

JAVA

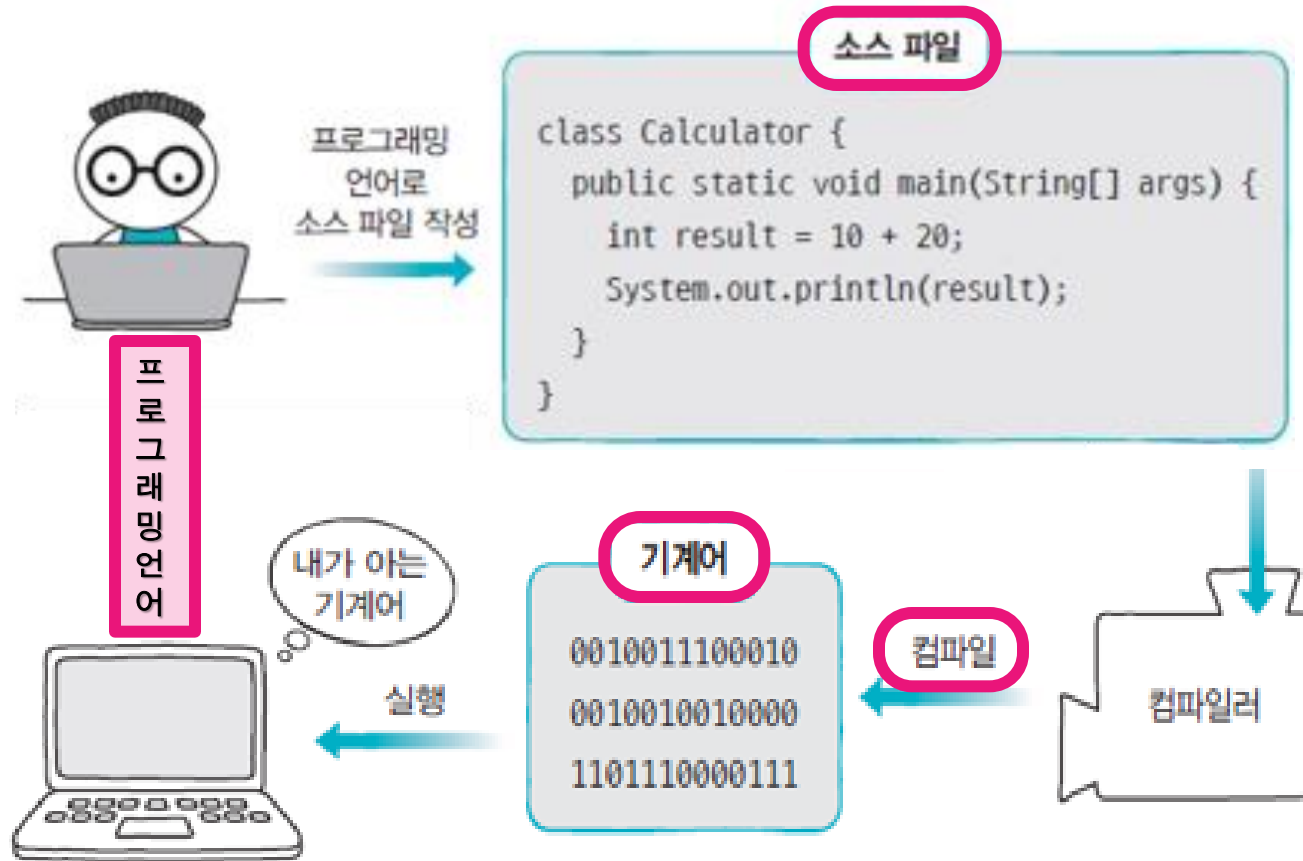
권정남

01

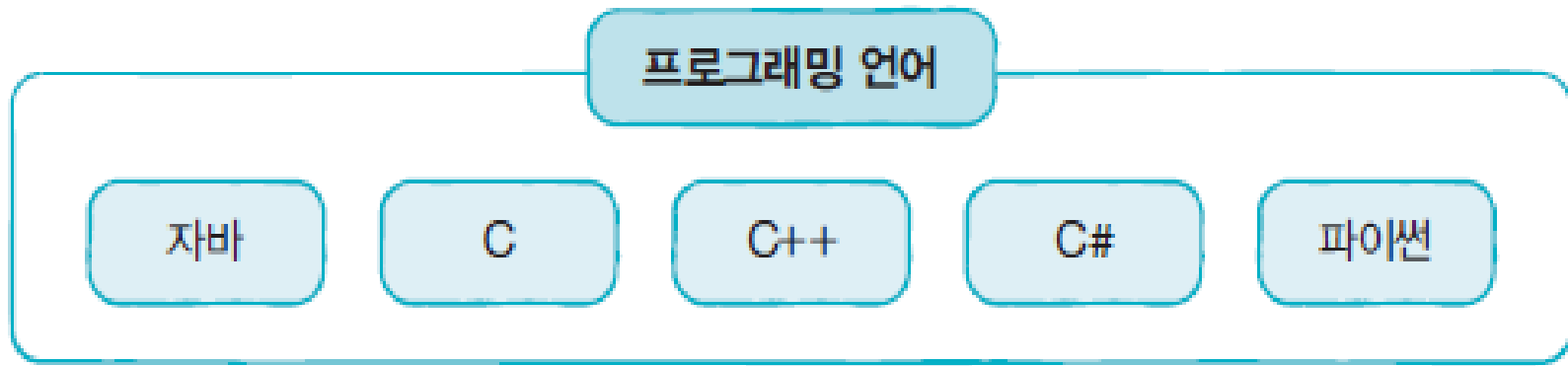
자바 시작하기

- 프로그래밍 언어와 자바
- 자바 개발환경 구축
- 자바 프로그래밍 개발 과정

❖ 시작하기 전에



프로그램은 특정 목적을 수행하도록 프로그래밍 언어로 작성된 소스를 기계어로 컴파일 한 것



자바(Java) : 임베디드, 웹, 안드로이드 소프트웨어(SW) 개발 분야

C / C++ : 하드웨어(HW) 장치 제어 및 임베디드 SW 분야

C++ / C# : 데스크탑 앱 또는 ASP.NET기반 웹 SW 분야

파이썬(Python) : 빅데이터 분석 및 머신러닝 SW 분야

❖ 자바 소개

- 1995년 마이크로시스템즈(Sun Microsystems)에서 발표
- 현재 웹사이트 및 다양한 애플리케이션 개발의 핵심 언어
- 오라클 (<http://www.oracle.com>) 라이선스
 - 자바 개발 도구(JDK)를 배포하여 개발에 필요한 기술적 지원을 함
- 특징
 - 모든 운영체제에서 실행 가능
 - 객체 지향 프로그래밍 (OOP) : 객체(부품)를 만들고 이들을 연결하여 더 큰 프로그램을 완성하는 기법
 - 메모리(RAM)를 자동 관리하므로 코드 작성에 집중할 수 있음
 - 풍부한 무료 라이브러리를 이용 → 개발 기간 단축 가능

❖ 자바 개발 분야



❖ 자바 개발 도구 (JDK : Java Development Kit)

■ JDK 역할

- 자바 언어로 소프트웨어를 개발할 때 필요한 환경 및 도구 제공

■ JDK 종류

- Open JDK: <https://openjdk.java.net>
 - 개발, 학습용 및 상업용 모두 무료로 사용
- Oracle JDK: <https://www.oracle.com>
 - 개발, 학습용은 무료로 사용
 - 상업용 목적으로 사용할 경우 연간 사용료 지불
 - 장기 기술지원(LTS: Long Term Support) 및 업데이트 제공으로 안정적

