

조영철 010-2390-9476 ren44@naver.com

# JAVA PROGRAMING

# 프로그래밍 기초 이론

목적지까지 버스를 운행하고 싶어.. 어떻게 하면 좋을까?



버스 / 도로 등 모든 기기는 제공되어 있는 상태에서 버스 운전사를 구현해야 함  
(즉, 모든 하드웨어가 구현되어 있는 상태에서 원하는 기능을 구현해야 함)

# 프로그래밍 기초 이론

## 코더? 프로그래머?

사전 상 Coder란 뜻은 "코딩(Coding)하는 사람"이란 뜻입니다.  
하지만 프로그래머(Programmer)도 코딩을 합니다.  
(소프트웨어 아키텍트(Software Architect) 및 구루(Guru) 라는 상위 계층도 존재합니다.)

과연 코더와 프로그래머를 나누는 기준은 무엇일까요?

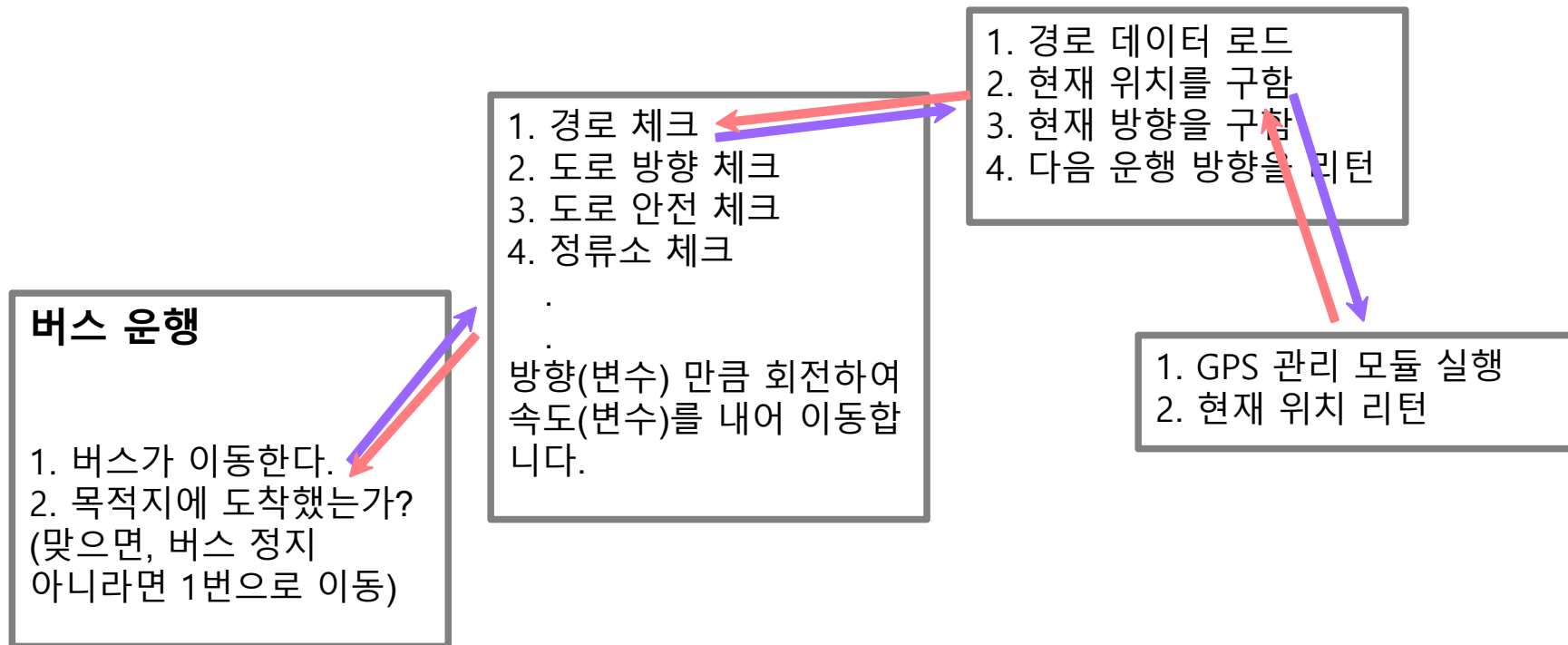
그 차이는 바로 가지고 있는 사상의 차이입니다. 첫 강의에서 이 부분을 강조하는 이유는 자신의 역할을 제대로 파악하기 위함입니다. 단순히 프로그래머가 되어야한다고 말하는 것이 아닙니다. 프로젝트 진행에 있어서 각각의 역할을 제대로 수행할 수 있을 때 제대로 된 프로젝트의 진행이 가능합니다.

프로그래머가 제 역할을 수행하지 못하면, 프로젝트 전체에서 문제가 생기며,  
코더가 제 역할을 수행하지 못하면, 해당 모듈에서 버그가 발생합니다.

즉, 코더란 어떠한 소프트웨어를 개발함에 있어 설계문서를 보면서 자신의 사상보다는 다른 사람들의 사상을 바탕으로 소프트웨어를 개발 또는 모듈을 제작하는 사람입니다.

# 프로그래밍 기초 이론

IF적인 사고



프로그래머는 바로 이러한 구조를 작성할 수 있어야 합니다. 기초 이론이라고 무시하지 마시고 실제로 자신이 버스 운행을 구현한다면 어떻게 구현해야 하는가? 진지하게 고민해 보시길 바랍니다. 단순히 상위 모듈로부터 하위 모듈까지의 구현으로 끝나는 것이 아니라 각 모듈간 교환하는 메시지는 어떻게 정의해야 하는가? 까지 정해져야 합니다.

현재 위 구조는 버그 투성이 입니다. 어떠한 버그가 발생할 수 있을까요?

# 프로그래밍 기초 이론

## 프로그램을 만들려면 필요한 것들

프로그램이 어떤 목적을 가지고 있는가?

프로그램에 필요한 데이터는 무엇인가?

프로그램에 필요한 로직은 무엇인가?

- 로직을 구현하는데 있어서의 걸림돌은 무엇인가?

프로그램의 결과는 어떤 형태인가?

## 어떤 목적을 가지고 만드는가?

목적(objectives)

- 프로그램을 사용하는 고객이 원하는 기능
- FR(Functional Requirements)
- NFR(Non Functional Requirements)

입문자들에게 가장 쉬운 목적 파악 방법?

- 실행되는 화면을 그려보는 일

화면에 숫자를 입력해 주세요!

13 <--입력후 엔터

당신이 입력한 숫자는 13

# 프로그래밍 기초 이론

## 프로그램에 필요한 데이터

필요한 데이터 = 변수

변수의 기준 적용

- 사용자의 입력은 변수

변수의 타입 고려

- 사용자의 입력은 숫자이므로 int

## 프로그램에 필요한 로직

로직의 순서를 세운다.

사용자에게 입력하라는 메시지를 띄운다.

사용자가 키보드에서 데이터를 입력한다.

입력한 데이터는 변수로 받아 둔다.

받아둔 변수를 이용해서 결과를 출력한다.

화면에 숫자를 입력해 주세요!

13 <--입력후 엔터

당신이 입력한 숫자는 13

# 프로그래밍 기초 이론

## 로직을 구현하는데 걸림돌?

모르는 내용은 별도의 테스트를 통해서 실험

실험된 결과를 프로그램에 반영

현실적으로 검색엔진을 통한 검색 혹은 API문서라는 것을 이용

‘키보드에서 들어오는 숫자 입력 받기’는 JDK1.5 이후에는 java.util.Scanner라는 것을 이용하면 가능하다.

## 키보드 입력 받는 테스트

```
import java.util.Scanner;

public class ScannerTest {

    public static void main(String[] args) {

        //키보드에서 데이터 입력받는 장치
        Scanner scanner = new Scanner(System.in);

        int value = scanner.nextInt();

        System.out.println(value);

    }
}
```



# 프로그래밍 기초 이론

## 자료구조, 알고리즘

자료구조는 **컴퓨터에 자료를 효율적으로 어떻게 저장할 것인가?**에 대해 저장하는 방식을 알려주는 학문입니다.

자료를 어떻게 저장할 것인가? 자료를 어떻게 쉽게 꺼내올 것인가?를 연구하는 것입니다.

현재 자료구조를 몰라도 코딩을 하는데 문제가 없습니다. 이유는 이미 구현되어 있는 라이브러리를 이용하기 때문입니다.

하지만, 좀더 프로그래밍적인 사고를 높이기 위해서 공부해야할 필요가 있습니다. 산수를 못해도 계산기를 이용하면 문제 없지만, 산수를 모르는 사람은 높은 수준의 깨우침을 알 수 없습니다.

아직도 초중고에서 배우는 학문이 실제 생활에는 필요 없으니 안배워도 돼~!! 라고 생각하시는 분은 없겠죠?

일단은 자료구조를 효율적으로 하기 위해서 **"알고리즘"**을 잘 짜는 것이 가장 중요합니다.

좋은 알고리즘에 대한 조건은 **"유한성, 유효성, 그리고 뚜렷한 결과"**를 창출해낼 수 있어야 한다는 것입니다.

1)  $1+2+3+...+n$

2)  $n*(n+1) / 2$

위 두가지 방법 중에 어느것이 쉽고 효율적인지는 이미 초등학교때 배우셨을겁니다.

알고리즘은 단순히 자료구조에만 사용되는 것이 아닙니다. 기초 이론의 대다수에는 알고리즘이 들어갑니다. 자신의 생각의 폭을 넓히기 위해서라도 다양한 기초 이론을 공부하시길 권장합니다.

# 프로그래밍 기초 이론

전산학에서 **자료 구조**(資料構造, 영어: data structure)는 자료를 효율적으로 이용할 수 있도록 컴퓨터에 저장하는 방법이다. 신중히 선택한 자료구조는 보다 효율적인 알고리즘을 사용할 수 있게 한다. 이러한 자료구조의 선택문제는 대개 추상적 자료구조의 선택으로부터 시작하는 경우가 많다. 효과적으로 설계된 자료구조는 실행시간 혹은 메모리 용량과 같은 자원을 최소한으로 사용하면서 연산을 수행하도록 해준다.

