

Java 프로그래밍 소개



1. Java 프로그래밍 소개

01 programming?



programming?

➡ 컴퓨터와 인간간의 커뮤니케이션

컴퓨터의 처리를 인간이 원하는 의도대로 하기 위해서
일련의 명령어로 작업을 지시하는 것

때문에 컴퓨터가 이해할 수 있는 커뮤니케이션 방식이 필요

1. Java 프로그래밍 소개

01 programming?

프로그래밍 언어



결국
커뮤니케이션용 언어일 뿐

- ✓ 실제 세계의 언어마다 특징이 있듯이
프로그래밍 언어들도 그 특징이 있음
- ✓ 사람과 사람 사이에 통역사가 있듯
컴퓨터와 사람 사이에도 통역사가 필요함
 - 자바에서는 JDK라는 통역

1. Java 프로그래밍 소개

02 프로그래밍 언어의 발전 방향

1 Low level 언어

- ✓ 기계어 중심의 프로그래밍 (0과 1로 이루어진 언어)
- ✓ CPU 인스트럭션
- ✓ 어셈블리어

1. Java 프로그래밍 소개

02 프로그래밍 언어의 발전 방향

2 High level 언어

- ✓ 사람이 이해하기 쉬운 언어
- ✓ 3세대 : fortran, cobol, c
- ✓ 4세대 : delphi, powerbuilder
- ✓ Web : 닷넷/ Java 등 현재 사용되는 언어들

1. Java 프로그래밍 소개

03 Low level의 언어의 장/단점

장점

- 세밀한 처리가 가능
- 최적화된
프로그래밍 구현 가능

단점

- 진입장벽이 높음
- 프로그램 구현을 위해서
많은 시간

1. Java 프로그래밍 소개

04 High level 언어의 장/단점

장점

- 인간이 이해하는 언어의 형태로 작성
- 빠른 개발
- 쉬운 유지보수

단점

- 완벽한 성능을 기대하기 어려움
- 개발자의 지식 수준이 상대적으로 낮음

1. Java 프로그래밍 소개

05 High level 언어와 컴파일러의 등장

High level



CPU인스트럭션으로
직접 변환은 불가

- ✓ 중간에 CPU의 언어로 해석할 수 있는
간접적인 장치가 컴파일러
- ✓ 보통의 경우 컴파일러를 통해서 운영체제와 통신하는
코드를 생성

1. Java 프로그래밍 소개

06 Interpreter 방식의 언어

컴파일을 통해서 운영체제와 통신하는 방식의 대안

- ✓ 특정한 프로그램이 실행되면서
프로그램에 필요한 로직이나 데이터를 해석
- ✓ 해석기의 존재만 있다면 운영체제에 영향을
덜 받을 수 있다는 장점

1. Java 프로그래밍 소개

07 자바 언어의 역사

1

- 1991년 썬사의 James Gosling에 의해
가전제품에 이용하기 위해 개발이 시작(Green
Project-컴퓨터기술을 통합한 통합 리모콘)
- 당시 WWW은 별로 알려지지 않았음

1. Java 프로그래밍 소개

07 자바 언어의 역사

2

- 초기에 개발된 언어를 Oak라 하였으며
전자기기의 내장된 프로그램을 위해 사용
- Oak는 별로 관심을 끌지 못하였음

1. Java 프로그래밍 소개

07 자바 언어의 역사

3

- 1994년 급격히 성장한 WWW에 자바를 적용 결정
- 처음 등장한 것이 웹러너였으며 뜨거운 자바 즉 핫자바로 이름을 바꾸었음

1. Java 프로그래밍 소개

07 자바 언어의 역사

4

- 처음으로 공식 회의에서 웹 문서 안에서 실행되는 3D 분자 구조모형을 선보였음
- 운영체제에 구애 받는 않는 프로그램을 작성할 수 있었음
- Gosling은 웹브라우저 안에서 실행되는 자바 프로그램을 작은 프로그램이라는 뜻에서 Applet이라 하였음

1. Java 프로그래밍 소개

07 자바 언어의 역사

5

- 1995년 5월 23일 넷스케이프 사의 브라우저가 자바 기술을 채택하게 되었음

6

- 플랫폼
: Sun SPARC Solaris, Windows NT, Windows95, Linux

1. Java 프로그래밍 소개

07 자바 언어의 역사

7

- Java beta1 발표(Sun Microsystems)

8

- Netscape 지원결정

1. Java 프로그래밍 소개

07 자바 언어의 역사

9

- Java beta2 발표

10

- JavaScript 발표(Sun & Netscape)

