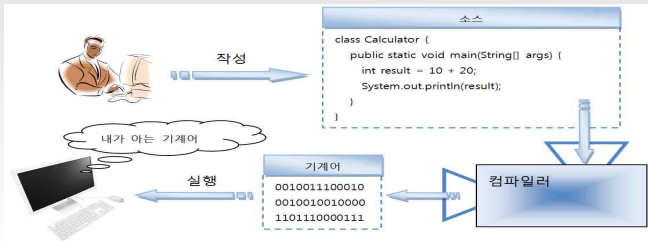


# 제1장 자바와 환경설정



# 1. 프로그래밍 언어의 역할은?

- 사람과 컴퓨터의 대화 도움
- 사람의 언어와 기계어 사이에서 다리와 같은 역할
- 고급 언어와 저급 언어로 구분

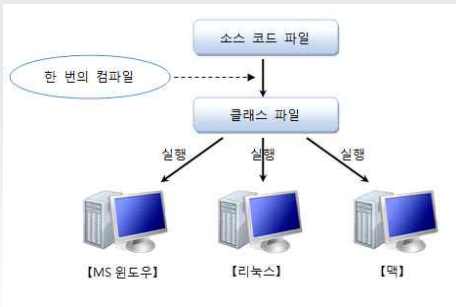


## 2. 자바가 무엇일까?

- 자바는 1991년에 썬 마이크로시스템즈(Sun Microsystems)사에서 **제임스 고슬링(James Gosling)**이 다른 연구원들과 함께 고안한 오크(Oak)라는 언어에서 시작.
- 1995년에 자바 1.0을 발표함.
- 1995년~1999년까지는 윈도우 프로그램 개발이 주류여서 C++언어에 비해 자바는 열세였다. 또, 메모리 및 CPU를 지나치게 많이 사용한다는 문제점이 존재함.
- 하지만, 1999년도부터 인터넷이 활성화되면서 웹 애플리케이션 구축을 위해 자바가 활성화되었다. 단, 한번 작성으로 모든 곳에서 실행 가능한 언어는 자바 뿐이었기 때문이다.
- 2010년 오라클에서 썬을 합병하여 Java 개발, 관리, 배포 주관하고 있음.

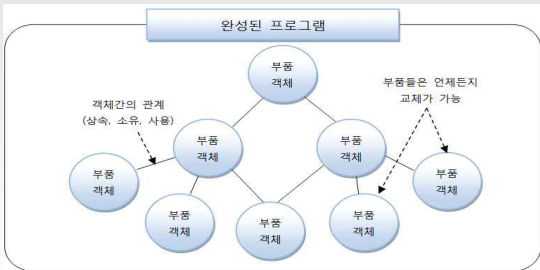
### 3.자바의 특징은?

- 자바는 이식성이 매우 높은 언어이다.



### 3.자바의 특징은?

- 객체 지향 언어(Object Oriented Programming)이다.
  - 작은 파일부터 만들고, 조합해 전체 프로그램을 완성하는 기법
- 자바는 캡슐화, 상속, 다형성 기능을 완벽하게 지원함.



### 3.자바의 특징은?

- 메모리의 자동 관리 기능이 강력해졌다.
  - 사용하지 않는 객체를 자동으로 메모리에서 제거
  - 핵심 기능 코드에 집중할 수 있도록 하는 기능
  - C, C++은 메모리를 개발자가 직접 조작을 해야 함.
- 함수적 스타일 코딩 지원
  - 함수적 스타일 프로그램 방식인 람다식(Lambda Expression)을 지원함(JDK1.8)
  - 람다식으로 인하여 코드가 매우 간결하게 작성이 가능해졌음.
  - 컬렉션 요소를 필터링,매핑,그룹핑,집계 처리 등 저용량 코드로 고성능을 냄.
- 웹, 앱 등 다양한 애플리케이션 프로그램의 개발 가능
- 현재도 계속 개발되어지고 있으며 JDK 1.15까지 발표함.

