

# Java 소개



# Java와 객체지향

**Key Point** Java는 안전한 프로그램을 개발하기 위해 설계된 객체지향 프로그래밍 언어다.

- Java는 객체지향 언어다.
  - **객체지향 프로그래밍:**  
객체를 중심으로 프로그램을 작성하는 방식
  - **객체:**  
상태를 유지하며 외부 요청에 반응하는 어떤 대상을 나타냄
- 고슬링은 왜 Java를 선택했는가?
  - 임베디드 시스템은 계산 자원이 충분하지 않음
  - 그에 비해 C++나 Smalltalk은 너무 덩치가 컸음
  - 그래서 Java의 저자 제임스 고슬링은 새로운 언어를 만들기로 결정함

# 프로그램을 통한 문제 해결 과정

---

## (1) 문제의 이해

- 먼저 그 문제를 제대로 이해해야 함.

## (2) 해법(solution) 설계 및 개선

- 문제 이해를 바탕으로 문제 해결 방법(해법)을 강구하고 이를 구체적으로 설계
- 설계된 해법은 지속적으로 개선

## (3) 해법 구현

- 설계된 해법을 실제로 프로그램 형태로 구현

## (4) 테스트

- 작성된 프로그램을 테스트하여 올바르게 동작하여 정확한 답을 주는지 반복적으로 테스트

# 객체지향 방법

---

## □ 기본 아이디어

- 주어진 문제를 실세계의 사물들을 나타내는 여러 개의 객체들로 분할하여 문제를 해결한다.

## □ 객체지향 프로그래밍(object-oriented programming)

- 객체지향 방법에 기초한 프로그래밍 방법으로
- 실세계의 사물 혹은 물체를 프로그램 상에 객체로 표현
- 이들 사이에 일어나는 상호작용을 프로그램 상에서 시뮬레이션
- 상호작용을 메소드(혹은 함수) 호출로 표현한다.

## □ 예: 은행에서 일어나는 일

- 프로그램 상에 고객,직원,계좌 등을 나타내는 객체들을 만듦
- 고객이 계좌에서 입출금하는 등의 객체들 간의 상호작용을 표현

# 객체

## □ 객체의 예

- 주사위, 학생, 은행계좌, 자동차, 교통카드, 신용카드 등

## □ 객체 표현

- 객체의 특성과 행동을 이해하여야 한다.

## □ Key Point

객체는 그 객체의 특성을 나타내는 상태(state)와 그 객체의 행동(behaviors)으로 이루어진다.



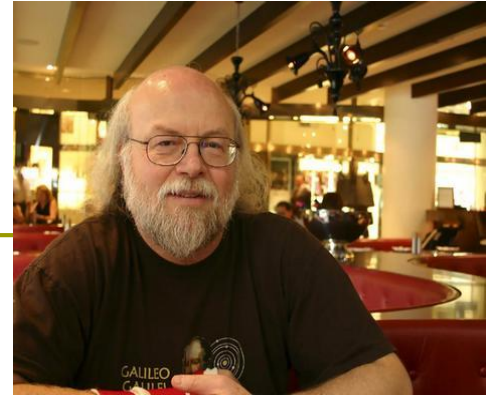
■ [그림 2.3] 대학생의 상태와 행동

# 클래스

---

- 프로그램 내에서 객체를 생성하기 위해서는 먼저 객체의 상태와 행동을 표현하는 클래스를 정의해야 한다.
- **Key Point**  
클래스는 객체에 대한 정의로 객체에 대한 설계도라고 생각할 수 있다.
- **Key Point**  
클래스 내부는 상태를 나타내기 위한 데이터(data) 선언들과 행동을 정의하는 메소드(method) 선언들로 구성된다.

# Java 언어의 유래



- Java는 커피 생산지 아닌가?
  - Oak 프로젝트를 개명하여 Java로 바꿈
  - Oak는 다른 회사의 등록상표였기 때문에 어쩔 수 없었음
  - Java 프로젝트의 팀원이 애용하던 커피숍 이름
- Oak 프로젝트란?
  - 제임스 고슬링이 주도한 프로젝트
  - 임베디드 시스템용 프로그램을 개발하기 위한 프로젝트
  - Java의 높은 안전성은 '임베디드 시스템'을 목표로 했기 때문

---

**Key Point** Java는 임베디드 시스템용 프로그램을 개발하기 위해 제작된 언어다.

---

# Java의 성공 요인

**Key Point** Java는 웹 브라우저에 내장(임베딩)됨으로써 성공을 거두었으며, Java 개발 환경은 현재까지 무료로 제공되고 있다.

- 초기 Java는 성공하지 못했음
  - 초기 임베디드 시스템은 Java를 받아들이기엔 소규모였음
  - Java 칩 등으로 반전을 꾀했지만 크게 성공하지 못했음
- Java의 성공 요인은 웹
  - 임베디드 시스템 대신 웹에 내장됨
  - 단순 문서 대신 움직이는 웹을 추구함
  - 웹 브라우저를 컴퓨팅 플랫폼으로 바꿈
  - 개발 환경을 무료로 제공함