1. Java 프로그래밍 소개

07 자바 언어의 역사

11

- 1996년 자바1.0 발표 Netscape2.0 자바 지원

12

- 2005년 자바 5.0 사용

1. Java 프로그래밍 소개

07 자바 언어의 역사

13

- 2009년 자바6.0 사용

14

- 2010년 Oracle인수

1. Java 프로그래밍 소개

07 자바 언어의 역사

15

- 2011년 자바 7.0 사용

16

- 2014년 자바 8.0 사용

1. Java 프로그래밍 소개

08 자바 언어의 특징

1 단순(simple)

- ✓ 자바는 C++ 언어를 기반으로 탄생하였지만,
C나 C++에서 프로그래머에게 많은 혼란을 주는
요소들을 제거

예시

포인터, 구조체

1. Java 프로그래밍 소개

08 자바 언어의 특징

2 객체지향(Object-Oriented)

- ✓ 객체지향 프로그램을 작성하는데에 자바만한 언어가 없다는 것도 큰장점
- ✓ 자바는 객체지향 개념을 쉽게 구현할 수 있도록 클래스와 상속을 지원하면서 C++와 달리 모든 프로그램이 클래스 안에서 구현되도록 규정되어 있으므로 진정한 객체지향 언어

1. Java 프로그래밍 소개

08 자바 언어의 특징

3 동적인 성능 확장 제공

- ✓ 자바가 세상에 결정적인 인상을 준 것은
미리 설치 되지 않은 프로그램이 서버로부터
동적으로 다운 로드 되어 웹 브라우저에서
실행할 수 있다는 것임

1. Java 프로그래밍 소개

08 자바 언어의 특징

4 멀티쓰레드의 지원

- ✓ 자바에서는 쓰레드라는 기능을 언어 차원에서 제공함
- ✓ 쓰레드란 프로그램 안에서 독립적으로 실행되는 작은 실행단위라고 할 수 있음
- ✓ 메모리의 공유가 가능하면서 프로그램이 보다 효율적으로 실행 될 수 있도록 해줌
- ✓ 수행시간도 빠를 뿐더러, 요청에 응답하는 시간도 빨라짐을 알 수 있음

1. Java 프로그래밍 소개

08 자바 언어의 특징

5 자동 메모리 관리

- ✓ 자바에서는 개발자가 메모리를 관리해 주지 않아도 됨
- 왜냐하면, 자바가상머신이 자동으로 해제해 주기 때문
- ✓ 따라서 개발자는 자기가 사용하고 싶을 때로 사용하다가 그냥 두어도 자바가상머신이 자동으로 해체를 해줌(가비지컬렉션)
- ✓ **자바가상머신(Java Virtual Machine--JVM)이란?**
인터프리터와 같은 자바 프로그램의 실행환경을
가상적인 운영체제라는 뜻에서 자바가상머신이라 함

1. Java 프로그래밍 소개

09 자바 언어의 특징 (인터프리터)

