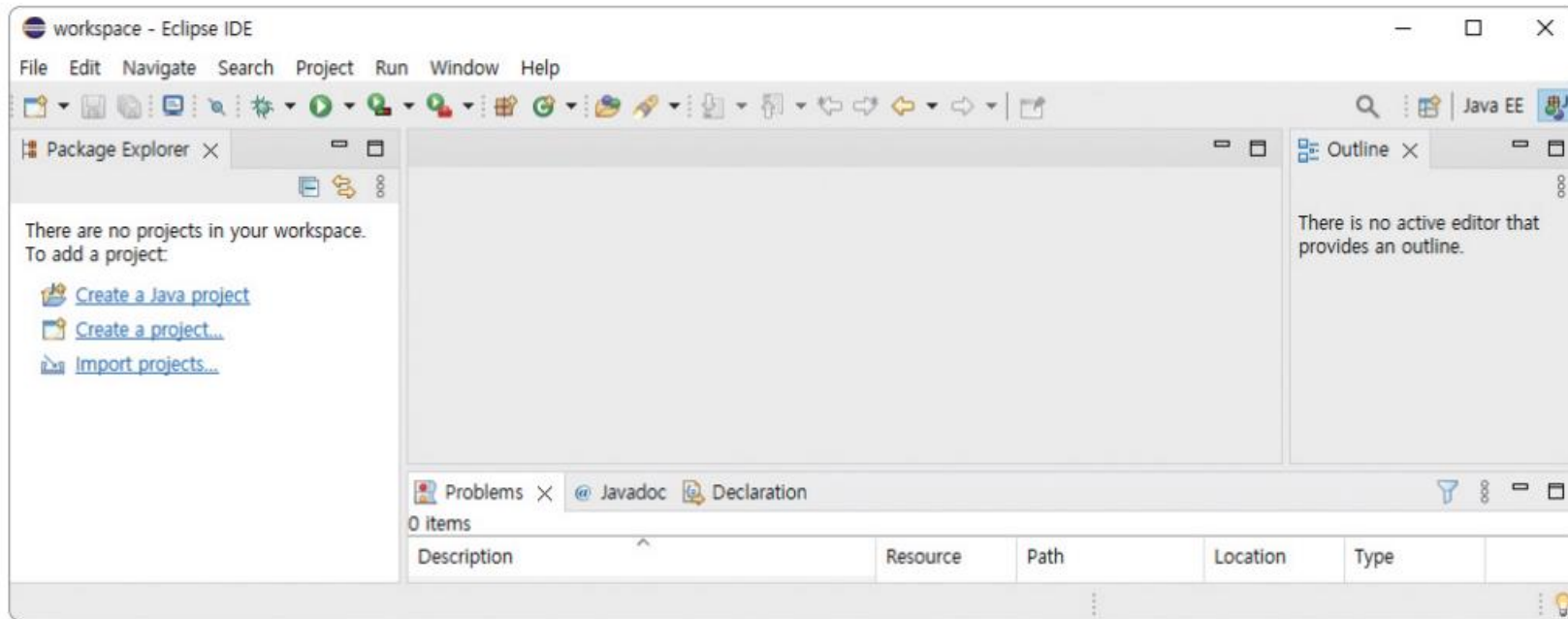
>>> Hello.java

```
1  package ch01.sec06;                //바이트코드 파일이 위치할 패키지 선언
2
3  public class Hello {                //Hello 클래스 선언
4      public static void main(String[] args) { //main() 메소드 선언
5          System.out.println("Hello, Java"); //콘솔에 출력하는 코드
6      }
7  }
```



