



# 1장. 자바 시작하기

이것이 자바다 (<http://cafe.naver.com/thisjava>)

## ❖ 목차

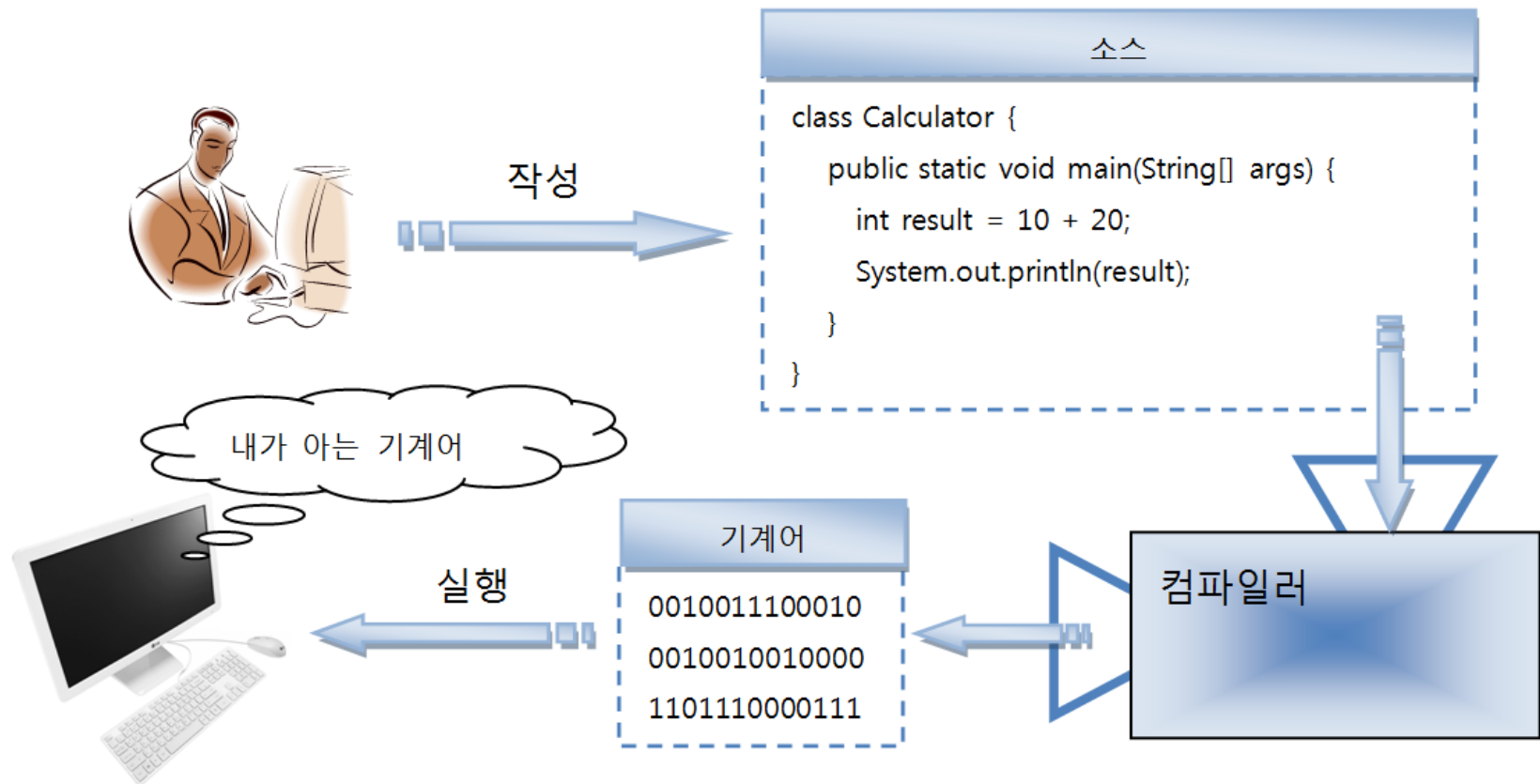
- 1절. 프로그래밍 언어란?
- 2절. 자바란?
- 3절. 자바 개발 환경 구축
- 4절. 자바 프로그램 개발 순서
- 5절. 주석과 실행문
- 6절. 이클립스 설치
- 7절. 풀인원 설치와 실행