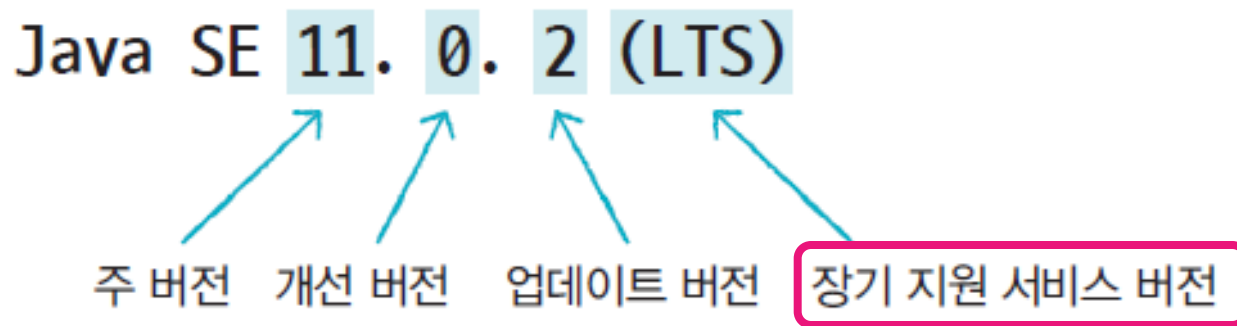
■ 학습용 JDK 선택

- 학습용은 안정적인 **Oracle JDK**를 사용하는 것이 좋음

자바 개발 도구 설치

❖ 오라클 사이트 : <https://www.oracle.com>

❖ JDK 버전 체계



LTS 버전 : 자바 8, 자바 11, 자바 17, **자바 21**

자바 개발 도구(JDK) 설치(p30 -37)

1. JDK 설치

- ① 오라클 웹 사이트 접속 - 제품 - 하드웨어 및 소프트웨어 - Java
- ② 우측 상단 - Java 다운로드 하기
- ③ JDK 21 - windows - x64 Installer 파일

163.91 MB

https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)

- ④ 파일 실행 - C:\Program Files\Java\jdk-21 에 저장
 - bin : 개발에 필요한 실행 파일이 있는 폴더
 - 자바 컴파일러 : javac.exe, 자바 실행 명령어 : java.exe

2. 환경변수 설정

- ① JAVA_HOME 환경 변수 등록
- ② Path 환경 변수 수정
- ③ 설치 확인 : cmd 실행 `javac -version => javac 21.0.x`

키워드로 끝내는 핵심 포인트

- **기계어**: 컴퓨터(운영체제)가 이해하고 실행할 수 있는 0과 1로 이루어진 코드를 말합니다.
- **프로그래밍 언어**: 사람이 기계어를 이해하는 것은 매우 어렵기 때문에 사람의 언어와 기계어의 다리 역할을 합니다. 종류로는 C, C++, 자바^{Java}, 파이썬^{Python} 등이 있습니다.
- **소스 파일**: 프로그래밍 언어로 작성된 파일을 말합니다.
- **컴파일**: 소스 파일을 기계어로 번역하는 것을 말합니다. 이 역할을 담당하는 소프트웨어를 컴파일러라고 합니다.
- **JDK**: 자바 개발 도구^{Java Development Kit}의 줄임말로, 자바로 프로그램을 개발할 수 있는 실행 환경(JVM)과 개발 도구(컴파일러) 등을 제공합니다.
- **환경 변수**: 운영체제가 실행하는 데 필요한 정보를 제공하는 변수를 말합니다. JDK를 설치한 후 명령 라인(명령 프롬프트, 터미널)에서 컴파일러(javac)와 실행(java) 명령어를 사용하려면 JAVA_HOME 환경 변수를 등록하고 Path 환경 변수를 수정하는 것이 좋습니다.

