

JAVA

# 자바개발도구





# 학습 목표

자바 프로그램의 형태를 학습합니다.

자바를 학습하기 위한 환경에 대해 알아봅니다.

JDK와 이클립스 설치 과정을 학습합니다.

자바를 실습하기 위한 준비 과정을 학습합니다.



# 자바를 사용하려면 무엇이 필요한가?

**명칭: JDK (Java Development Kit)**

**설명: 자바 개발 도구**

**다운로드 위치:**

**<https://www.oracle.com/java/technologies/javase-downloads.html>**



# JDK와 JRE

## JRE(Java Runtime Environment)

- JRE는 자바 프로그램을 실행하기 위한 라이브러리, 자바 가상 기계, 기타 컴포넌트들을 제공한다.

## JDK(Java Development Kit)

- JDK는 JRE에 추가로 자바 프로그램을 개발하는데 필요한 컴파일러, 디버거와 같은 명령어행 개발 도구를 추가한 것이다.

*JDK = JRE + 컴파일러 + 디버거  
+ ...*





# JDK와 JRE

## JDK(Java Development Kit)

- 자바 응용 개발 환경. 개발에 필요한 도구 포함
  - 컴파일러, 컴파일된 자바 API 클래스들이 들어 있는 모듈 파일들, 샘플 등 포함

## JRE(Java Runtime Environment)

- 자바 실행 환경. JVM 포함
- 컴파일된 자바 API 들이 들어 있는 모듈 파일
- 개발자가 아닌 경우 JRE만 따로 다운 가능

## JDK와 JRE의 개발 및 배포

- 오라클의 Technology Network의 자바 사이트에서 다운로드
  - <http://www.oracle.com/technetwork/java/index.html>

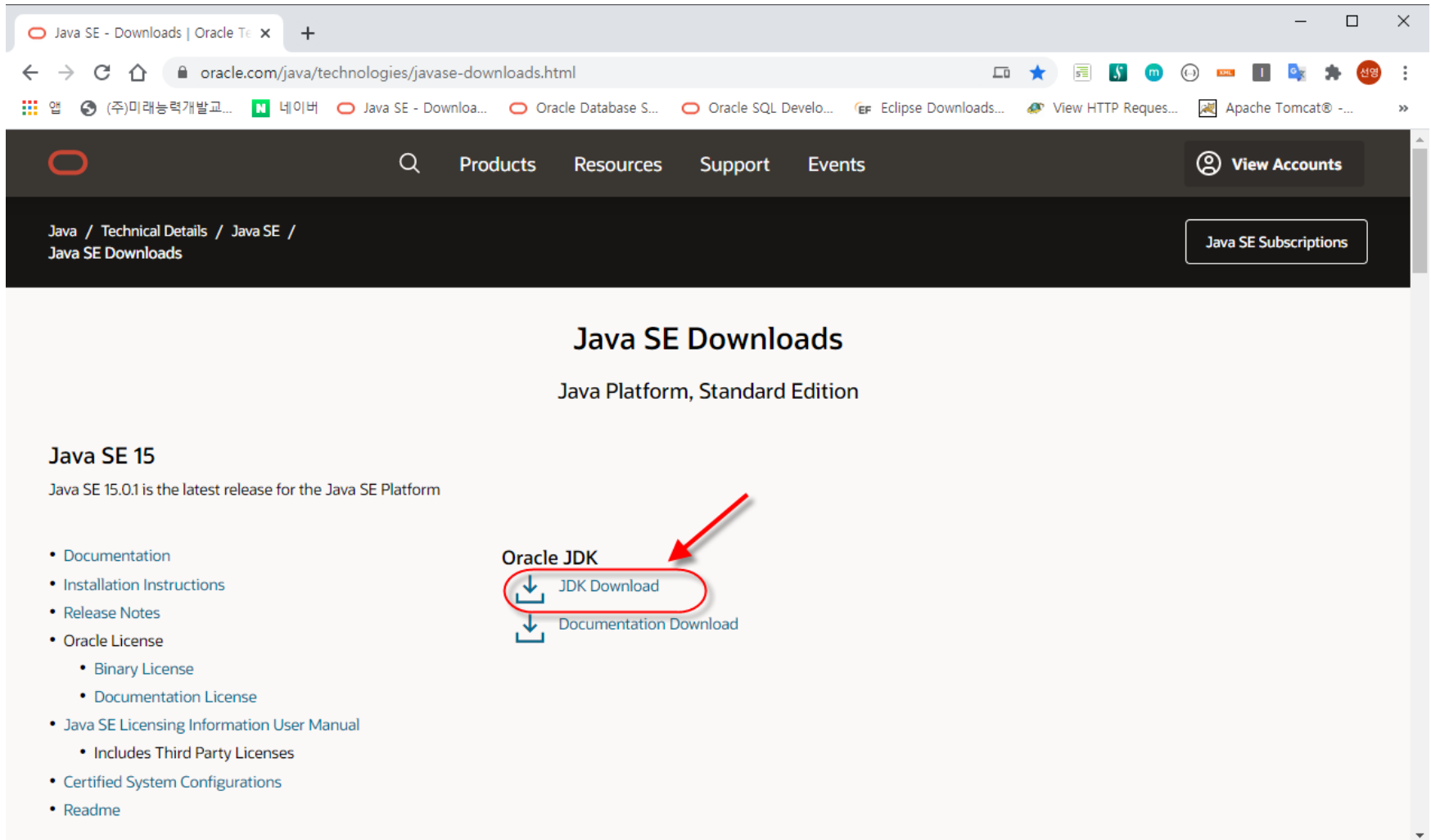
## JDK의 bin 디렉터리에 포함된 주요 개발 도구(jdk-11를 기준으로)

- javac - 자바 소스를 바이트 코드로 변환하는 컴파일러
- java - 자바 응용프로그램 실행기. 자바 가상 기계를 작동시켜 자바프로그램 실행
- javadoc - 자바 소스로부터 HTML 형식의 API 문서 생성
- jar - 자바 클래스들(패키지포함)을 압축한 자바 아카이브 파일(.jar) 생성 관리
- jmod: 자바의 모듈 파일(jmod)을 만들거나 모듈 파일의 내용 출력
- jlink: 응용프로그램에 맞춘 맞춤형(custom) JRE 제공
- jdb - 자바 응용프로그램의 실행 중 오류를 찾는 데 사용하는 디버거
- javap - 클래스 파일의 바이트 코드를 소스와 함께 보여주는 디어셈블러



# 다운로드

<https://www.oracle.com/java/technologies/javase-downloads.html>



Java SE - Downloads | Oracle Te x

oracle.com/java/technologies/javase-downloads.html

Products Resources Support Events View Accounts

Java / Technical Details / Java SE / Java SE Downloads

Java SE Subscriptions

## Java SE Downloads

Java Platform, Standard Edition

### Java SE 15

Java SE 15.0.1 is the latest release for the Java SE Platform

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

Oracle JDK

- [JDK Download](#)
- [Documentation Download](#)

# 다운로드










<https://www.oracle.com/java/technologies/javase-downloads.html>

Java SE Development Kit 15 - D x +

oracle.com/java/technologies/javase-jdk15-downloads.html

Java SE Development Kit 15.0.1

This software is licensed under the [Oracle Technology Network License Agreement](#) for Oracle Java SE

Product / File Description	File Size	Download
Linux ARM 64 RPM Package	141.81 MB	 <a href="#">jdk-15.0.1_linux-aarch64_bin.rpm</a>
Linux ARM 64 Compressed Archive	157.01 MB	 <a href="#">jdk-15.0.1_linux-aarch64_bin.tar.gz</a>
Linux x64 Debian Package	154.79 MB	 <a href="#">jdk-15.0.1_linux-x64_bin.deb</a>
Linux x64 RPM Package	162.02 MB	 <a href="#">jdk-15.0.1_linux-x64_bin.rpm</a>
Linux x64 Compressed Archive	179.33 MB	 <a href="#">jdk-15.0.1_linux-x64_bin.tar.gz</a>
macOS Installer	175.94 MB	 <a href="#">jdk-15.0.1_osx-x64_bin.dmg</a>
macOS Compressed Archive	176.53 MB	 <a href="#">jdk-15.0.1_osx-x64_bin.tar.gz</a>
Windows x64 Installer	159.69 MB	 <a href="#">jdk-15.0.1_windows-x64_bin.exe</a>
Windows x64 Compressed Archive	179.27 MB	 <a href="#">jdk-15.0.1_windows-x64_bin.zip</a>



# 중간 점검 문제

1. 자바 프로그램을 개발하는 데 개발 도구를 JDK 라고 한다.

2. JDK와 JRE의 차이점은 무엇인가?

**JRE**는 자바 프로그램을 실행만 시킬 수 있는 환경이고 **JDK**는 개발도 할 수 있는 환경이다.



# 자바 프로그램 개발 단계

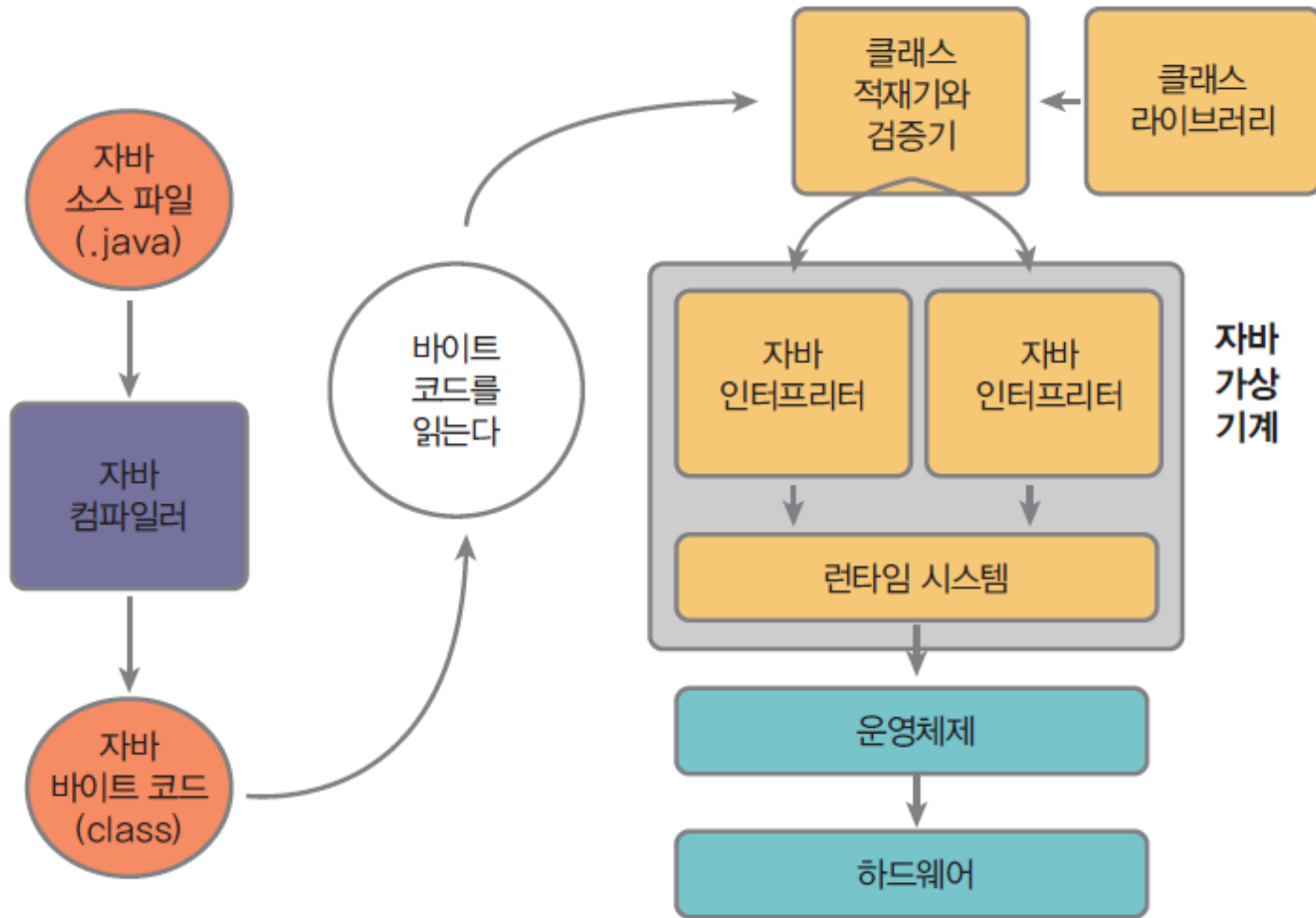


그림 1-12 • 자바 프로그램 개발 단계



# 자바 프로그램 개발 단계

## 소스 파일의 생성

- 에디터를 사용
- 소스 파일은 .java 확장자
- 메모장, 이클립스 ([www.eclipse.org](http://www.eclipse.org)), 넷빈([www.netbeans.org](http://www.netbeans.org)), JBuilder ([www.borland.com](http://www.borland.com)) 사용 가능

## 컴파일

- 컴파일러로 컴파일
- 컴파일러는 자바 소스 코드를 바이트 코드로 변환
- 바이트 코드는 확장자가 .class로 끝나는 파일에 저장

## 클래스 적재

- 바이트 코드 파일을 메모리로 적재
- 실행에 필요한 다른 클래스 파일도 함께 적재한다.