# 1절. 프로그래밍 언어란?

## ❖ 프로그래밍 언어의 역할은?

- 사람과 컴퓨터의 대화 도움
- 사람의 언어와 기계어 사이에서 다리와 같은 역할
- 고급 언어와 저급 언어로 구분 (p.2)



## 2절. 자바란?

### ❖ 1.2.1 자바 소개

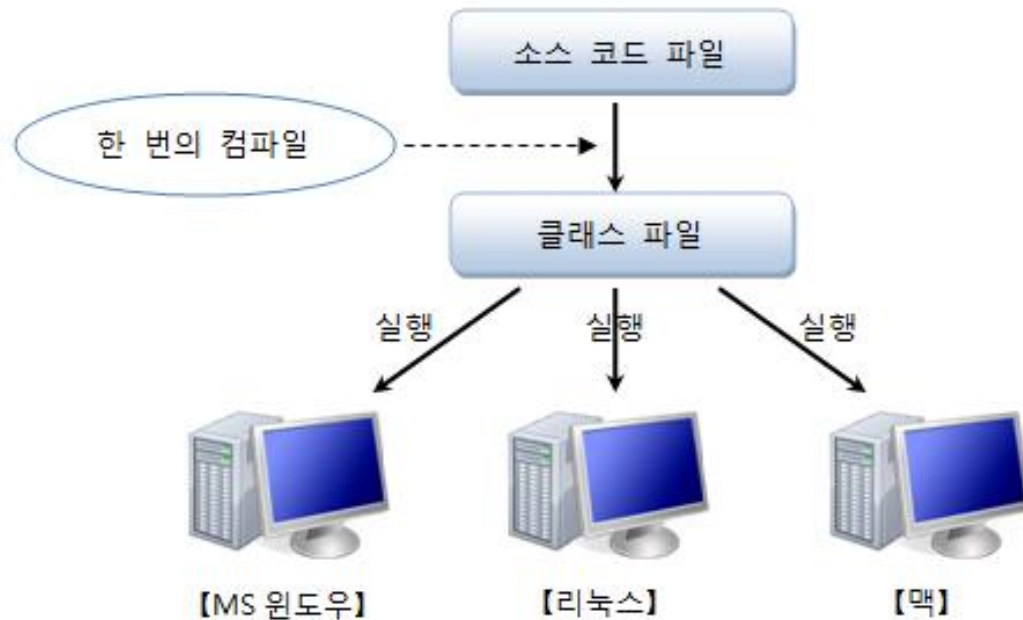
- 1995년 썬마이크로시스템즈(Sun Microsystems)에서 최초 발표한 언어
- 1991년 가전 제품에서 사용할 목적인 오크(Oak) 언어에서부터 시작
- 인터넷 프로그래밍 언어로 발전하면서 자바라는 이름으로 변경
- 2010년 오라클에서 썬을 인수하여 Java 개발,관리,배포 주관