확인 문제 풀기 p38

01-2. 이클립스 개발 환경 구축

시작하기 전에

❖ 이클립스 (eclipse) : <http://www.eclipse.org>

■ 통합 개발 환경(IDE: Integrated Development Environment)

- IDE: 프로젝트 생성, 자동 코드 완성, 디버깅 등과 같이

개발에 필요한 여러 가지 기능을 통합적으로 제공해주는 툴

■ 기본적으로 자바 프로그램을 개발하도록 구성

- 플러그인(plugin)을 설치하면
웹 애플리케이션 개발,
C, C++ 애플리케이션 개발 등
다양한 개발 환경을 구축할 수 있음



좋은 편집 툴이란
개발자에게 보다
정확한 코딩을 유
도하도록 도와주
는 것!

■ 학습자 뿐만 아니라 고급 개발자에 이르기까지 광범위하게 사용

이클립스 설치

1. 이클립스 홈페이지 접속

① 우측 상단 - 다운로드



The image shows a screenshot of the Eclipse IDE download page. At the top center is the Eclipse logo, a purple circle with three horizontal white lines and an orange crescent on the left. Below the logo, the text "선호하는 데스크탑 IDE 패키지를 설치하세요" (Install the IDE package you prefer) is displayed. Underneath, there are two buttons: "더 알아보기" (Learn more) and "x86_64 다운로드" (Download x86_64). Below these buttons, the text "패키지 다운로드 | 도움이 필요하다" (Package download | Need help) is shown. On the right side, there is a large orange button with a download icon and the text "다운로드" (Download). Below this button, the text "다운로드 출처: 대한민국 - 카카오(https)" (Download source: South Korea - Kakao(https)) is displayed. Underneath, the file name "파일: eclipse-inst-jre-win64.exe" is shown in orange, followed by a box containing "SHA-512". At the bottom, the text ">> 다른 미리 선택" (See other pre-selects) is displayed.

선호하는 데스크탑 IDE 패키지를 설치하세요

더 알아보기 x86_64 다운로드

패키지 다운로드 | 도움이 필요하다

다운로드

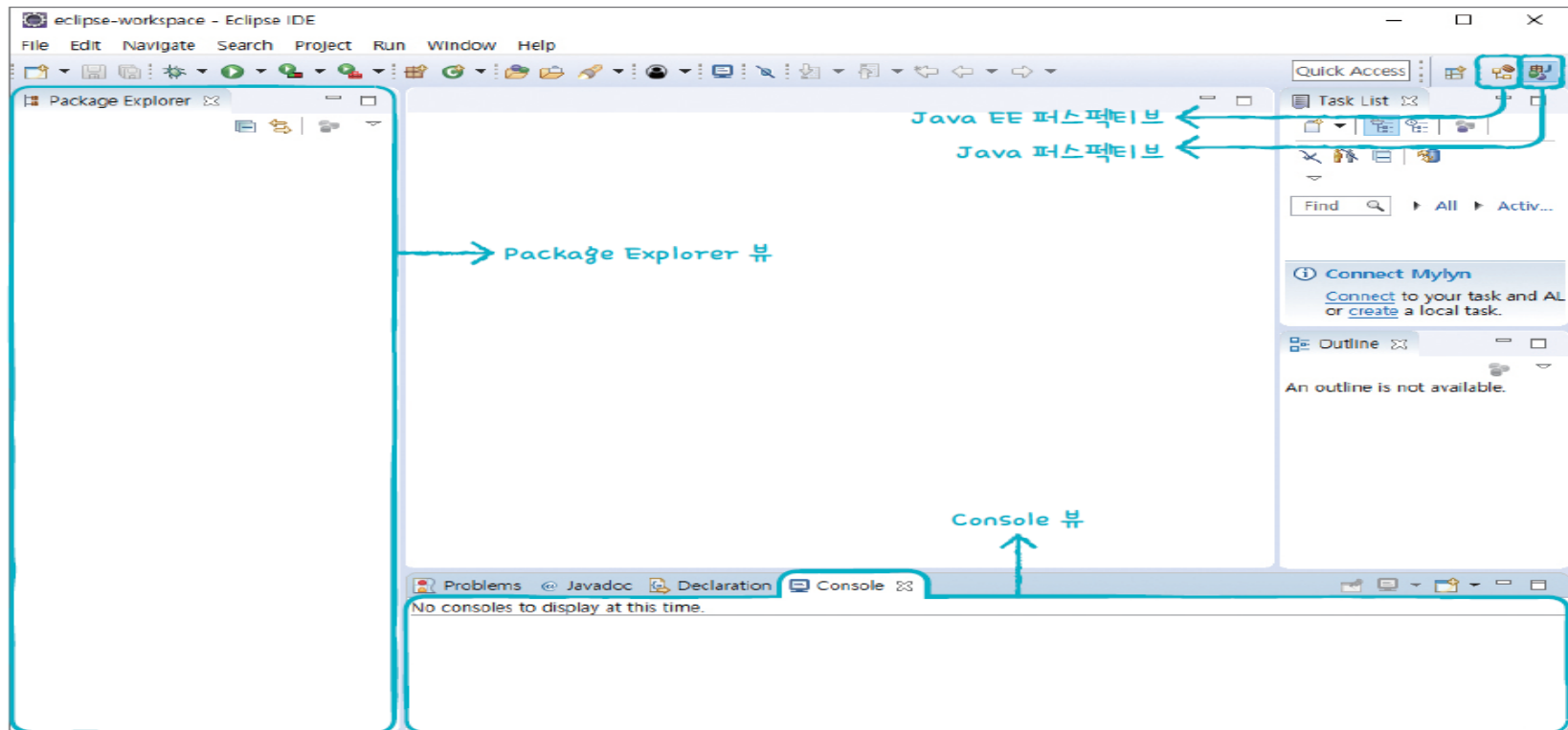
다운로드 출처: 대한민국 - 카카오(https)