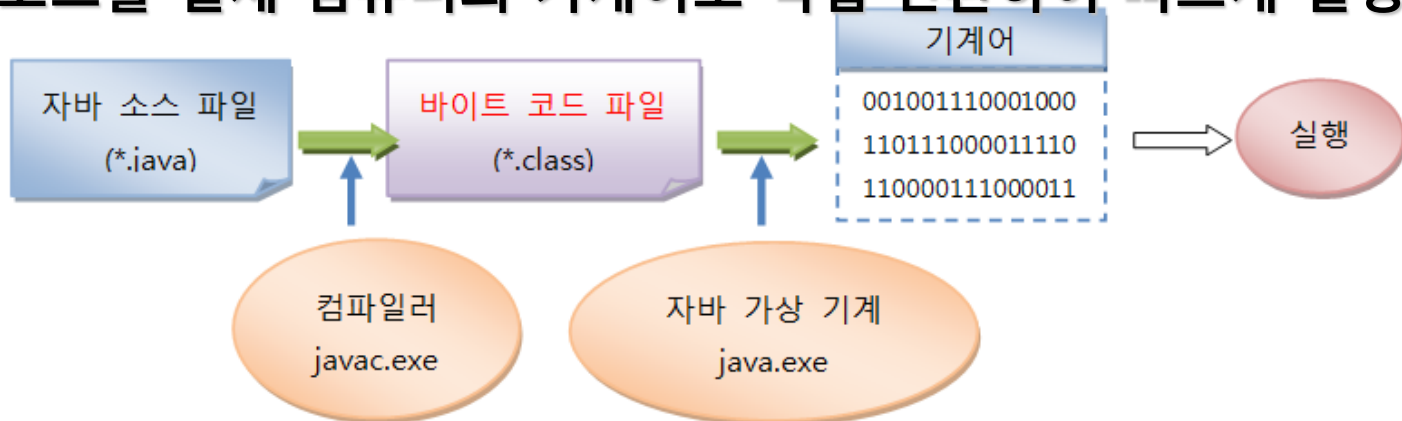
# 자바 프로그램 개발 단계

## 바이트 코드 검증

- 바이트 코드들이 이상이 없으며 자바의 보안 규칙을 위배하지 않는지를 검사

## 실행

- 자바 가상 기계(JVM)가 바이트 코드를 실행
- 초기 JVM은 단순히 한번에 하나의 바이트 코드를 해석하여서 실행하였기 때문에 상당히 실행 속도가 느림.
- 동시에 어떤 부분이 가장 빈번하게 실행되는지를 분석하여 가장 빈번하게 실행되는 부분(HotSpot)에서는 JIT 컴파일러가 바이트 코드를 실제 컴퓨터의 기계어로 직접 변환하여 빠르게 실행





# 자바 API

## 자바 API(Application Programming Interface)란?

- JDK에 포함된 클래스 라이브러리
  - 주요한 기능들을 미리 구현한 클래스 라이브러리의 집합
- 개발자는 API를 이용하여 쉽고 빠르게 자바 프로그램 개발
  - API에서 정의한 규격에 따라 클래스 사용

## 자바 패키지(package)

- 서로 관련된 클래스들을 분류하여 묶어 놓은 것
- 계층구조로 되어 있음
  - 클래스의 이름에 패키지 이름도 포함
  - 다른 패키지에 동일한 이름의 클래스 존재 가능
- 자바 API(클래스 라이브러리)는 JDK에 패키지 형태로 제공됨
  - 필요한 클래스가 속한 패키지만 import하여 사용
- 개발자 자신의 패키지 생성 가능



# Java 9부터 시작된 모듈 프로그래밍

## 모듈화(modularity)

- Java 9에서 정의된 새로운 기능, 2017년 9월 21일 출시
- **모듈**
  - 자바 패키지들과 이미지, XML 파일 등의 자원들을 묶은 단위
- 모듈 프로그래밍
  - 자바 응용프로그램을 마치 직소 퍼즐(jigsaw)을 연결하듯이 필요한 모듈을 연결하는 방식으로 작성

## 자바 플랫폼의 모듈화

- 실행 시간에 사용되는 자바 API의 모든 클래스들을 모듈들로 분할
- 모듈화의 목적
  - 세밀한 모듈화, 자바 응용프로그램이 실행되는데 필요없는 모듈 배제
  - 작은 크기의 실행 환경 구성
  - 하드웨어가 열악한 소형 IoT 장치 지원

## 모듈 방식이 아닌, 기존 방식으로 자바 프로그래밍 해도 무관

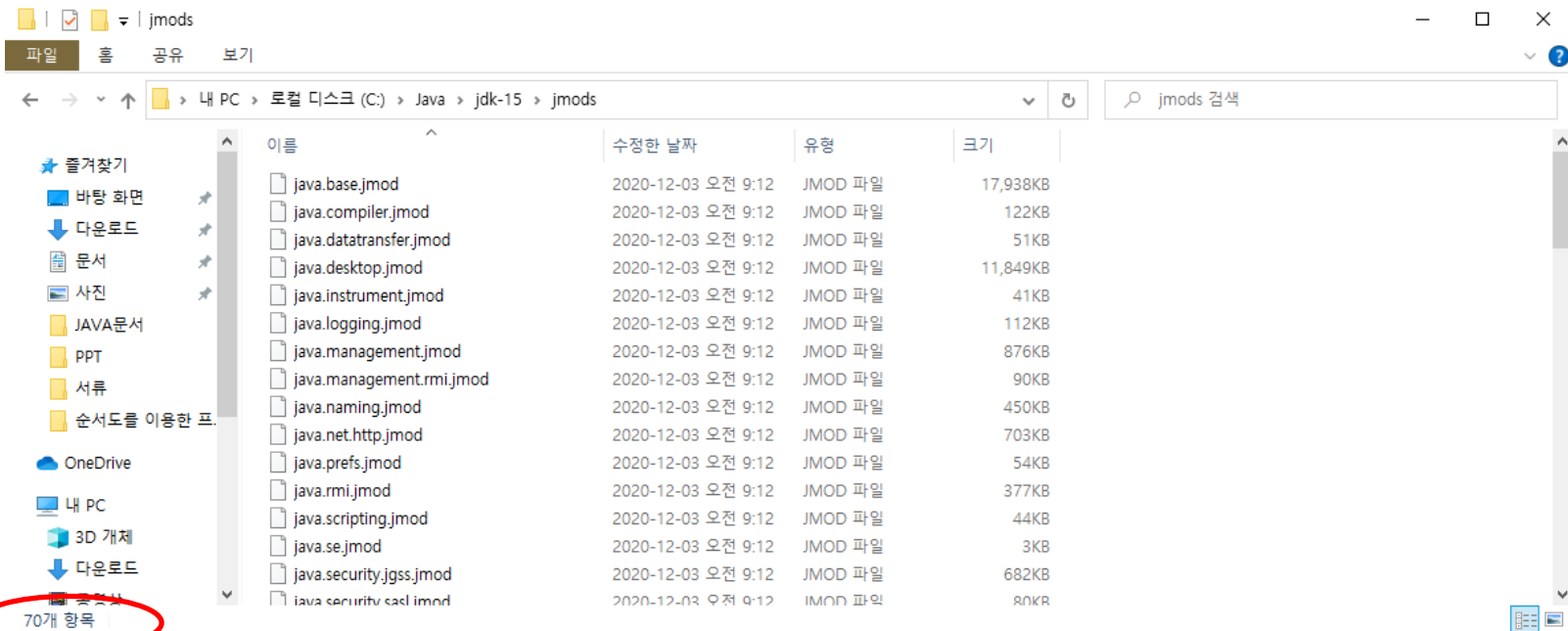
- 자바 플랫폼이 모듈 방식으로 바뀌었지만,
- 굳이 응용프로그램을 모듈 방식으로 작성할 필요 없음
  - 모듈 설계자들도 이런 사실 강조



# 자바에서 제공하는 전체 모듈 리스트(Java SE)

## Java 9부터 플랫폼을 모듈화함

- Java SE의 모든 클래스들을 모듈들로 재구성
- JDK의 설치 디렉터리 밑의 jmods 디렉터리에 있음

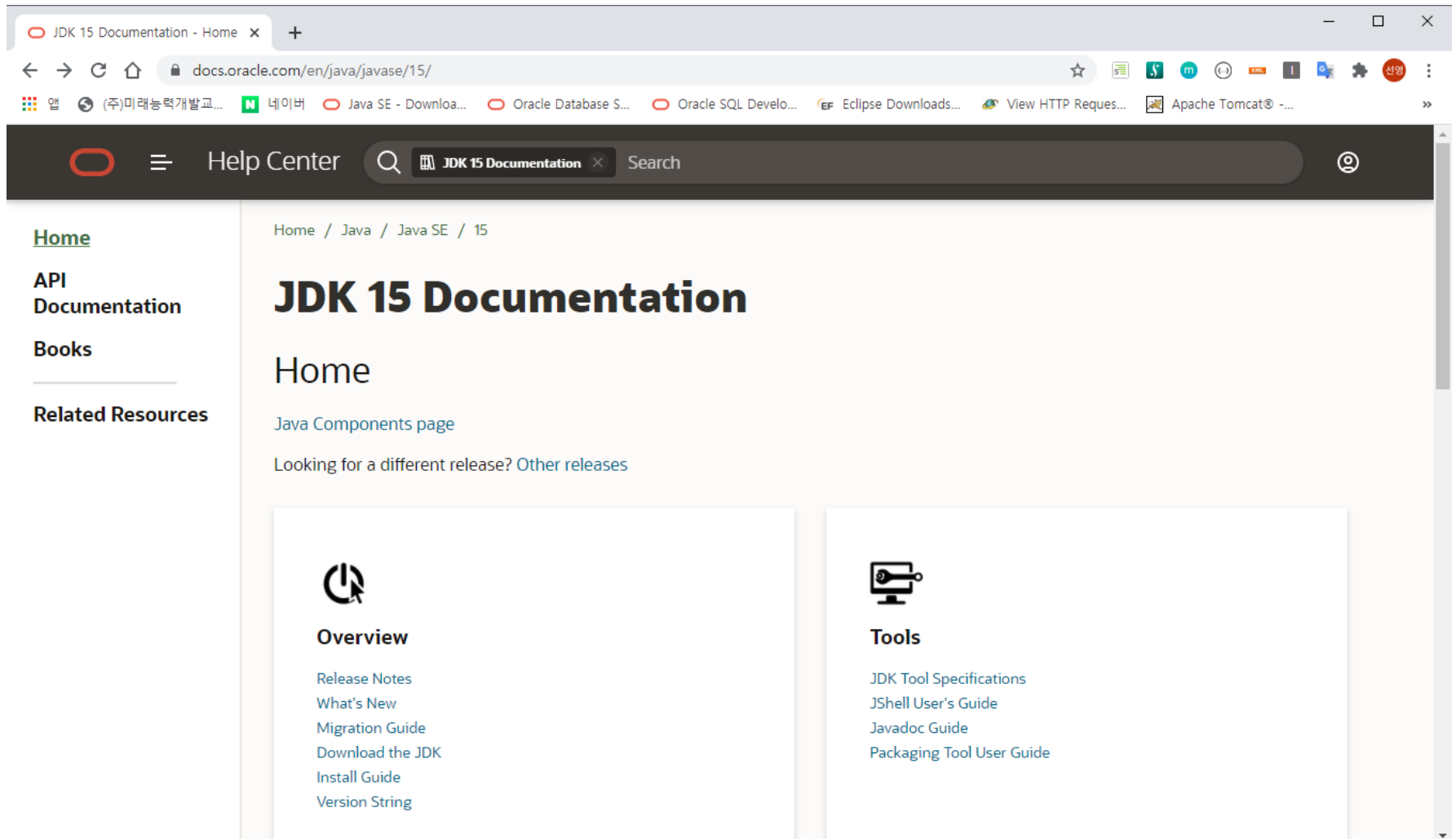


JDK 15에서는 자바 API의 클래스 패키지를 수십개의 모듈 파일에 분산

# 자바 문서 참조 하기

URL

<https://docs.oracle.com/en/java/javase/15/>



The screenshot shows a web browser displaying the JDK 15 Documentation page. The browser's address bar shows the URL <https://docs.oracle.com/en/java/javase/15/>. The page has a dark header with the Oracle logo, a hamburger menu, the text "Help Center", a search bar containing "JDK 15 Documentation", and a user profile icon. On the left side, there is a sidebar with links: "Home" (highlighted), "API Documentation", "Books", and "Related Resources". The main content area shows the breadcrumb "Home / Java / Java SE / 15", the title "JDK 15 Documentation", and the subtitle "Home". Below this, there are links for "Java Components page" and "Looking for a different release? Other releases". The page is divided into two columns. The left column, titled "Overview" with a power icon, contains links: "Release Notes", "What's New", "Migration Guide", "Download the JDK", "Install Guide", and "Version String". The right column, titled "Tools" with a monitor icon, contains links: "JDK Tool Specifications", "JShell User's Guide", "Javadoc Guide", and "Packaging Tool User Guide".