자료 구조에는 여러 종류가 있으며, 이러한 각각의 자료구조는 각자의 연산 및 목적에 맞추어져 있다. 예를 들어 B-트리는 데이터베이스에 효율적이며, 라우팅 테이블은 네트워크에 일반적이다.

다양한 프로그램을 설계할 때, 어떠한 자료 구조를 선택할지는 가장 우선적으로 고려되어야 한다. 이는 큰 시스템을 제작할 때 구현의 난이도나 최종 결과물의 성능이 자료 구조에 크게 의존한다는 것을 많은 경험 뒷받침하기 때문이다. 일단 자료구조가 선택되면 적용할 알고리즘은 상대적으로 명확해지기 마련이다. 때로는 반대 순서로 정해지기도 하는데, 이는 목표로 하는 연산이 특정한 알고리즘을 반드시 필요로 하며, 해당 알고리즘은 특정 자료구조에서 가장 나은 성능을 발휘할 때와 같은 경우이다. 어떠한 경우든, 적절한 자료 구조의 선택은 필수적이다.

이러한 관점은 알고리즘보다 자료 구조가 보다 중요한 요소로 적용되는 많은 정형화된 개발론 그리고 프로그래밍 언어의 개발을 촉발시켰다. 대부분의 언어는 일정 수준의 모듈개념을 가지고 있으며, 이는 자료구조가 검증된 구현은 감춘 채 인터페이스만을 이용하여 다양한 프로그램에서 사용되는 것을 가능하게 해준다. C++나 자바와 같은 객체지향 프로그래밍 언어는 특별히 이러한 목적으로 객체를 사용한다.

이러한 자료 구조의 중요성으로 말미암아, 최근의 프로그래밍 언어 및 개발 환경은 다양한 표준 라이브러리를 제공하고 있다. 예로, C++의 표준 템플릿 라이브러리나 자바의 자바 API, 마이크로소프트 .NET과 같은 것을 들 수 있다.

\* 위 내용을 전부 이해할 수 있어야 합니다. (출처 위키)

# 객체지향 언어

## 절차 지향 언어 [ procedure-oriented language, 節次指向言語]

프로그램을 작성할 때 실행 순서를 지정하게 되는 프로그램 작성 언어.

문제의 해결 순서와 절차의 표현과 해결이 쉽도록 설계된 프로그램 언어로서  
고수준 언어에는 C 언어, 파스칼, 코볼, 포트란, 알골, PL/1 등이 있다.

이에 대하여 리스프(LISP)와 같은 함수형 언어나 프롤로그와 같은 논리형 프로그램 작성 언어  
는  
실행 순서를 지정할 필요가 없어 비절차형, 또는 비절차 지향 언어라고 한다.

## 객체 지향 언어 [ object-oriented language, 客體指向言語 ]

객체 지향 프로그래밍(OOP) 환경에서 사용되는 프로그램 언어의 총칭.

대표적인 객체 지향 언어로는 추상 데이터형 프로그램 언어의 시조라고 할 수 있는  
시뮬레이션 언어 시뮬라, 스몰토크 80 등이 있다.

또한 기존 언어에서 객체 지향으로 확장된 것으로는 표와 같은 것이 있다.

# 객체지향 언어

**객체 지향 프로그래밍 [ OOP, objective-oriented programming, 客體指向- ]**

실세계의 현상을 컴퓨터상에 객체로 실현(모델화)함으로써,  
컴퓨터를 자연스러운 형태로 사용하여 다양한 문제를 해결하기 위한 프로그램 기법.  
여기에서 객체는 실체(데이터)와 그 실체와 관련되는 동작(절차, 방법, 기능)을 모두 포함한다.

기차역에서 승차권 발매의 예를 들면,  
실체인 '손님'과 절차인 '승차권 주문'은 하나의 객체이고,  
실체인 '역무원'과 절차인 '승차권 발매'도 하나의 객체이다.

어떤 과제를 처리하기 위해 객체 간에는 메시지(지시)를 주고받는다.  
메시지를 받은 객체는 동작(절차)을 실행한다.

코볼 L이나 베이식과 같은 기존 프로그램에서는 동작과 절차를 중심으로 하고 실체는 종속적으로 취급했으나, 객체 지향 프로그램에서는 실체와 동작을 객체로 정의하고 객체 간의 메시지 교환에 주안점을 두어 정보를 처리한다.

즉, 객체 지향은 과정을 중시하는 절차 중심의 설계가 아니고, 실체를 중시하는 설계이다.  
객체 지향 프로그램의 또 하나의 중요한 특징은 공통의 성질을 갖는 객체는 객체 등급으로 정의한다는 점인데, 같은 등급에 속하는 객체들은 그들이 받는 메시지에 대하여 비슷하게 반응한다.

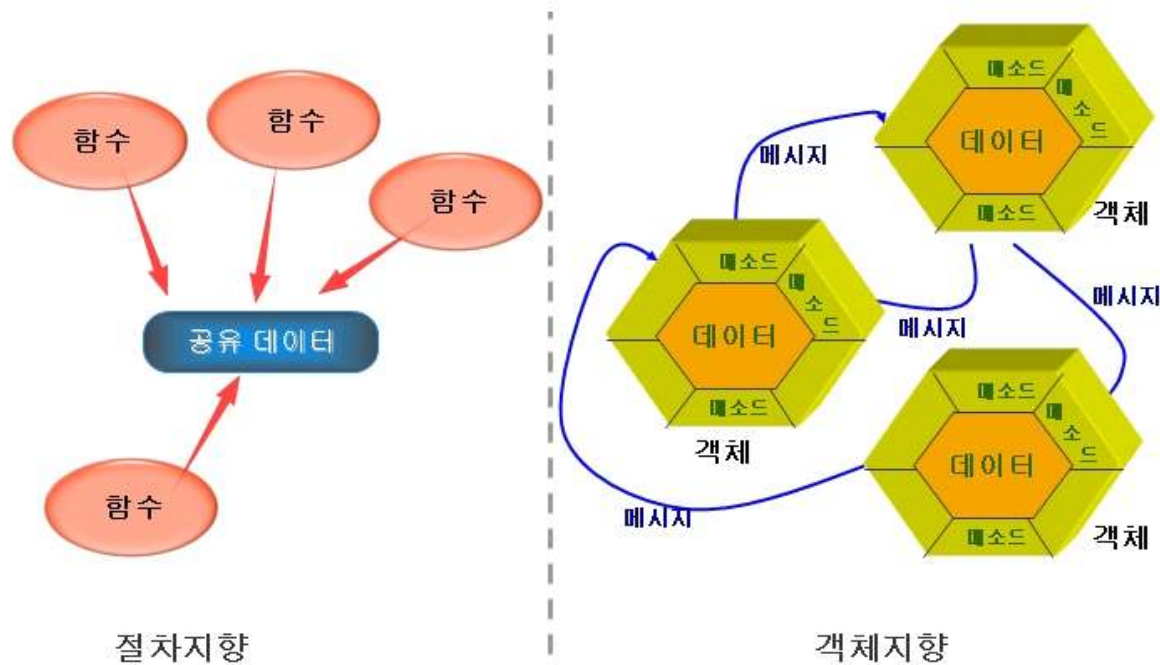
# 객체지향 언어

객체 등급은 계층화할 수 있으며, 하위 계층의 객체 등급은 상위 등급의 성질과 기능을 계승한다.

따라서 객체 지향은 시스템의 모듈화, 캡슐화를 촉진하여 복잡화, 거대화되는 소프트웨어를 사용하기 쉽고,

작성하기 쉬우며, 유지 보수하기 쉬운 방향으로 재구축하는 새로운 기법으로 각광받고 있다.

객체 지향 프로그램은 스몰토크와 같은 객체 지향 언어로 작성된다.



# 객체지향 언어

## 객체지향이란??

실제 세계를 모델링하여 소프트웨어를 개발하는 방법

## 객체란 무엇인가??

우리 주변에 실제 존재하는 컴퓨터, 휴대폰, TV, 사람 등의 사물을 말한다.

프로그래밍에서의 객체는 클래스에 정의된 내용대로 메모리에 생성된 것을 말한다.

이러한 객체는 상태(state)와 동작(behavior)을 가지고 있다.

이런 소프트웨어 객체들은 다른 소프트웨어 객체들과 서로 상호작용하기 위하여 메시지(message)를 사용한다.

## 클래스란??

객체를 만드는 설계도라 할 수 있다.(ex - 자동차의 설계도, 봉어빵의 기계틀)

이 클래스로부터 만들어지는 각각의 객체들을 인스턴스(instance)라 한다.

# 객체지향 언어

## 객체지향의 특징

- 캡슐화 : 객체 내부에 구현된 정보를 외부에 노출하지 않는 것  
메소드(method) 호출을 통하여 결과를 알 수 있다.  
(ex - 캡슐에 쌓인 약 안에 든 성분은 알 수 없지만, 그 약이 지닌 효능만 알고 그 약을 복용하는 경우)
- 정보은닉 : 객체 내부의 데이터와 구현의 세부 사항을 외부 세계에 감추는 것  
외부 세계에 영향을 끼치지 않으면서 쉽게 객체 내부를 업그레이드 할 수 있다  
(ex - 컴퓨터의 외부단자와 스피커를 연결한 경우 내부의 사운드 카드를 변경 할 수 있지만, 내부의 오디오 단자와 스피커를 직접 연결한 경우 내부의 사운드 카드를 변경할 수 없다.  
전자 - 정보은닉의 경우)
- 상속 : 부모클래스에 이미 정의된 코드를 자식클래스가 그대로 이어받아 사용하는 것  
(ex - 부모클래스 : 자동차, 자식클래스 : 승용차, 트럭, 버스)
- 다형성 : 객체가 취하는 동작이 상황에 따라서 달라지는 것  
(ex - 같은 speak() 라는 함수를 호출한다 하더라도, 어떠한 객체가 호출했느냐에 따라 결과가 달라진다.  
"개"의 경우 "멍멍", "고양이"의 경우 "야옹")

# 객체지향 언어

객체지향언어와 절차지향언어의 가장 뚜렷한 차이점은 클래스와 객체가 있느냐 없느냐입니다. 객체지향언어는 그 언어의 기본요소가 객체로 되어있습니다.