파일: eclipse-inst-jre-win64.exe SHA-512

>> 다른 미리 선택

이클립스 화면(p44)

- ❖ 워크스페이스(workspace) : 프로젝트 폴더 저장
 - .metadata : 개발 환경 정보와 관련된 메타 데이터 저장 폴더
- ❖ 퍼스펙티브(perspective)
 - 프로젝트를 개발할 때 유용하게 사용할 수 있는 뷰View 묶음
- ❖ 뷰(view)
 - 이클립스 내부에서 사용되는 작은 창

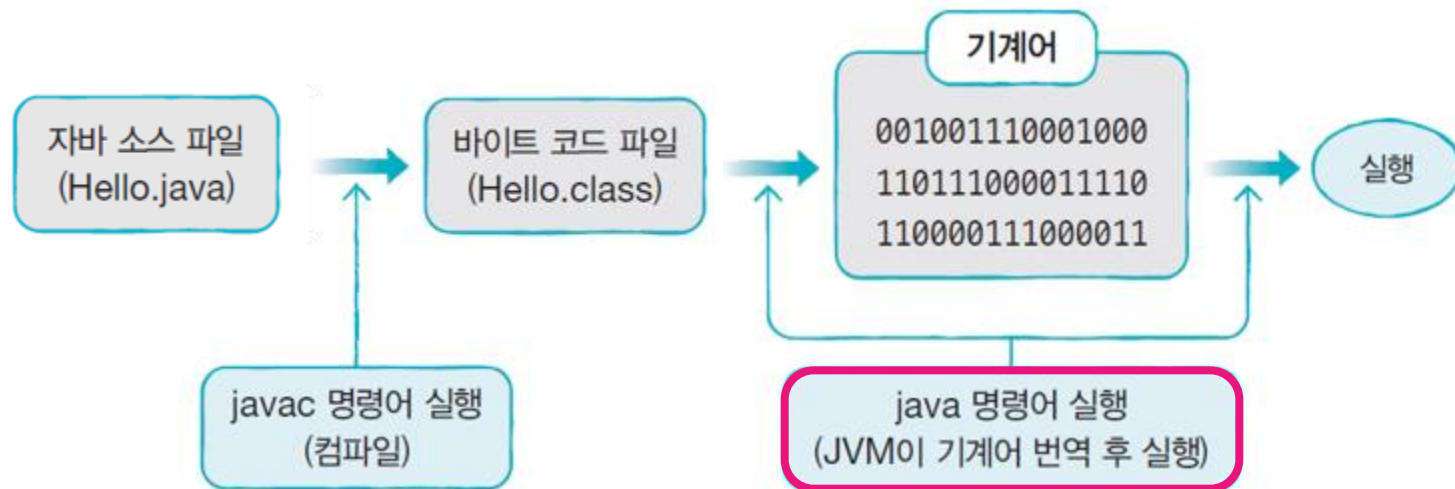


키워드로 끝내는 핵심 포인트

- **이클립스**: 무료로 사용할 수 있는 오픈 소스 통합 개발 환경^{IDE: Integrated Development Environment}입니다. IDE란 프로젝트 생성, 자동 코드 완성, 디버깅 등과 같이 개발에 필요한 여러 가지 기능을 통합적으로 제공해주는 툴을 말합니다.