JDK 15 Documentation - Home x +

docs.oracle.com/en/java/javase/15/

Help Center Search JDK 15 Documentation

Home / Java / Java SE / 15

## JDK 15 Documentation

### Home

[Java Components page](#)

Looking for a different release? [Other releases](#)

#### Overview

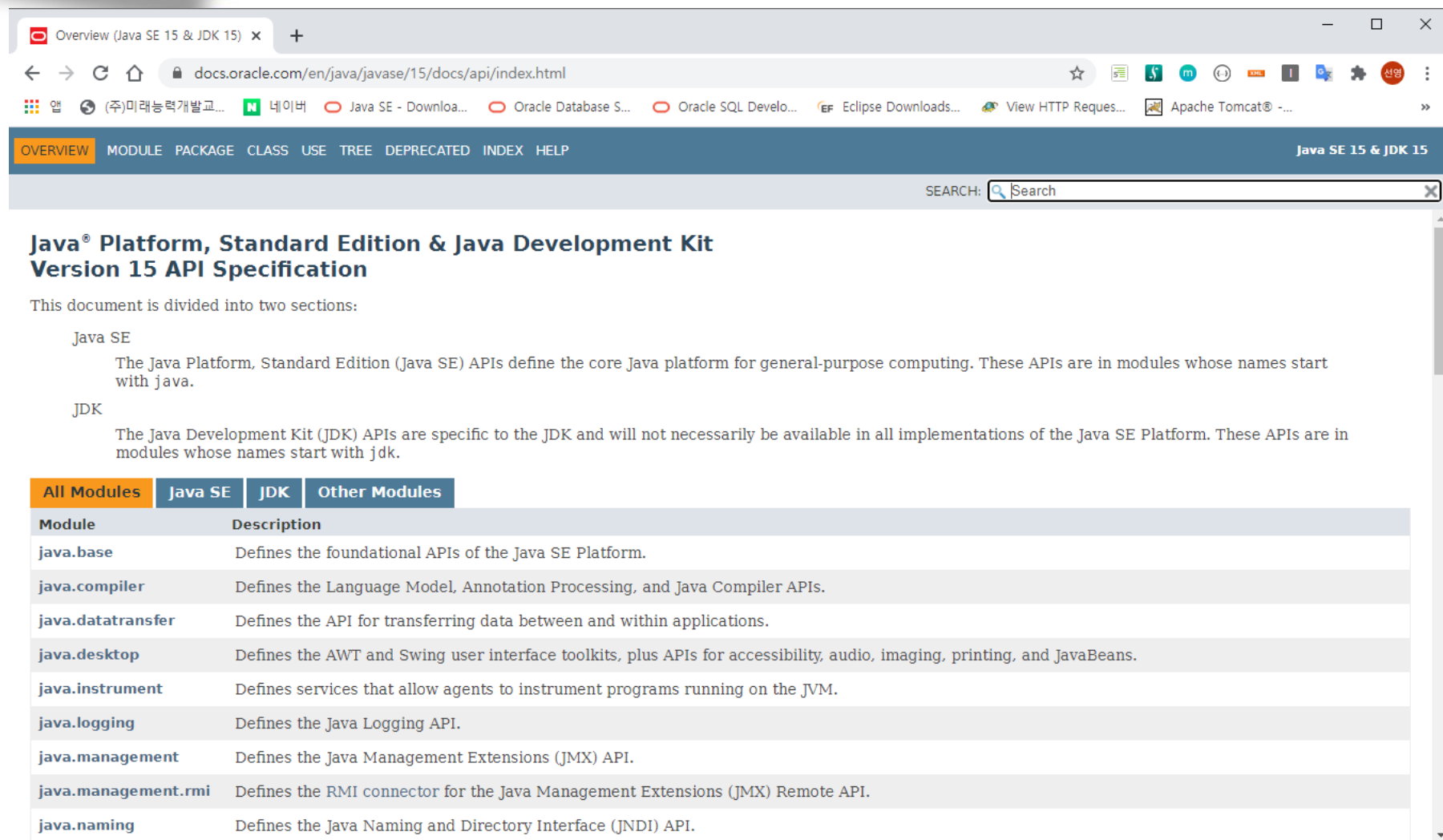
- [Release Notes](#)
- [What's New](#)
- [Migration Guide](#)
- [Download the JDK](#)
- [Install Guide](#)
- [Version String](#)

#### Tools

- [JDK Tool Specifications](#)
- [JShell User's Guide](#)
- [Javadoc Guide](#)
- [Packaging Tool User Guide](#)

# 자바 문서 참조 하기

## 자바 API 문서 : 자바 라이브러리에 대하여 설명하는 문서



The screenshot shows the Oracle Java SE 15 API Specification page. The browser address bar shows the URL `docs.oracle.com/en/java/javase/15/docs/api/index.html`. The page has a navigation bar with tabs for OVERVIEW, MODULE, PACKAGE, CLASS, USE, TREE, DEPRECATED, INDEX, and HELP. The OVERVIEW tab is selected. Below the navigation bar is a search bar. The main content area is titled "Java® Platform, Standard Edition & Java Development Kit Version 15 API Specification". It states that the document is divided into two sections: Java SE and JDK. Under Java SE, it says "The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with java." Under JDK, it says "The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with jdk." Below this text is a table with four columns: All Modules, Java SE, JDK, and Other Modules. The table lists various modules and their descriptions.

All Modules	Java SE	JDK	Other Modules
Module	Description		
<code>java.base</code>	Defines the foundational APIs of the Java SE Platform.		
<code>java.compiler</code>	Defines the Language Model, Annotation Processing, and Java Compiler APIs.		
<code>java.datatransfer</code>	Defines the API for transferring data between and within applications.		
<code>java.desktop</code>	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.		
<code>java.instrument</code>	Defines services that allow agents to instrument programs running on the JVM.		
<code>java.logging</code>	Defines the Java Logging API.		
<code>java.management</code>	Defines the Java Management Extensions (JMX) API.		
<code>java.management.rmi</code>	Defines the RMI connector for the Java Management Extensions (JMX) Remote API.		
<code>java.naming</code>	Defines the Java Naming and Directory Interface (JNDI) API.		

# 자바 문서 참조 하기

## 자바 API 문서 : 자바 라이브러리에 대하여 설명하는 문서

java.base (Java SE 15 & JDK 15) x +

docs.oracle.com/en/java/javase/15/docs/api/java.base/module-summary.html

OVERVIEW MODULE PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

MODULE: DESCRIPTION | MODULES | PACKAGES | SERVICES

SEARCH: Search

Java SE 15 & JDK 15

### Module java.base

Defines the foundational APIs of the Java SE Platform.

**Providers:**

The JDK implementation of this module provides an implementation of the `jrt` file system provider to enumerate and read the class and resource files in a run-time image. The `jrt` file system can be created by calling `FileSystems.newFileSystem(URI.create("jrt:/"))`.

**Module Graph:**

java.base

**Tool Guides:**

java launcher, keytool

**Since:**


9

### Packages

Exports

Package	Description
java.io	Provides for system input and output through data streams, serialization and the file system.
java.lang	Provides classes that are fundamental to the design of the Java programming language.
java.lang.annotation	Provides library support for the Java programming language annotation facility.
java.lang.constant	Classes and interfaces to represent <i>nominal descriptors</i> for run-time entities such as classes or method handles, and classfile entities such as

# String 클래스에 대한 설명



String (Java SE 15 & JDK 15)

docs.oracle.com/en/java/javase/15/docs/api/java.base/java/lang/String.html

OVERVIEW MODULE PACKAGE **CLASS** USE TREE DEPRECATED INDEX HELP

Java SE 15 & JDK 15

SUMMARY: NESTED | FIELD | CONSTR | METHOD    DETAIL: FIELD | CONSTR | METHOD    SEARCH:

**Module** java.base  
**Package** java.lang  
**Class** String

java.lang.Object  
  java.lang.String

**All Implemented Interfaces:**  
Serializable, CharSequence, Comparable<String>, Constable, ConstantDesc

---

public final class **String**  
extends Object  
implements Serializable, Comparable<String>, CharSequence, Constable, ConstantDesc

The String class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

```
String str = "abc";
```

is equivalent to:

```
char data[] = {'a', 'b', 'c'};  
String str = new String(data);
```

Here are some more examples of how strings can be used:

```
System.out.println("abc");  
String cde = "cde";  
System.out.println("abc" + cde);
```

