

자바 시작하기

자바 개발환경

자바 개발환경

자바 개발환경 설치 순서

- 1 - **STEP1.** JDK(Java Development Kit) [[오라클](#)]
- 2 - **STEP2.** 이클립스(Eclipse) 설치 [[이클립스](#)]

자바 개발환경

2

☞ 자바 개발환경 설치 순서

- **STEP1.** JDK(Java Development Kit) [[오라클](#)]

1

Java SE 8

Java SE 8u261 is the latest release for the Java SE 8 Platform.

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

Oracle JDK

- [JDK Download](#)
- [Server JRE Download](#)
- [JRE Download](#)
- [Documentation Download](#)
- [Demos and Samples Download](#)

Java SE Development Kit 8u261

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

| Product / File Description | File Size | Download |
|-------------------------------------|-----------|---|
| Linux ARM 32 Hard Float ABI | 73.4 MB | jdk-8u261-linux-arm32-vfp-hflt.tar.gz |
| Linux ARM 64 Hard Float ABI | 70.3 MB | jdk-8u261-linux-arm64-vfp-hflt.tar.gz |
| Linux x86 RPM Package | 121.92 MB | jdk-8u261-linux-i586.rpm |
| Linux x86 Compressed Archive | 136.81 MB | jdk-8u261-linux-i586.tar.gz |
| Linux x64 RPM Package | 121.53 MB | jdk-8u261-linux-x64.rpm |
| Linux x64 Compressed Archive | 136.48 MB | jdk-8u261-linux-x64.tar.gz |
| macOS x64 | 203.94 MB | jdk-8u261-macosx-x64.dmg |
| Solaris SPARC 64-bit (SVR4 package) | 125.77 MB | jdk-8u261-solaris-sparcv9.tar.Z |
| Solaris SPARC 64-bit | 88.72 MB | jdk-8u261-solaris-sparcv9.tar.gz |
| Solaris x64 (SVR4 package) | 134.23 MB | jdk-8u261-solaris-x64.tar.Z |
| Solaris x64 | 92.47 MB | jdk-8u261-solaris-x64.tar.gz |
| Windows x86 | 154.52 MB | jdk-8u261-windows-i586.exe |
| Windows x64 | 166.28 MB | jdk-8u261-windows-x64.exe |

자바 개발환경

☞ 자바 개발환경 설치 순서

- **STEP1.** JDK(Java Development Kit) [[오라클](#)]

1

Java SE 11 (LTS)

Java SE 11.0.8 is the latest release for the Java SE 11 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

2

Java SE Development Kit 11.0.8

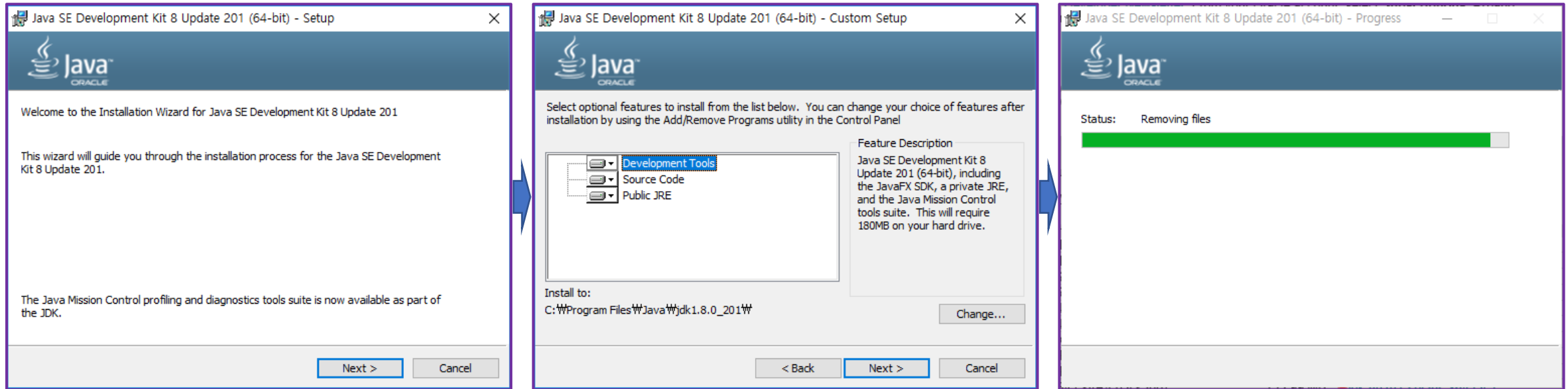
This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE

| Product / File Description | File Size | Download |
|----------------------------------|-----------|---------------------------------------|
| Linux Debian Package | 148.77 MB | jdk-11.0.8_linux-x64_bin.deb |
| Linux RPM Package | 155.45 MB | jdk-11.0.8_linux-x64_bin.rpm |
| Linux Compressed Archive | 172.66 MB | jdk-11.0.8_linux-x64_bin.tar.gz |
| macOS Installer | 166.84 MB | jdk-11.0.8_osx-x64_bin.dmg |
| macOS Compressed Archive | 167.23 MB | jdk-11.0.8_osx-x64_bin.tar.gz |
| Solaris SPARC Compressed Archive | 186.49 MB | jdk-11.0.8_solaris-sparcv9_bin.tar.gz |
| Windows x64 Installer | 151.73 MB | jdk-11.0.8_windows-x64_bin.exe |
| Windows x64 Compressed Archive | 171.16 MB | jdk-11.0.8_windows-x64_bin.zip |

자바 개발환경

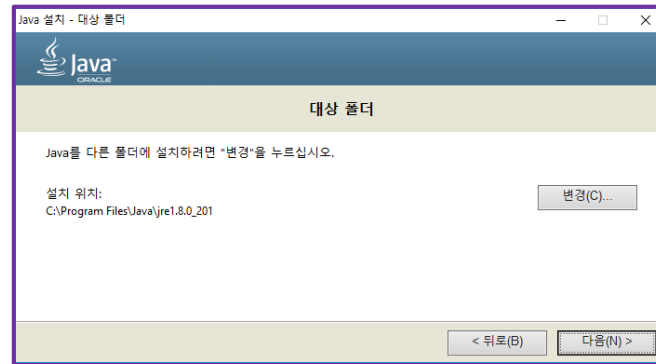
☞ 자바 개발환경 설치 순서 **STEP1**. JDK(Java Development Kit) [[오라클](#)]

1

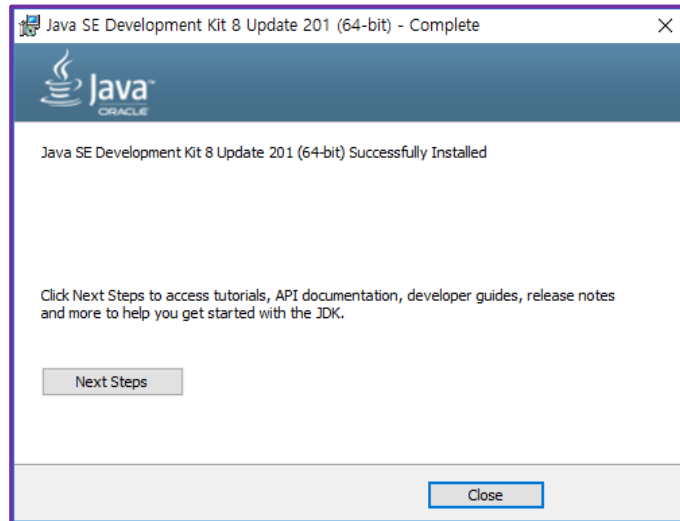


자바 개발환경

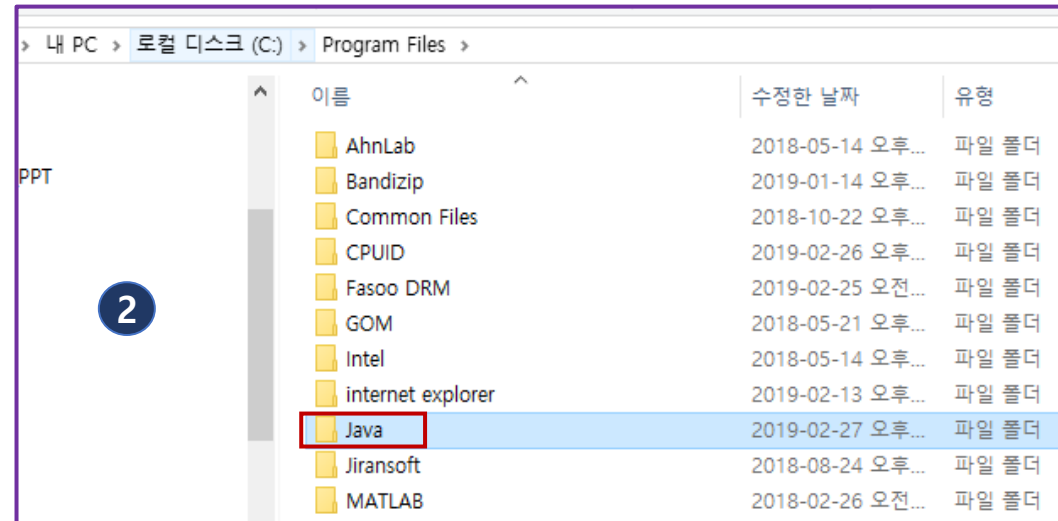
☞ 자바 개발환경 설치 순서 - **STEP1**. JDK(Java Development Kit) [[오라클](#)]



1



2

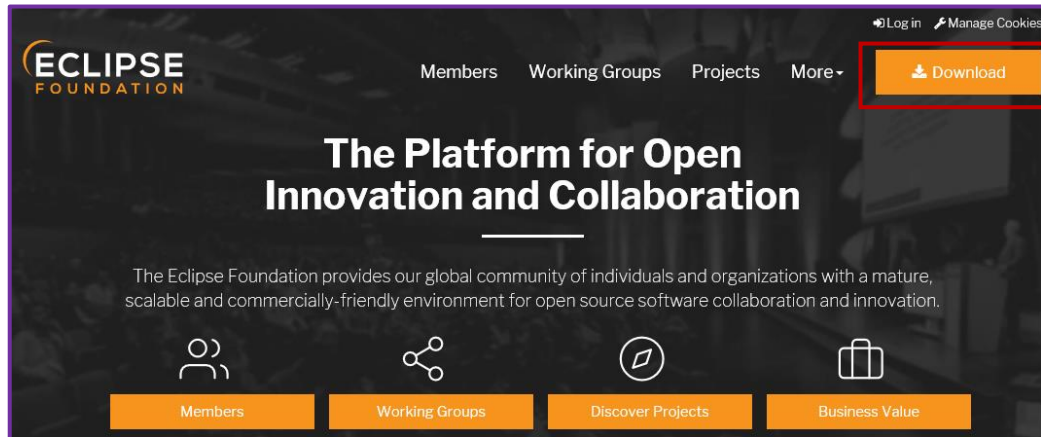


자바 개발환경

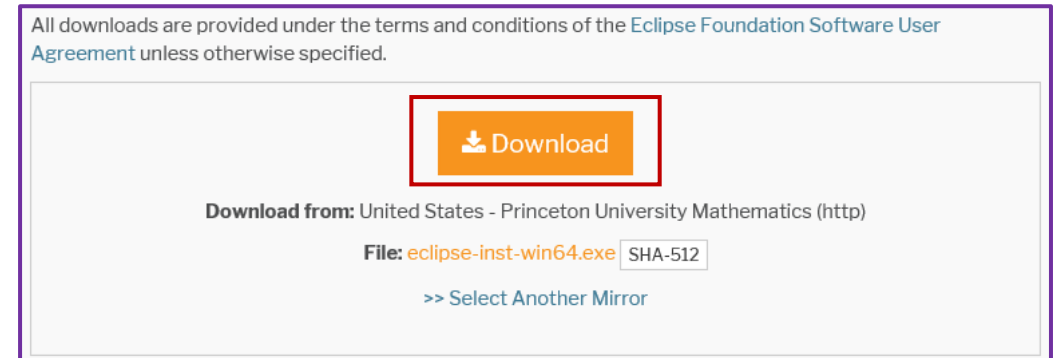
☞ 자바 개발환경 설치 순서

- **STEP2.** 이클립스(Eclipse) 설치 [[이클립스](#)]

1



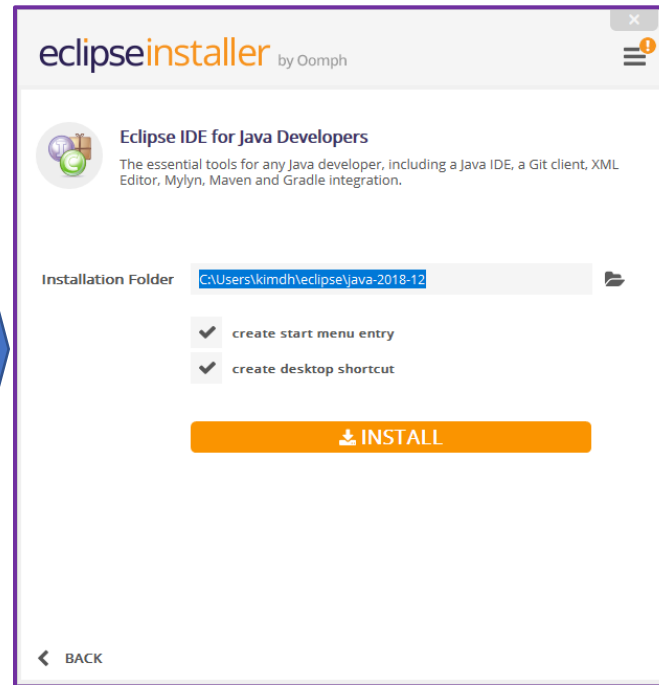
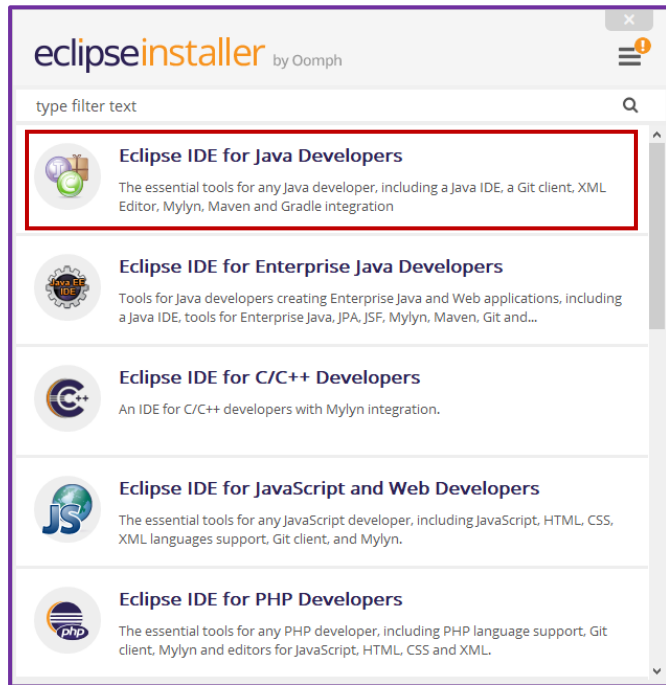
2