클래스는 객체를 생성하기위한 틀에 해당합니다. 그래서 클래스를 정의하는 것은 곧 객체를 설계하는 것입니다.

사실 객체라는 것은 여러가지 정보를 가진 정보의 덩어리에 불과합니다.

하지만 객체는 자신만의 전용 함수들을 갖고 있습니다. 이 것들을 멤버함수 또는 메서드라고 부르죠.

이 메서드는 객체가 가진 멤버변수들에 대해서는 따로 입력받지 않고도 접근할 수 있어서 주로 객체가 특정 입력을 받았을 때 알맞는 반응을 보이도록하는데에 사용합니다.

그리고 이런 식으로 객체를 설계하는 것이 프로그래밍 작업의 주요한 부분입니다. 그래서 객체지향언어라고 부릅니다.

하지만 절차지향언어로도 객체지향 프로그래밍을 하는 것이 가능합니다.

절차지향 언어의 대표격인 C언어는 구조체라는 도구를 제공해서 정보 덩어리를 생성할 수 있게 해줍니다.

또한 포인터를 제공하므로 구조체를 함수에 넘기기 쉽도록 해줍니다.

비록 언어차원에서 직접적인 지원은 못받지만

어떤 구조체가 자신만의 전용함수들을 가지고 있다면 객체지향 언어의 객체와 다를 바가 없습니다.



# Java

Java언어는 C/C++언어에서 문법적인 내용을 취해서 개발 진입장벽을 낮춘 언어

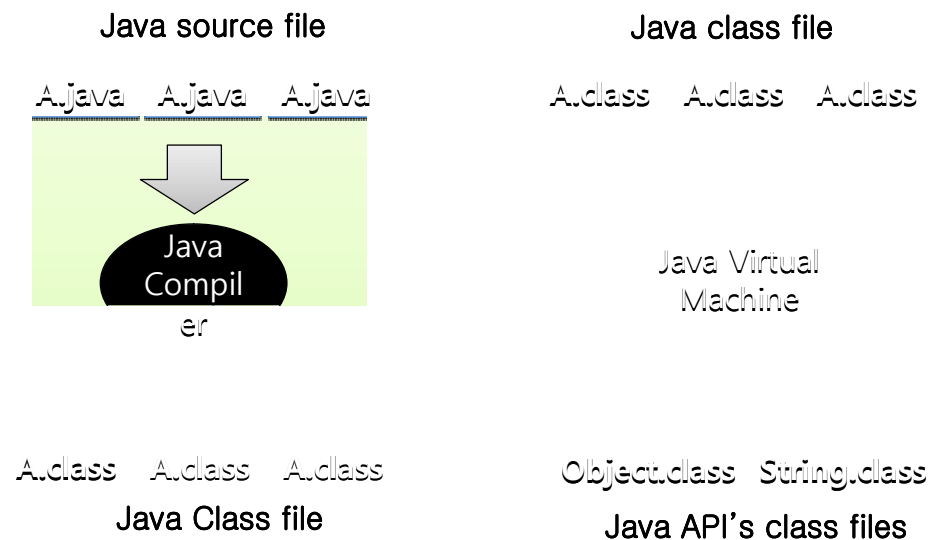
C/C++에서 개발자들이 혼란을 가질 만한 부분을 제거 – 메모리의 관리/ 운영체제에 대한 독립

객체지향 패러다임을 성공적으로 활용한 언어

Java Virtual Machine(JVM)이라는 독특한 장치를 가지고 있다.

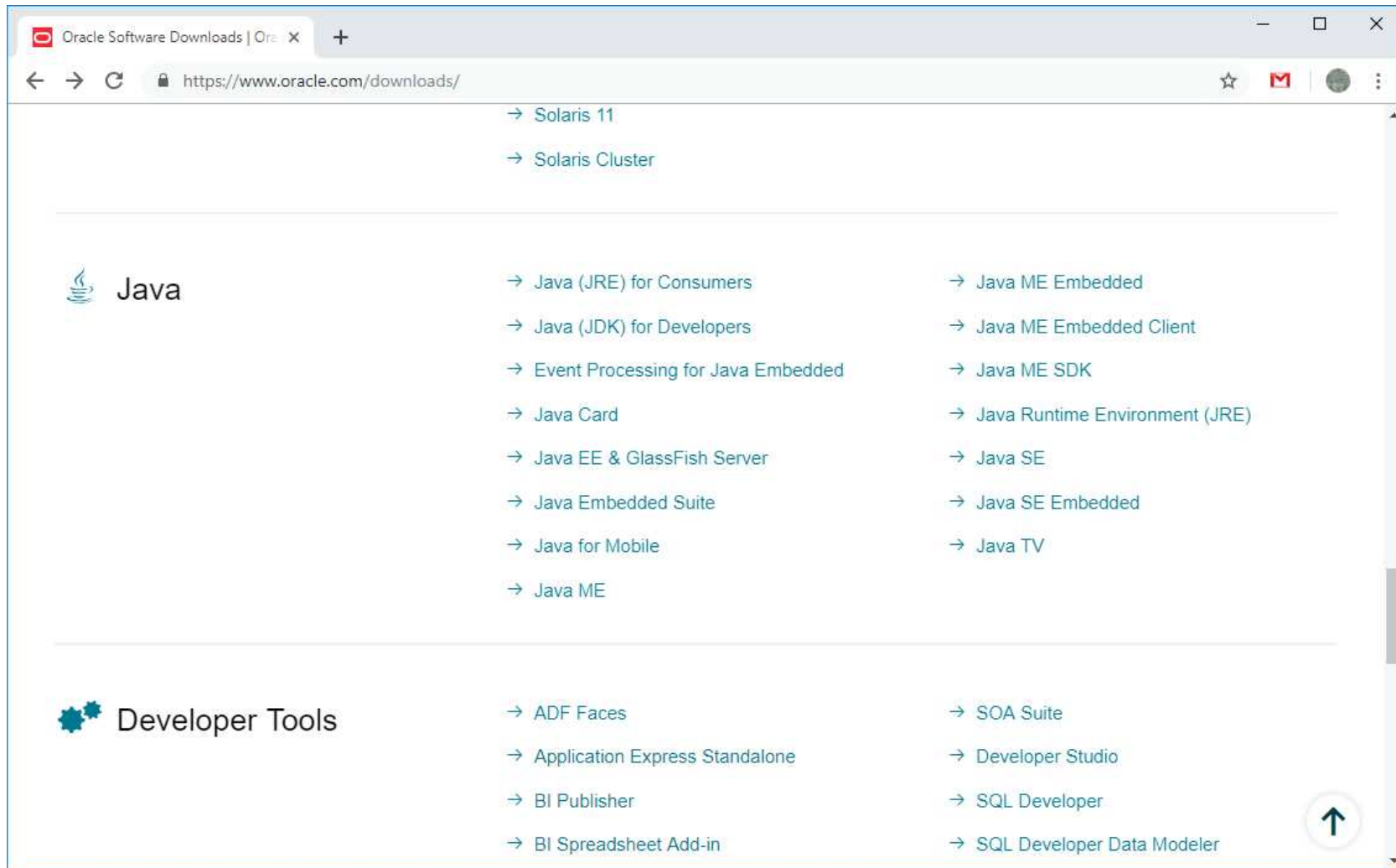
인간이 만든 코드를 JVM이 인식하도록 컴파일 과정을 거친다.