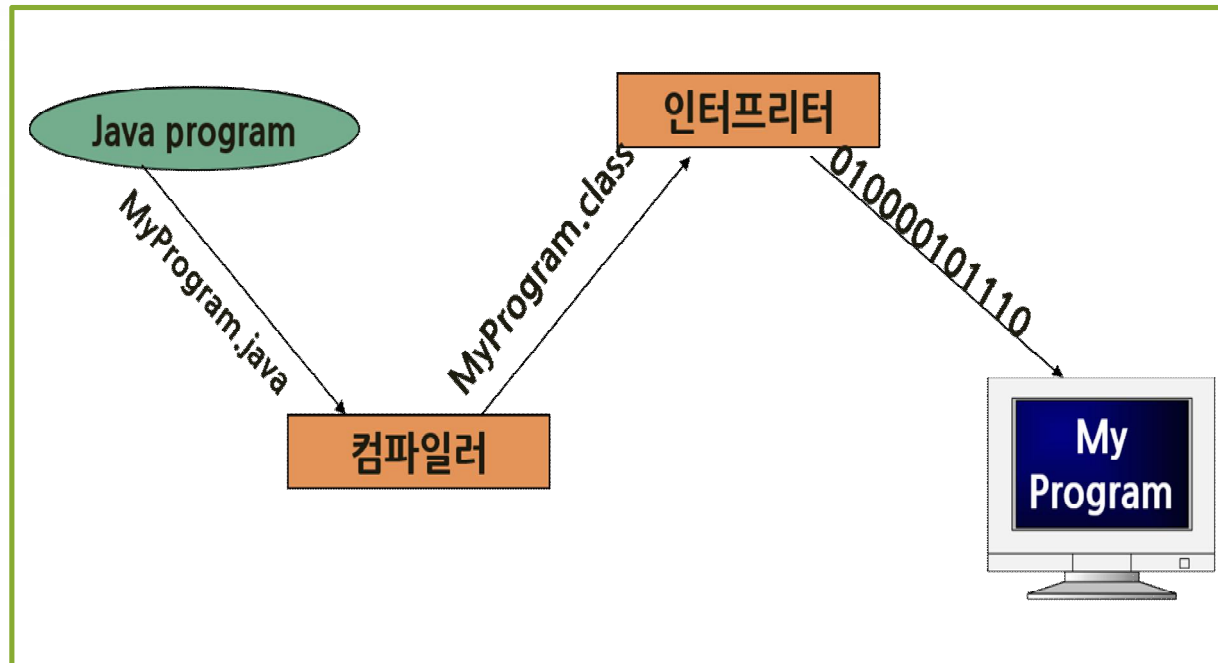
인터프리터(Interpreted)에 의해 실행

- 자바는 독특한 실행 방식을 취하고 있음
- 자바 소스는 자바 가상 머신에서 실행될 수 있는 바이트 코드로 번역되며, 번역된 바이트 코드는 바이트 코드 해석기가 있는 어떠한 종류의 컴퓨터 환경에서도 실행될 수 있음

1. Java 프로그래밍 소개

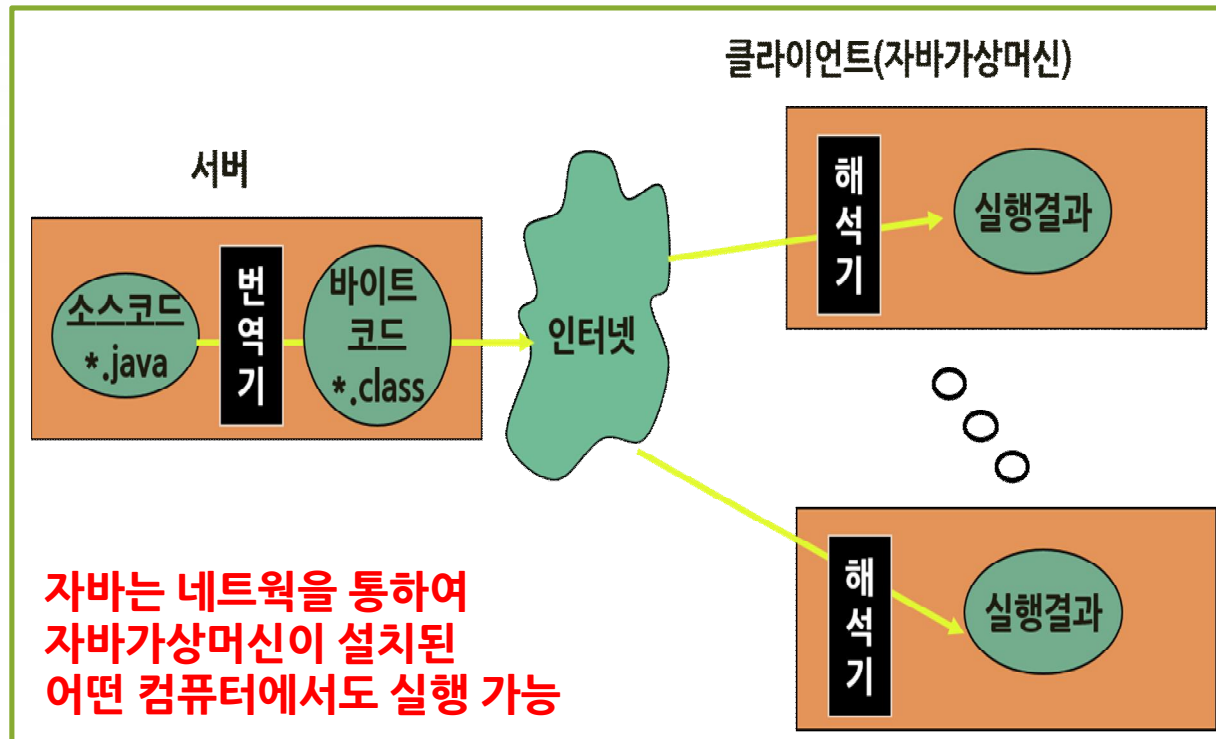
09 자바 언어의 특징 (인터프리터)