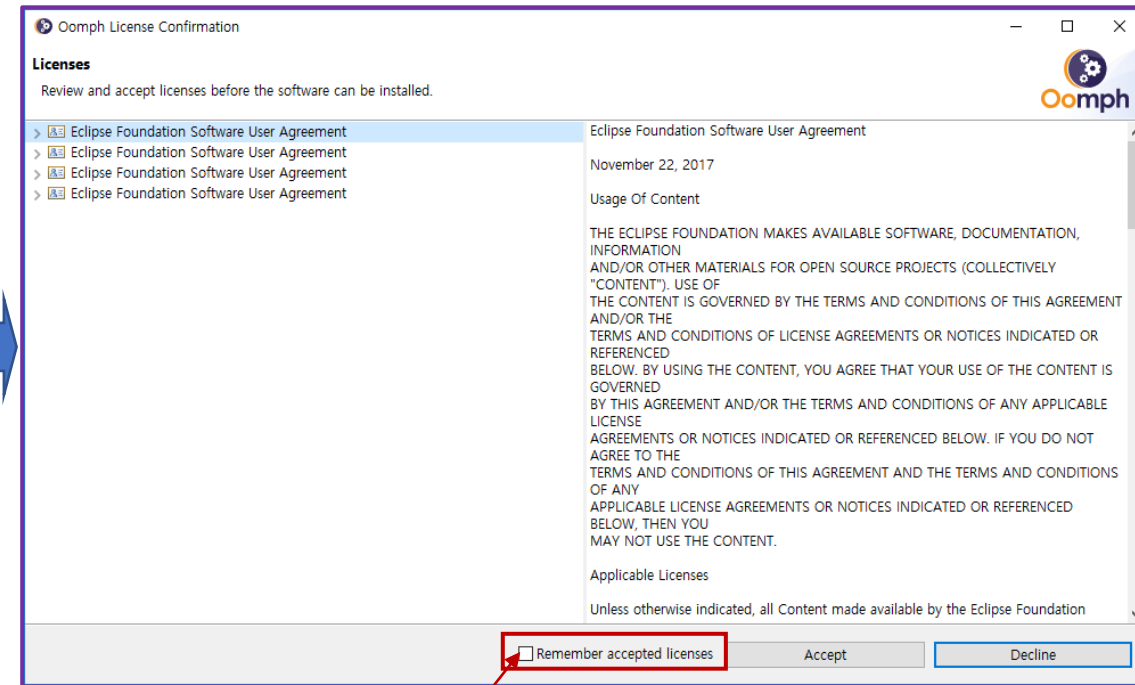
자바 개발환경

☞ 자바 개발환경 설치 순서 - **STEP2.** 이클립스(Eclipse) 설치 [[이클립스](#)]

1



2



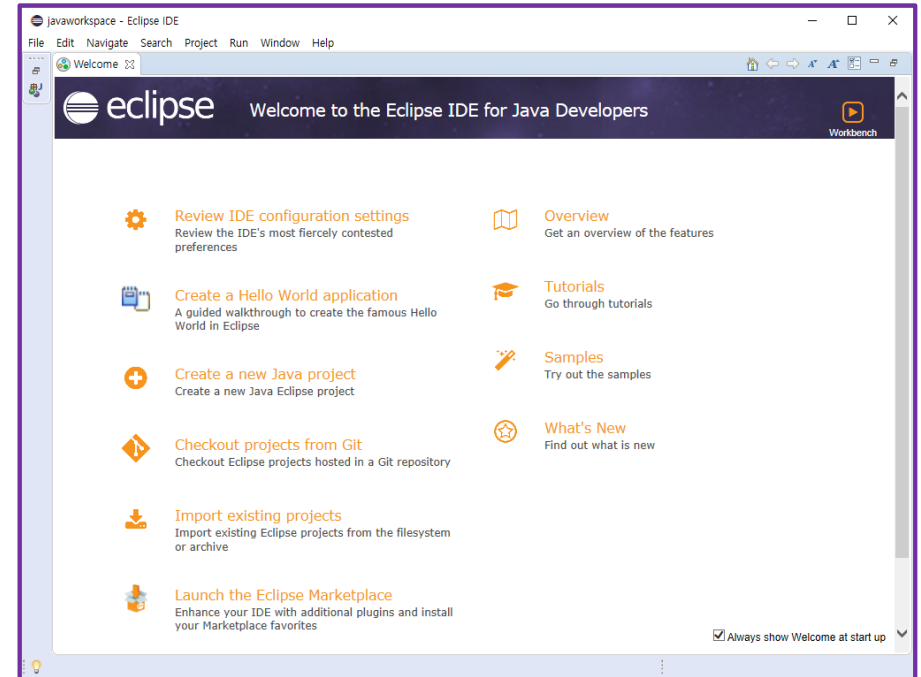
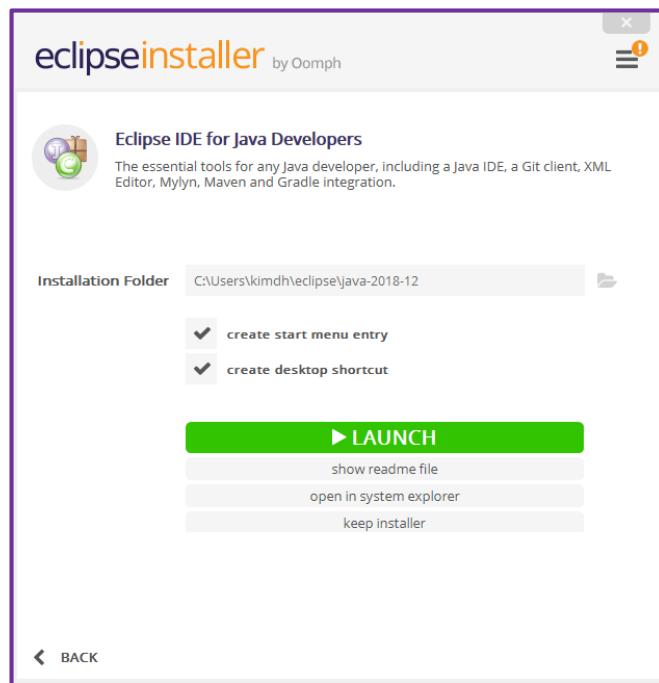
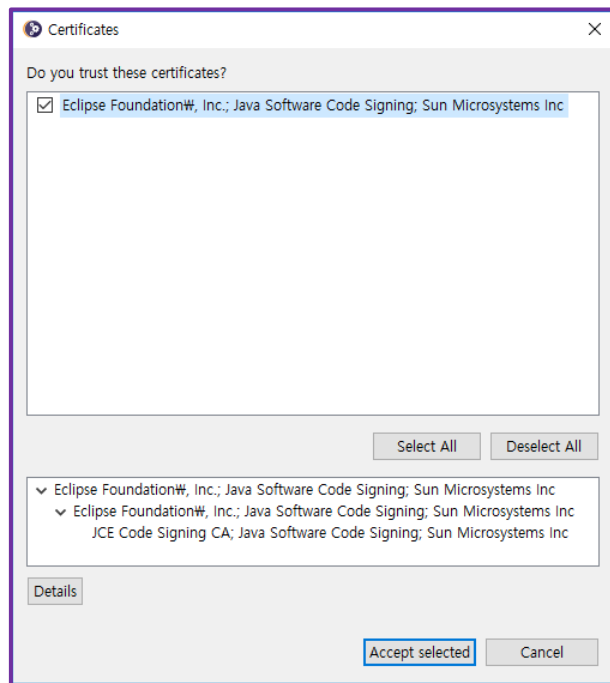
Check

3

자바 개발환경

☞ 자바 개발환경 설치 순서 - **STEP2**. 이클립스(Eclipse) 설치 [[이클립스](#)]