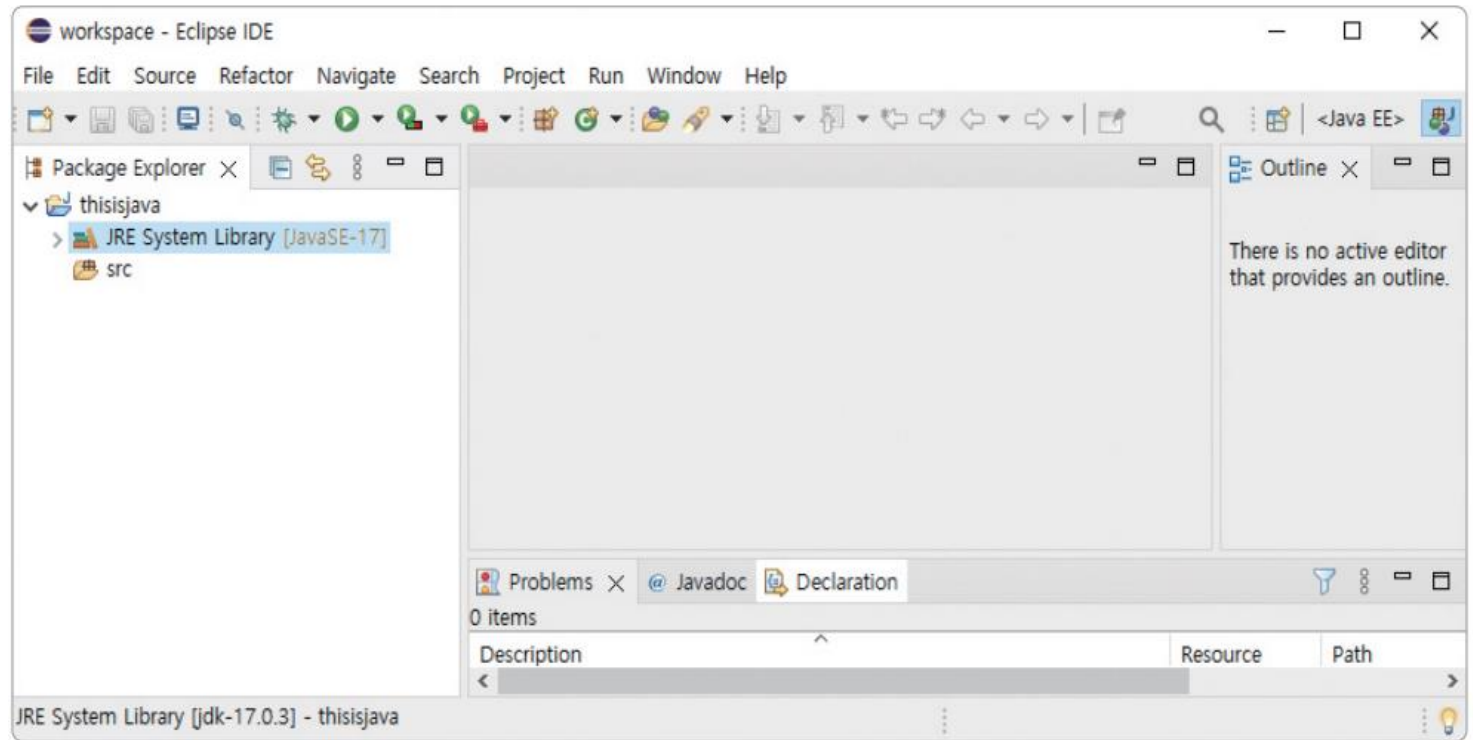
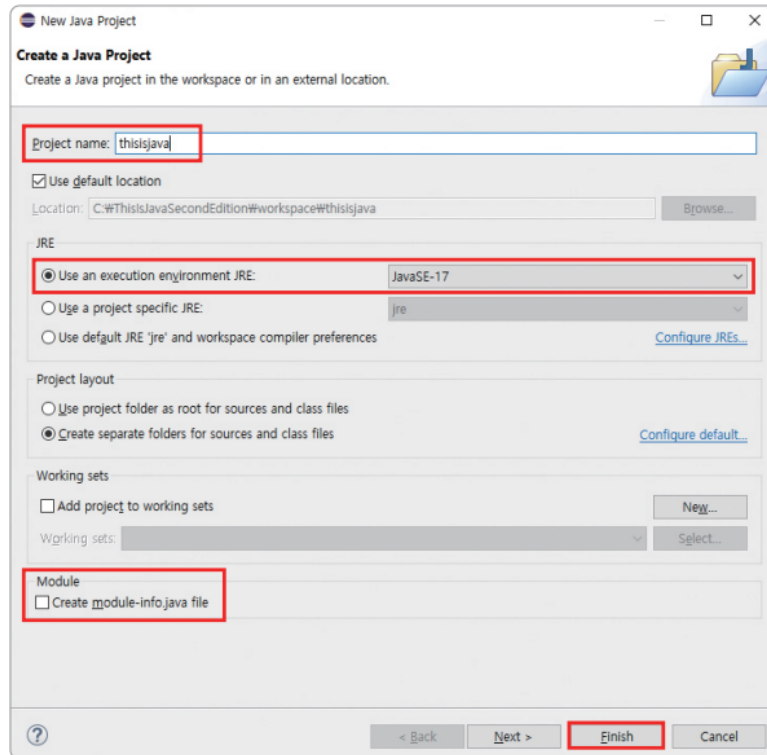
### 이클립스 다운로드 및 설치