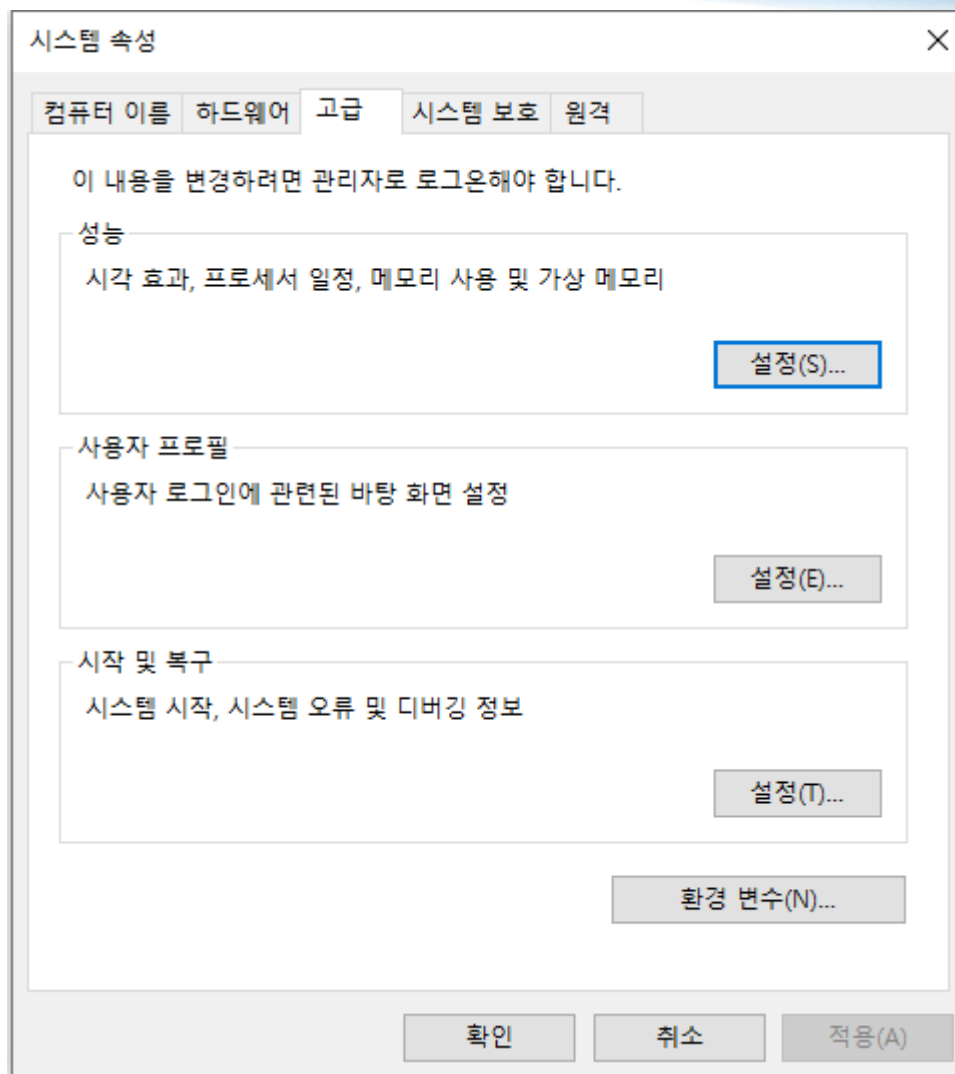




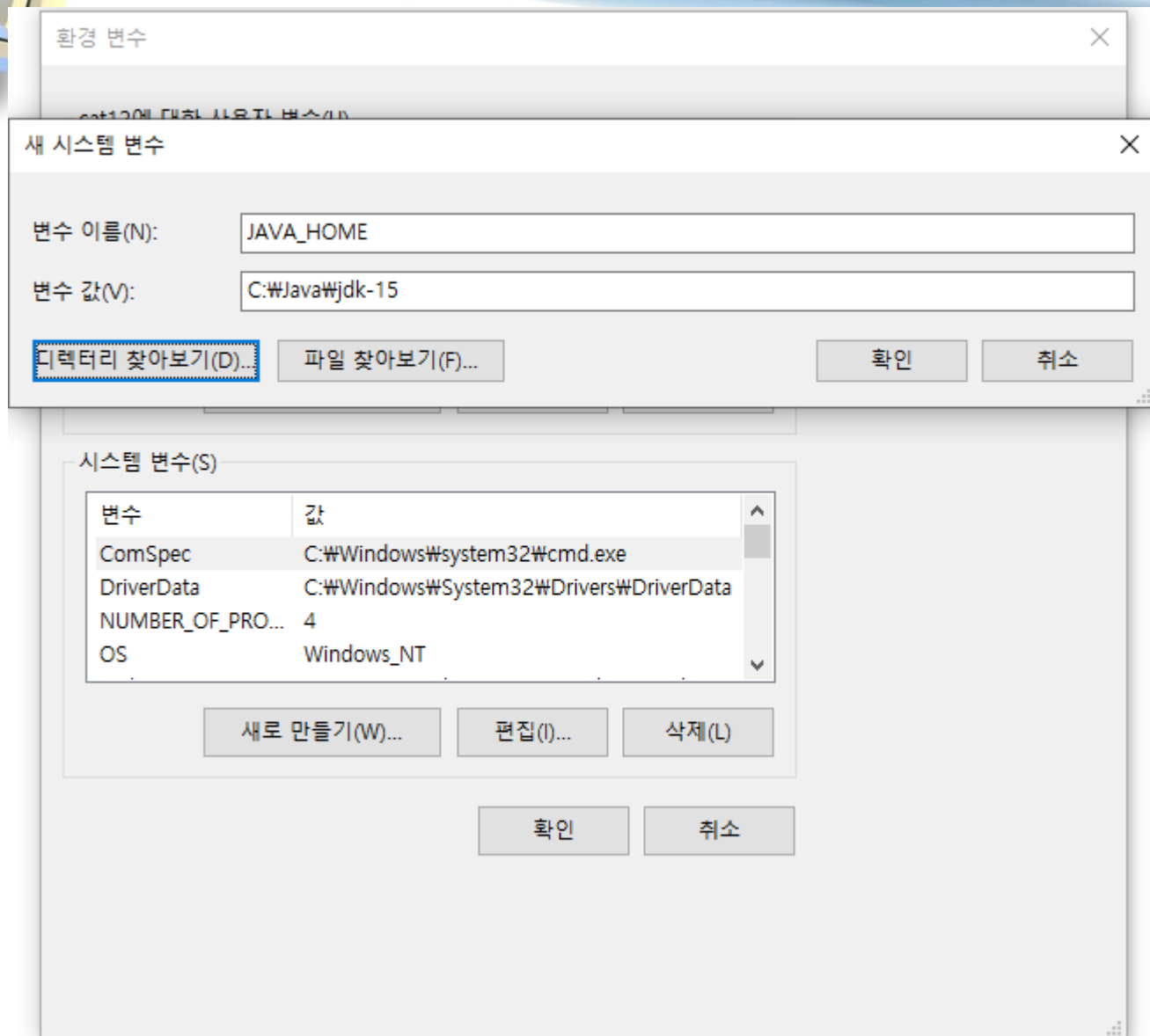
# 중간 점검 문제

1. 자바에서 제공하는 기본적인 클래스 라이브러리를 자바 API 라고 한다.
2. 자바 API 문서에서 System 클래스를 찾아서 어떤 내용들이 포함되어 있는지를 조사하여 보자.

# 설치 후 경로 지정하기



# 설치 후 경로 지정하기



# 설치 후 경로 지정하기

환경 변수

cat12에 대한 사용자 변수(U)

변수	값
OneDrive	C:\Users\cat12\OneDrive
OneDriveConsumer	C:\Users\cat12\OneDrive
Path	C:\Users\cat12\AppData\Local\Microso...
TEMP	C:\Users\cat12\AppData\Local\Temp

새로 만들기(N)...

편집(E)...

삭제(D)

시스템 변수(S)

변수	값
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\W...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.W...
PROCESSOR_ARC...	AMD64

새로 만들기(W)...

편집(I)...

삭제(L)

확인

취소

환경 변수 편집

C:\Program Files\Common Files\Oracle\Java\javapath  
%SystemRoot%\system32  
%SystemRoot%  
%SystemRoot%\System32\Wbem  
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\  
%SYSTEMROOT%\System32\OpenSSH\  
C:\Program Files\Intel\WiFi\bin\  
C:\Program Files\Common Files\Intel\WirelessCommon\  
C:\Java\jdk-15\bin

새로 만들기(N)

편집(E)

찾아보기(B)...

삭제(D)

위로 이동(U)

아래로 이동(O)

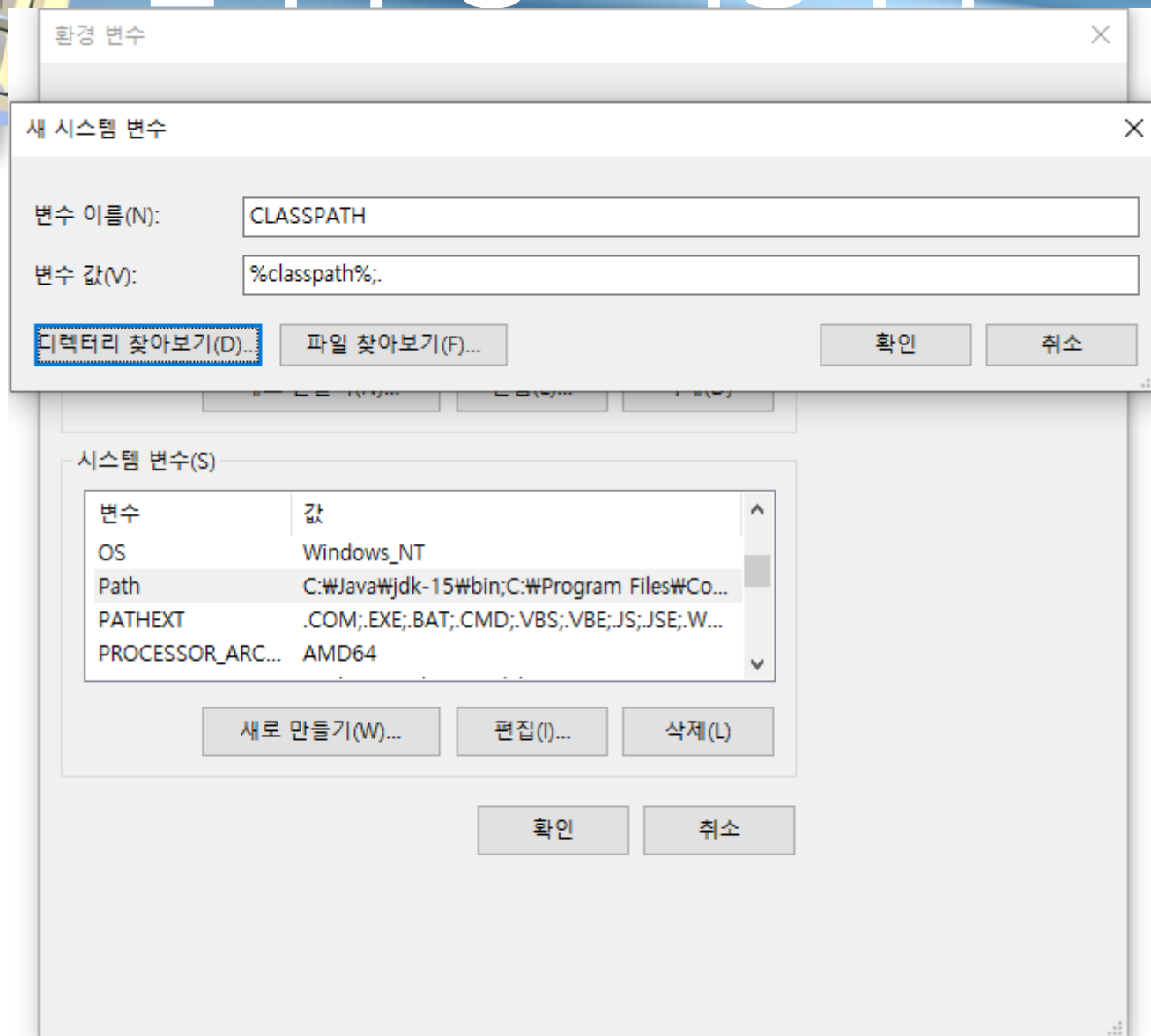
텍스트 편집(T)...

확인

취소

Path에 특정 폴더를 지정해 놓으면 어느곳에서도 해당 폴더에 있는 프로그램을 실행할 수 있어서 편리하다.

# 설치 후 경로 지정하기





# 설치 후 경로 지정

운영체제가 JVM을 올바르게 인식하도록 JAVA\_HOME이라는 이름으로 시스템 변수에 JDK가 설치된 폴더를 지정해야 한다

JDK에 포함된 명령어 도구를 사용하기 위해서는 먼저 어떤 디렉토리에서도 JDK명령어들을 찾을 수 있도록 운영체제에게 알려주는 것이 필요하다. 다음과 같이 Path라는 환경 변수를 적절하게 설정하여 주면 된다.

- ① 제어판을 열고 "시스템 " 아이콘을 더블클릭한다.
- ② "고급"탭을 선택하고 하단의 "환경변수"버튼을 클릭한다.
- ③ "시스템변수"중에서 "새로 만들기" 버튼을 클릭하여  
**JAVA\_HOME** 을 새로 만들고 **C:\Java\jdk-15** 추가한다.
- ④ " 시스템변수 " 중에서 " Path " 를 선택하고 " 편집 " 버튼을 누른다.
- ⑤ **Path** 변수의 값 첫 부분에 JDK bin 폴더를 추가한다. 새로 만들기를 클릭하여 **C:\Java\jdk-15\bin**(또는 %JAVA\_HOME%\bin) 으로 작성한다.
- ⑥ "새로 만들기"로 변수이름은 **CLASSPATH**, 변수값은 **%classpath%;**로 추가한다.

Path에 특정 폴더를 지정해 놓으면 어느곳에서도 해당 폴더에 있는 프로그램을 실행할 수 있어서 편리하다.