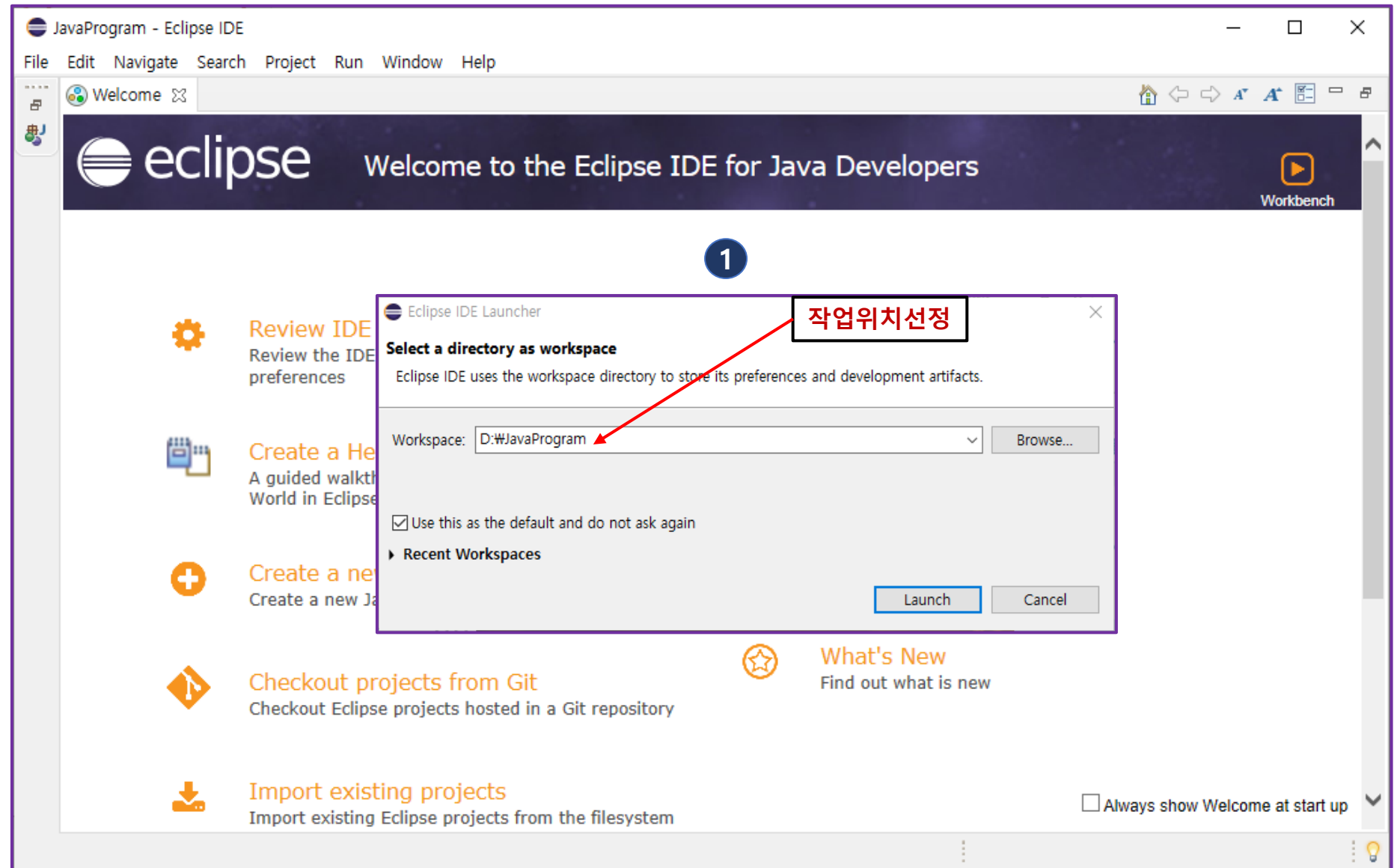
1



이클립스 구성 및 기본설정

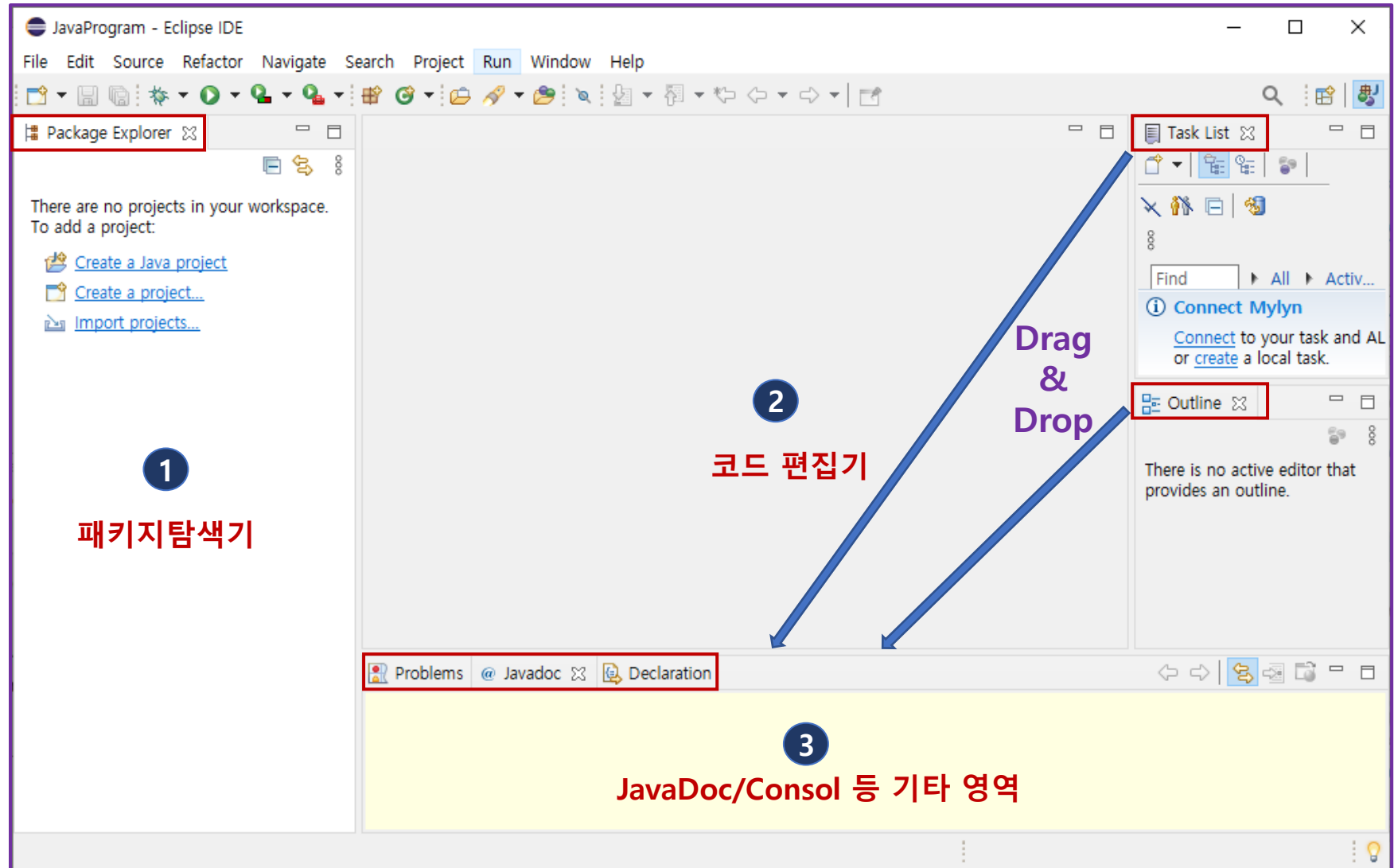
이클립스 구성 및 기본설정

👉 **작업위치** 설정



이클립스 구성 및 기본설정

👉 퍼스펙티브 설정



이클립스 구성 및 기본설정

TIP

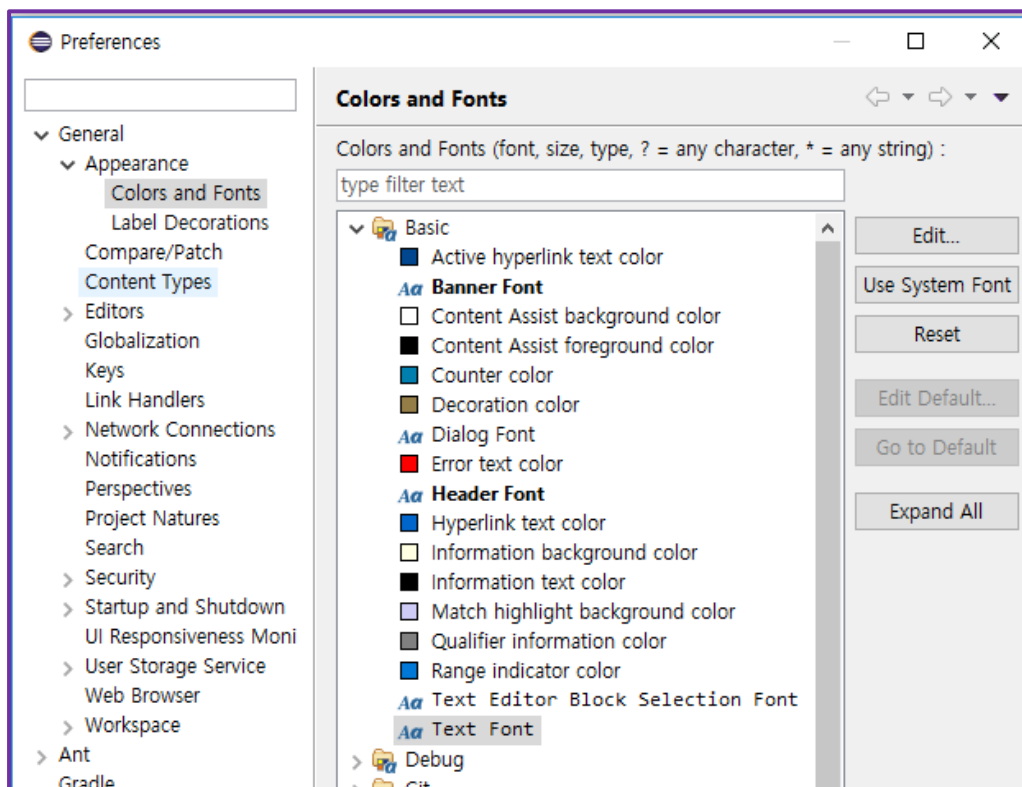
3

폰트 크기만 변경하는 하는 경우
[ctrl + (+)] | [ctrl + (-)]

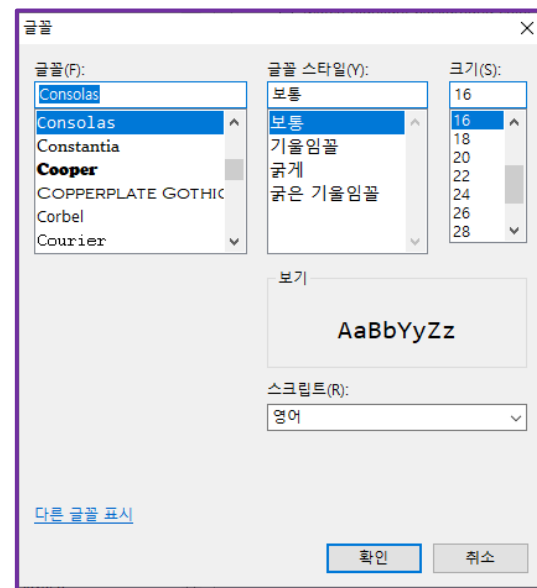
☞ 이클립스 **폰트** 조정

- **폰트 조정** : Window → Preference → General → Appearance → Color and Fonts → Basic → Text Font

1



2



이클립스 구성 및 기본설정

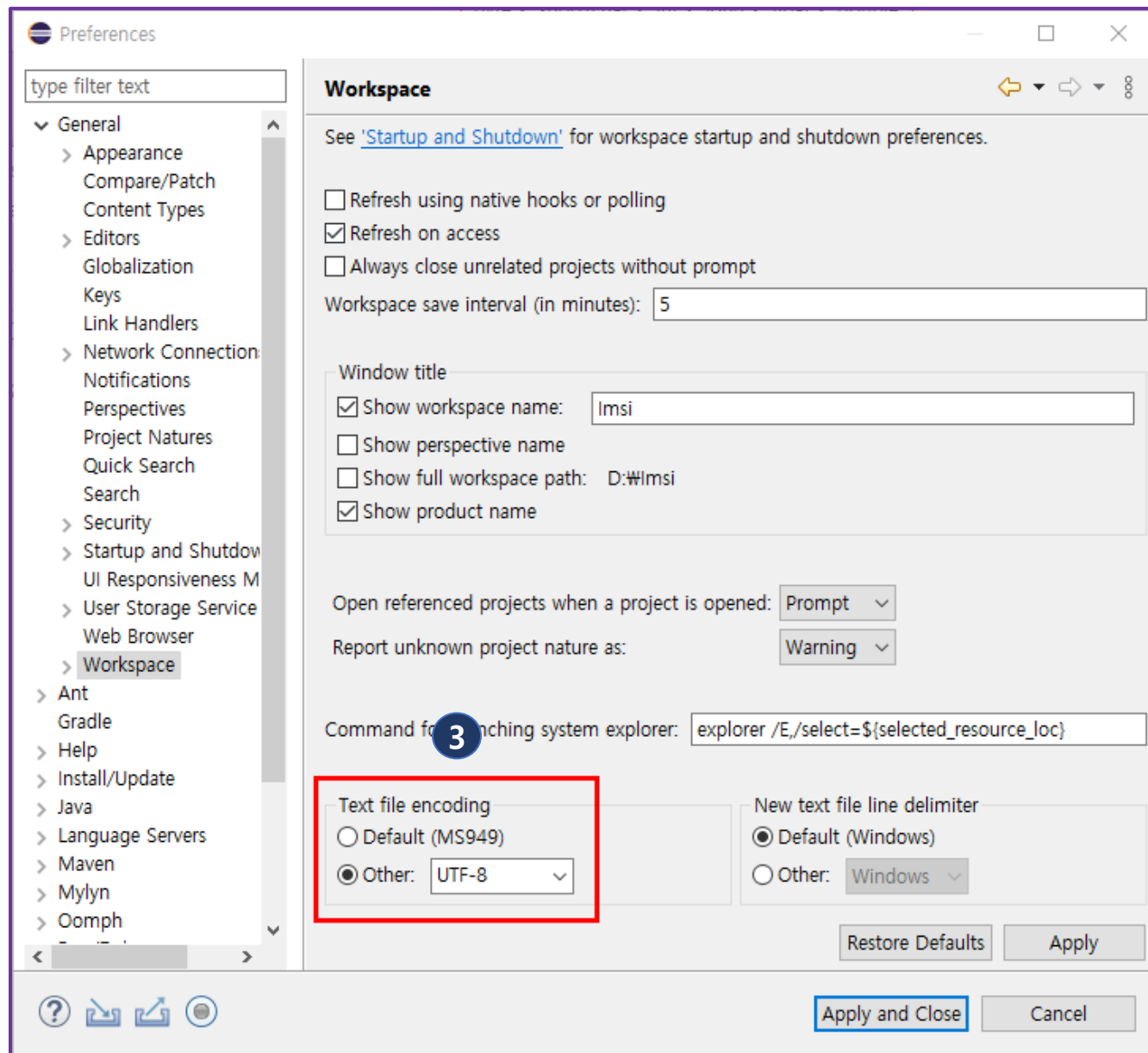
☞ 텍스트 인코딩(encoding) **UTF-8** 설정

1

Workspace 단위

2

Window → Preference →
General → Workspace →
Text file encoding → Other : **UTF-8**



이클립스 구성 및 기본설정

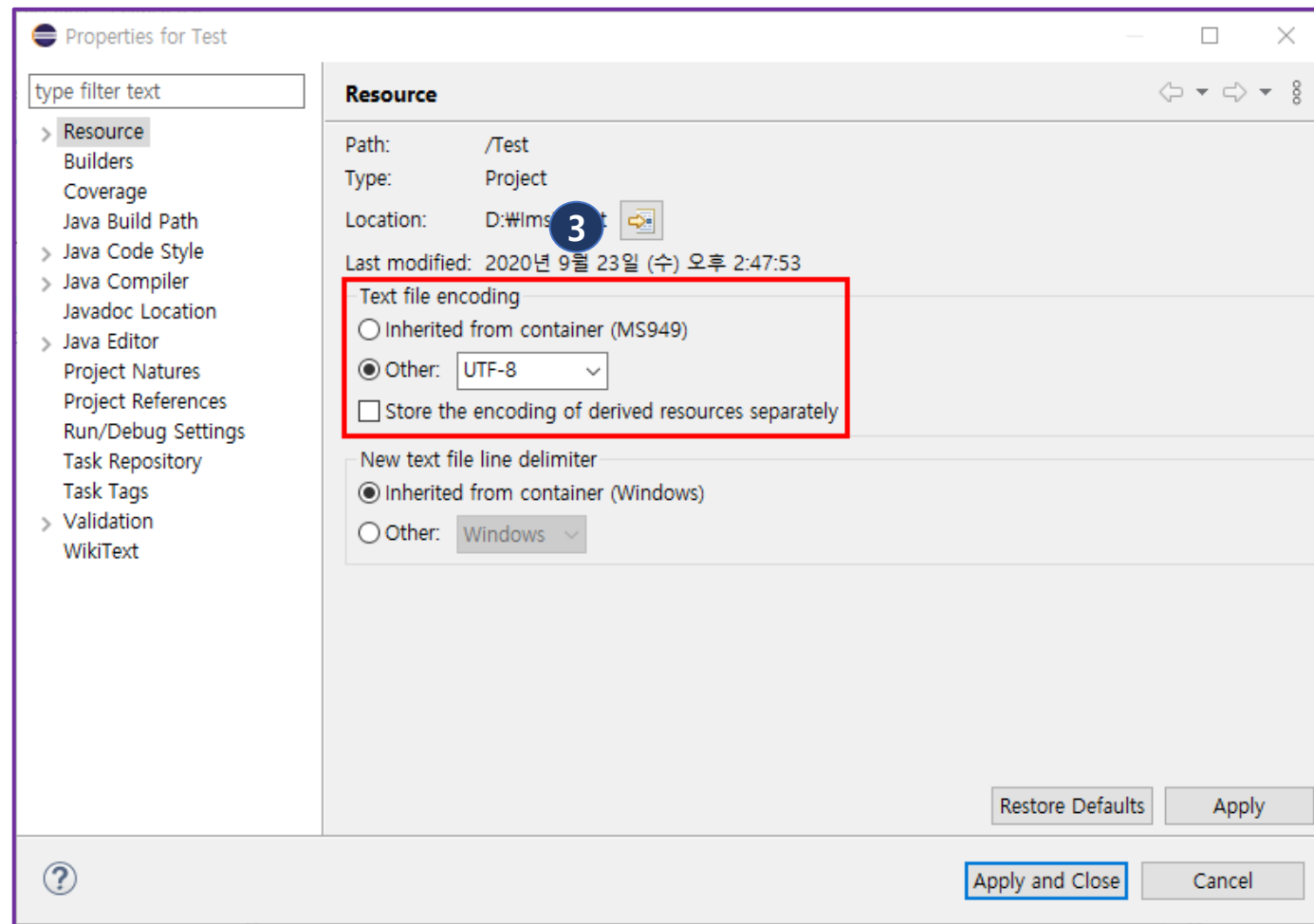
☞ 텍스트 인코딩(encoding) **UTF-8** 설정

1

Project 단위

2

Project 선택 → 마우스 오른쪽 클릭
Properties → Resource →
Text file encoding → Other : **UTF-8**