JVM이라는 해석기가 컴파일된 코드를 해석하면서 실행

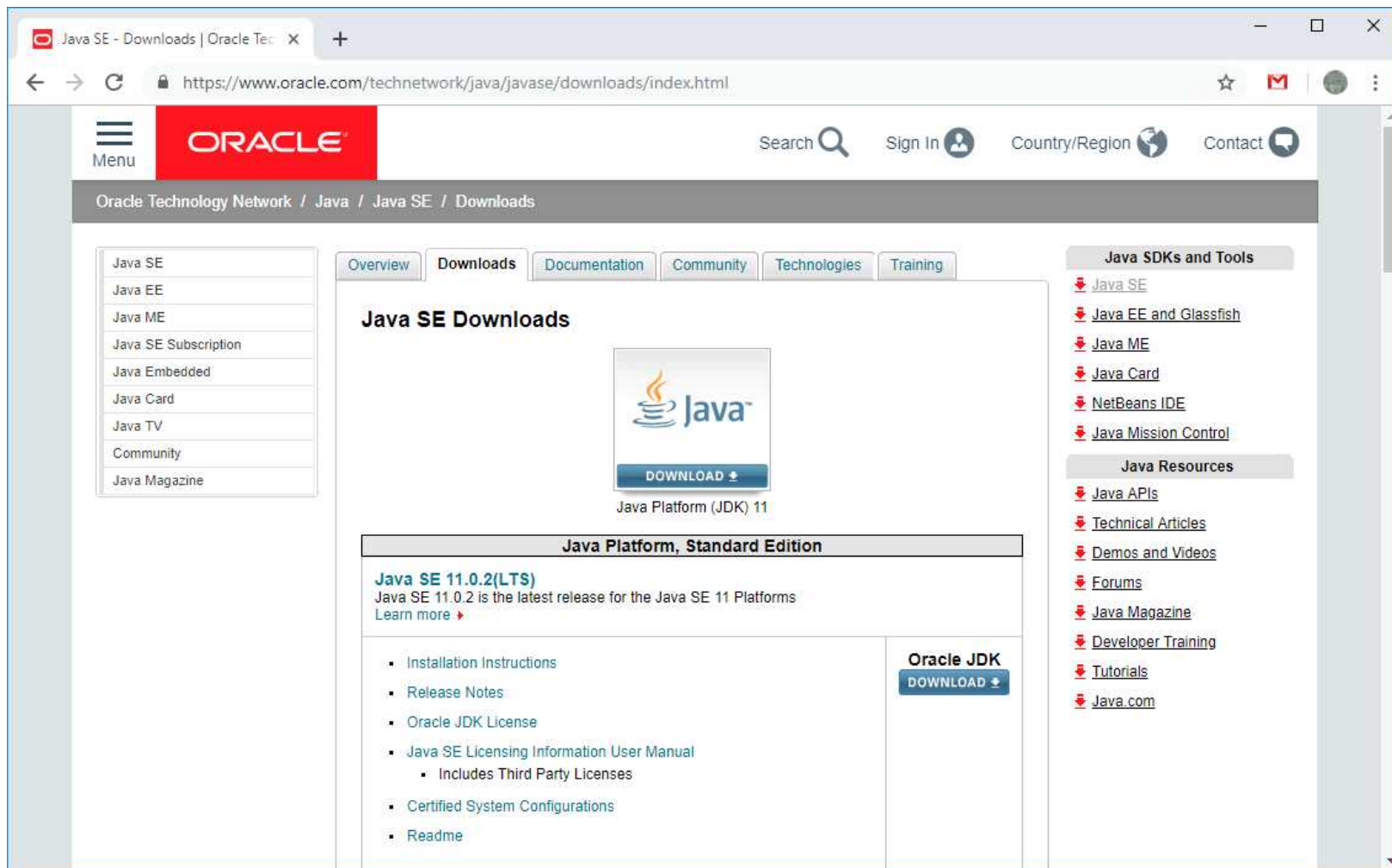


# Install JDK

<https://www.oracle.com/downloads/>



# Install JDK



The screenshot shows the Oracle Java SE Downloads page in a web browser. The browser's address bar displays the URL: <https://www.oracle.com/technetwork/java/javase/downloads/index.html>. The page features the Oracle logo and a navigation menu. The main content area is titled "Java SE Downloads" and includes a "DOWNLOAD" button for the Java Platform (JDK) 11. Below this, a section for "Java Platform, Standard Edition" highlights "Java SE 11.0.2(LTS)" as the latest release. A list of links for installation instructions, release notes, and licenses is provided. On the right side, there are sections for "Java SDKs and Tools" and "Java Resources" with various links.

Java SE - Downloads | Oracle Tec x


https://www.oracle.com/technetwork/java/javase/downloads/index.html

Menu ORACLE Search Sign In Country/Region Contact

Oracle Technology Network / Java / Java SE / Downloads

Overview Downloads Documentation Community Technologies Training

**Java SE Downloads**

  
DOWNLOAD

Java Platform (JDK) 11

**Java Platform, Standard Edition**

**Java SE 11.0.2(LTS)**  
Java SE 11.0.2 is the latest release for the Java SE 11 Platforms  
[Learn more](#)

- Installation Instructions
- Release Notes
- Oracle JDK License
- Java SE Licensing Information User Manual
  - Includes Third Party Licenses
- Certified System Configurations
- Readme

**Oracle JDK**  
DOWNLOAD

**Java SDKs and Tools**

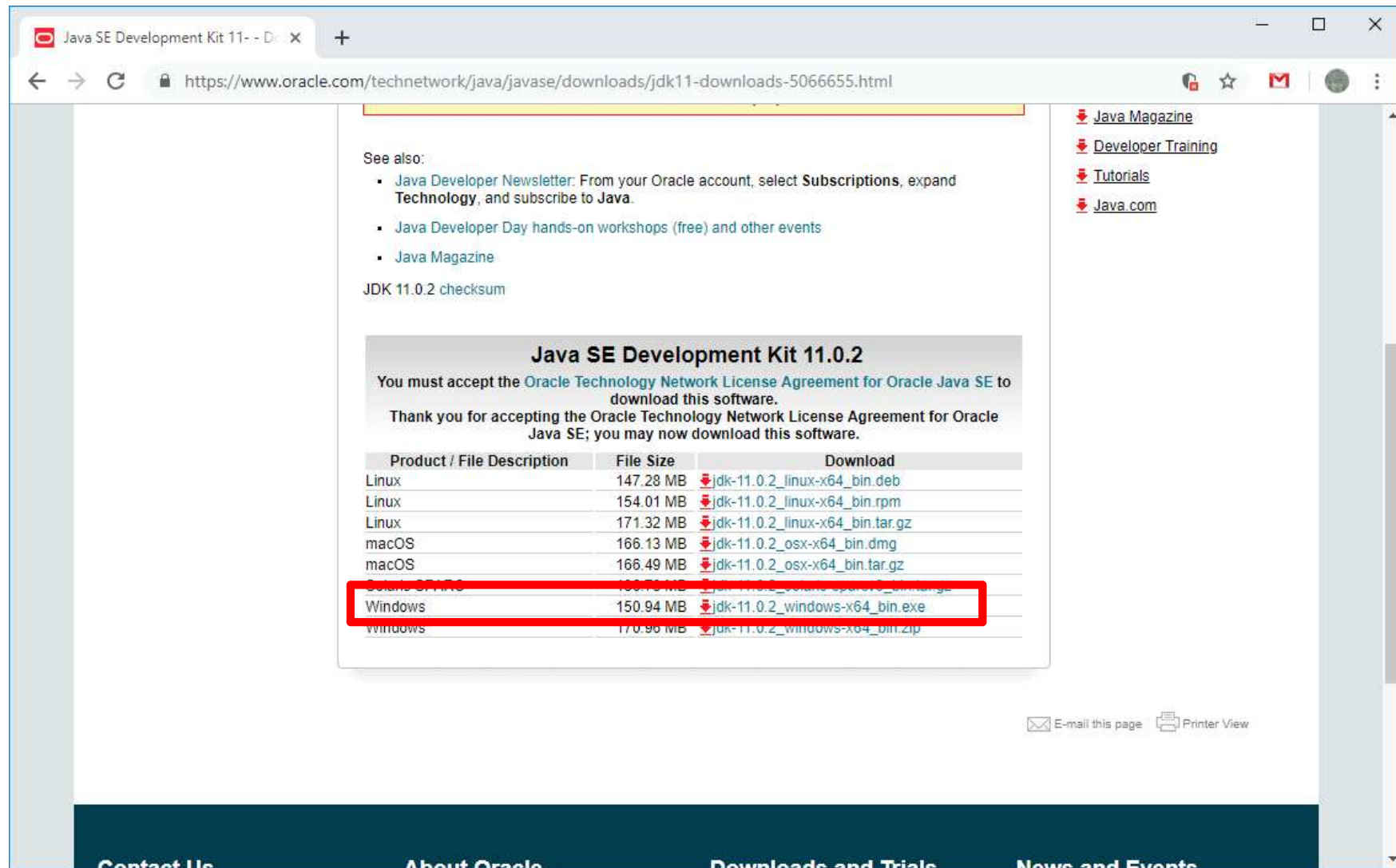
- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

**Java Resources**

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)

Java SE  
Java EE  
Java ME  
Java SE Subscription  
Java Embedded  
Java Card  
Java TV  
Community  
Java Magazine

# Install JDK



Java SE Development Kit 11 - - D x +

https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 11.0.2 checksum

**Java SE Development Kit 11.0.2**

You must accept the [Oracle Technology Network License Agreement for Oracle Java SE](#) to download this software.

Thank you for accepting the Oracle Technology Network License Agreement for Oracle Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux	147.28 MB	<a href="#">jdk-11.0.2_linux-x64_bin.deb</a>
Linux	154.01 MB	<a href="#">jdk-11.0.2_linux-x64_bin.rpm</a>
Linux	171.32 MB	<a href="#">jdk-11.0.2_linux-x64_bin.tar.gz</a>
macOS	166.13 MB	<a href="#">jdk-11.0.2_osx-x64_bin.dmg</a>
macOS	166.49 MB	<a href="#">jdk-11.0.2_osx-x64_bin.tar.gz</a>
Solaris SPARC	133.73 MB	<a href="#">jdk-11.0.2_solaris-sparc64_bin.tar.gz</a>
Windows	150.94 MB	<a href="#">jdk-11.0.2_windows-x64_bin.exe</a>
Windows	170.96 MB	<a href="#">jdk-11.0.2_windows-x64_bin.zip</a>