# 소스 파일 작성

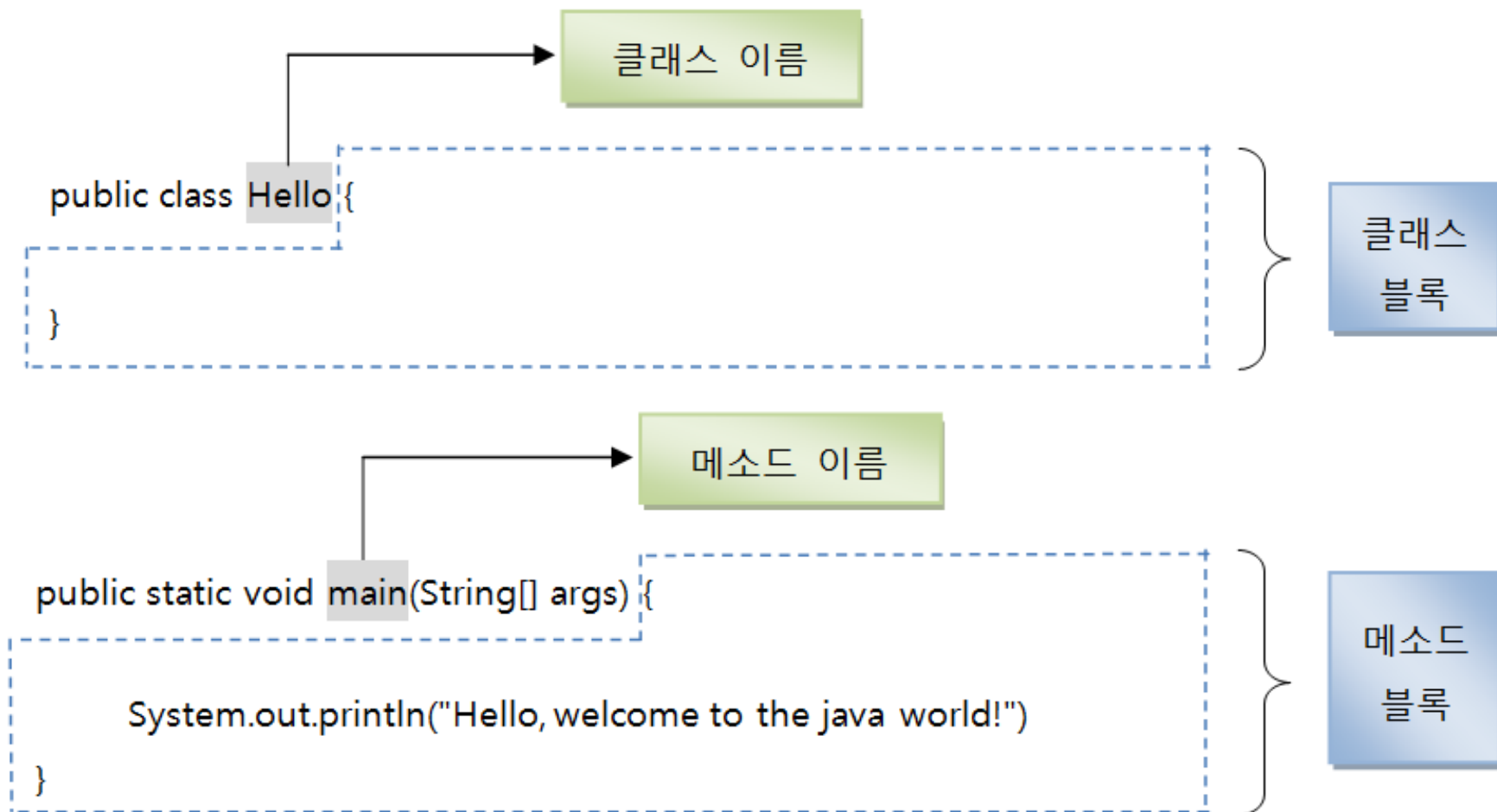
윈도우 보조 프로그램의 메모장 사용하여 코딩한다.  
파일의 확장자는 .java로 한다.

```
public class Hello {  
    public static void main(String[] arg){  
        System.out.print("안녕하세요. ");  
        System.out.print("자바 수업 시간에");  
        System.out.println("예제 파일입니다.");  
    }  
}
```

클래스 : 하나의 이상의 필드 또는 메서드를 포함하는 블록.  
메서드 : 어떤 일을 처리하는 실행문들을 모아 놓은 블록.

# 프로그램 소스 분석

자바 프로그램은 클래스들의 모임으로 구성된다. 객체 지향 언어에서는 프로그램 개발의 단위가 클래스이다. 자바에서는 클래스를 하나씩 만들어감으로써 전체 프로그램이 완성된다.





# 컴파일

명령 프롬프트(CMD)를 열어서 아래와 같이 코딩하여 확인한다

```
C:\Temp>javac Hello.java
```

```
C:\Temp>
```

```
C:\Temp>dir
```

```
...
```

```
2009-06-05 오후 04:06 454 Hello.class
```

```
2009-06-05 오후 02:53 144 Hello.java
```

```
2개 파일 598 바이트
```

```
C:\Temp>javap -c Hello > Hello.m
```

```
C:\Temp>type "Hello.m"
```

javap 커맨드는 어떤 바이트코드 파일이 어디서부터 나왔으며, 어떠한 필드와 메소드를 갖고 있는 파일인지를 알려주는 기능을 수행하게 된다.



# 실행

```
C:\WTemp>java Hello
```

안녕하세요. 자바 첫 수업 시간에 첫 예제 실행입니다.

```
C:\WTemp>
```

명령어 도구	설 명
java	자바 가상머신을 구동시키고 그 위에 자바 프로그램이 실행되도록 돕는 프로그램.
javac	자바 컴파일러는 소스파일에 저장되어 있는 소스 코드를 가상머신이 이해할 수 자바 바이트코드로 변환해주는 프로그램
javadoc	도큐먼트 생성기, 자바 소스 파일 안에 <code>/** */</code> 로 주석을 만들면 이 주석을 추출하여서 문서로 만들어 준다.





# 통합 개발 환경

**통합 개발 환경(integrated development environment)**

**: 소스 에디터, 컴파일러, 디버거가 결합**

**비주얼 컴포넌트를 마우스로 드래그하고 드롭하여 응용 프로그램을 생성할 수 있는 비주얼 개발 도구를 포함하는 것도 있다.**

**(예) 이클립스(Eclipse), 넷빈(NetBeans)**

**통합 개발 환경도 내부적으로는 명령어 도구들을 이용한다.**



# 중간 점검 문제

1. 컴파일하는 명령어 버전 이름은 javac 이다.
2. Hello.java를 컴파일하면 Hello.class 파일이 생성된다.
3. 자바 가상 기계는 java 프로그램으로 구현한다.

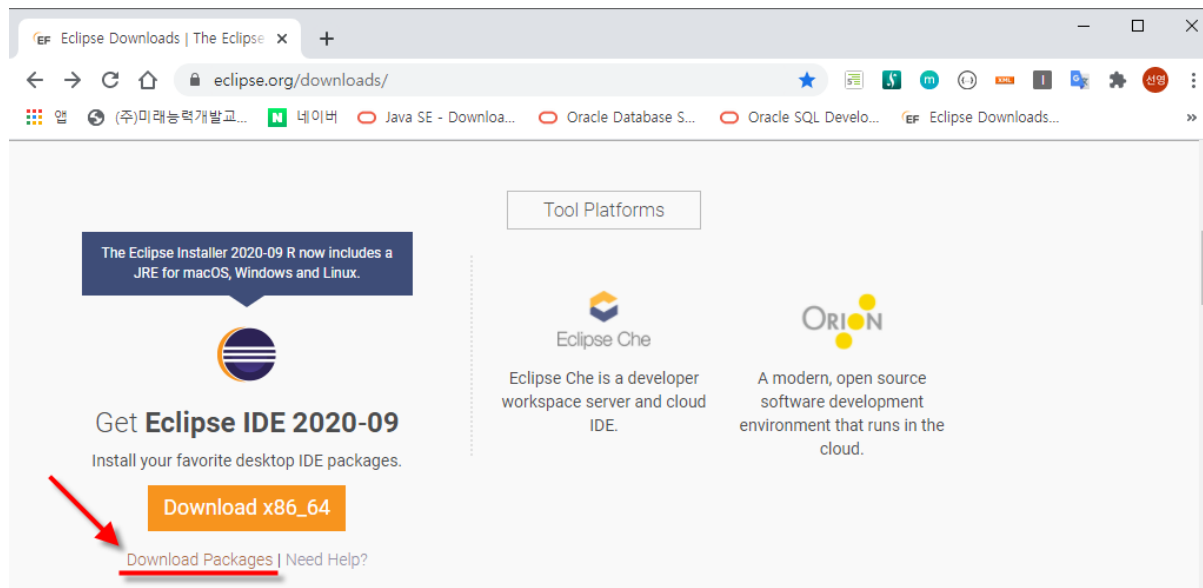
# 자바 통합 개발 환경-이클립스(Eclipse)

## IDE(Integrated Development Environment )란?

- 통합 개발 환경
- 편집, 컴파일, 디버깅을 한번에 할 수 있는 통합된 개발 환경

## 이클립스(Eclipse)

- 자바 응용 프로그램 개발을 위한 통합 개발 환경
- IBM에 의해 개발된 오픈 소스 프로젝트
- <http://www.eclipse.org/downloads/> 에서 다운로드