# Java의 특징

**Key Point** 객체란 스스로 상태를 유지하며 외부 요청에 반응하는 어떤 대상이다. 객체는 능동적인 데이터라고 볼 수 있다.

- 객체지향성
  - 프로그램 설계와 유지보수가 쉬움
- 하드웨어 독립성
  - 한 번 작성하여 아무곳에서나 실행(WORA)
- 강력한 라이브러리
  - 다양한 응용분야를 지원할 수 있음
- 다양한 플랫폼
  - 데스크톱은 물론, 휴대폰, 서버까지 지원함
- 자동 메모리 관리
  - 불안정한 상태가 되는 현상을 줄일 수 있음

## 1.2 컴퓨터와 프로그램



# 컴퓨터 시스템

**Key Point** 컴퓨터는 하드웨어와 소프트웨어로 구성된다. 컴퓨터 하드웨어란 물리적인 장치를 의미한다.

컴퓨터 = 하드웨어 + 소프트웨어

## □ 컴퓨터 하드웨어

- 컴퓨터를 구성하는 물리적 장치
- 본체, 모니터, 키보드, 마우스 등
- 본체 내에는 CPU, 메모리(RAM) 등이 존재함

## □ 컴퓨터 소프트웨어

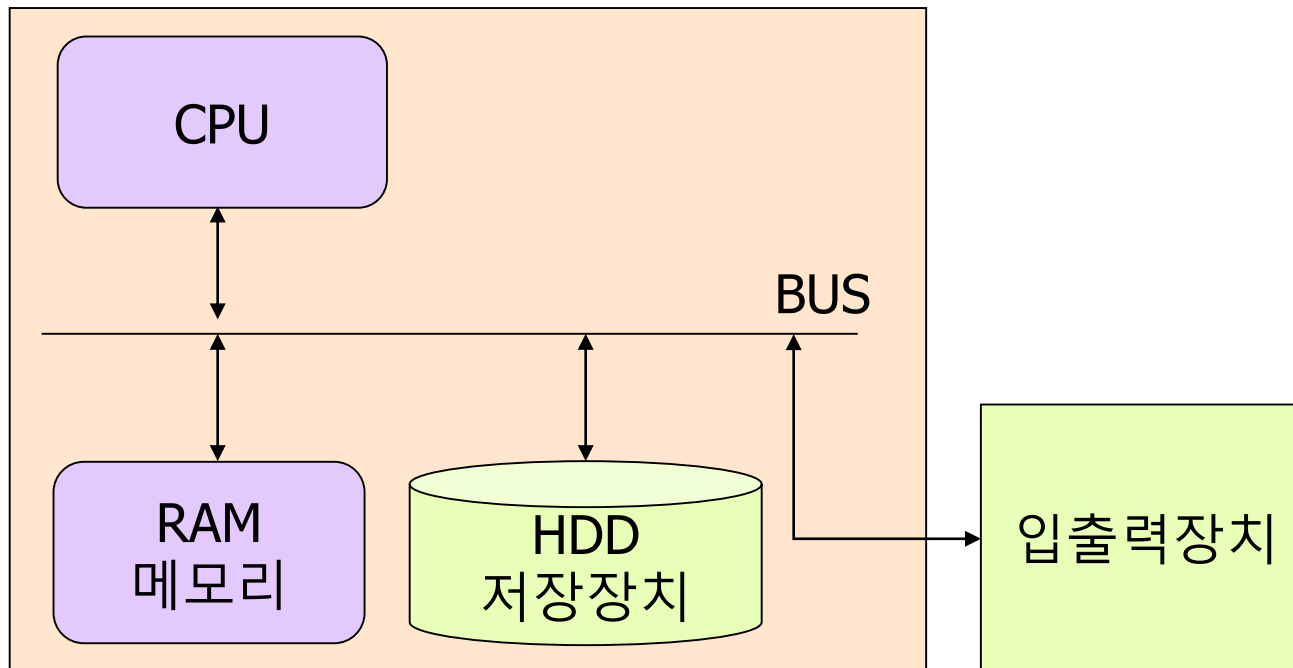
- 컴퓨터가 수행할 프로그램과 필요한 데이터
- 매우 중요하긴 하지만 하드웨어 없이는 볼 수 없는 형태임
- 시스템 소프트웨어와 응용 소프트웨어로 구성됨

