섹션0 - Java 개발 시작하기

- 기업에서 백엔드 개발 시 가장 많이 사용하는 언어

- 기본적으로 객체지향 언어

++ 버전이 증가하면서 다양한 장점들 흡수

(람다, stream API, 병렬 프로그래밍 등)

Hello.java로 java program 생성

-> Hello class안에 Hello를 출력하는 main함수 작성

=> 이후 javac Hello.java 명령어로 컴파일 -> Hello.class 생성됨

-> java Hello를 명령어로 입력하면 JVM이 실행되면서 Hello.class를 실행

(자바프로그램을 실행하는 것은 .class 파일을 실행하는 것)

- 자바는 운영체제 dependent하지 않게 모든 운영체제에서 동일한 프로그램이 돌아가기를 원했음

-> 소스코드를 기계어로 변환할 때, 바로 변환하지 말고 그 중간에 바이트 코드를 생성하자

=> .class file들이 byte코드 <-> C언어는 컴파일하면 바로 기계어 파일이 생성

=> java는 JVM을 다양한 운영체제와 CPU에 맞게 제공(intel 용, MAX용 등 -> 각 개발자는 자신의 CPU에 알맞은 JVM을 설치해야 함)

-> 해당 PC에 맞는 JVM이 설치되어 있다면 .class 바이트 코드가 어느 운영체제에서 작성되었든 상관없이 실행이 가능

=> JVM은 바이트코드를 한 줄씩 읽어들이면서 인터프리터 방식으로 기계어로 변환하면서 실행

결론: 자바에서 컴파일은 기계어를 생성하는 것이 아니라 바이트 코드를 생성하게 된다 -> 이후 이러한 바이트 코드는 JDK안에 JVM이 한 줄씩 읽으면서 실행

섹션1 - 자바 기본 문법

- 기본형 type: int, short, long, char, Boolean 등

- 참조형 type: 기본형이 아닌 type

=> class나 interface가 해당됨(대문자로 시작하는 것이 좋음)

기본형은 메모미를 직접 할당받고 값을 저장하고 변수는 해당 메모리를 가르키는 것

참조형은 값을 직접 갖는 것이 아니라 참조하는 형식

++ i = 1과 같이 선언하면 1은 정수 리터럴이라고 함(리터럴: 변수에 입력되는 값)