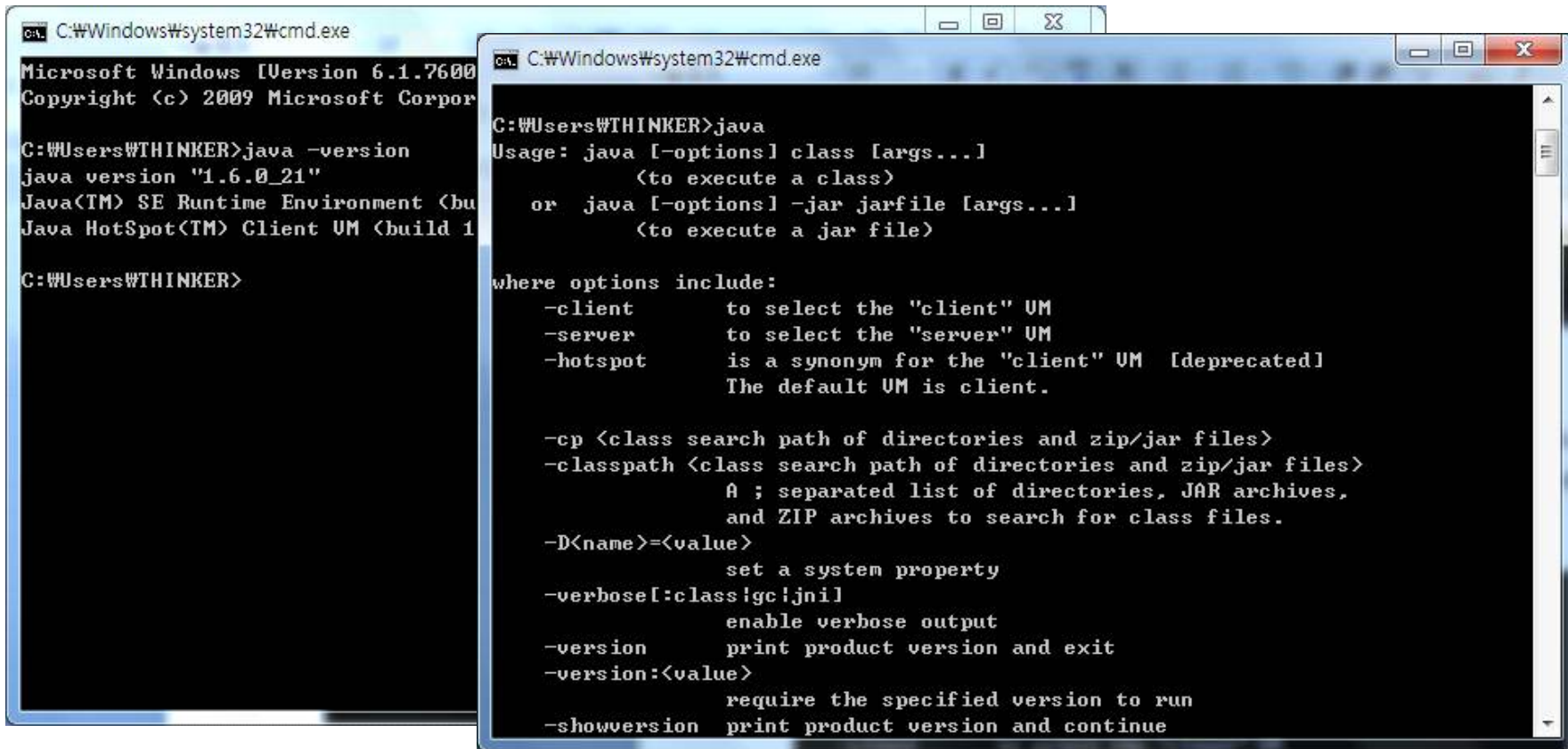
E-mail this page Printer View

Contact Us About Oracle Downloads and Trials News and Events

# JDK 설치 후 확인

java 명령어 실행

java -version 명령어 실행



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600.16385]
Copyright (c) 2009 Microsoft Corporation

C:\Users\THINKER>java -version
java version "1.6.0_21"
Java(TM) SE Runtime Environment (build 1.6.0_21-b05)
Java HotSpot(TM) Client VM (build 1.6.0_21-b05, mixed mode)

C:\Users\THINKER>
```

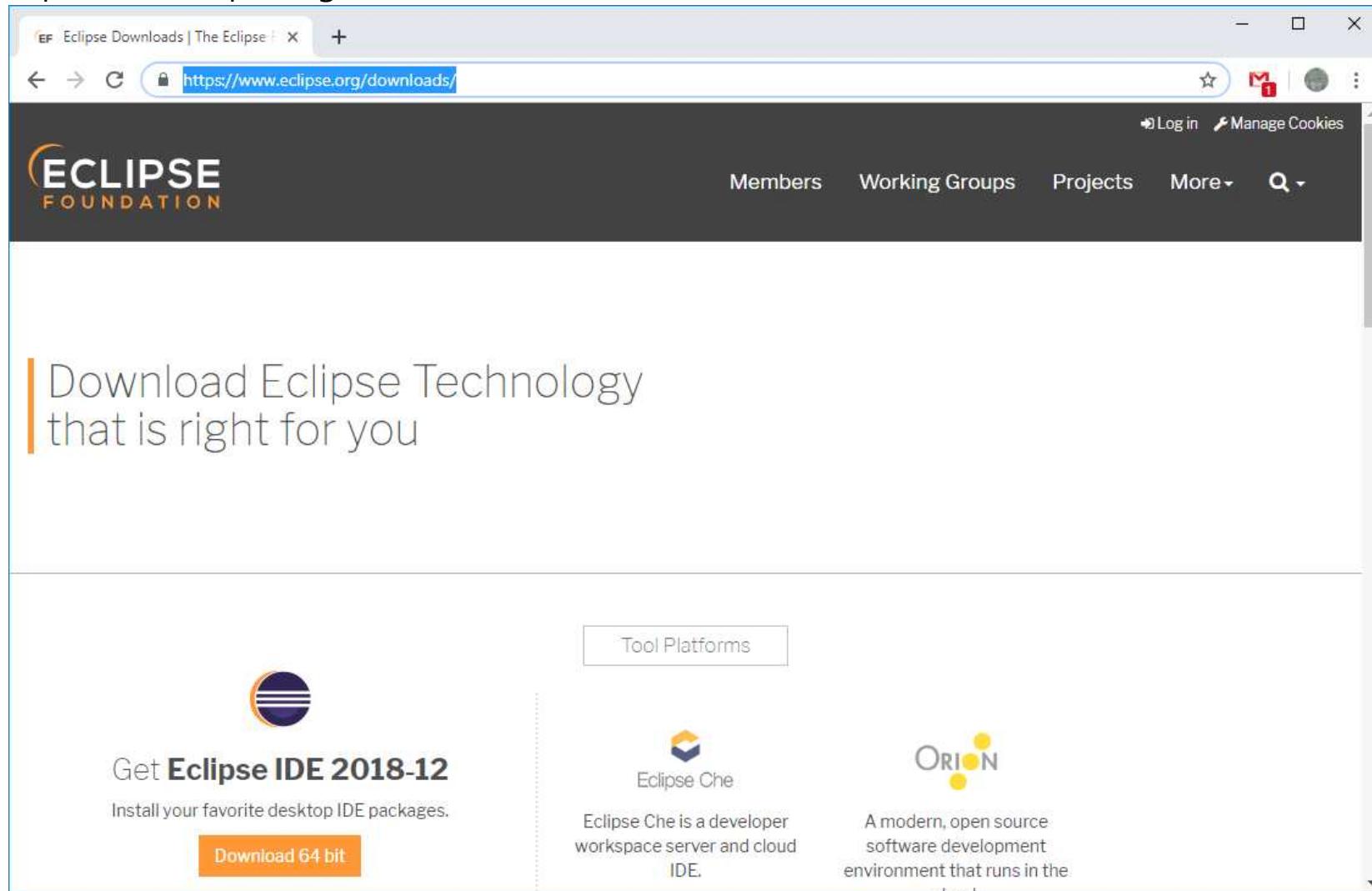
```
C:\Windows\system32\cmd.exe
C:\Users\THINKER>java
Usage: java [-options] class [args...]
           (to execute a class)
 or  java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
    -client      to select the "client" VM
    -server      to select the "server" VM
    -hotspot     is a synonym for the "client" VM [deprecated]
                 The default VM is client.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                 A ; separated list of directories, JAR archives,
                 and ZIP archives to search for class files.
    -D<name>=<value>
                 set a system property
    -verbose[:class!gc!jni]
                 enable verbose output
    -version      print product version and exit
    -version:<value>
                 require the specified version to run
    -showversion  print product version and continue
```

# 이클립스

<https://www.eclipse.org/downloads/>



# 이클립스

1. 프로젝트를 만듦
2. 패키지를 만듦
3. 클래스 파일을 만듦
4. 코딩

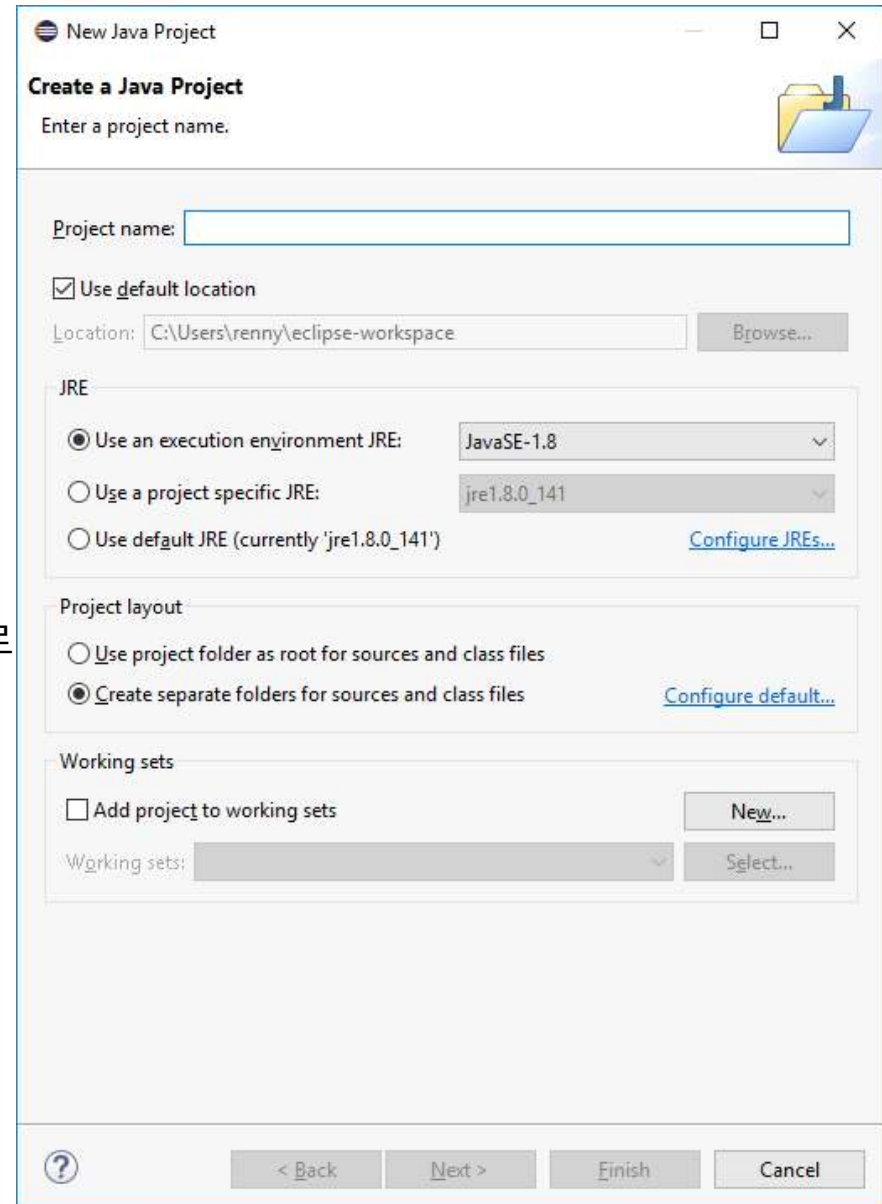
## -프로젝트 만들기

일단 좌측 상단에서 File -> New -> Java Project  
Alt + Shift + N

좌측 Package Explorer 에서  
우클릭 -> New -> Java Project

Project name : 프로젝트의 이름을 만드는 칸  
Use default location : 이 프로젝트가 저장될 경로  
JRE : 가상 컴파일러의 버전을 설정

Finish



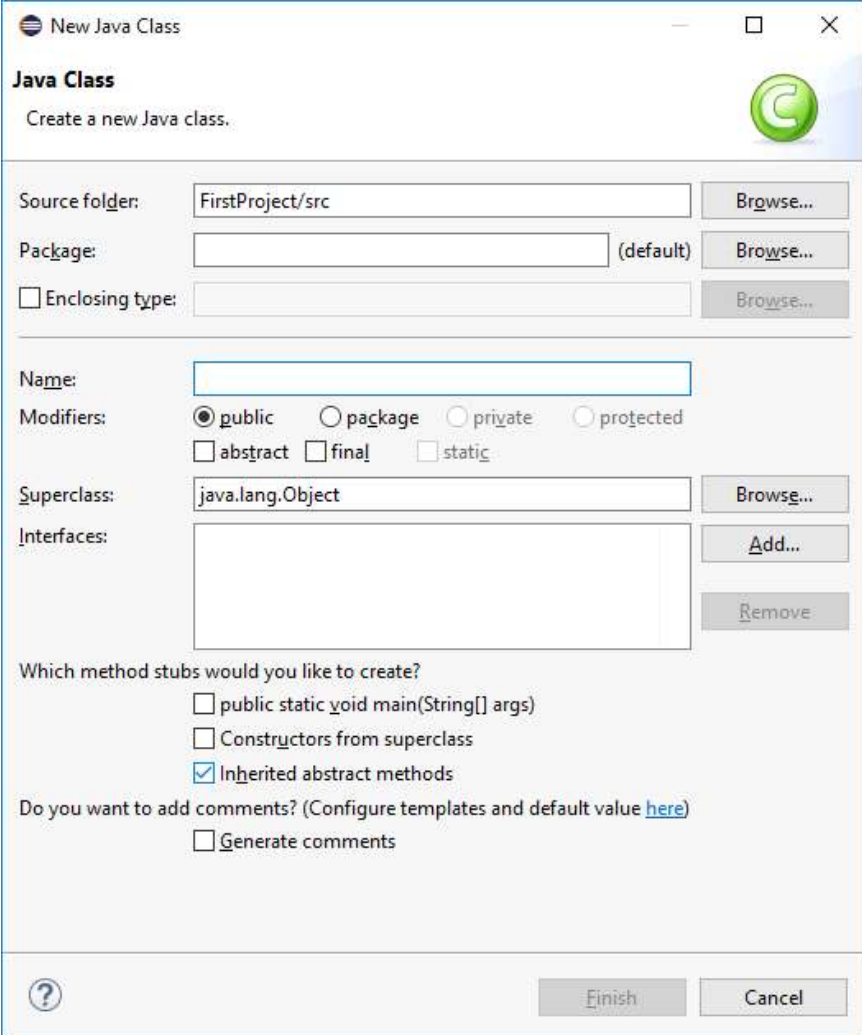