# 프로그램 실행 구조

**Key Point** 컴퓨터 CPU는 메모리에 있는 프로그램에 대하여 인출·실행 주기를 반복한다.

## □ 하드웨어와 프로그램 실행 구조

- CPU와 주변장치가 BUS로 연결되어 있음
- BUS는 데이터 전송로임
- CPU는 메모리에 있는 명령어를 불러와서 해석하고 실행하는 주기를 반복함(인출·해석·실행 주기 또는 인출·실행 주기)



# 프로그램과 프로세스

**Key Point** 프로그램과 프로세스의 가장 큰 차이는 실행 중인가 여부다. 실행 중인 프로그램을 프로세스라고 부른다.

소프트웨어 = 프로그램 + 데이터

## □ 프로그램

- 컴퓨터가 수행할 명령어들을 나열한 것
- 하드디스크에 파일 형태로 저장되어 있음
- 실행 시에는 메모리로 읽어 들임(적재, loading)

## □ 프로세스

- 메모리에 적재되어 수행되고 있는 프로그램
- 명령어 뿐만 아니라 수행 상태 정보도 포함하고 있음
- 현대적 운영체제는 여러 프로세스를 동시에 수행함

# 프로그램 실행

- 두 가지 프로그램 실행 방식
  - 소스코드를 기계어로 바꾸거나
  - 소스코드를 해석하여 실행함

- 컴파일러
  - CPU가 알아듣는 명령어를 기계어(machine language)라고 함
  - 프로그래밍 언어로 작성된 프로그램을 기계어로 번역해 주는 프로그램

- 인터프리터
  - 프로그램의 한 문장을 필요할 때마다 해석하여 수행하는 프로그램
  - CPU의 인출-실행 주기와 같은 방식으로 동작함

---

**Key Point** 컴파일러는 소스코드를 번역하여 실행파일을 만들어 준다. 일단 실행파일이 만들어지면 소스코드와 컴파일러가 없어도 프로그램을 실행시킬 수 있다.

---

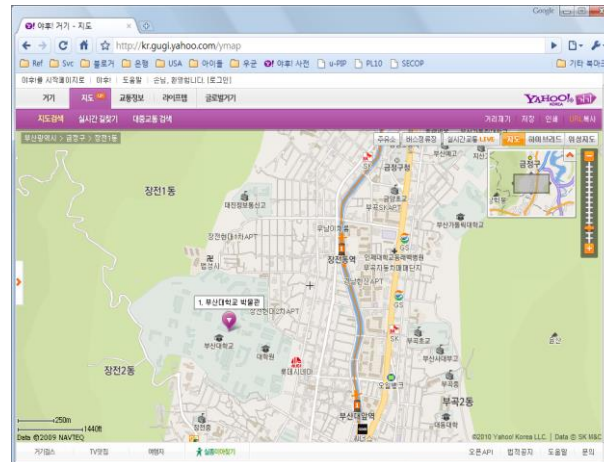
# 실행 환경에 따른 프로그램 분류

## □ 데스크톱 프로그램

- 독립적인 한 컴퓨터에서 수행되는 프로그램
- 흔히 응용 프로그램이라고 부름

## □ 웹 프로그램

- 웹 브라우저를 통하여 수행되는 프로그램
- 전통적인 클라이언트 서버 프로그램의 일종임



# 웹 프로그램과 Java

**Key Point** 웹 프로그램은 한 번 작성하면 어떤 곳에서든 수행될 수 있는데, 이러한 특성을 WORA(write once run anywhere)라고 부른다.

## □ 웹 프로그램의 특징

- 웹 브라우저 상에서 수행되므로 설치할 필요가 없음
- 한 번만 작성하면 아무 곳에서도 수행될 수 있음
- WORA(write once, run anywhere)

## □ Java와 웹의 관계

- Java 프로그램은 웹 브라우저에서 실행되도록 작성할 수 있음
- 웹 브라우저에 Java 실행 엔진을 탑재시키고 Java 코드를 네트워크로 전송함