### 3.자바의 특징은?

- 멀티스레드(Multi-Thread) 기능 구현이 쉬워짐
  - 대용량 작업을 빨리 처리할 때
  - 동시에 여러 기능을 처리해야 할 때
- 동적 로딩(Dynamic Loading) 지원
  - 미리 객체를 만들어 놓지 않고 필요한 시점에 동적 로딩해 객체 생성
  - 유지 보수 시 특정 객체만 쉽게 수정 및 교체해 사용
- 풍부하고 엄청난 오픈소스 라이브러리를 제공함

## 4.자바 개발 환경 구축

### ● 자바 개발 도구(JDK) 설치

#### ● Java Standard Edition 구현체의 종류

- JDK (Java Development Kit) = JRE + 개발 도구  
(자바 프로그램 개발하고 실행하기 위해 반드시 설치해야 함)
- JRE (Java Runtime Environment) = JVM + 표준 클래스 라이브러리  
(자바 프로그램을 실행만 할 경우 설치함)

#### ● JDK (Java Development Kit) 설치

- 설치 파일 다운로드 사이트: <http://www.oracle.com>
- 유지 보수 시 특정 객체만 쉽게 수정 및 교체해 사용

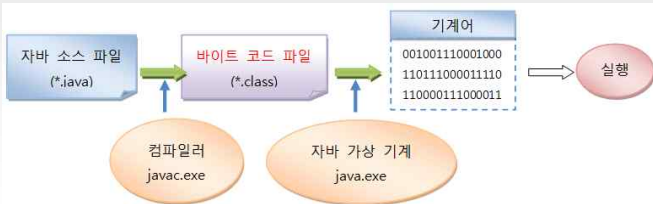
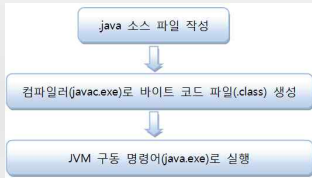
#### ● 자바 닥스(java docs) 문서

- JDK에서 제공하는 표준 클래스 라이브러리 설명해 놓은 HTML 페이지 집합
- <https://docs.oracle.com/en/java/javase/11/docs/api/index.html>



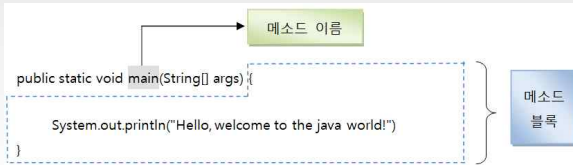
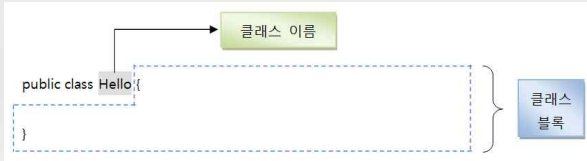
## 5. 자바로 하는 프로그램 개발의 순서

### 소스 작성부터 실행까지



## 5. 자바로 하는 프로그램 개발의 순서

### ○ 프로그램 소스의 분석



## 6. 주석문과 실행문

### ○ 주석(comments) 사용하기

- 프로그램에는 영향을 끼치지 않고, 코드에 설명을 붙여 주는 것.
- 컴파일 과정에는 주석은 무시되며, 실행문들만 바이트 코드로 변환됨.

### ○ 주석문의 종류

- // 주석 : 한 줄만 주석 처리함(행 주석)
- /\* ~ \*/ : /\*와 \*/ 사이에 있는 모든 범위를 주석으로 처리한다.(범위 주석)

### ○ 실행문과 세미콜론(;)

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

```
int x = 1;           //변수 x를 선언하고 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); //콘솔에 출력하는 메소드 호출
```

## 7.개발 플랫폼 설치(eclipse)

- 이클립스(Eclipse) 소개
  - 2003년 IBM에서 개발
  - 자바 통합 개발 환경(IDE: Integrated Development Environments) 제공(프로젝트 생성, 자동 코드 완성, 디버깅 기능 등 다양한 기능 제공)
- 지속적으로 버전업과 배포가 이루어짐.
- 다양한 개발 환경을 구축할 수 있도록 플러그 인 프로그램 설치 가능함.
- 안드로이드 개발, 스프링, JSP, C++ 등 통합개발 환경 구축 가능
- 이클립스는 자바로 개발된 도구임.
- 다운로드 사이트 : <http://www.eclipse.org>

## 7.개발 플랫폼 설치(Work-Space)

- 워크 스페이스란?
  - 이클립스에서 생성한 프로젝트가 기본적으로 저장되는 폴더를 의미함
- 최초로 실행 시 워크스페이스 런처에서 설정 가능함.- 상황별로 다르게 지정가능.
- .metadata 폴더
  - 자동적으로 생성이 되며, 이클립스 실행 시 메타데이터 저장함.
  - 이 폴더를 삭제하면, 이클립스 재 실행 시 초기 상태로 됨.

## 8. Java 설치



**감사합니다.**