# 이클립스

왼쪽에 보시면 프로젝트가 생성된 것을 알 수 있습니다  
이제 클래스파일을 생성할 때 입니다

File - New - Class

왼쪽에서 src(source)를 우클릭 한 후 New - Class

Name만 설정 후 Finish



The screenshot shows the 'New Java Class' dialog box in the Eclipse IDE. The dialog has a title bar 'New Java Class' and a 'Java Class' header. Below the header, it says 'Create a new Java class.' and features the Eclipse logo. The dialog is divided into several sections: 'Source folder' with a text field containing 'FirstProject/src' and a 'Browse...' button; 'Package' with a text field and '(default)' and a 'Browse...' button; 'Enclosing type' with a text field and a 'Browse...' button. Below these is a 'Name' text field. The 'Modifiers' section includes radio buttons for 'public' (selected), 'package', 'private', and 'protected', and checkboxes for 'abstract', 'final', and 'static'. The 'Superclass' section has a text field containing 'java.lang.Object' and a 'Browse...' button. The 'Interfaces' section has a text area and 'Add...' and 'Remove' buttons. The 'Which method stubs would you like to create?' section has checkboxes for 'public static void main(String[] args)', 'Constructors from superclass', and 'Inherited abstract methods' (checked). The 'Do you want to add comments? (Configure templates and default value [here](#))' section has a checkbox for 'Generate comments'. At the bottom, there is a help icon, and 'Finish' and 'Cancel' buttons.

New Java Class

Java Class

Create a new Java class.

Source folder: FirstProject/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name:

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

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

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




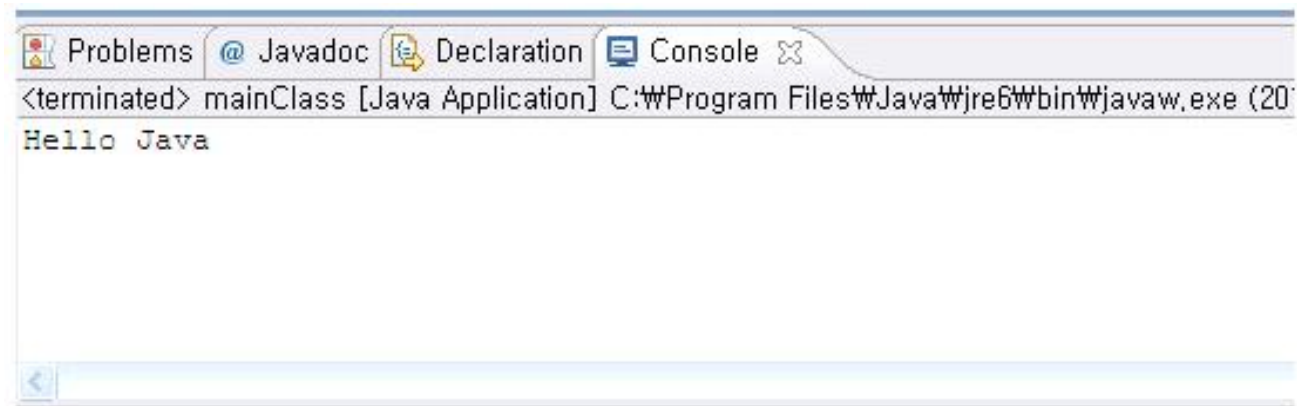
# 이클립스

모두 지운 후