Tip : 자바는 compiled interpreted language



1. Java 프로그래밍 소개

09 자바 언어의 특징 (인터프리터)



1. Java 프로그래밍 소개

10 자바 언어의 특징 (독립성)

플랫폼의 독립성

- 자바는 자바가상머신에서 실행되는 바이트코드를 사용함으로써 특정 컴퓨터에 종속되지 않음
- 자바는 한번 작성되면 JVM이 설치된 모든 종류의 컴퓨터에서 실행할 수 있음
- 자바는 언어의 문법 뿐만 아니라 시스템 종속적인 부분까지도(int는 항상 32비트 등) 시스템 구조 중립적으로 설계되었음

1. Java 프로그래밍 소개

10 자바 언어의 특징 (독립성)



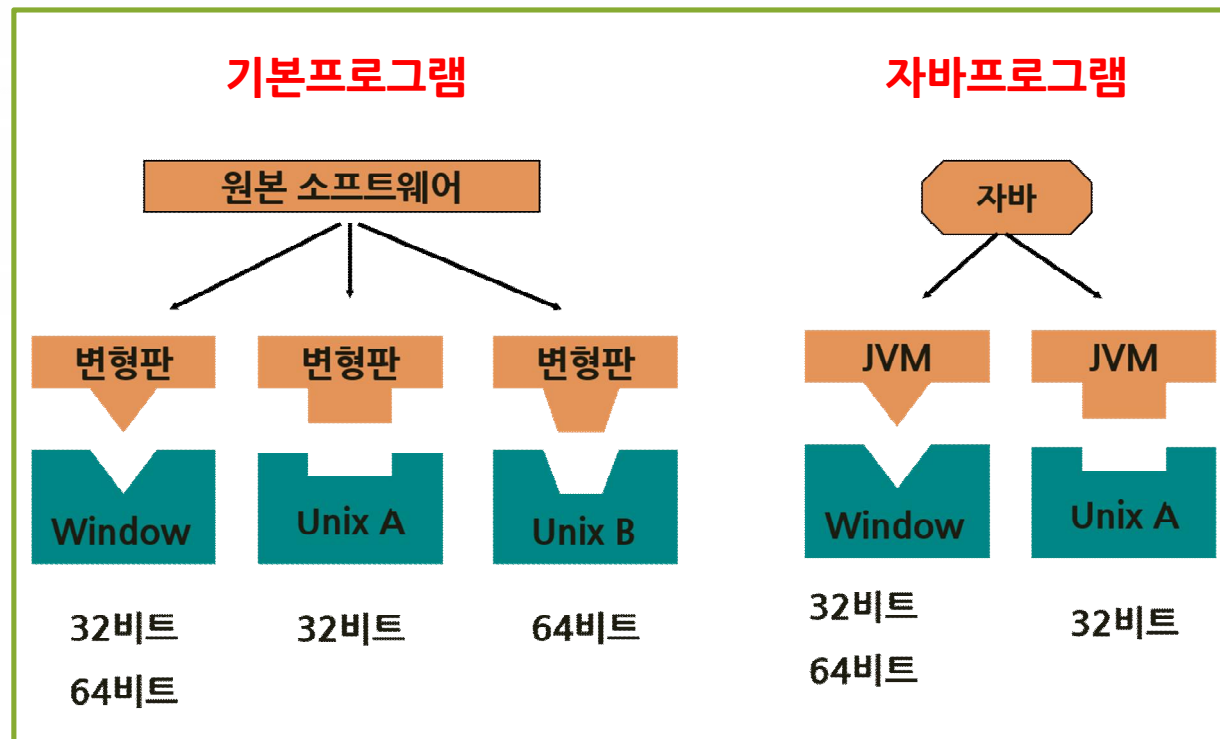
플랫폼의 독립성이란?

→ 자바에 대한 특징을 언급할 때
항상 플랫폼의 독립성이란 말이 나오는데,
플랫폼이란 우리나라말로 하면
“운영체제”를 말하고 독립적이란
“상관없다” 라고 말함

즉, 운영체제에 상관없이 실행됨

1. Java 프로그래밍 소개

10 자바 언어의 특징 (독립성)



1. Java 프로그래밍 소개

11 자바의 장점

- 1 빨리 배울 수 있음
- 2 적은 코드로 프로그램을 작성 할 수 있음
- 3 좋은 품질의 프로그램을 작성할 수 있음
- 4 프로그램 개발을 빨리 할 수 있음
- 5 플랫폼에 독립적인 프로그램을 작성할 수 있음
- 6 분산 소프트웨어를 쉽게 작성할 수 있음