- 현재 기업체에서 자바 프로그램을 개발할 때 가장 많이 사용하는 통합 개발 환경(IDE)
- 다운로드: <https://www.eclipse.org>
- JDK 17을 지원하는 Eclipse IDE 2021-12 이상 버전 사용. 가능하면 최신 버전 권장



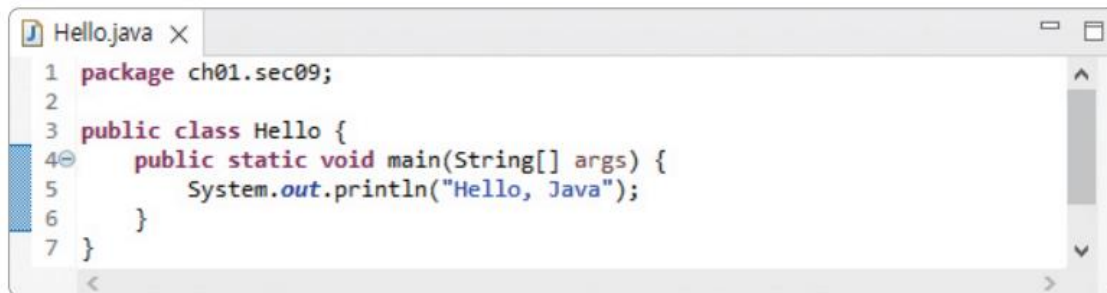
### 이클립스 프로젝트 생성

- Java Project 먼저 생성하고 JRE는 'JavaSE-17' 선택
- 윈도우인 경우 [Preferences]에서 Text file encoding을 'UTF-8' 선택

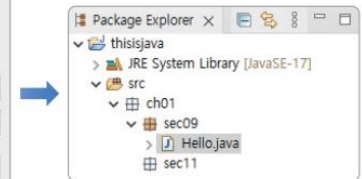
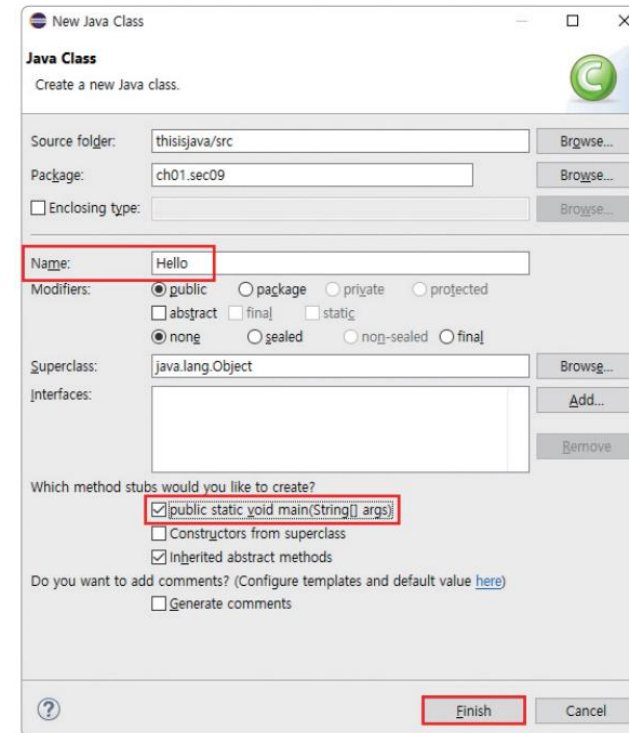