- **워크스페이스**: 이클립스 실행과 관련된 메타 데이터^{metadata}와 프로젝트 폴더가 저장되는 폴더를 말합니다.
- **뷰**: 이클립스 내부에서 사용되는 작은 창을 말합니다.
- **퍼스펙티브**: 프로젝트를 개발할 때 유용하게 사용할 수 있는 뷰^{view}들을 미리 묶어 이름을 붙여 놓은 것을 말합니다.

01-3. 자바 프로그램 개발 과정

시작하기 전에



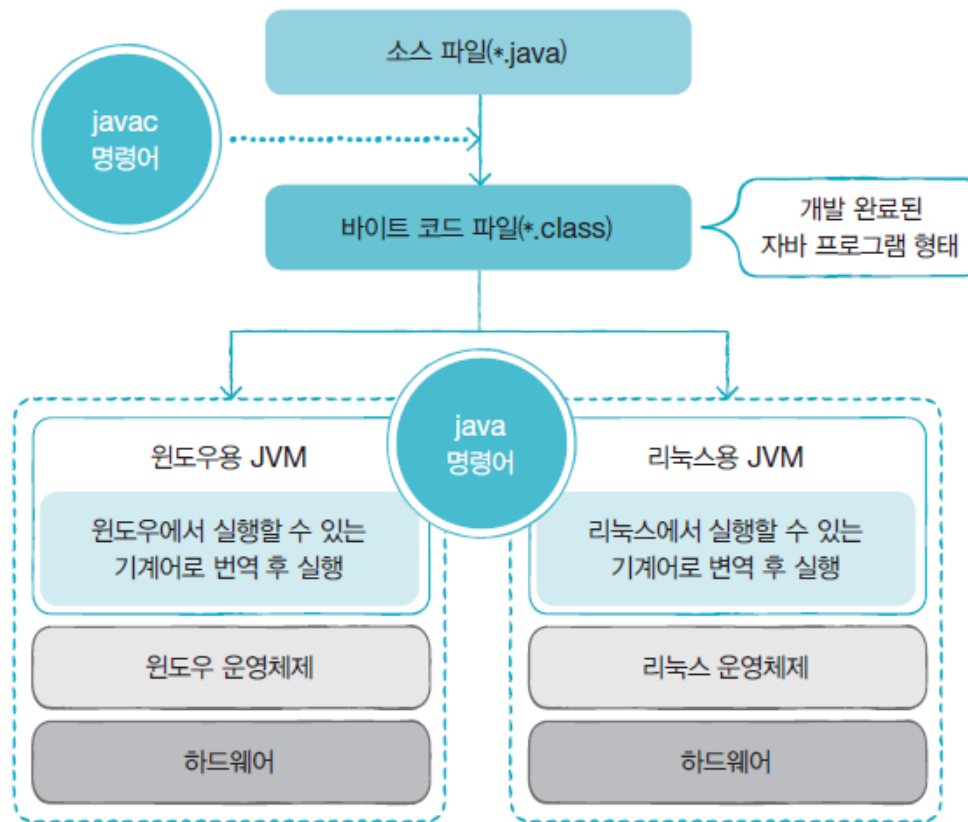
바이트 코드와 자바 가상 기계

❖ 바이트 코드 파일과 자바 가상 기계

- 자바 소스 파일 – compile - byte code 파일(.class)