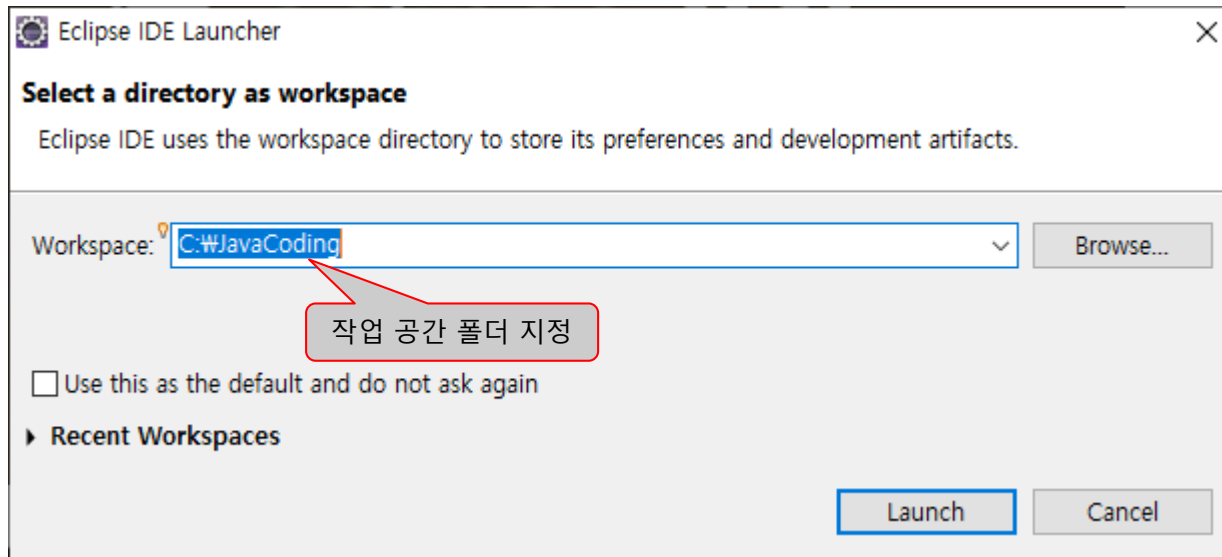


# 이클립스 실행



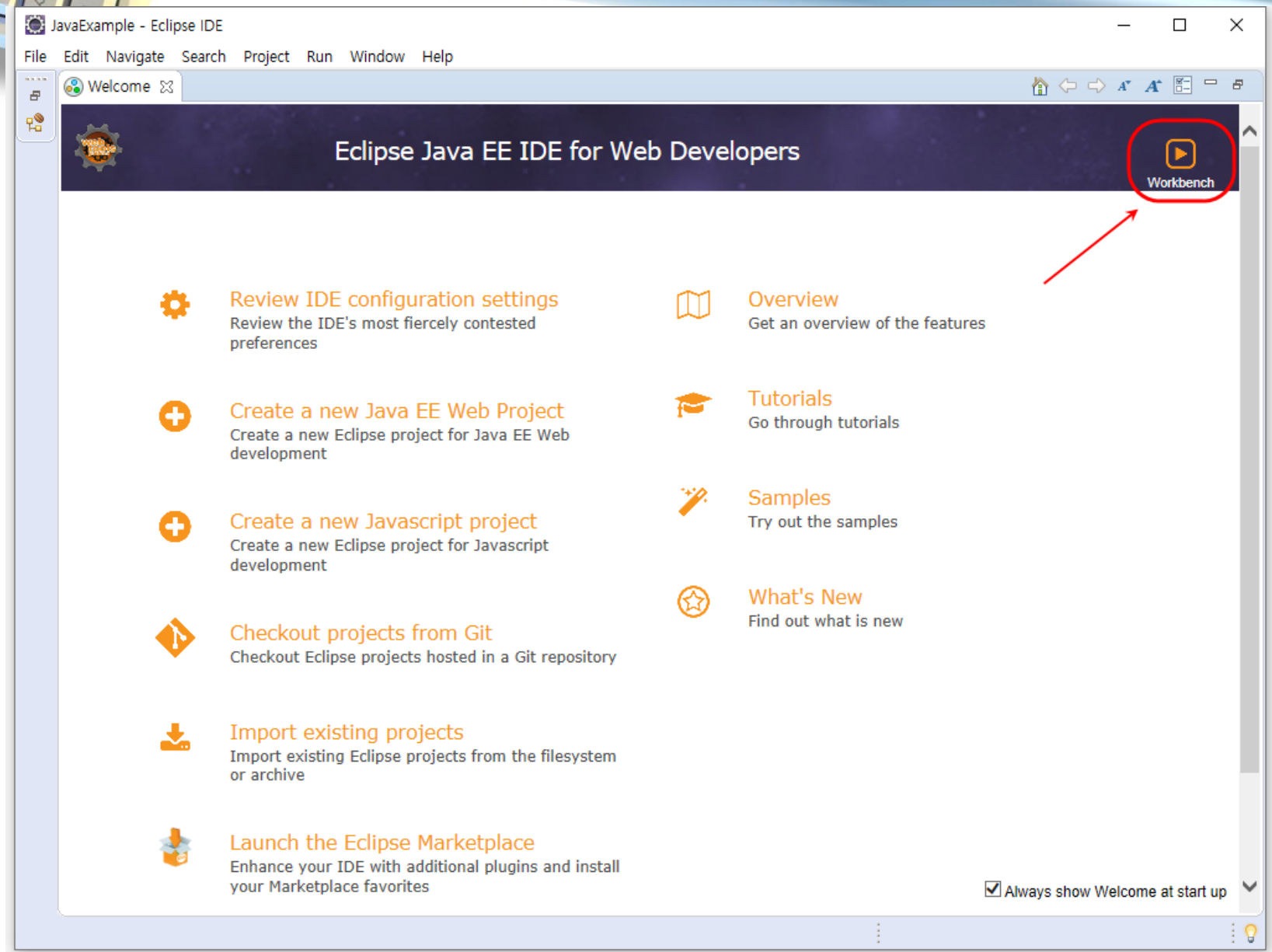
이클립스 시작 화면

작업 공간을 물어본다 → C:\JavaCoding으로 변경한다.

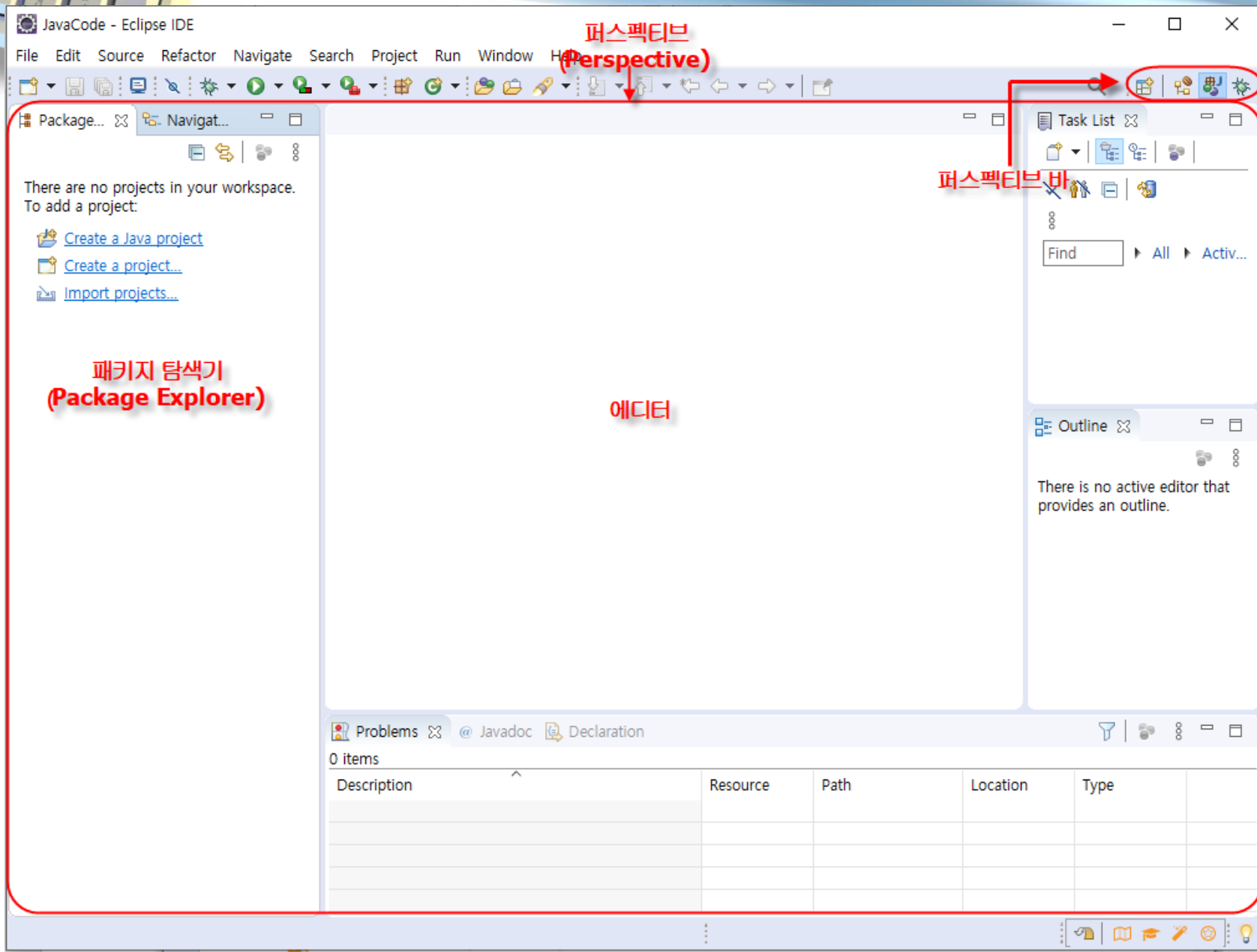


작업 공간 폴더 지정

# 이클립스이 첫화면



# 이클립스의 사용

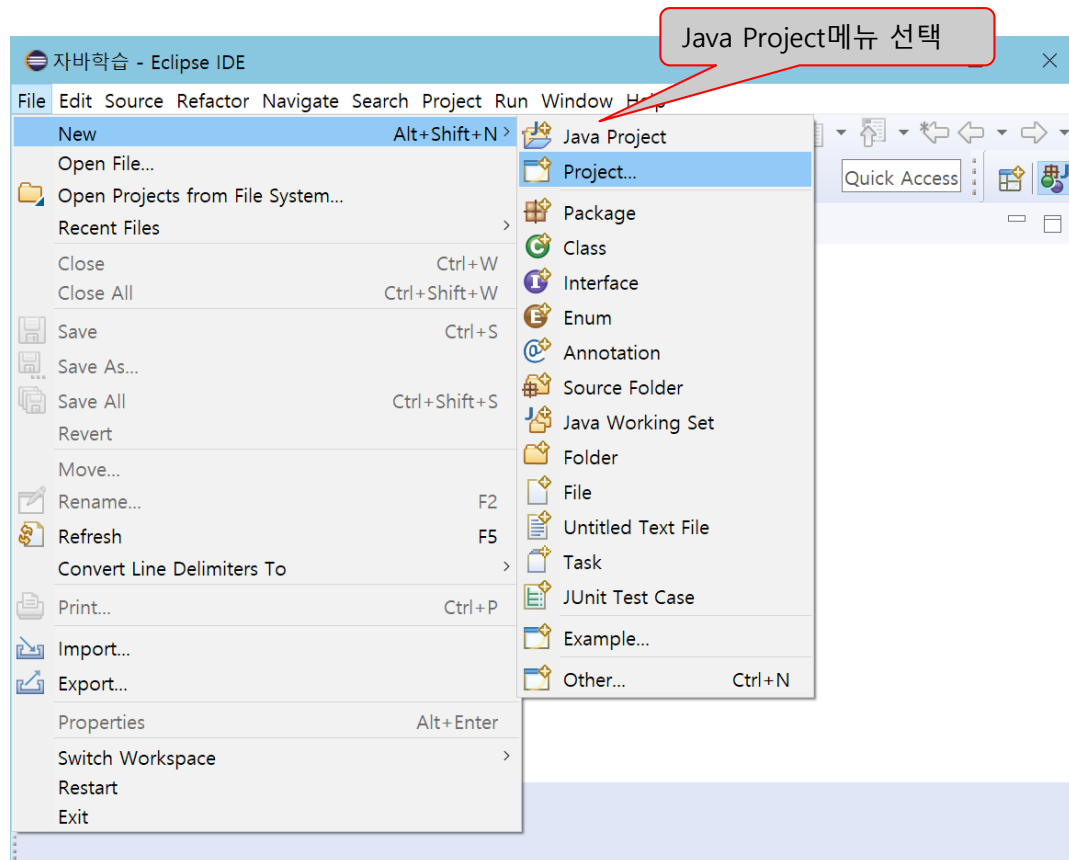




# 간단한 프로그램의 작성

1. 프로젝트의 생성
2. 클래스 생성
3. 소스 코드 입력
4. 실행

컴파일은 자동  
으로 됩니다.



# 프로젝트의 생성

## File->New-> Java Project 메뉴 사용

New Java Project

Create a Java Project

프로젝트명 입력

Project name:

☒ Use default location

Location:  [Browse...](#)

JRE

☐ Use an execution environment JRE:

☒ Use a project specific JRE:  [Configure JREs...](#)

☐ Use default JRE 'jdk-15' and workspace compiler preferences

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets [New...](#)

Working sets:  [Select...](#)

[? < Back](#) [Next >](#) [Finish](#) [Cancel](#)

New Java Project

Java Settings

Define the Java build settings.

Source Projects Libraries Order and Export Module Dependencies

☒ Java\_project\_day1

> ☒ src

Details

[Create new source folder](#): use this if you want to add a new source folder to your project.

[Link additional source](#): use this if you have a folder in the file system that should be used as additional source folder.

[Add project 'Java project day1' to build path](#): Add the project to the build path if the project is the root of packages and source files. Entries on the build path are visible to the compiler and used for building.