### 이클립스 소스 작성 및 실행

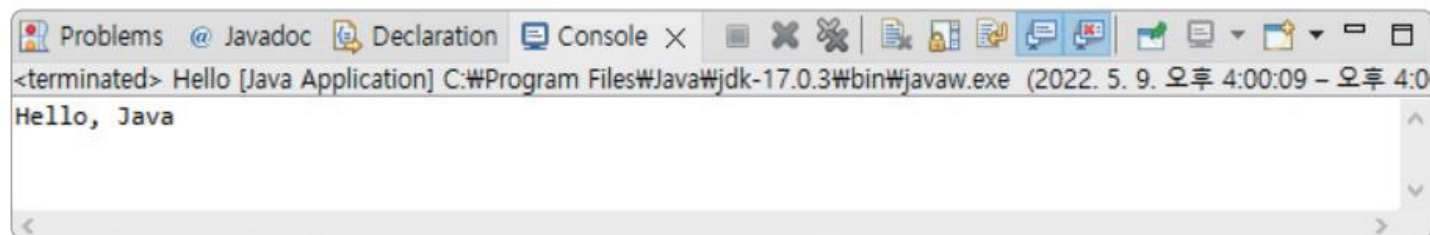
- 소스 파일과 바이트코드 파일을 관리하기 위한 패키지 생성
- Class를 추가해 Hello.java 소스 파일 작성



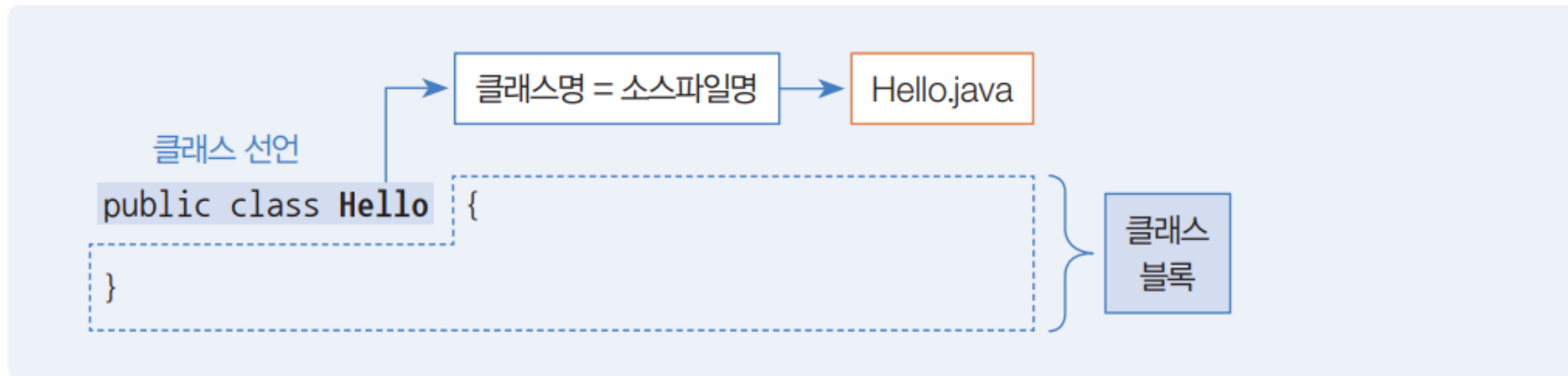
```
1 package ch01.sec09;  
2  
3 public class Hello {  
4     public static void main(String[] args) {  
5         System.out.println("Hello, Java");  
6     }  
7 }
```



