**프로그래밍 언어**

1.프로그램 작성 언어

-기계어(machine language)

0, 1의 이진수로 구성된 언어

컴퓨터의 CPU는 기계어만 이해하고 처리가능 => “lnstruction set”

ex) 01011001000…..

-어셈블리어

기계어 명령을 ADD, SUB, MOVE 등과 같은 표현하기 쉬운 상징적인 단어인 나모닉 기호

(mnemonic symbol)로 일대일 대응시킨 언어

ex) add 001, sub 101, mov 110 ….

-고급언어

사람이 이해하기 쉽고, 복잡한 작업, 자료 구조, 알고리즘을 표현하기 위해 고안된 언어

Pascal, Basic, C/C++, Java, C#

절차 지향 언어와 객체 지향 언어

ex) a = b + c;, if, while……

2.컴파일

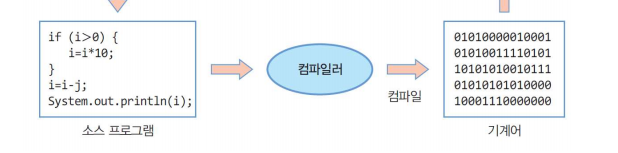
-소스 : 프로그래밍 언어로 작성된 텍스트 파일

-컴파일 : 소스 파일을 컴퓨터가 이해할 수 있는 기계어로 만드는 과정

소스 파일 확장자와 컴파일 된 파일의 확장자

자바 : .java ->(컴파일) .class

C : .c ->(컴파일) .obj ->(링크) .exe

 C++ : .cpp ->(컴파일) .obj ->(링크) .exe

3. 자바의 태동

-목적

*플랫폼 호환성 문제 해결*

기존 언어로 작성된 프로그램은 PC, 유닉스, 메인 프레임 등 플랫폼 간에 호환성 없음

소스를 다시 컴파일 하거나 프로그램을 재 작성해야 하는 단점

*플랫폼 독립적인 언어 개발*

모든 플랫폼에서 호환성을 갖는 프로그래밍 언어 필요

네트워크, 특히 웹에 최적화된 프로그래밍 언어의 필요성 대두

4. WORA

(Write Once Run Anywhere)

한번 작성된 코드는 모든 플랫폼에서 바로 실행

5. 플랫폼 종속성(platform dependency)

플랫폼 = 하드웨어 + 운영체제 플랫폼

프로그램의 플랫폼 호환성 없는 이유

-기계어가 CPU마다 다름

-운영체제마다 API가 다름 [Application Programming Interface ex) Library]

-운영체제마다 실행파일 형식이 다름