## 2절. 자바란?

### ❖ 자바의 특징

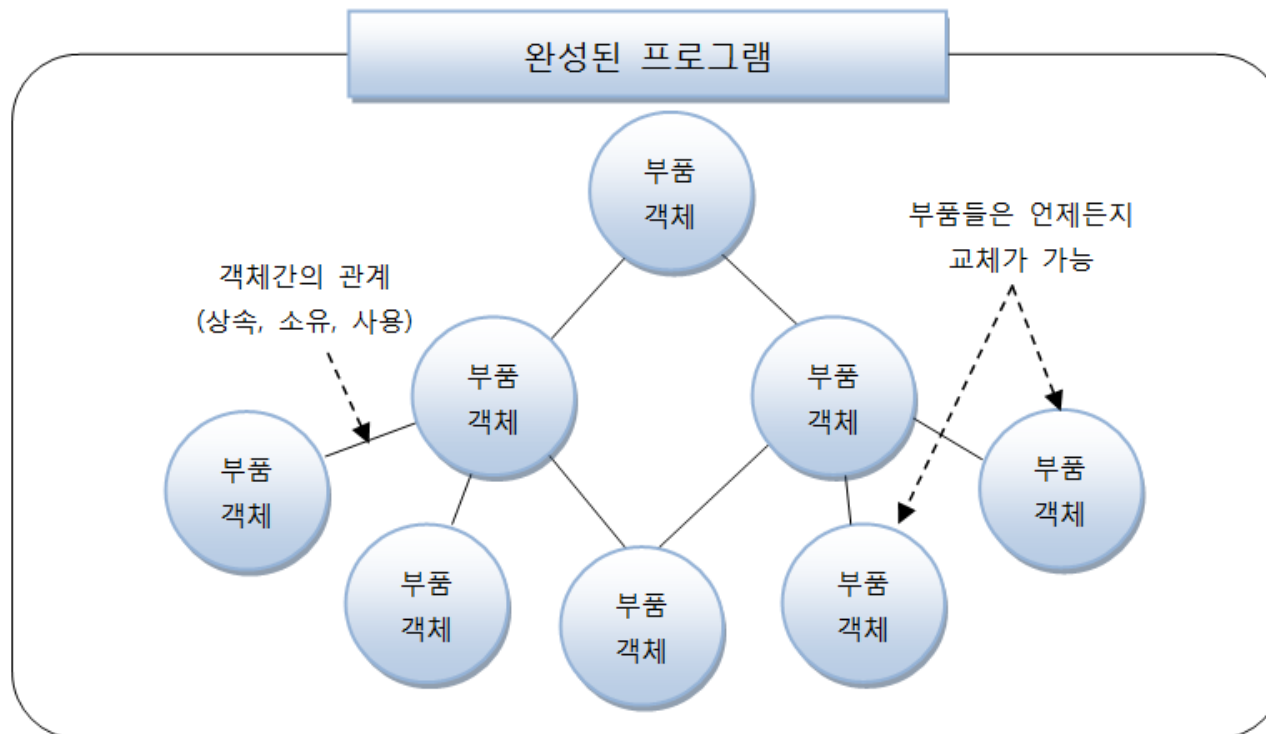
- 이식성이 높은 언어



## 2절. 자바란?

### ❖ 자바의 특징

- 객체 지향 언어 -> OOP(Object Oriented Programming)란?
  - 부품 객체를 먼저 만들고, 이것들을 조합해 전체 프로그램을 완성하는 기법
- 자바는 처음부터 OOP 개발용 언어로 설계
  - 캡슐화, 상속, 다형성 기능 완벽하게 지원



## 2절. 자바란?

### ❖ 자바의 특징

- 함수적 스타일 코딩 지원
  - 함수적 스타일 코딩 방식인 람다식(Lambda Expressions) 지원
    - 코드 간결하게 작성 가능
    - 컬렉션 요소를 필터링, 매핑, 그룹핑, 집계 처리시 주로 사용
- 메모리를 자동으로 관리
  - 사용하지 않는 객체를 자동으로 메모리에서 제거
  - 핵심 기능 코드에 집중할 수 있도록 하는 기능
- 다양한 애플리케이션 개발 가능



## 2절. 자바란?

### ❖ 자바의 특징

- 멀티 스레드(Multi-Thread) 쉽게 구현 가능
  - 동시에 여러 가지 작업을 할 경우
  - 대용량 작업을 빨리 처리할 경우
- 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
- 막강한 오픈 소스 라이브러리 풍부





# 3절. 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 설치
- JRE(Java Runtime Environment) = JVM + 표준 클래스 라이브러리
  - 자바 프로그램을 실행만 할 경우 설치