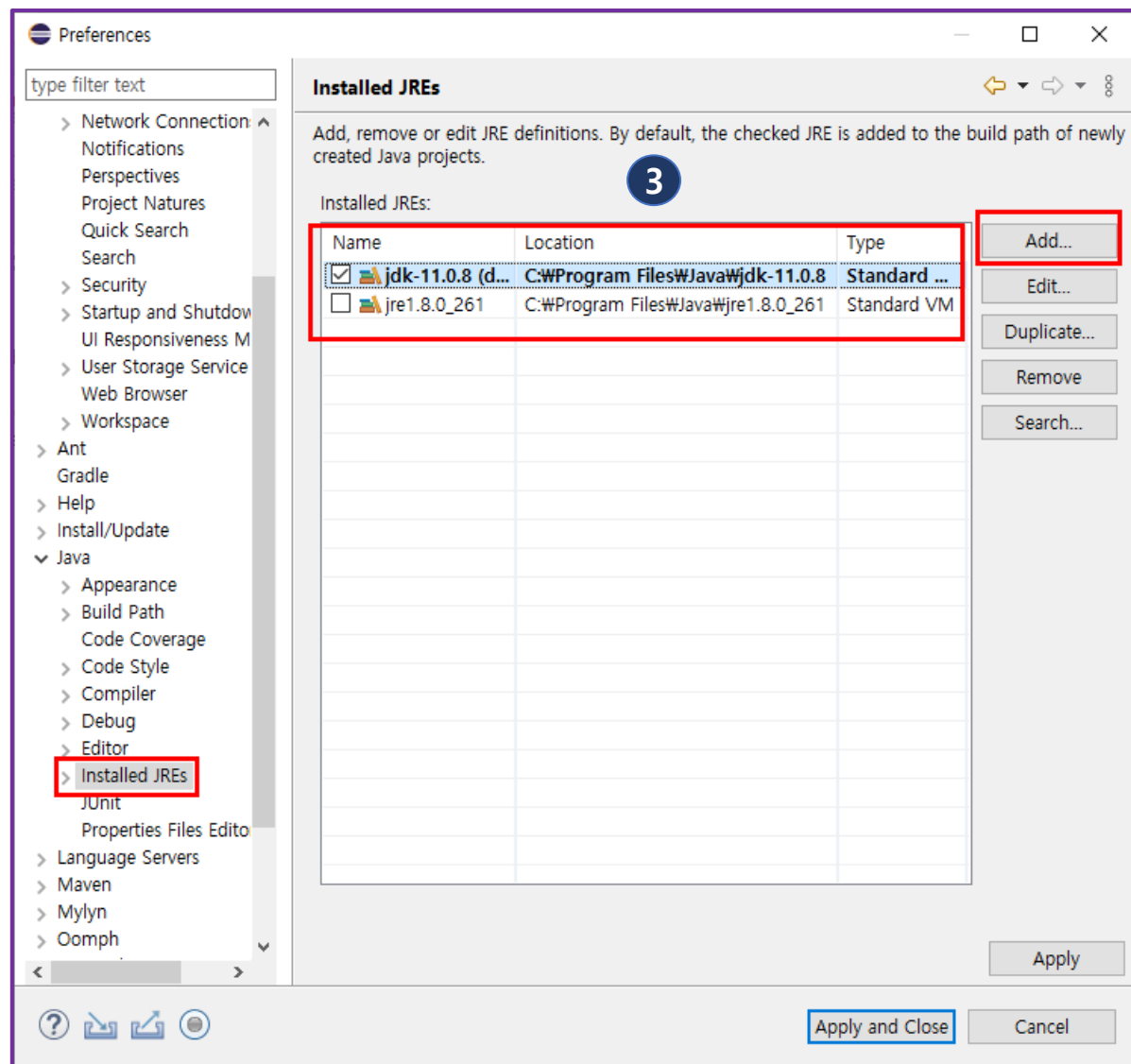


이클립스 구성 및 기본설정

1 JRE(Java Runtime Environment) 버전 설정

2

Window → Preference → Java → Installed JREs



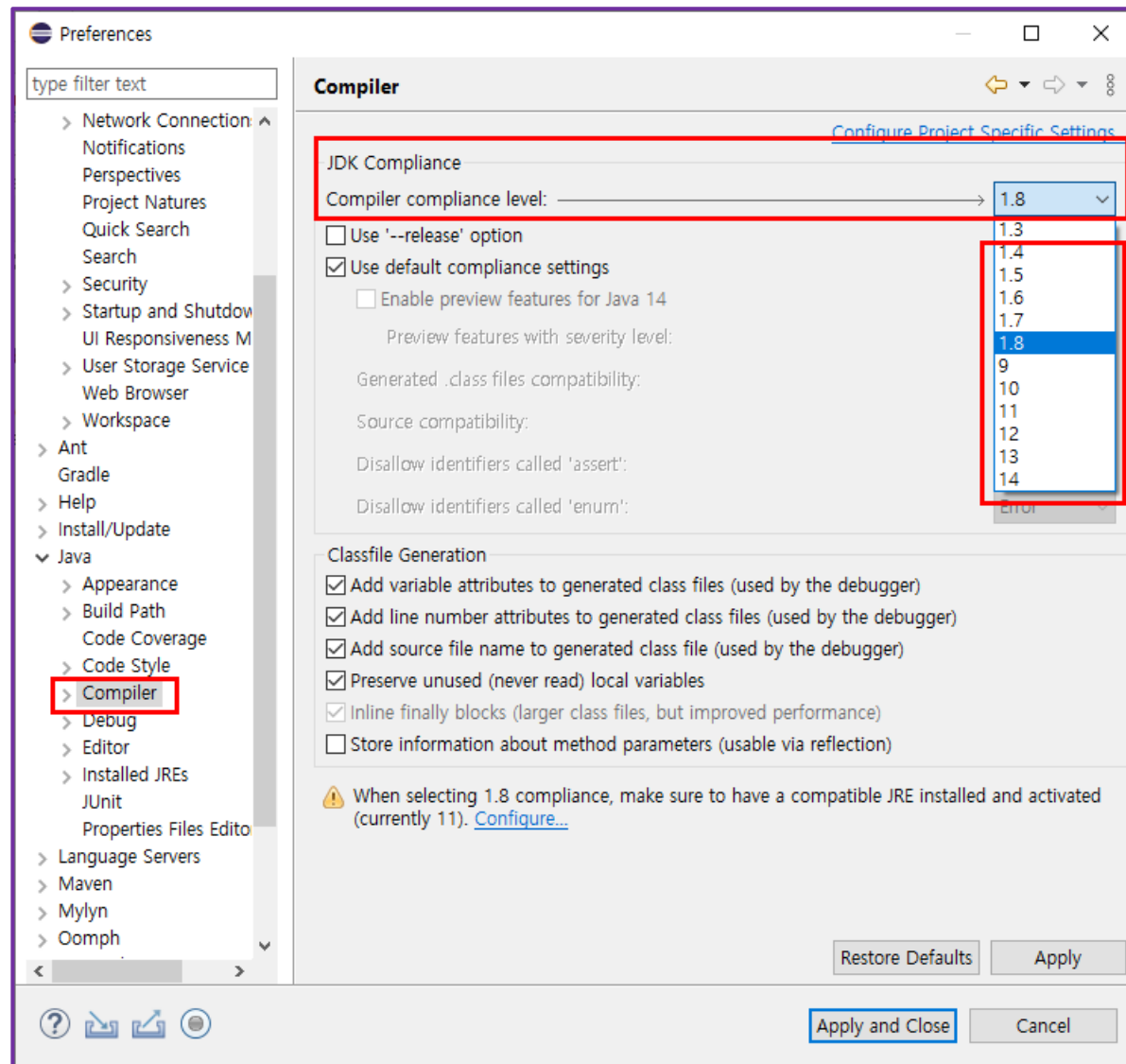
이클립스 구성 및 기본설정

3

1 컴파일러 버전 설정

1

Window → Preference → Java → Compiler



이클립스 구성 및 기본설정

☞ 이클립스 대표적 단축키

1

| | | |
|--------------------|----|---|
| 주요 Function Key | F2 | 패키지/클래스 이름 바꾸기 (rename) |
| | F3 | 자바 API 클래스 및 함수는 class source 연결 필요 (클래스 및 함수 정의로 이동) : API class source 연결방법 참고 |
| | F4 | 클래스 정의 확인 (상속관계 포함) |

| | | |
|----|---|--|
| 기능 | [ctrl + shift + O] | 자동 임포트 |
| 기능 | [ctrl + (/)] [ctrl + (\)] | 한줄주석 설정 및 해제 (동일 방식) |
| 기능 | [ctrl + shift+(/)] [ctrl + shift+(W)] | 블록주석 설정 및 해제 |
| 기능 | (클래스 이름 선택 후) [ctrl+t] | 상속관계 표현 (한번더 ctrl+t 슈퍼타입 확인) |
| 편집 | [ctrl + shift + f] | 자동정렬 |
| 편집 | [alt + (+)] [alt + (-)] | 블록선택시: 블록전체를 위아래로 이동 블록미선택시: 커서위치 라인을 위아래로 이동 |
| 편집 | (한글(ㄱ~ㅎ) 입력 후) [한자키] | 특수키 입력 |
| 보기 | [ctrl + (+)] [ctrl + (-)] | 폰트 확대 및 축소 |
| 보기 | [ctrl+shift + (-)] [ctrl+shift + (I)] | 화면 가로 나누기 및 세로 나누기 |
| 보기 | [alt + (←)] [alt + (→)] | 이전 자바파일 히스토리로 이동 (자바파일) |

The End

자바 개요

자바 개요

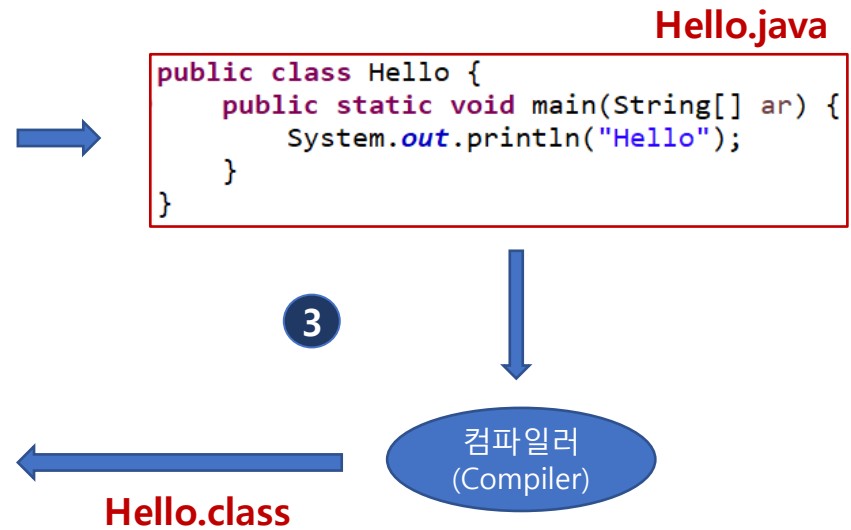
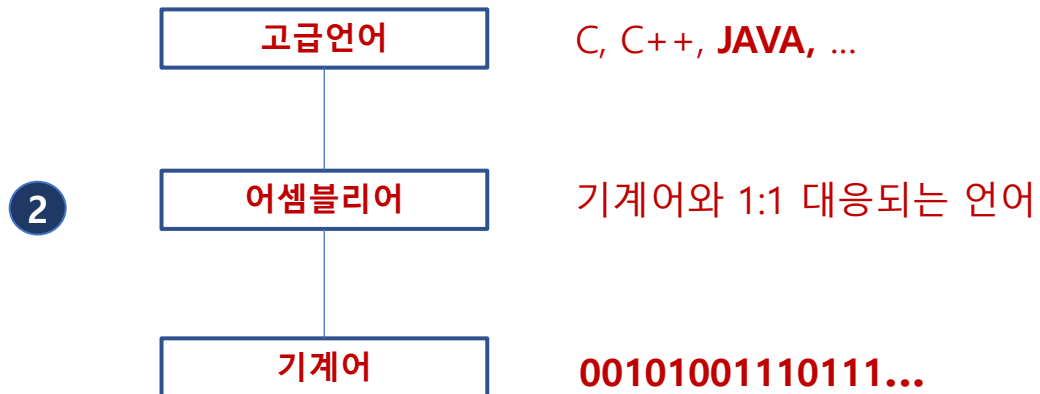
☞ 프로그래밍 언어

- 인간이 컴퓨터가 이해할 수 있는 명령을 작성하는 일련의 도구

1

(기계어) 00101001110111...

- 프로그램 언어의 종류



자바 개요



- 제임스 고슬링(James Gosling)

2

☞ 자바의 역사

1

- **1991**. 그린프로젝트(Green Project)로 탄생한 오크(Oak): 자바의 전신
- **1993**. 인터넷과 웹의 급속한 발전
- **1995**. 인터넷 환경에 적합하도록 오크를 새롭게 설계 (Java 발표)
- **1996**. 자바 1.0 발표
- **2014**. 자바 8 (LTS) 발표 (개인사용자에 한하여 2020년 12월까지 지원)
- **2018**. 자바 11 (LTS) 발표 (구독모델)
- **2020.09** 자바 15

| Version | Release date | End of Free Public Updates ^{[1][4]} | Extended Support Until |
|--|----------------|---|------------------------|
| JDK Beta | 1995 | ? | ? |
| JDK 1.0 | January 1996 | ? | ? |
| JDK 1.1 | February 1997 | ? | ? |
| J2SE 1.2 | December 1998 | ? | ? |
| J2SE 1.3 | May 2000 | ? | ? |
| J2SE 1.4 | February 2002 | October 2008 | February 2013 |
| J2SE 5.0 | September 2004 | November 2009 | April 2015 |
| Java SE 6 | December 2006 | April 2013 | December 2018 |
| Java SE 7 | July 2011 | April 2015 | July 2022 |
| | | January 2019 for Oracle (commercial) | |
| Java SE 8 (LTS) | March 2014 | December 2020 for Oracle (personal use) At least May 2026 for AdoptOpenJDK At least May 2026 ^[5] for Amazon Corretto | December 2030 |
| | | | |
| Java SE 9 | September 2017 | March 2018 for OpenJDK | N/A |
| Java SE 10 | March 2018 | September 2018 for OpenJDK | N/A |
| | | | |
| Java SE 11 (LTS) | September 2018 | At least October 2024 for AdoptOpenJDK At least September 2027 ^[5] for Amazon Corretto | September 2026 |
| | | | |
| Java SE 12 | March 2019 | September 2019 for OpenJDK | N/A |
| Java SE 13 | September 2019 | March 2020 for OpenJDK | N/A |
| Java SE 14 | March 2020 | September 2020 for OpenJDK | N/A |
| Java SE 15 | September 2020 | March 2021 for OpenJDK | N/A |
| Java SE 16 | March 2021 | September 2021 for OpenJDK | N/A |
| Java SE 17 (LTS) | September 2021 | TBA | TBA |
| Legend: ■ Old version ■ Older version, still maintained ■ Latest version ■ Future release | | | |