* 운영체제에서 바로 실행할 수 없음

- 바이트 코드 - 자바 가상 기계(JVM: Java Virtual Machine) – 기계어 - 실행



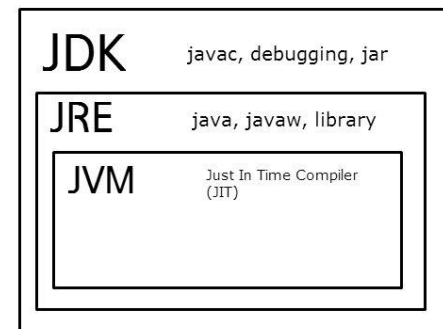
❖ 용어 정리

❖ JDK분류

- Java SE : java Platform Standard Edition : 자바 표준안
- Java EE : java Platform Enterprise Edition : 기업용 개발 환경
- Java ME : java Platform Micro Edition : 모바일 개발 환경

❖ **JDK**(Java Development Kit) : 개발자를 위한 자바 버전

- Java SE의 표준안에 따른 구체적인 소프트웨어
- Java 코드 컴파일러와 개발에 필요한 각종 도구 포함



❖ **JRE**(Java Runtime Environment) : 일반인을 위한 자바 버전

- 자바가 실제로 동작하는 데 필요한 JVM, 라이브러리, 각종 파일들 포함.

❖ **JVM**(Java Virtual Machine) : 자바가 실제 구동되는 환경

- 자바로 만들어진 소프트웨어는 JVM이라는 가상화된 환경에서 운영체제 버전에 따라 구동
- 하나의 자바 프로그램을 어떤 환경에서도 실행할 수 있도록 하는 역할

프로젝트 생성부터 실행까지

❖ 이클립스에서 실습 (p48)

■ [1단계] 프로젝트 생성

- [file] - [new]-[java project] - project 명 생성

■ [2단계] 소스 파일 생성과 작성

- src 폴더 – 우클릭 [new]-[package]
 - 패키지 생성(파일 관리를 위한 기능 => 폴더(디렉토리)와 비슷한 개념)
- 패키지 – [new]-[class] - 클래스 생성 : **클래스 명이 파일 이름이 됨**
- 소스 파일 저장 => 컴파일 수행 됨

■ [3단계] 바이트 코드 실행

- 실행버튼 Run 
- 소스파일 – 우 클릭 – **Run As – Java Application**

명령 라인에서 실행

1. bin 폴더 안에 패키지와 파일 삭제
2. 컴파일 : 프로젝트 폴더 안에서 명령 프롬프트 실행

프로젝트 저장 위치

옵션

[소스경로]

```
C:\selfJava\Whongongjava> javac -d bin src/chap01/*.java
```

컴파일명령

byte code 저장 위치]

3. 실행 : 바이트 코드 파일 저장 위치

바이트 코드 저장 위치

ClassPath

패키지.파일명

```
C:\selfJava\Whongongjava\bin> java -cp . chap01.Hello
```

실행
명령어

자바 명령어
실행위치

4. 결과

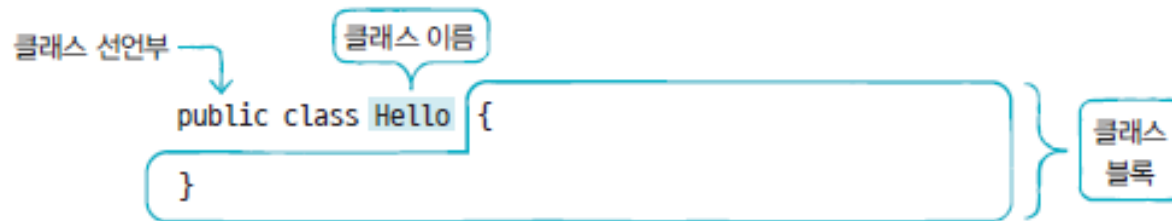
```
C:\selfJava\hongongjava\bin> java -cp . chap01.Hello
hello java
```