☐ Allow output folders for source folders

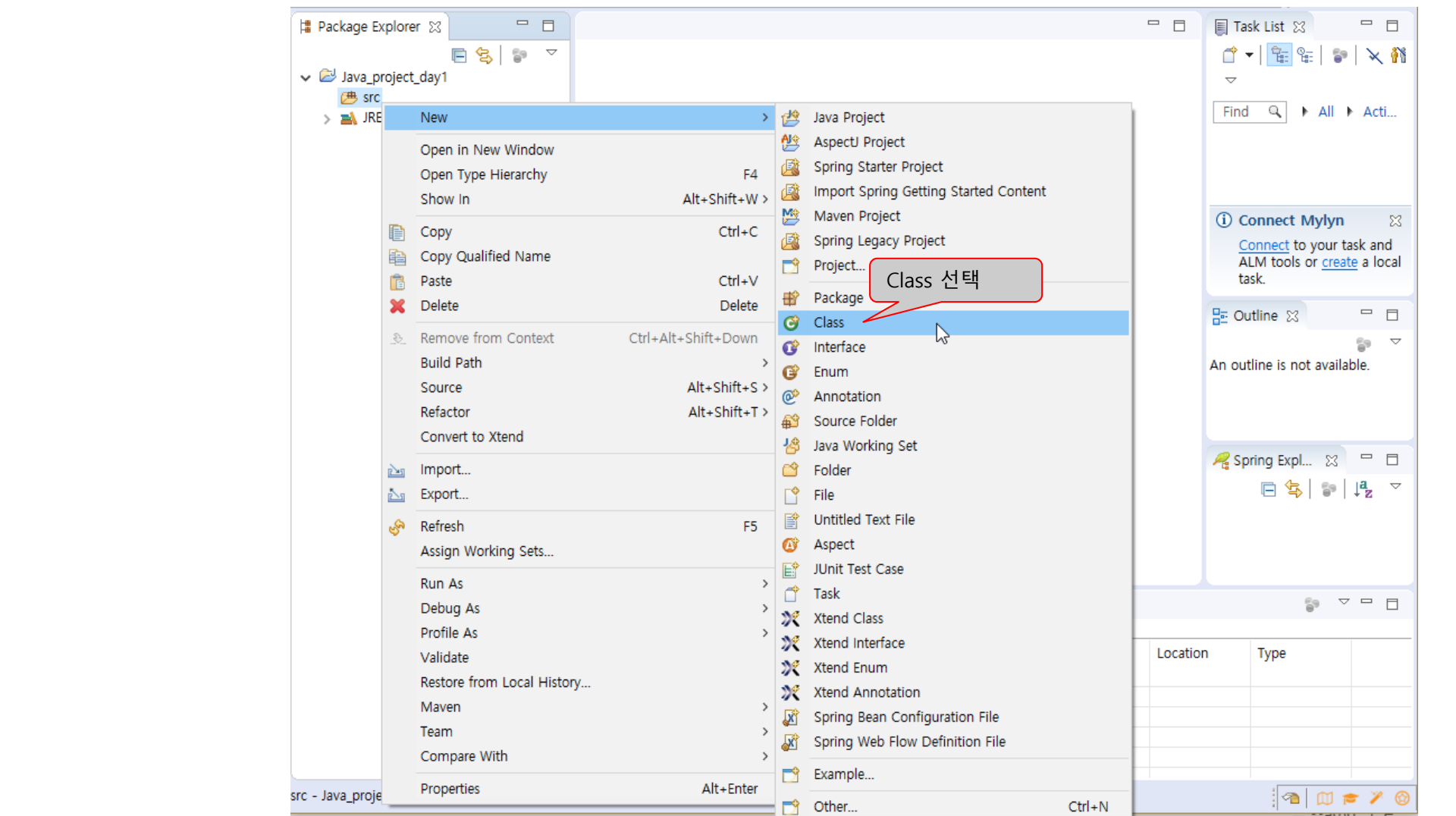
☐ Create module-info.java file

Default output folder:

[Browse...](#)

[? < Back](#) [Next >](#) [Finish](#) [Cancel](#)

# File->New



Location	Type	

# 클래스의 생성

**New Java Class**

Java Class  
Create a new Java class.

Source folder: Java\_project\_01/src Browse...

Package: firstExam Browse...

☐ Enclosing type: Browse...

Name: HelloWorld

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add...

Which method stubs would you like to create?

☐ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

? Finish Cancel

firstExam

HelloWorld

패키지명 입력

클래스명 입력

Finish 선택



# 소스 코드 입력

```
package firstExam;
```

```
public class HelloWorld {
```

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

```
        System.out.println("자바를 처음  
        시작하지만 열심히 하겠습니다.");
```

```
        System.out.println("오늘 하루도 화이팅  
        합시다.");
```

```
    }
```

```
}
```

`System.out.println("~~");` 문장은 한 줄로 작성해 주세요

# 프로그램의 실행

javaTest - Java - Java\_project\_day1/src/firstExam/HelloWorld.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- Java\_project\_day1
  - src
    - firstExam
      - HelloWorld.java
- JRE System Library [JavaSE-1.8]

HelloWorld.java

```
1 package firstExam;
2
3 public class HelloWo
4     public static vo
5         System.out.p
6         System.out.p
7     }
8 }
9
```

파일명

Java Application 클릭

오른쪽 마우스 클릭 -> Run 실행 메뉴

1 Java Application Alt+Shift+X, J

Run Configurations...

Run As

Undo Ctrl+Z

Revert File

Save Ctrl+S

Open Declaration F3

Open Type Hierarchy F4

Open Call Hierarchy Ctrl+Alt+H

Show in Breadcrumb Alt+Shift+B

Quick Outline Ctrl+O

Quick Type Hierarchy Ctrl+T

Open With

Show In Alt+Shift+W

Cut Ctrl+X

Copy Ctrl+C

Copy Qualified Name

Paste Ctrl+V

Quick Fix Ctrl+1

Source Alt+Shift+S

Refactor Alt+Shift+T

Local History

References

Declarations

Add to Snippets...

AspectJ Refactoring

Debug As

Profile As

Validate

Team

Compare With

Replace With

Preferences...

Remove from Context Ctrl+Alt+Shift+Down



# 프로그램의 실행

실행 버튼

Run -> Run 실행 메뉴

File Edit Source Refactor Navigate Search Project Run Window Help



## Run->Run As->Java Application 메뉴 사용

콘솔 윈도우

Problems Javadoc Declaration Search Console Data Source Explorer SQL Results

<terminated> HelloWorld (1) [Java Application] C:\Java\jdk1.8\bin\javaw.exe (2017. 4. 10. 오전 5:45:14)

자바를 처음 시작하지만 열심히 하겠습니다.  
오늘 하루도 화이팅 합시다.

실행 결과





# 컴파일 오류

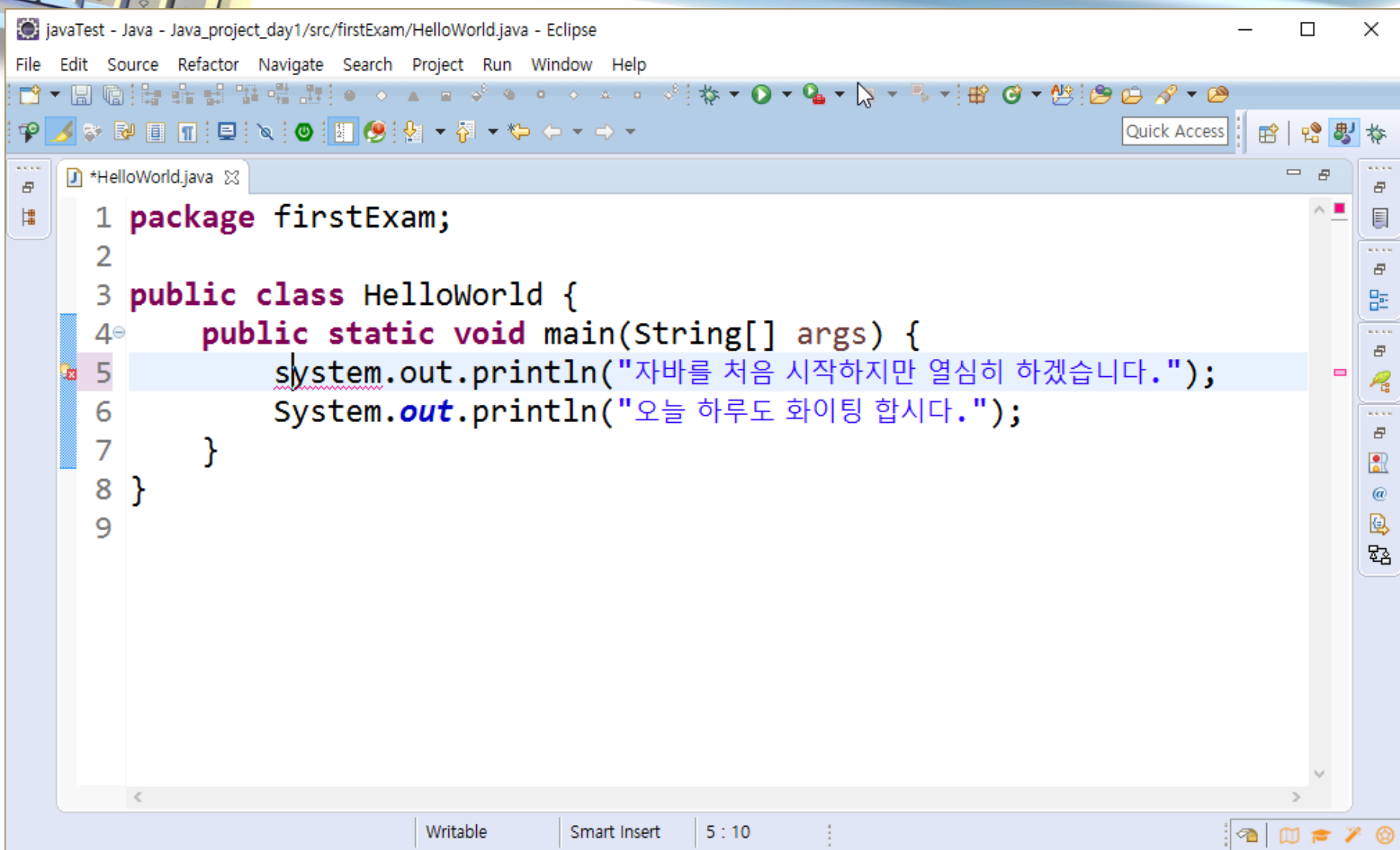
소스를 입력하는 과정에서 다음과 같이 System으로 하여야 할 것을 system으로 잘못 입력하였다고 가정.

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

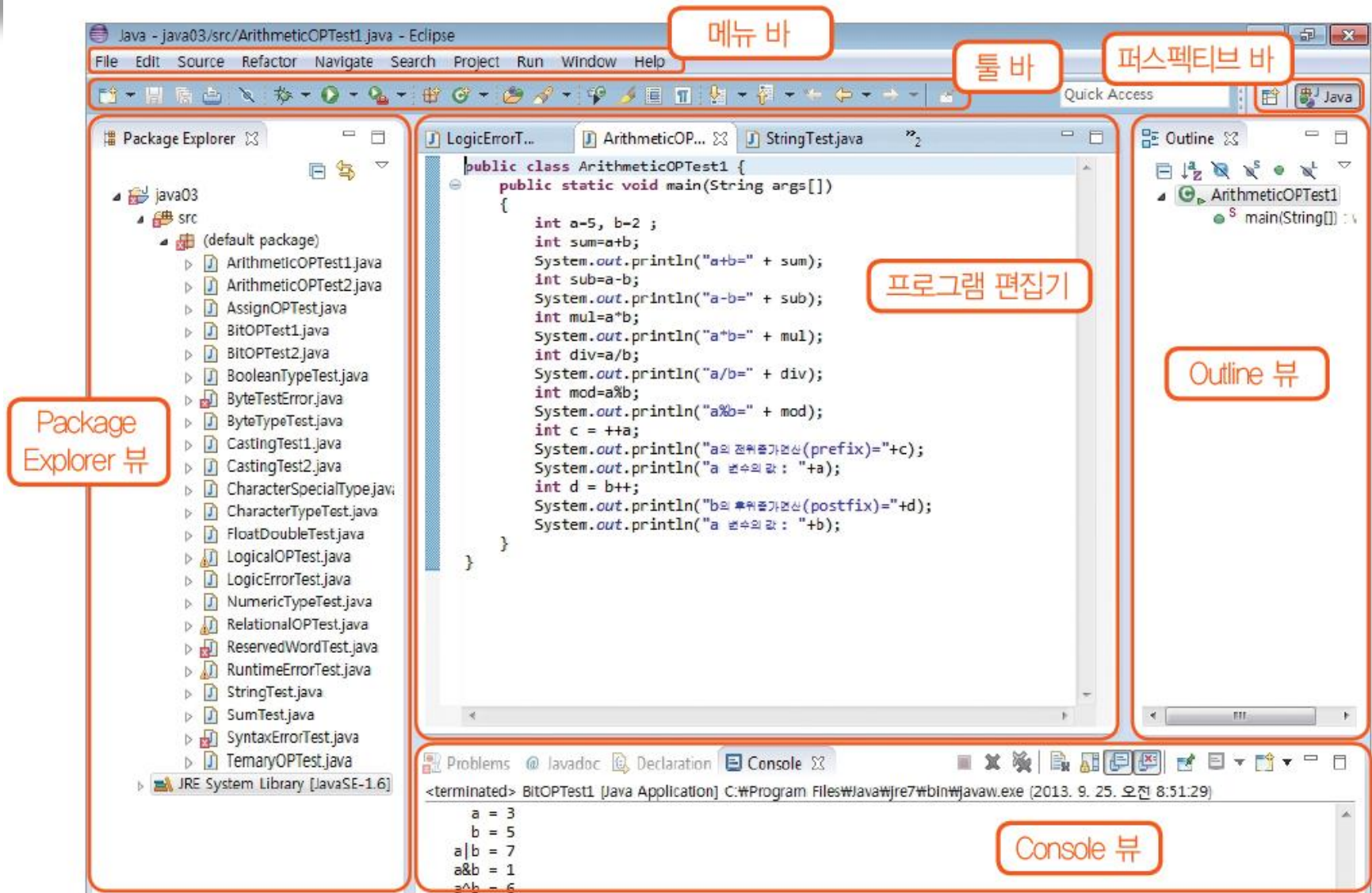
System이 system을 잘못 표기되었다.