```
class '클래스이름' {  
    public static void main(String []args){  
        System.out.println("Hello Java");  
    }  
}
```

\* 클래스 이름은 앞에서 설정한 이름 사용

 클릭 또는 F11을 눌러주세요



# 이클립스 - 단축키

Ctrl + 1 : 빠른 교정

문맥에 맞는 소스코드의 교정을 도와준다. 예를 들어 인터페이스를 구현하는 클래스에서 인터페이스에서 선언된 메소드를 구현하고 있지 않을 때, 빨간색 에러가 표시되는데 구현 클래스 명에서 이 단축키를 적용하면 인터페이스에 선언된 메소드의 원형이 구현 클래스에 추가된다.

Ctrl + Shift + O : import!!문 자동 생성

자동으로 import!!문을 추가해 준다. 그것도 패키지 유형의 분류에 따라. 예를 들어 java.net.URL 클래스를 사용하고자 어딘가에서 URL url = new URL("http://..."); 이런식으로 작성했을 경우, import!!문을 통해 URL이라는 클래스가 인식되지 않기 때문에 에러가 표시되지만 이 단축키를 적용하게 되면 자동으로 문맥에 맞는 import!!문이 삽입된다. 만약 동일한 이름의 클래스가 여러 개 있을 경우 적절한 패키지를 선택할 수 있는 대화상자가 나타난다.

Ctrl + Space bar : Content Assist

Alt + Shift + R : 이름변경(클래스, 패키지, 변수, 메소드 등등...)

나는 변수명이나 클래스명, 패키지명, 메소드명 등을 상당히 자주 바꾸는 타입이다. 문맥에 맞지 않거나 조금 의미가 약한 경우 여지없이 이 단축키를 눌러댄다. 거의 Content Assist 만큼이나 자주 쓰는 단축키. 자동으로 참조 정보까지 변경해 주기 때문에 너무 편리하다. ^^

Alt + Shift + V : 이동(클래스, 인터페이스 등..)

역시나 이름변경 단축키와 함께 자주 사용되는 단축키이다. 해당 자원을 원하는 곳으로 이동할 수 있게 해주며, 역시나 자동으로 참조 정보까지 변경해 준다.

# 이클립스 - 단축키

**Ctrl + Page Up / Ctrl + Page Down : 좌/우 창 이동**

코드 편집 창에 여러개의 창이 띄워져 있을 경우, 각 창 사이를 이 단축키로 왔다리 갔다리 할 수 있다.

**F3 : 선언 위치로 이동**

이 단축키를 눌러 변수의 경우에는 변수가 선언된 위치로, 메소드의 경우에는 메소드의 본문이 있는 곳으로 이동할 수 있다.

**Alt + ↑ / ↓ : 줄 위치 바꿈**

문장의 위/아래 위치를 바꾼다. 즉 한 라인을 위로 올리고 싶을 경우, Alt + ↑를 눌러 이동시킬 수 있다.

**Ctrl + D : 한줄 삭제**

현재 커서가 위치한 라인 전체를 삭제한다.

**Ctrl + Shift + Delete : 현재 위치에서 라인 끝까지 삭제**

현재 커서가 위치한 곳에서부터 그 라인의 마지막까지의 내용을 삭제한다.

# 이클립스 - 단축키

## Ctrl + J : Incremental Find

이 단축키를 누르게 되면, 이클립스 창의 하단 상태 표시줄 부분에 Incremental Find라고 표시되며, 한글자씩을 누를 때마다 그곳에 입력된 글자가 보이고 코드내의 일치하는 문자열로 이동하게 된다. 입력을 끝내고 다시 Ctrl + J를 누르면 그 문자열과 일치하는 곳으로 커서가 이동하게 된다.

## Alt + Shift + S : Source 메뉴

아래와 같은 source 메뉴의 메뉴가 쿼 메뉴로 나타난다.

Toggle Comment	Ctrl+/ 
Remove Block Comment	Ctrl+Shift+W
Generate Element Comment	Alt+Shift+J
Correct Indentation	Ctrl+I
Format	Ctrl+Shift+F
Add Import	Ctrl+Shift+M
Organize Imports	Ctrl+Shift+O
Sort Members...	
Clean Up...	
Override/Implement Methods...	
Generate Getters and Setters...	
Generate Delegate Methods...	
Generate hashCode() and equals()...	
Generate Constructor using Fields...	
Generate Constructors from Superclass...	
Externalize Strings...	

# 이클립스 - 단축키

**Ctrl + / : 주석처리 토글**

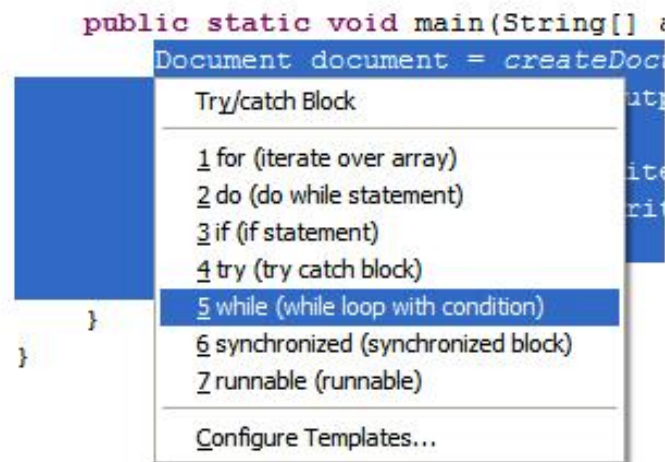
현재 라인을 주석처리 하거나 주석처리된 부분의 주석을 제거해 준다.

**'sysout' + (Ctrl + Space) : System.out.println() 문장 삽입**

출력결과 확인할 때 상당히 편리하다. ^^;;

**Alt + Shift + Z : Surround With 메뉴**

코드를 블록으로 선택한 다음 이 단축키를 적용하면 다음과 같은 쿼 메뉴가 나타나며 선택한 문장으로 블록으로 선택했던 문장들을 감싼다



# 이클립스 - 단축키

## **Ctrl + Shift + F : 코드 포매팅**

코드의 내용을 문법 템플릿에 맞게 포매팅 해준다. 자바 표준 코딩 스타일을 준수시켜 줄 때 좋다. ^^

## **Ctrl + F11 : 마지막 Run 구성 실행**

마지막으로 실행시켰던 Run 구성을 실행시킨다.

## **Ctrl + Alt + ↓ : 한줄 복사후 붙여넣기**

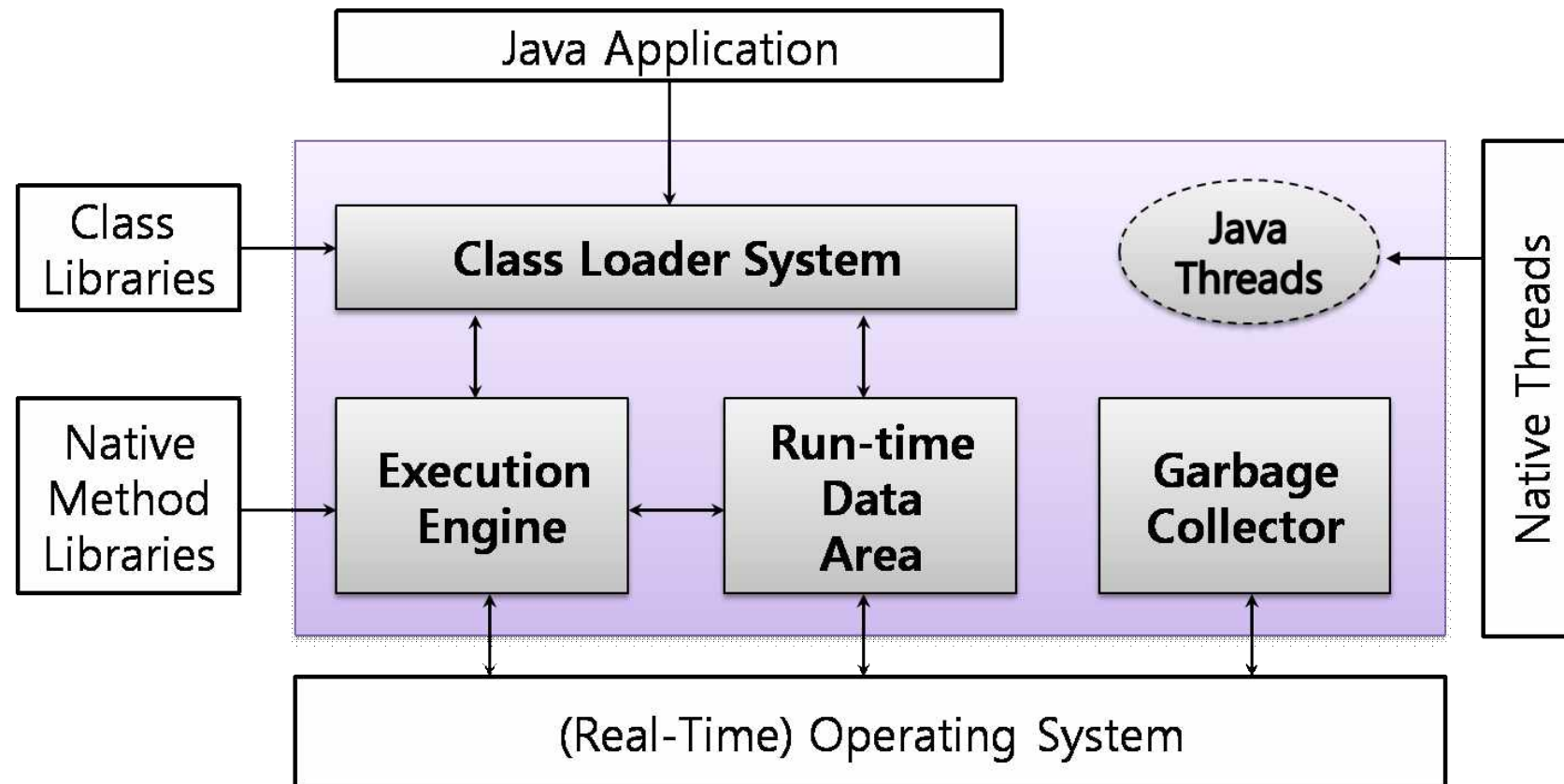
이 단축키를 적용하게 되면 현재 커서가 위치한 라인의 문장 전체를 복사해서 바로 아랫줄에 붙여넣는다. C&P 하던 것보다 간편하다.

## **Ctrl + Shift + X : 대문자로 / Ctrl + Shift + Y : 소문자로**

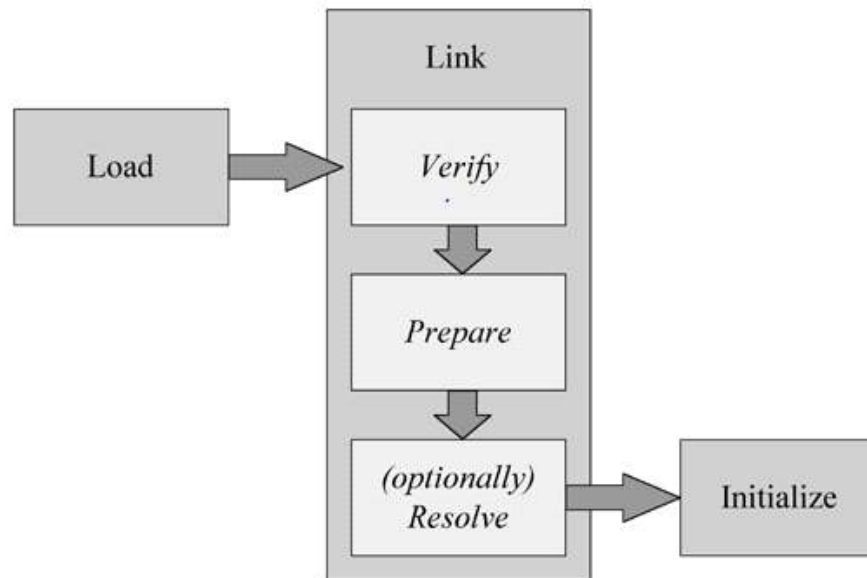
### **\* 이클립스 기본 설정**

- ▶ Windows - Preferences - Java - Editor - Syntax Coloring - java==>
  - Keyword 'return' , Keywords excluding 'return' = (blue)
  - Classes - (red)
- ▶ Windows - Preferences - General - Appearance - Colors and Fonts - Basic - Text Font ->Tahoma,Verdana, 굵게, 크기12로 조정

# JVM의 구성



# Class로더의 실행



실행코드를 인스트럭션으로 변환하고 해석해서 실행

일반 Interpreter

JIT(Just – In – Time)

- 반복 수행되는 코드를 기존 해석한 내용을 활용하는 방식(성능향상)

Hotspot

- 성능의 저하를 가져오는 부분을 프로파일링해서 집중처리



# 기타 특징

## Garbage Collector

메모리의 회수를 담당

Heap영역에서 동작(Heap이란 메모리를 동적으로 사용하는 공간)  
사용되지 않는 메모리를 판단해서 자체적으로 삭제하는 매커니즘  
JDK의 major 버전을 결정하는 가장 큰 요인

## 환경변수의 설정

JDK 버전이 낮았을 때는 필수적인 과정

JDK1.4 이후에는 기본적으로 java 명령어는 사용가능  
javac 와 같은 개발자에게 필요한 기능을 위해서 세팅  
이클립스를 사용하는 경우에는 별도의 설정은 옵션

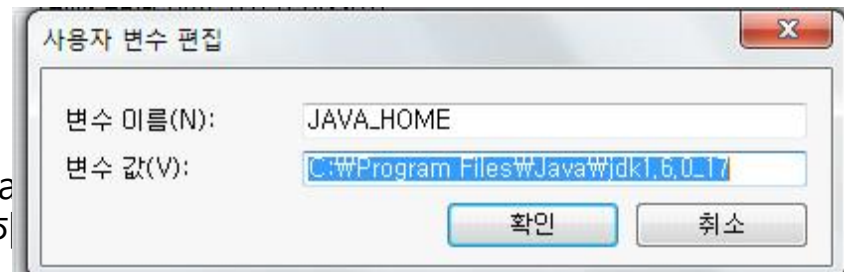
## 환경변수에 나오는 용어

JAVA\_HOME

- Java가 설치된 폴더를 의미하는 가명(a)
- Java 관련 시스템들이 공통으로 참조하

PATH

- 운영체제에서 별도의 경로 없이 사용하기 위한 설정



# 변수

데이터를 담는 상자의 의미

데이터를 가리키는 지시대명사의 의미 (포인터)

메모리상 공간의 의미

종합: 메모리상에 데이터를 보관하는 메모리의 조각

메모리상에 데이터를 보관할 수 있는 공간을 확보(bit/byte)

적절한 메모리 공간을 확보하기 위해서 변수의 타입이라는 것이 등장

‘=’ 을 통해서 CPU에게 연산작업을 의뢰

## 변수의 선언 방식

변수의 타입에 대해 느슨한 언어