출처: https://en.wikipedia.org/wiki/Java_version_history

자바 개요

☞ 자바의 특징 및 점유율

1

플랫폼독립적

객체지향언어

함수형코딩지원

분산처리지원

멀티쓰레드지원

2

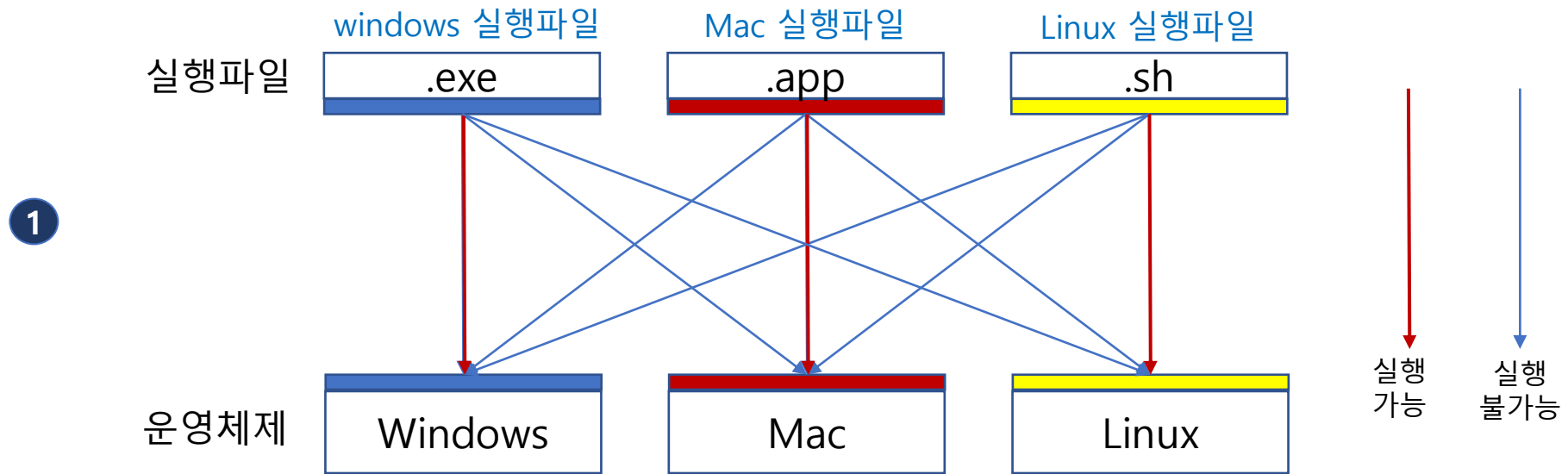
TIOBE Index for September 2020

| Sep 2020 | Sep 2019 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 2 | ▲ | C | 15.95% | +0.74% |
| 2 | 1 | ▼ | Java | 13.48% | -3.18% |
| 3 | 3 | | Python | 10.47% | +0.59% |
| 4 | 4 | | C++ | 7.11% | +1.48% |
| 5 | 5 | | C# | 4.58% | +1.18% |
| 6 | 6 | | Visual Basic | 4.12% | +0.83% |
| 7 | 7 | | JavaScript | 2.54% | +0.41% |
| 8 | 9 | ▲ | PHP | 2.49% | +0.62% |
| 9 | 19 | ▲▲ | R | 2.37% | +1.33% |
| 10 | 8 | ▼ | SQL | 1.76% | -0.19% |
| 11 | 14 | ▲ | Go | 1.46% | +0.24% |
| 12 | 16 | ▲▲ | Swift | 1.38% | +0.28% |
| 13 | 20 | ▲▲ | Perl | 1.30% | +0.26% |
| 14 | 12 | ▼ | Assembly language | 1.30% | -0.08% |
| 15 | 15 | | Ruby | 1.24% | +0.03% |
| 16 | 18 | ▲ | MATLAB | 1.10% | +0.04% |
| 17 | 11 | ▼▼ | Groovy | 0.99% | -0.52% |
| 18 | 33 | ▲▲ | Rust | 0.92% | +0.55% |
| 19 | 10 | ▼▼ | Objective-C | 0.85% | -0.99% |
| 20 | 24 | ▲▲ | Dart | 0.77% | +0.13% |

출처: <https://www.tiobe.com/tiobe-index/>

자바 개요

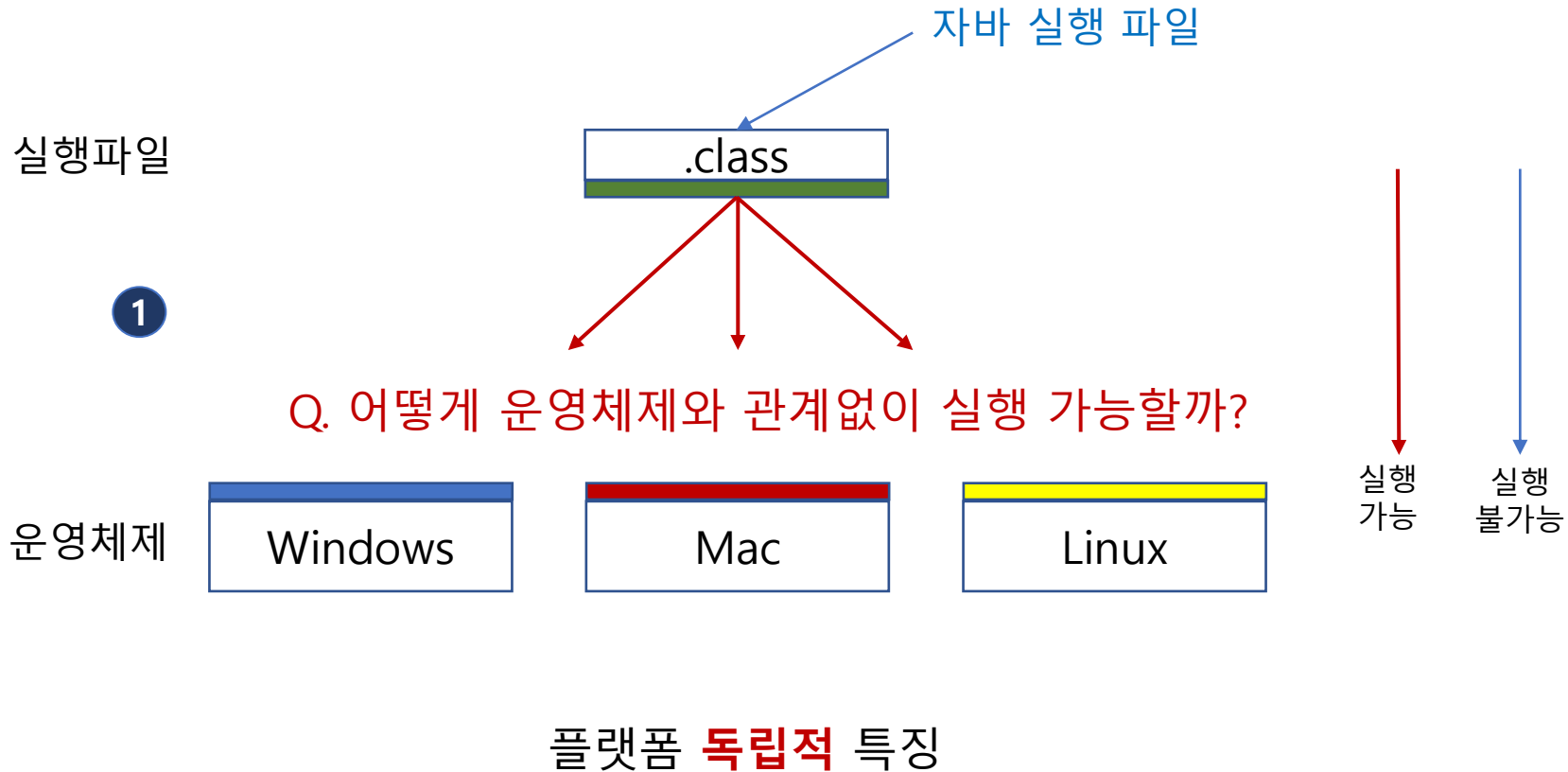
👉 플랫폼 종속 vs. 플랫폼 독립



플랫폼 종속적 특징

자바 개요

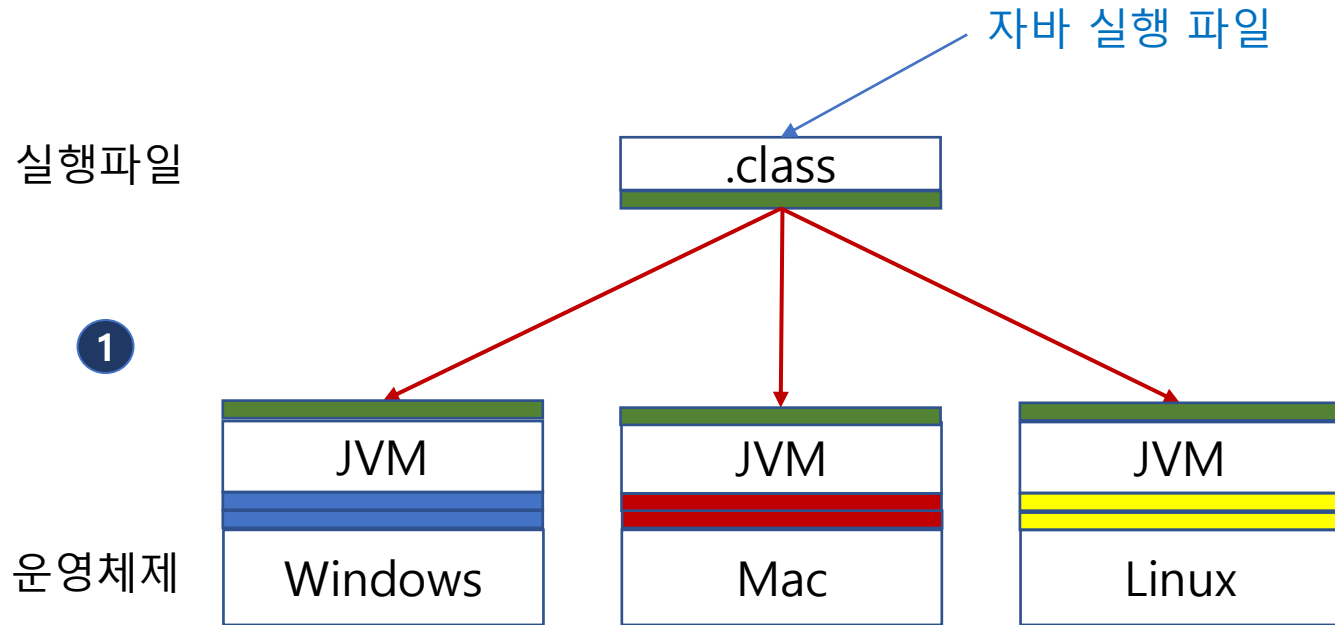
☞ 플랫폼 종속 vs. 플랫폼 독립



자바 개요

JVM: **J**ava **V**irtual **M**achine

👉 플랫폼 종속 vs. 플랫폼 독립

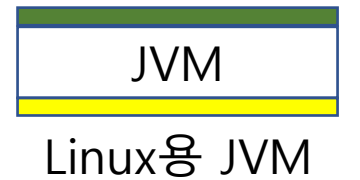
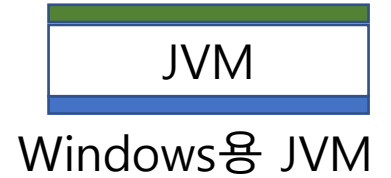


플랫폼 **독립적** 특징

TIP

- 운영체제 별 JVM은 오라클에서 제공

2



실행
가능

실행
불가능

3

| Java SE Development Kit 11.0.8 | | |
|--|-----------|---|
| This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE | | |
| Product / File Description | File Size | Download |
| Linux Debian Package | 148.77 MB | jdk-11.0.8_linux-x64_bin.deb |
| Linux RPM Package | 155.45 MB | jdk-11.0.8_linux-x64_bin.rpm |
| Linux Compressed Archive | 172.66 MB | jdk-11.0.8_linux-x64_bin.tar.gz |
| macOS Installer | 166.84 MB | jdk-11.0.8_osx-x64_bin.dmg |
| macOS Compressed Archive | 167.23 MB | jdk-11.0.8_osx-x64_bin.tar.gz |
| Solaris SPARC Compressed Archive | 186.49 MB | jdk-11.0.8_solaris-sparcv9_bin.tar.gz |
| Windows x64 Installer | 151.73 MB | jdk-11.0.8_windows-x64_bin.exe |
| Windows x64 Compressed Archive | 171.16 MB | jdk-11.0.8_windows-x64_bin.zip |

자바 개요

☞ JDK vs. JRE

1

- JDK: **J**ava **D**evelopment **K**it (자바개발도구)
- JRE: **J**ava **R**untime **E**nvironment (자바 실행환경)

TIP

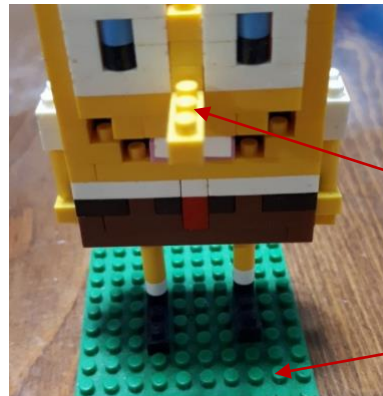
2

- JDK는 사용가능한 블록을 모아 놓은 개념
- Java 실행파일은 완성된 블록 개념
- JRE는 완성된 블록을 세워(실행) 놓을 수 있는 널찍한 판