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

# 컴파일 오류



# 이클립스의 기본 구조





# 이클립스의 기본 구조 설명

## 메뉴 바

- 메뉴를 통하여 이클립스의 모든 기능들을 사용할 수 있습니다.

## 툴 바

- 자주 쓰이는 기능들을 편리하게 사용할 수 있도록 버튼으로 제공하는 바입니다. 다양한 형태의 버튼으로 제공되고 있습니다.

## Package Explorer 뷰

- 프로젝트를 중심으로 패키지과 클래스 파일, 라이브러리를 관리하는 뷰입니다. 프로젝트에 소속된 클래스와 패키지, 라이브러리 등을 볼 수 있는 창입니다.

## 프로그램 편집기

- 자바 프로그램을 작성할 수 있는 편집기 창입니다. 자바 프로그램의 편집기는 자동 완성 기능을 포함한 다양한 기능들이 제공되고 있습니다.





## Outline 뷰

- 현재 편집되고 있는 프로그램 코드의 개요와 트리 구조를 나타냅니다.

## Console 뷰

- 프로그램의 실행 결과나 프로그램에 오류가 있을 경우 오류를 나타내는 창입니다.

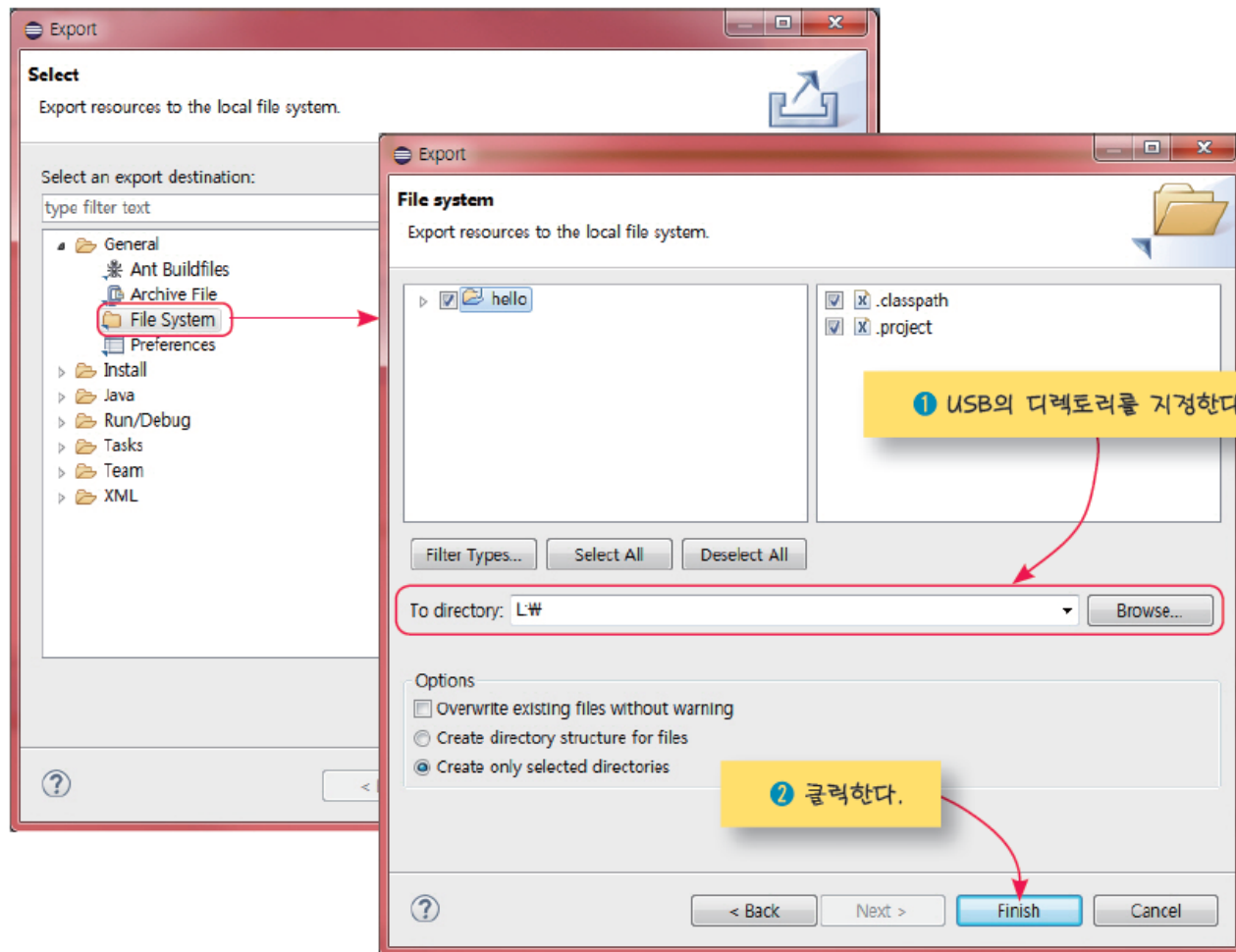
## 퍼스펙티브 바

- 퍼스펙티브는 이클립스를 사용할 때의 뷰나 창의 배열 형태를 의미합니다. 이클립스에서는 다양한 형태의 퍼스펙티브를 제공하고 있고[그림 2-13], 사용자가 자신만의 퍼스펙티브를 만들어 저장한 다음 사용할 수도 있습니다.



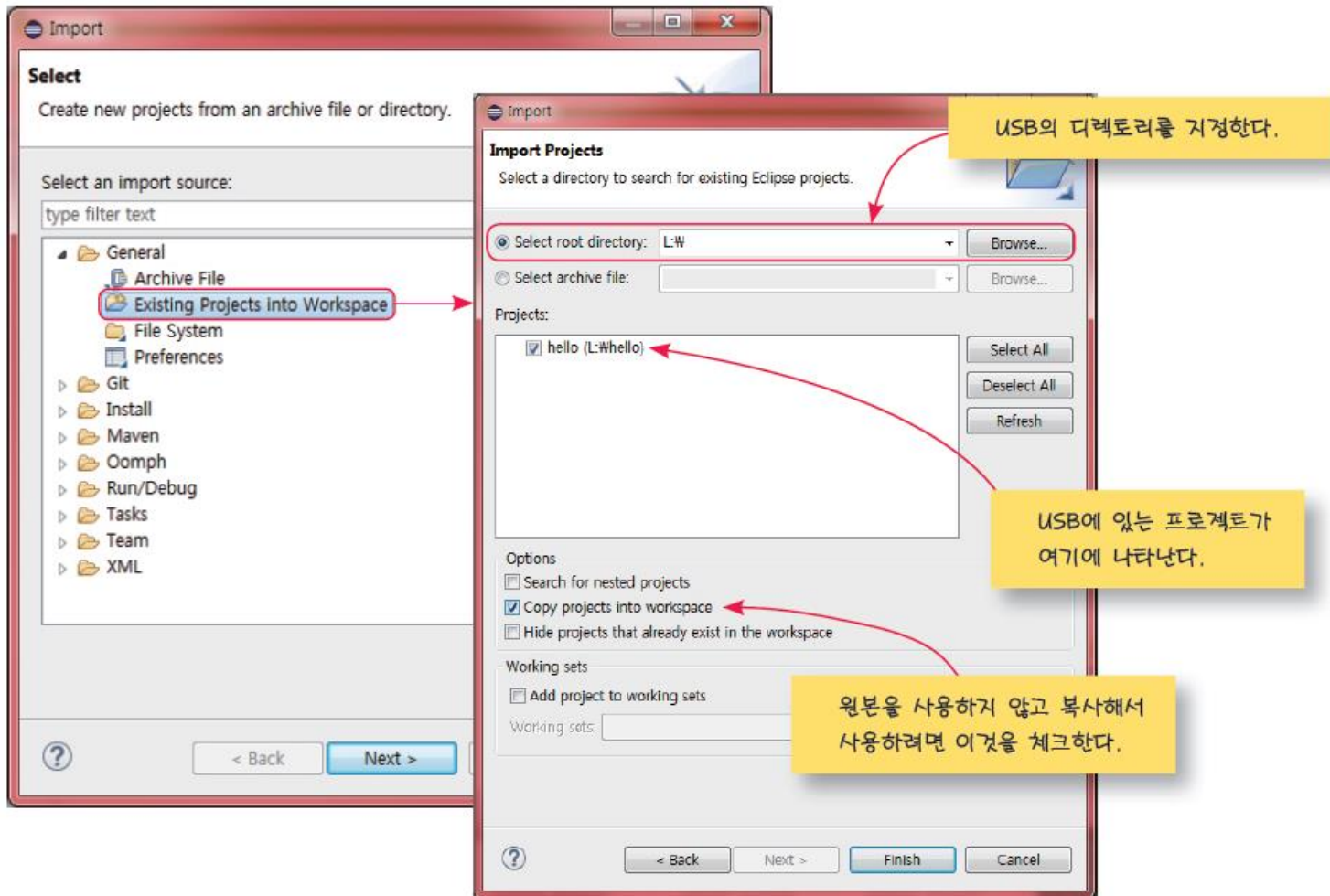
# 이클립스 프로젝트 내보내기

앞에서 작성한 hello 프로젝트를 USB에 저장하려면 hello 프로젝트를 선택한 상태에서 [File] -> [Export] 메뉴를 선택한다.



# 이클립스 프로젝트 읽기

[File] -> [Import] 메뉴를 선택한다



# 자바 API 참조하기

## Java SE 최근 버전을 선택

Overview [Java Platform: X]

docs.oracle.com/javase/8/docs/api/

Overview [Java Platform: X]

Overview PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

### Java™ Platform, Standard Edition 8 API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

See: Description 패키지 창

#### Profiles

- compact1
- compact2
- compact3

#### Packages

Package	Description
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.

#### All Classes

클래스 창

- AbstractAction
- AbstractAnnotationValueVisitor6
- AbstractAnnotationValueVisitor7
- AbstractAnnotationValueVisitor8
- AbstractBorder
- AbstractButton
- AbstractCellEditor
- AbstractChronology
- AbstractCollection
- AbstractColorChooserPanel
- AbstractDocument
- AbstractDocument.AttributeContext
- AbstractDocument.Content
- AbstractDocument.ElementEdit
- AbstractElementVisitor6
- AbstractElementVisitor7
- AbstractElementVisitor8
- AbstractExecutorService
- AbstractIncorruptibleChannel
- AbstractLayoutCache
- AbstractLayoutCache.NodeDimension
- AbstractList
- AbstractListModel
- AbstractMap
- AbstractMap.SimpleEntry
- AbstractMap.SimpleImmutableEntry
- AbstractMarshallerImpl
- AbstractMethodError
- AbstractObservableSynchronizer
- AbstractPreferences
- AbstractProcessor
- AbstractQueue
- AbstractQueue.LongSynchronizer
- AbstractQueue.Synchronizer
- AbstractRegionPainter
- AbstractRegionPainter.PaintContext
- AbstractRegionPainter.PaintContext2D

# 자바 API 참조하기

원하는 클래스를 선택

Object (Java Platform Standard Ed. 8)

Overview PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

compact1, compact2, compact3

java.lang

**Class Object**

java.lang.Object

public class Object

Class Object is the root of the class hierarchy. Every class has Object as a superclass. All objects, including arrays, implement the methods of this class.

Since:  
JDK1.0

See Also:  
Class

**Constructor Summary**

Constructors

Constructor and Description

Object()

**Method Summary**

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
protected Object	<b>clone()</b> Creates and returns a copy of this object.
boolean	<b>equals(Object obj)</b> Indicates whether some other object is "equal to" this one.
protected void	<b>finalize()</b> Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.





실패하더라도 무언가 배우는 것이  
아무것도 시도하지 않는 것보다 낫다.

You are better off trying something and having it not work  
and learning from that than not doing anything at all.



**마크 저커버그**  
**Mark Zuckerberg, 페이스북 설립자**