프로그램 소스 분석

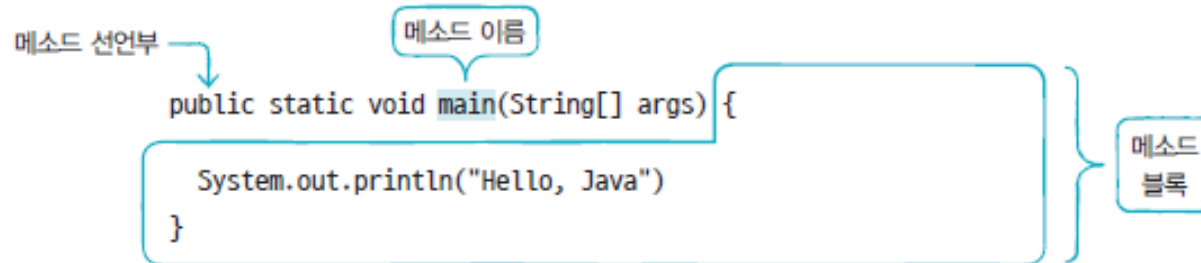
❖ 패키지 선언

```
package sec03.exam01;
```

❖ 클래스 선언



❖ 메소드 선언 : main() 메소드 – 프로그램 진입점



주석 사용하기

❖ 주석 : 코드에 설명 붙인 것

- 전체 코드를 이해하기 쉽고, 수정이 용이 하도록 설명을 붙이는 것.
- 컴파일 과정에서 무시 됨

❖ 주석의 종류

구분	주석 기호	설명
라인 주석	// ...	//부터 라인 끝까지 주석으로 처리합니다.
범위 주석	/* ... */	/*와 */ 사이에 있는 내용은 모두 주석으로 처리합니다.
도큐먼트 주석	/** ... */	/**와 */ 사이에 있는 내용은 모두 주석으로 처리합니다. 주로 javadoc 명령어로 API 도큐먼트를 생성하는 데 사용합니다.

실행문과 세미콜론(;)

❖ 실행문

- main() 메소드 블록 내부에 실행문 작성
- 실행문 끝에는 반드시 **세미콜론(;)** 을 붙여서 실행문이 끝났음을 표시

```
System.out.println("Hello, Java");
```

❖ 실행문의 종류

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

키워드로 끝내는 핵심 포인트

- **바이트 코드 파일:** 자바 소스 파일을 javac 명령어로 컴파일한 파일을 말합니다.
- **JVM:** 자바 가상 기계 Java Virtual Machine는 바이트 코드 파일을 운영체제를 위한 완전한 기계어로 번역하고 실행하는 역할을 합니다. JVM은 java 명령어에 의해 구동됩니다.
- **클래스 선언:** 자바 소스 파일은 클래스 선언부와 클래스 블록으로 구성됩니다. 이렇게 작성하는 것을 클래스 선언이라고 합니다.
- **main() 메소드:** java 명령어로 바이트 코드 파일을 실행하면 제일 먼저 main() 메소드를 찾아 블록 내부를 실행합니다. 그래서 main() 메소드를 프로그램 실행 진입점 entry point이라고 부릅니다.
- **주석:** 주석은 프로그램 실행과는 상관없이 코드에 설명을 붙인 것을 말합니다. 주석은 컴파일 과정에서 무시되고 실행문만 바이트 코드로 번역됩니다.
- **실행문:** 변수 선언, 값 저장, 메소드 호출에 해당하는 코드를 말합니다. 실행문 끝에는 세미콜론(;)을 붙여야 합니다.

확인 문제 풀기 p67