### ■ JDK(Java Development Kit) 설치

- 설치 파일 다운로드 사이트: <http://www.oracle.com>

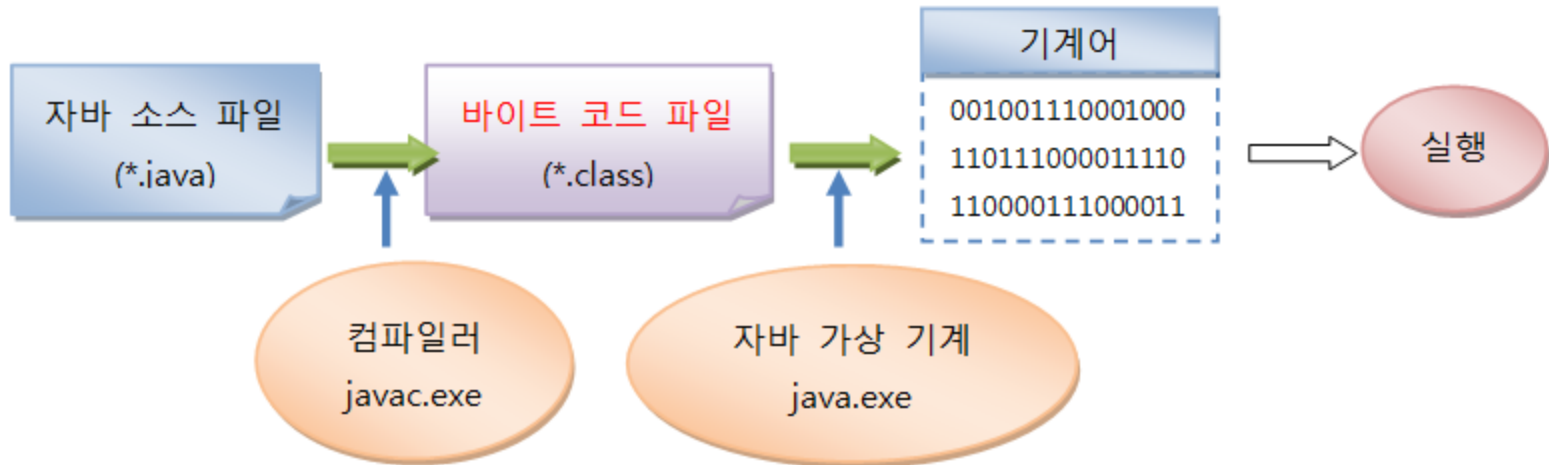
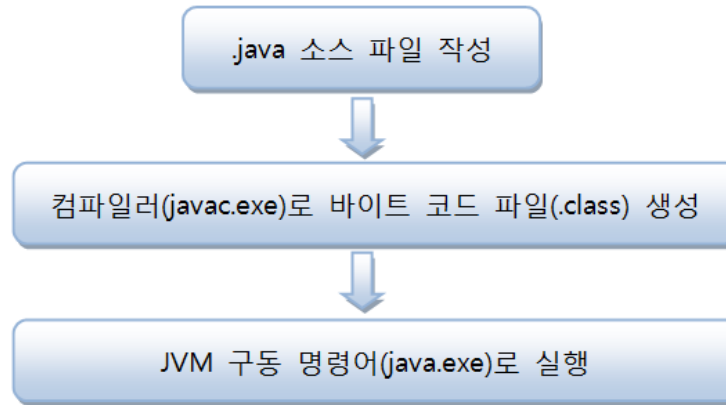
### ■ API 도큐먼트

- JDK에서 제공하는 표준 클래스 라이브러리 설명해 놓은 HTML 페이지들
- <http://docs.oracle.com/javase/버전/docs/api/>