```
var a = 10;
```

```
var a = "aaaa";
```

```
a = "123";
```

- 변수의 타입에 느슨하면 개발시점에는 유리하지만 실행시에는 불리한 방식

변수의 타입에 대해 엄격한 언어

- 정확한 변수의 타입을 명시해서 메모리의 공간을 효과적으로 사용

- 다른 타입의 데이터로 변환할 때에는 별도의 처리

# 상수

누구나 같은 의미로 해석해야 하는 데이터  
문자  
숫자  
리터럴 역시 메모리상에 공간을 차지한다.

```
public static void main<java.lang.String[]>(  
  Code:  
    Stack=2, Locals=2, Args_size=1  
    0:  bipush  10  
    2:  istore_1  
    3:  getstatic      #16; //Field java/lang/System.out:Ljava/io/PrintStream;  
    6:  iload_1  
    7:  invokevirtual  #22; //Method java/io/PrintStream.println:<I>V  
   10:  return  
LineNumberTable:  
  line 6: 0  
  line 8: 3  
  line 10: 10  
  
LocalVariableTable:  
  Start  Length  Slot  Name   Signature  
  0       11      0    args   [Ljava/lang/String;  
  3        8      1     i      I
```

# 기본 자료형

정수를 표현할 때는 int를 사용한다.

소수를 표현할 때는 double을 쓴다.

참, 거짓(불 타입)인 경우에는 boolean으로 쓴다.

글자 하나를 의미할 때는 char를 쓴다.

**int value = 10;**

메모리에 정수가 들어갈 상자를 만들어라

상자의 이름은 value이다.

value라는 상자에 10을 넣어라

```
public class VarEx {  
    public static void main(String[] args) {  
        //정수를 표현할 때에는 int라는 타입을 쓴다.  
        int i = 100;  
        //소수를 표현할 때에는 double이라는 타입을 쓴다.  
        double d = 10.5555;  
        //참 거짓을 표현할 때에는 boolean을 쓴다.  
        boolean b = true;  
        //문자 하나를 의미할 때에는 char를 쓴다.  
        char c = '가';  
  
        System.out.println(i);  
        System.out.println(d);  
        System.out.println(b);  
        System.out.println(c);  
    }  
}  
  
100  
10.5555  
true  
가
```

# 기본 자료형

```
public class VarEx {  
    public static void main(String[] args) {  
        // 정수를 표현할 때에는 int라는 타입을 쓴다.  
        int i = 100;  
        // 소수를 표현할 때에는 double이라는 타입을 쓴다.  
        double d = 10.5555;  
        // 참 거짓을 표현할 때에는 boolean을 쓴다.  
        boolean b = true;  
        // 문자 하나를 의미할 때에는 char를 쓴다.  
        char c = '가';  
  
        System.out.println(i);  
        System.out.println(d);  
        System.out.println(b);  
        System.out.println(c);  
    }  
}
```

100  
10.5555  
true  
가

주의) 자바에서는 참/거짓 표현시 true/false를 사용합니다. (c처럼 1/0이 아님)

특수문자 사용방법

char c='t' 탭간격, \n 줄바꿈, \b 백스페이스, \r 캐리지리턴, \' 작은따옴표, \" 큰따옴표

# 변수

변수의 타입마다 다른 크기를 사용하기 때문에 메모리 공간을 필요한 만큼 사용 가능

상자의 크기마다 표현 가능한 최대값과 최소값

- byte a = 130; (컴파일 에러)

부호비트를 이용한 음수/양수 처리

상자의 크기 + 상자의 처리방식을 Type(타입) 정보로 표현

이름	종류	크기
char	문자	2 Byte
short	정수	2 Byte
int	정수	4 Byte
double	실수	8 Byte
float	실수	4 Byte
long	정수	8 Byte
boolean	참과 거짓( true, false) true = 1, false = 0	0 or 1



# 변수

숙제) 프로그램을 이용해서 다음을 표현하려고 합니다.

화면에 숫자를 입력해 주세요!

13

당신이 입력한 숫자는 13

프로그램 구성시 필요한 부분

1. 키보드로 입력하는 글자를 입력받기  
- `Scanner s = new Scanner(System.in);`

2. 화면에 출력하기  
- `System.out.println(s.nextInt());`

3. Scanner를 사용하기 위해서 코드 제일 첫줄에 (class바깥 부분)  
`import java.util.Scanner;`  
를 적어줍니다.

# 형 변환

## 묵시적(Automatic) 형 변환

형 변환이 일어나도 기존의 데이터에는 별다른 해가 없기 때문에 개발자들에게 알려주지 않는 변환

작은 상자에서 큰 상자로  
byte -> short    int -> long

데이터를 다른 방식으로 보관하지만 유지하는 경우  
long(8byte) -> float

## 명시적 형 변환

기존의 데이터를 다른 타입으로 쓰거나 데이터가 훼손될 가능성이 있는 경우에 개발자가 반드시 인식할 수 있게 하는 장치

큰 상자의 데이터를 작은 상자로  
int ( 4byte) -> byte (1 byte)

동일한 상자크기라고 해도 변수의 용도가 변하는 경우  
short(2 bytes) -> char(2bytes)



# 형 변환

명시적 형 변환시 데이터가 잘릴 수 있습니다.

```
public class DownCastingEx {  
    public static void main(String[] args) {  
        int xx = 128;  
        byte yy = (byte)xx;  
        System.out.println(yy);  
    }  
}
```

-128 ← 128이라는 데이터가 변질됩니다.

작은 공간으로 옮기고 나니 2의 7승이지만 앞자리가 1이므로 음수가 됩니다.

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0
-128 +	0 +	0 +	0 +	0 +	0 +	0 +	0

# 산술 연산자

## '=' : 대입 연산자

- 변수에 값을 대입

ex) `int l = 534;` 'l' 라는 이름을 가진 int 형 변수의 메모리값에 534라는 int 형 정수값을 대입한다

## 다항 연산자

- + , - , \* , % , /

'+' 덧셈 연산자 : 더하기 연산자입니다

'-' 뺄셈 연산자 : 빼기 연산자입니다

'/' 나누기 연산자 : 나누기 연산자입니다

'\*' 곱셈 연산자 : 곱하기 연산자입니다

'%' 나머지 연산자 : 왼쪽 값 나누기 오른쪽 값의 나머지를 반환합니다

'+' 와 '-' 는 부호 연산자로도 쓰입니다

-- , ++ , += , \*= , %= , /= , -=

1, '--' 과 '++'

변수에 쓰이는 연산자, 반복문에서 유용하게 사용

'--' 는 변수가 가진 값을 1 감소

'++' 는 변수가 가진 값을 1 증가

# 산술 연산자

변수의 앞과 뒤에 쓰일 수 있는데 ex) ++val; val++;  
- '++/--' 기호가 앞에 있으면 변수가 사용되기 전에 증가한다.  
- '++/--'가 뒤에 있으면 변수가 사용된 후에 증가한다.

간단히 보면

```
int num = 0;  
int sum1 = num++ ;    //지금은 0  
int sum2 = num;        //이때는 1
```

2. ' += ', ' \*= ', ' %= ', ' /= '

이연산자들은 각각

a = a + b ( a += b 와 같음)

a = a - b ( a -= b 와 같음)

a = a \* b ( a \*= b 와 같음)

a = a / b ( a /= b 와 같음)

a = a %b ( a %=b 와 같음)

## 주의사항

'%' 나머지 기호 주의

'/' 연산시에는 항상 실수가 나올 수 있는 상황을 주의

- int와 int의 연산은 int로 처리됨

- Java에서 정수는 int로 간주됨

연산자에는 우선 순위가 있다 : 우선 순위 보다는 가독성을 위해서 '(' )'를 이용하라.

연산 작업시에 두 변수의 타입에 주의하라 : 정수와 정수의 연산 = 정수, 정수와 실수의

연산 = 실수

# 산술 연산자

두개의 변수로 간단한 연산을 하는 코드

```
public class VarTest2 {  
  
    public static void main(String[] args) {  
  
        int a = 100;  
        int b = 20;  
  
        System.out.println( a + b);  
        System.out.println( a - b);  
        System.out.println( a * b);  
        System.out.println( a / b);  
    }  
}
```

120  
80  
2000  
5

# 산술 연산자

숙제) 임의의 값을 발생하여 홀수 짝수를 판단하는 프로그램 만들기  
- 나머지값을 출력하여 홀수인지 짝수인지 판단

임의의 값은: 64

나머지 값은: 0

필요한 내용

1. 임의의 값을 생성하는 방법

```
import java.util.Random;
public class RandomEx {
    public static void main(String[] args) {
        Random r = new Random();
        int randomValue = r.nextInt(100);
        System.out.println(randomValue);
    }
}
```

import로 해당 메소드를 사용할 수 있게 한다.

nextInt(100)은 0~100미만의 임의의 정수를 발생시켜줌