3



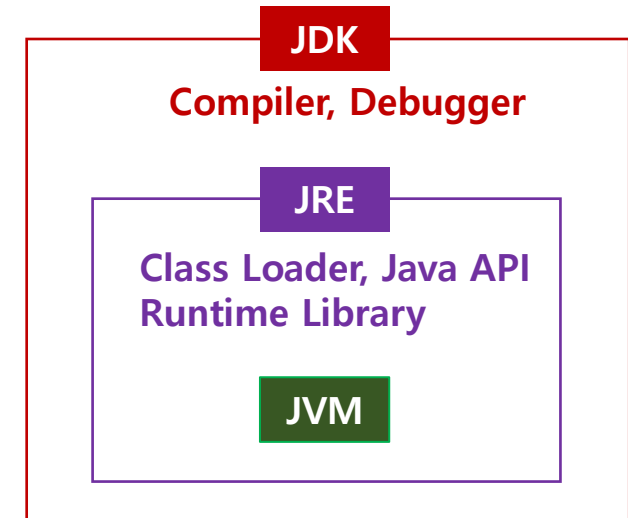
JDK



Java
프로그램

JRE

4



The End

자바프로젝트 생성 및 실행

자바프로젝트 생성 및 실행

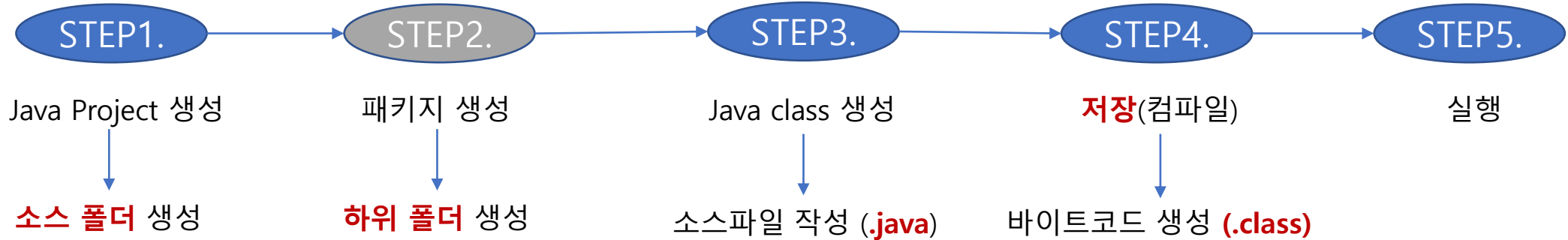
☞ 자바 프로젝트 작성 및 실행

3

TIP

STEP2와 STEP3을 묶어서 한번에 처리 가능

1



2

TIP

- 패키지 생성을 생략하는 경우 default package 설정 (하위폴더 생성 안함)
- 가능한 패키지를 지정

자바프로젝트 생성 및 실행

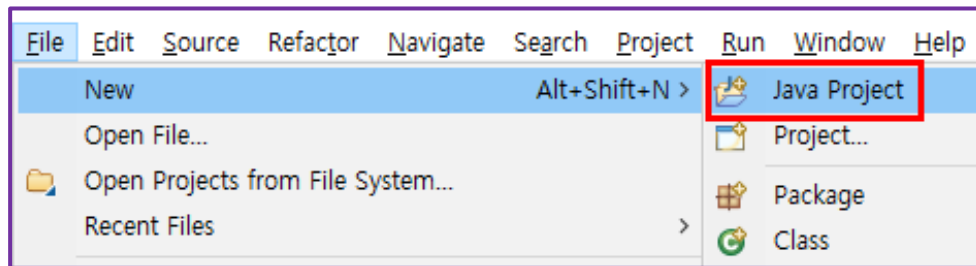
3

☞ 자바 프로젝트 작성 및 실행

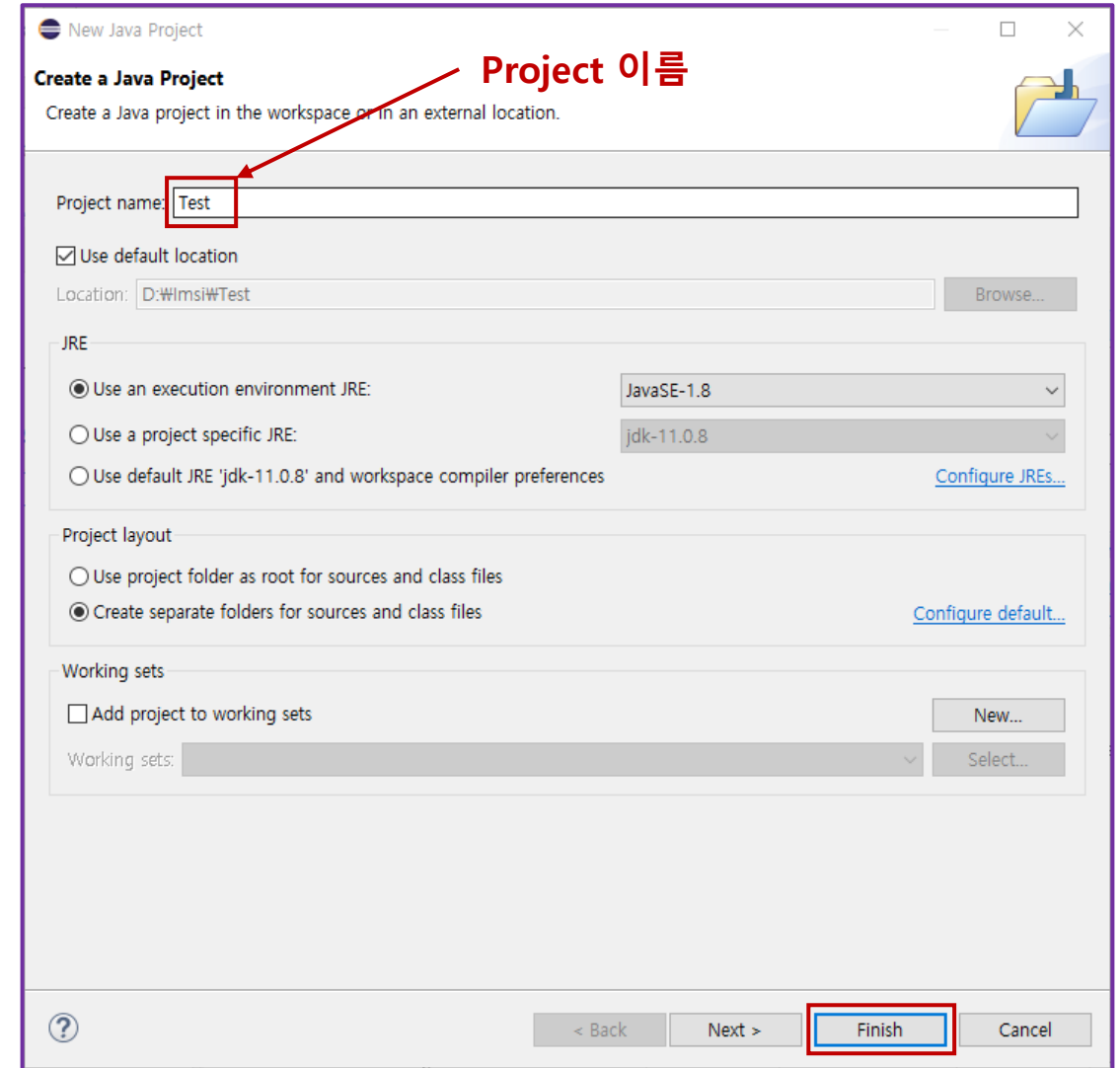
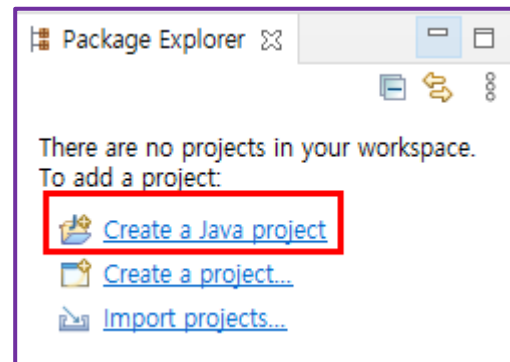
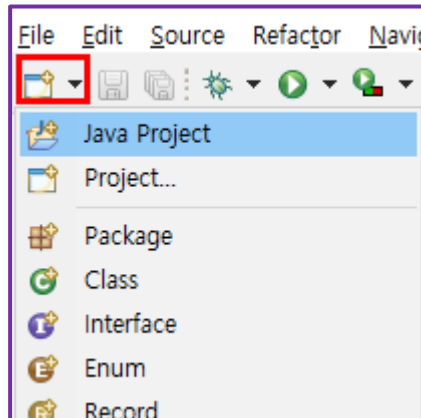
1

STEP1.

Java Project 생성



2



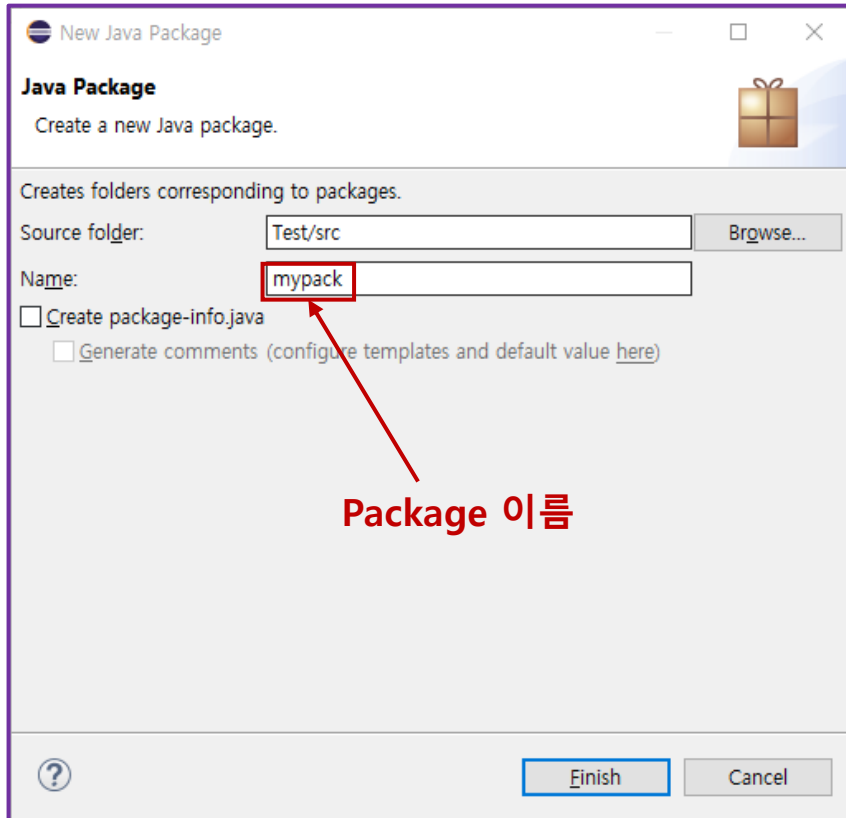
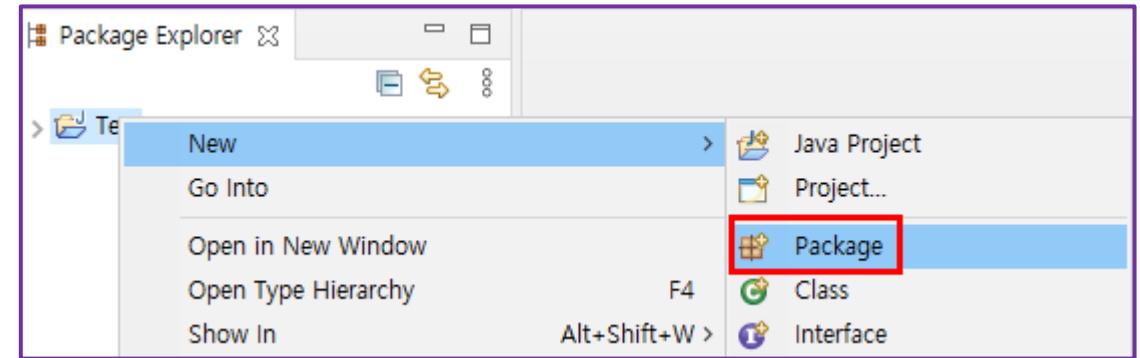
자바프로젝트 생성 및 실행

☞ 자바 프로젝트 작성 및 실행

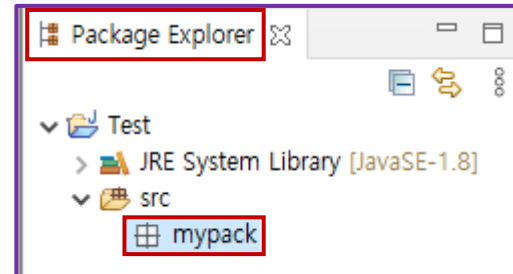
1

STEP2.

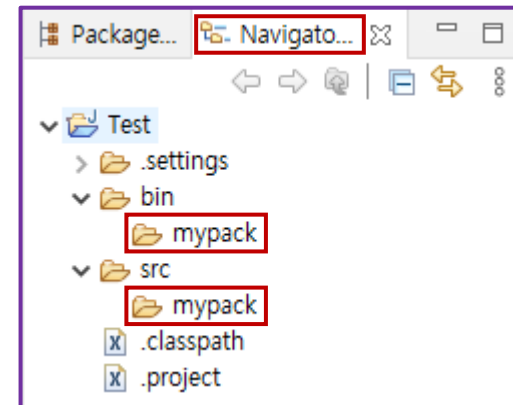
패키지 생성
↓
하위 폴더 생성



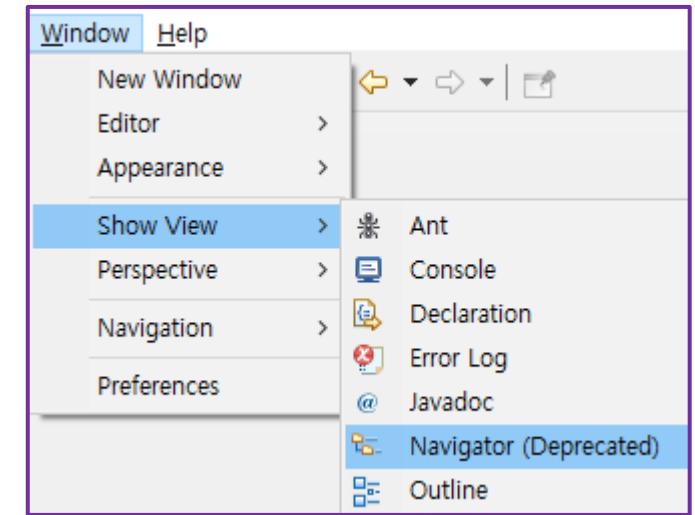
4



6



5



자바프로젝트 생성 및 실행

☞ 자바 프로젝트 작성 및 실행

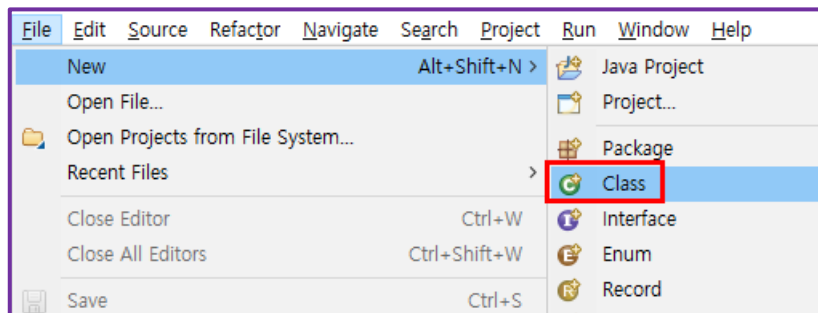
1

STEP3.