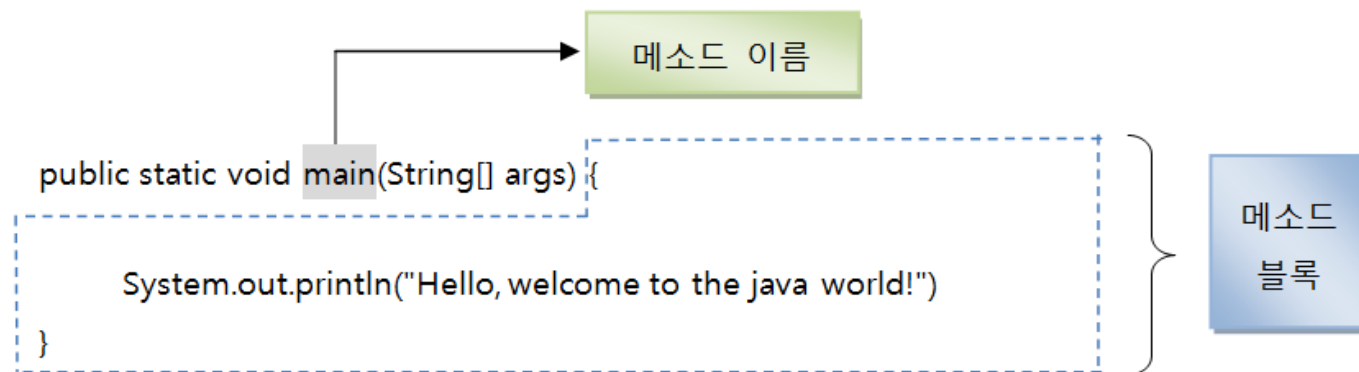
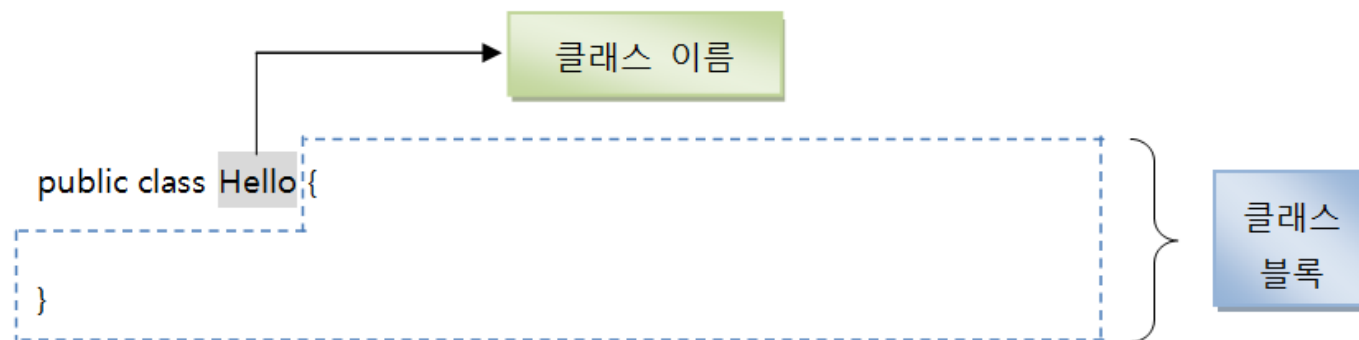
## 4절. 자바 프로그램 개발 순서

### ❖ 소스 작성에서부터 실행까지



## 4절. 자바 프로그램 개발 순서

### ❖ 프로그램 소스 분석



## 5절. 주석과 실행문

### ❖ 주석 사용하기

- 프로그램 실행과는 상관없이 코드에 설명 붙인 것
- 컴파일 과정에서 주석은 무시되고 실행문만 바이트 코드로 번역
- 코드에서 사용하는 주석문의 종류

주석 기호	설명
//	//부터 라인 끝까지 주석으로 처리한다. (행 주석)
/* ~ */	/*와 */ 사이에 있는 모든 범위를 주석으로 처리한다. (범위 주석)



# 5절. 주식과 실행문

## ❖ 실행문과 세미콜론(;)

### ■ 실행문

- 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
```

```
int x = 1; int y = 2;
int result =
    x + y;
```