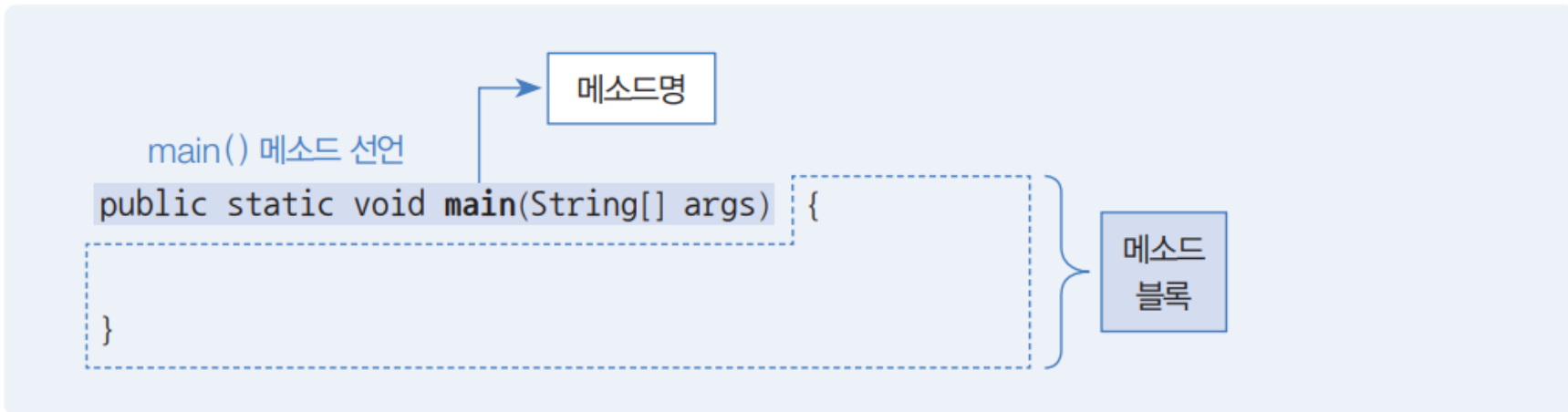
- 소스 파일을 실행하고 Console 뷰에서 결과 확인



### 클래스 선언



### 메소드



### 주석 기호와 설명

- 프로그램 실행과 상관없이 코드에 설명 붙임

구분	주석 기호	설명
행 주석	// ...	//부터 행 끝까지 주석으로 처리한다.
범위 주석	/* ... */	/*와 */ 사이에 있는 내용은 모두 주석으로 처리한다.
도큐먼트 주석	/** ... */	/**와 */ 사이에 있는 내용은 모두 주석으로 처리한다. javadoc 명령어로 API 도큐먼트를 생성하는 데 사용한다.

- 주석 기호는 문자열(" ") 내부에 작성하면 안 됨

```
System.out.println("Hello, /*주석이 될 수 없음*/ welcome to the java world!");
```

### 실행문

- 실행문은 변수 선언, 변수값 저장, 메소드 호출에 해당하는 코드
- 실행문 끝에는 반드시 세미콜론(;)을 붙여 실행문의 끝 표시

```
int x;           //변수 x 선언
x = 1;          //변수 x에 1 값을 저장
int y = 2;       //변수 y를 선언하고 2 값을 저장
int result = x + y; //변수 result를 선언하고 변수 x와 y를 더한 값을 저장
System.out.println(result); //콘솔에 변수의 값을 출력하는 println() 메소드 호출
```

Thank you!