# 1.4 Java 실행 환경과 개발 환경

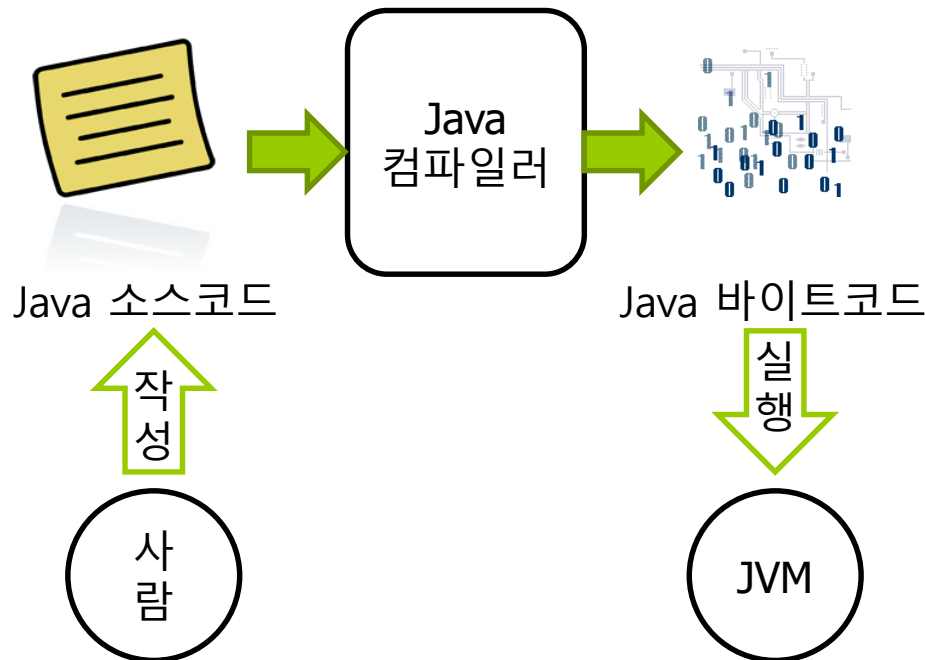


# Java 가상기계

**Key Point** Java 가상기계가 이해하는 프로그램을 Java 바이트코드라고 한다.

## □ Java 가상기계란?

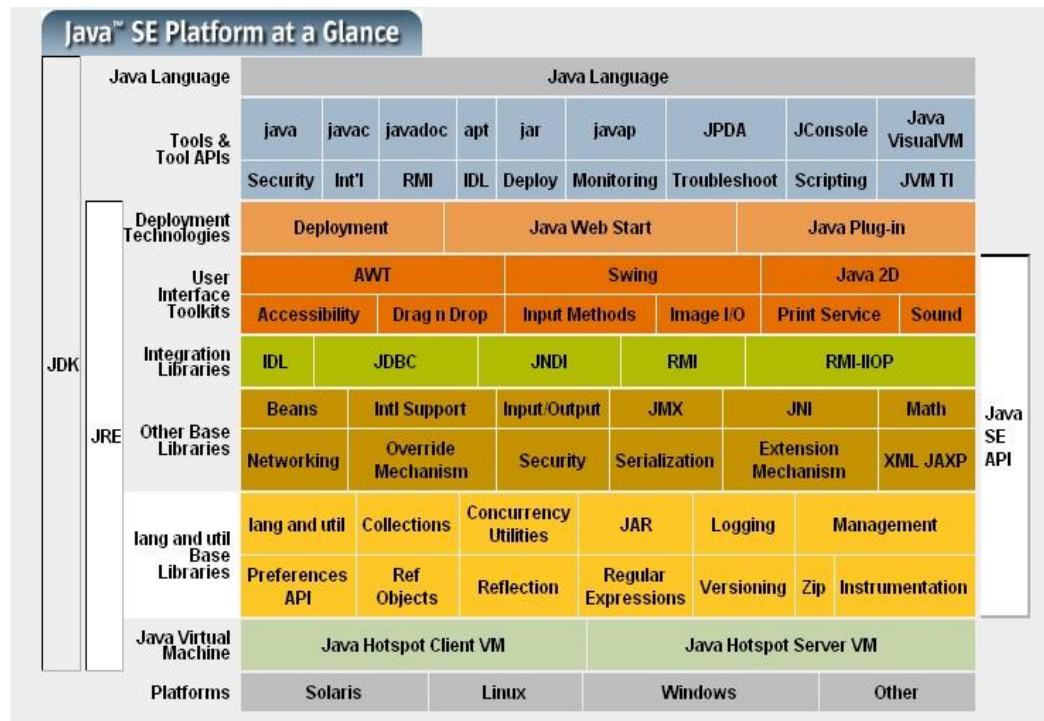
- 실제 기계가 아니라 프로그램으로 구현한 기계
- Java 바이트 코드를 이해하고 수행해 주는 인터프리터 프로그램



# Java 플랫폼

## □ Java 플랫폼이란?

- Java 가상 기계와 클래스 라이브러리
- Java SE, Java EE, Java ME 등이 있음



# Java 개발환경

**Key Point** Java 프로그램을 개발하는 데 필요한 제반 프로그램들을 모두 합쳐서 Java 개발 환경이라고 부른다.

## □ Java 개발환경이란?

- 컴파일러, 문서 생성기, 디버거 등 개발에 필요한 제반 프로그램
- 주로 사용할 도구는 java(Java 가상기계)와 javac(Java 컴파일러)임

## □ 두 가지 형태의 개발 환경

- IDE 환경: 여러 도구가 하나의 윈도우 인터페이스 형태로 제공됨
- 명령줄 환경: 필요한 도구를 일일이 명령어로 호출하여 사용함