Java class 생성

↓
소스파일 작성 (.java)

2



5

STEP2에서
설정한
패키지이름

STEP2를 SKIP 한 경우 여기에
직접 패키지 이름 입력 가능
(STEP2+STEP3)

4

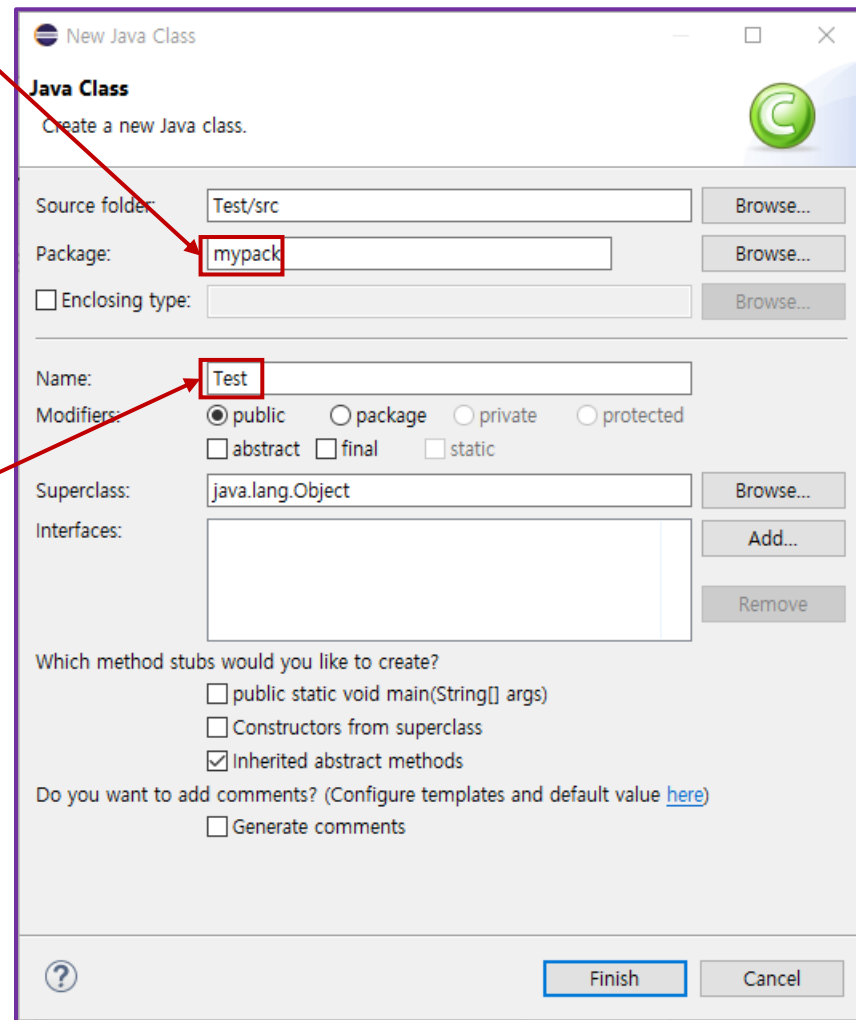
클래스 이름

TIP

6

- 자바 소스파일 이름과 클래스 이름은
반드시 동일하여야 함

3



자바프로젝트 생성 및 실행

☞ 자바 프로젝트 작성 및 실행

TIP

- 명령프롬프트(CMD) 창에서는 javac 를 이용하여 직접 컴파일을 하여야 함
- 이클립스에서는 **저장**버튼을 누를 때마다 자동 컴파일 수행

2

TIP

- **Package Explorer**는 자바 코드를 작성하기 위한 파일만 보임
- 윈도우탐색기와 유사하게 보려면 **Navigator View**를 사용

3

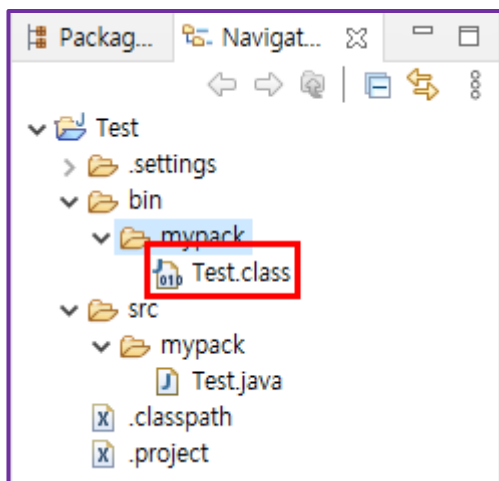
1

STEP4.

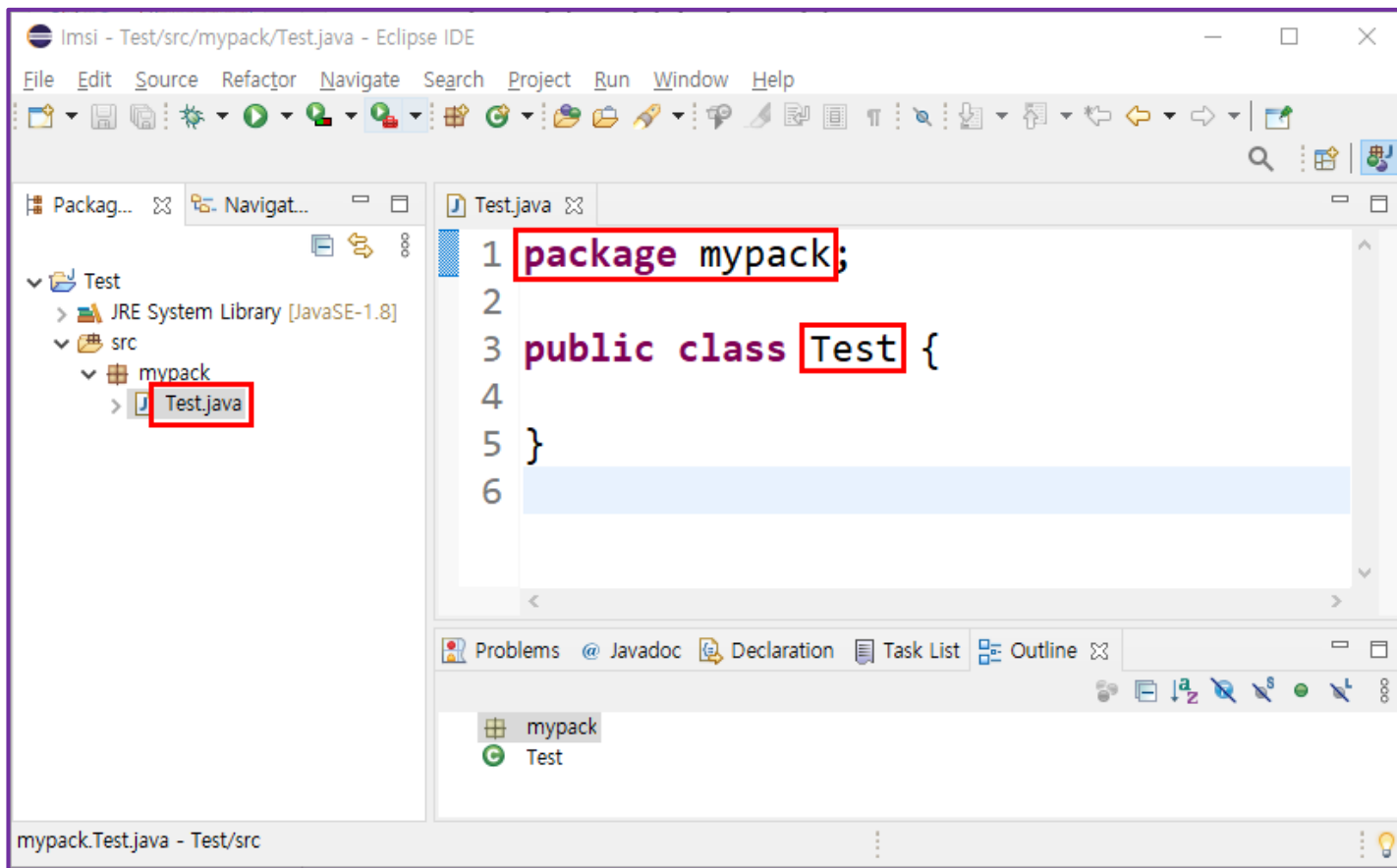
저장(컴파일)

↓
바이트코드 생성 (.class)

4



5



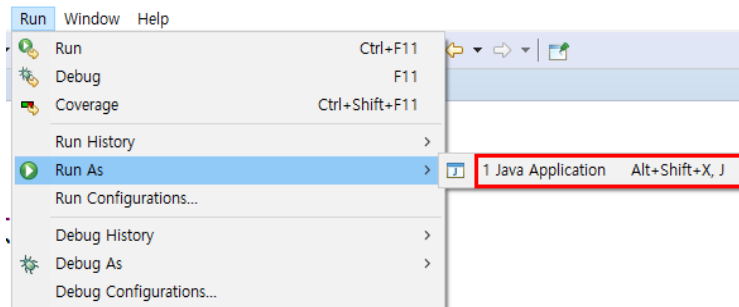
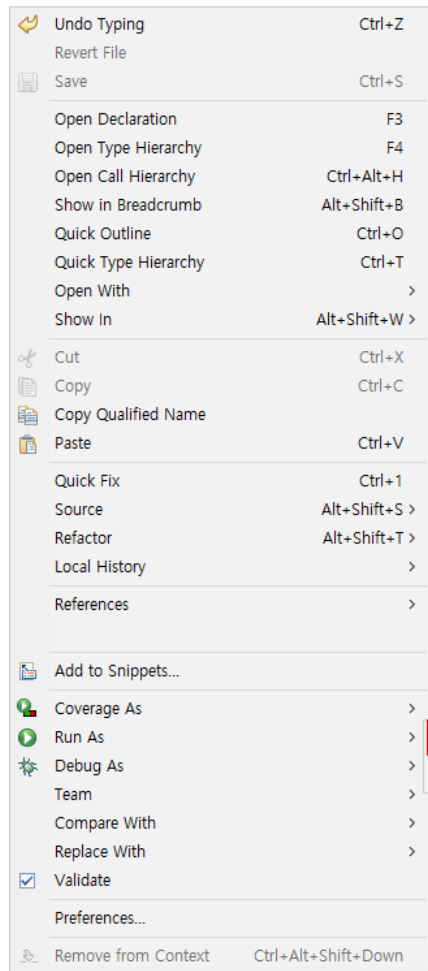
자바프로젝트 생성 및 실행

☞ 자바 프로젝트 작성 및 실행

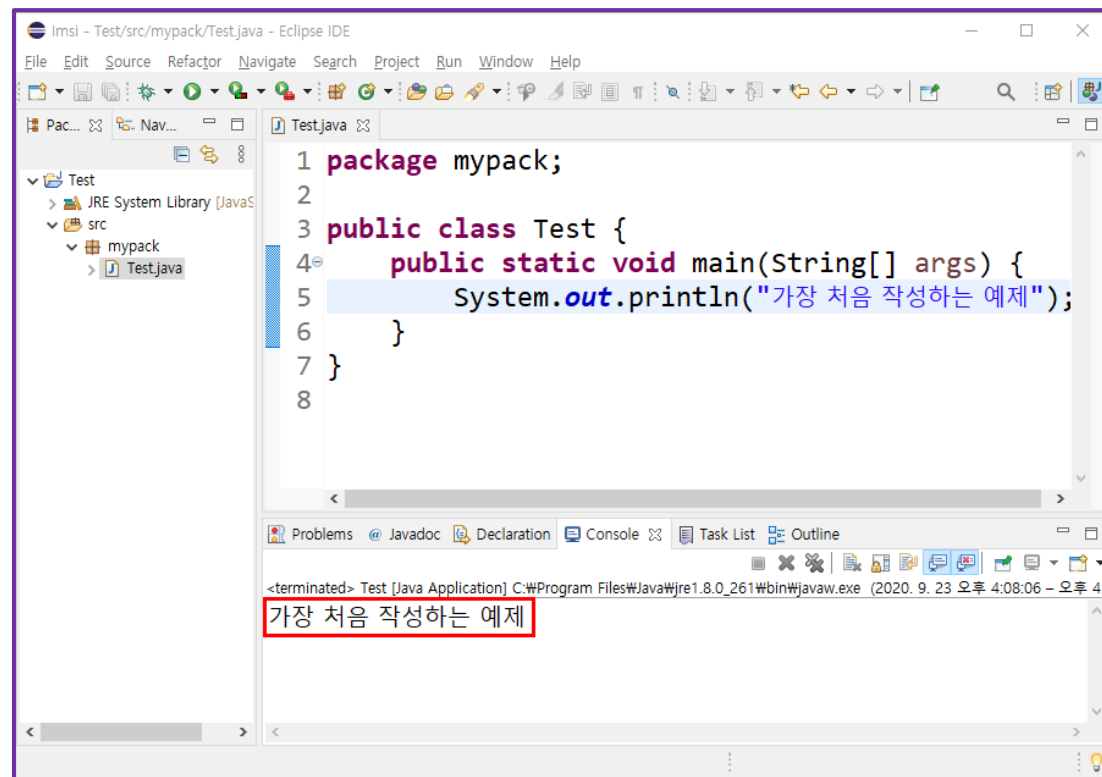
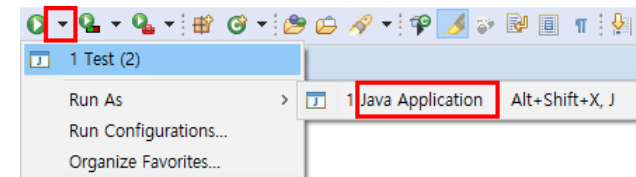
1

STEP5.

실행



2



자바프로젝트 생성 및 실행

☞ 자바 프로젝트 생성 실습

- 예제1. 프로젝트 생성 및 실행 (패키지 미설정)

1

- 프로젝트 이름: Test01
- 패키지 이름 : 디폴트 패키지 (패키지를 지정하지 않음)
- 소스파일 이름: Test.java

- 예제2. 프로젝트 생성 및 실행 (패키지 설정)

2

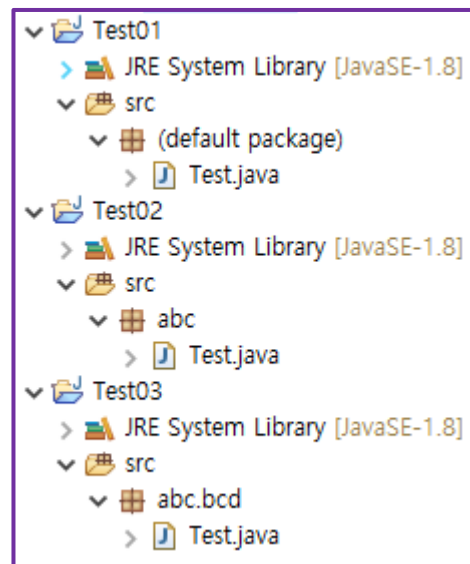
- 프로젝트 이름: Test02
- 패키지 이름 : abc
- 소스파일 이름: Test.java

- 예제3. 프로젝트 생성 및 실행 (패키지 설정)

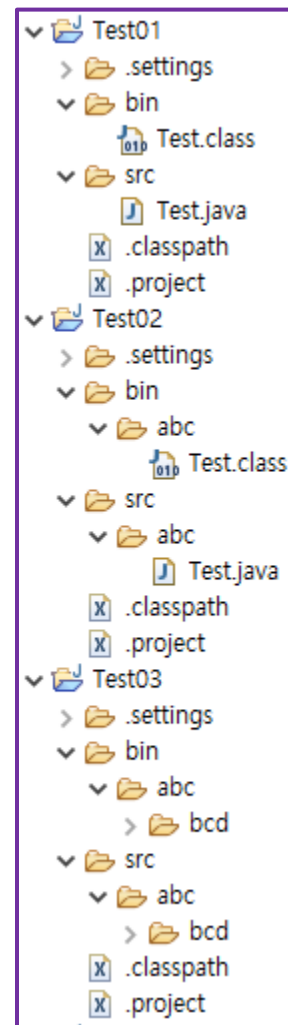
3

- 프로젝트 이름: Test03
- 패키지 이름 : abc.bcd
- 소스파일 이름: Test.java

4



5



6

Quiz.

디폴트 패키지와 패키지를 지정했을 때 소스코드 상에서의 차이점은?

자바프로젝트 생성 및 실행

👉 자바 프로젝트 생성 실습

1

Quiz.

디폴트 패키지와 패키지를 지정했을 때 소스코드 상에서의 차이점은?

2

```
public class Test {  
}
```

3

```
package abc;  
  
public class Test {  
}
```

4

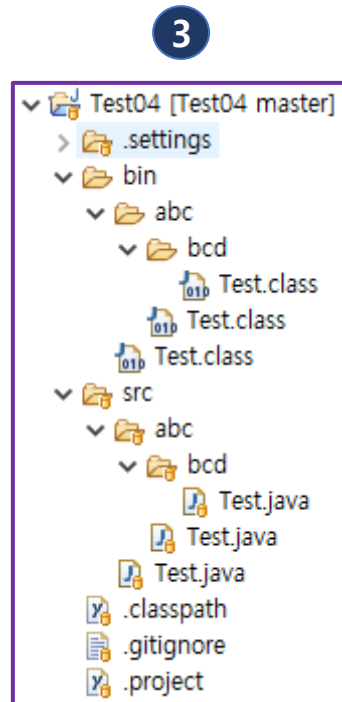
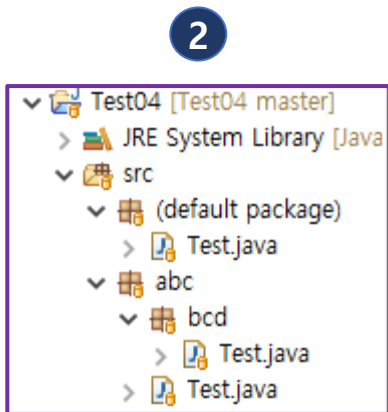
```
package abc.bcd;  
  
public class Test {  
}
```

자바프로젝트 생성 및 실행

☞ 자바 프로젝트 생성 실습

- **예제4. 패키지만 구분하여 예제1~예제3을 하나의 프로젝트에 작성**

- 1 - 프로젝트 이름: Test04

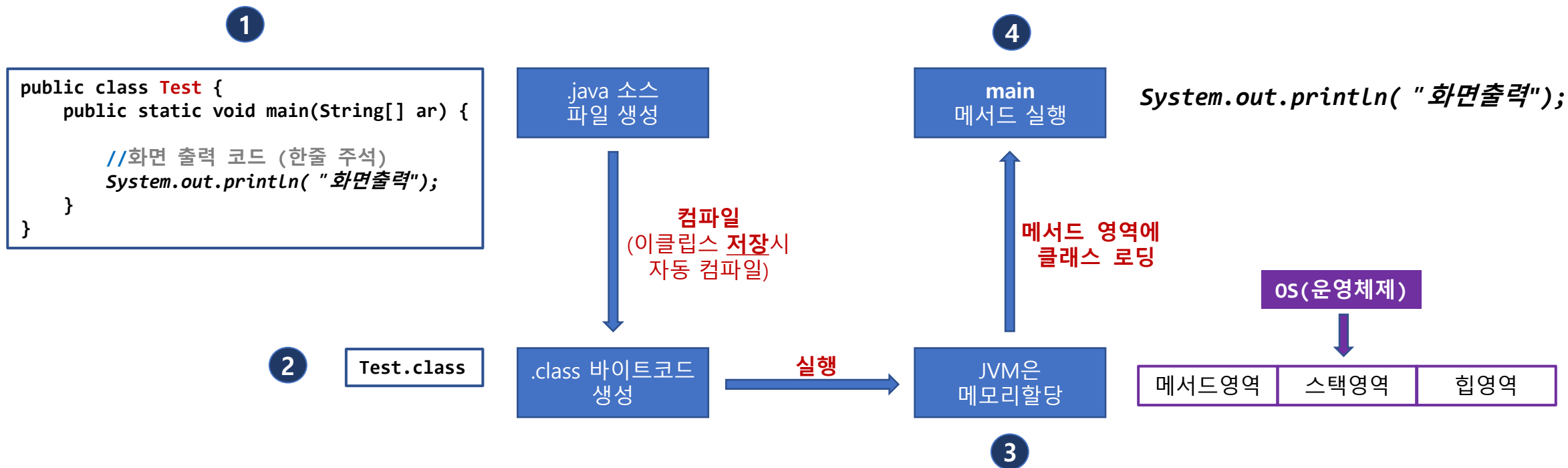


The End

기본 구조 및 문법

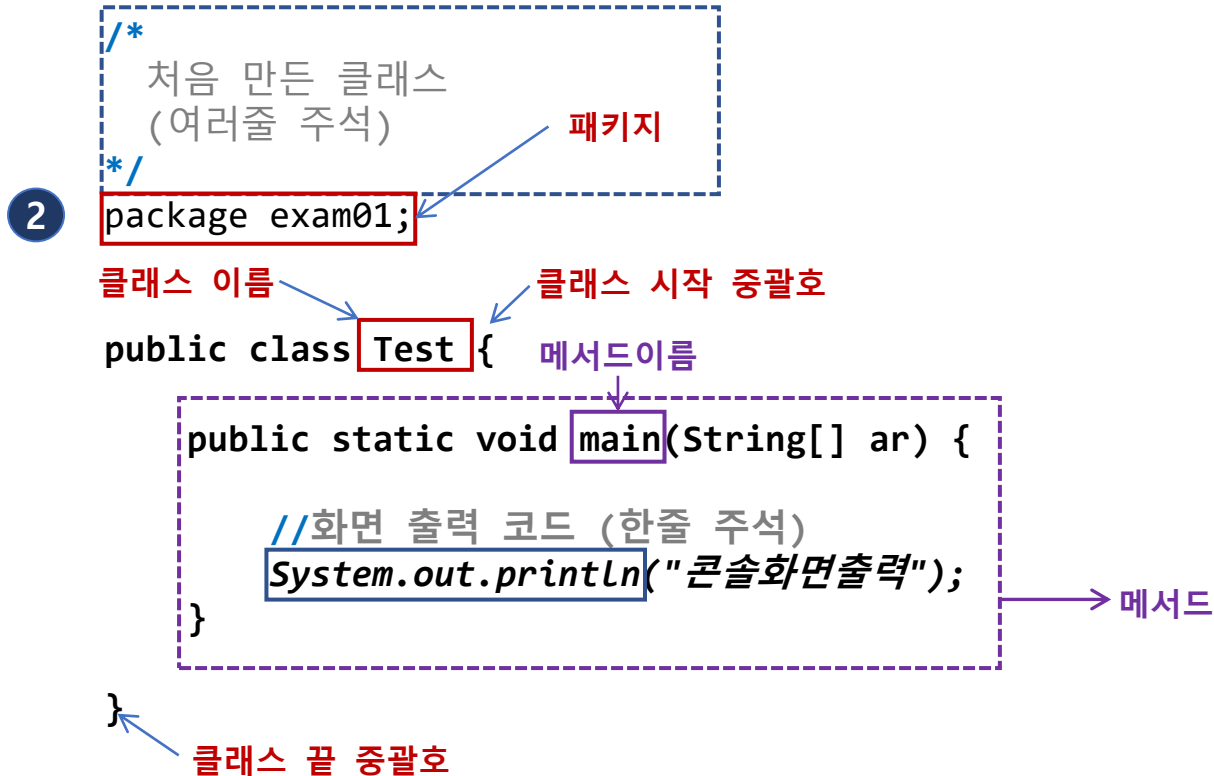
기본 구조 및 문법

☞ Java의 실행 프로세스



기본 구조 및 문법

☞ 처음 클래스 생성시 클래스 분석



3 TIP
패키지를 지정하지 않은 경우 클래스 외부에는 아무것도 오지 않음

- **주석**

1
/* 주석내용 */ : 여러 줄 주석
// 주석내용 : 한 줄 주석 (//이후 줄 마지막까지)

- **클래스** `public class Test {`
....
}

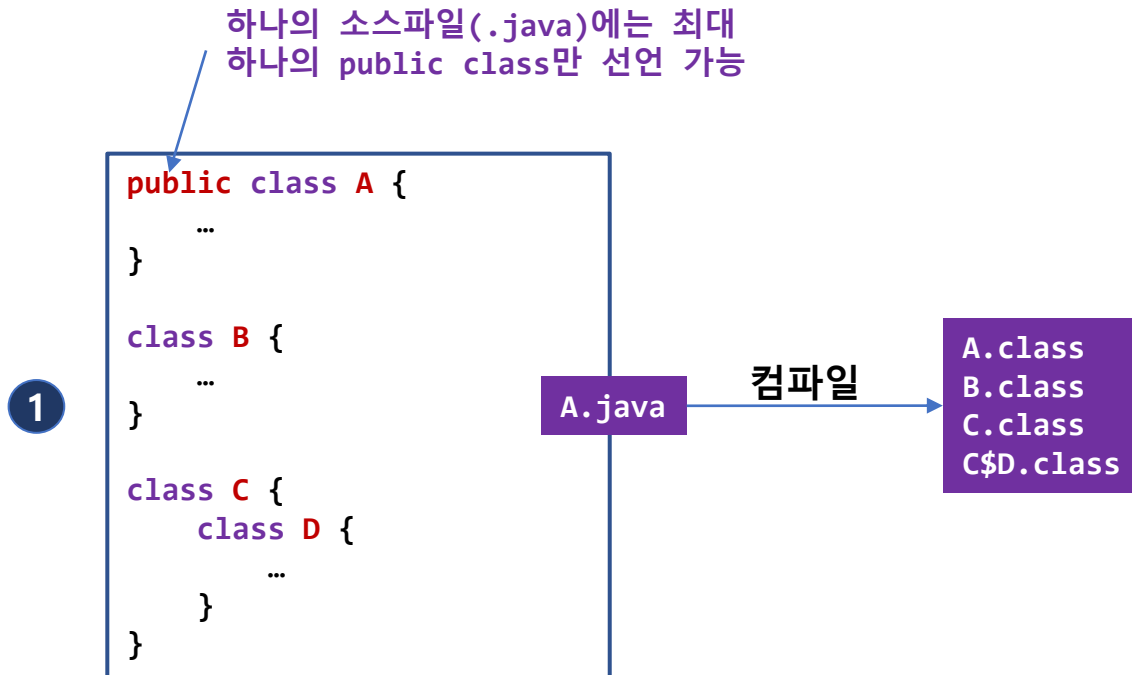
4
- public: 접근지정자 키워드 (파일당 최대 1개)
- class 키워드는 클래스의 이름 앞에 오는 예약어
- 클래스 이름(Test)은 자유롭게 선정 가능하지만 소스 파일 이름과 동일해야 함 (Test.java)

- **메서드** `public static void main(String[] ar) {`
....
}

5
- public : 접근지정자 키워드
- static : 정적 메서드 키워드
- 나머지부분(`void main(...){ }`):
메서드의 원형- **반환타입 메서드이름 (...){...}**

기본 구조 및 문법

☞ 소스파일 컴파일과 바이트코드 생성



CHECK 1

2 소스파일별 바이트코드 생성이 아닌 **클래스별 바이트 코드(.class)**가 생성됨
(class 키워드 개수와 동일하게 생성)

CHECK 2

3 소스파일에는 public class가 최대 1개만 존재할 있으며 public class의 이름이 파일 이름과 동일하여야 함

CHECK 3

- 4
- 외부클래스(external class)
클래스이름.class
 - 내부클래스(inner class)
외부클래스이름\$내부클래스이름.class

기본 구조 및 문법

👉 실습예제

프로젝트이름: Chap01

패키지이름: exam01

클래스이름: A.java

1

```
*A.java
1 package exam01;
2
3 public class A {
4
5 }
6 class B{
7
8 }
9 class C{
10     class D{
11
12     }
13 }
```

2

```
Chap01
├── .settings
├── bin
│   ├── abc
│   └── exam01
│       ├── A.class
│       ├── B.class
│       ├── C.class
│       ├── C$D.class
│       └── Test.class
├── src
│   ├── .classpath
│   └── .project
```

기본 구조 및 문법

TIP

문자열은 쌍따옴표(" ")안에만 표기 가능

1

TIP

2

String 자료형과의 '+' 연산

- String + String = String
- String + 기본자료형 = String
- 기본자료형 + String = String

3 콘솔출력 System.out.xxx() 메서드와 문자열(String)

- **println() 메서드:** 괄호안의 내용 출력 + 줄바꿈

4

```
System.out.println("화면출력");  
System.out.println("화면"+"출력");  
System.out.println(3.8);  
System.out.println(3+5);  
System.out.println("화면"+3);  
System.out.println("화면"+3+5);  
System.out.println(3+5+"화면");
```

화면출력
화면출력
3.8
8
화면3
화면35
8화면

5

```
int a = 3;  
String b = "화면";  
  
System.out.println(a);  
System.out.println(b);  
System.out.println(b+"출력");  
System.out.println(a+b+"출력");
```

3
화면
화면출력
3화면출력

- **print() 메서드:** 괄호안의 내용 출력 6

```
System.out.print("화면");  
System.out.print("출력");  
System.out.print(3);
```

화면출력3

- **printf() 메서드:** 자료형 포맷에 따라 출력 7

```
System.out.printf("%d\n",30); (10진수)  
System.out.printf("%o\n",30); (8진수)  
System.out.printf("%x\n",30); (16진수)
```

```
System.out.printf("%s\n","출력");  
System.out.printf("%f\n",5.8);  
System.out.printf("%4.2f\n",5.8);  
System.out.printf("%d와 %4.2f\n",4,5.8);
```

30
36
1e

출력
5.800000
5.80
4와 5.80

The End