**OOP 객체지향프로그래밍**

1. **소프트웨어 개발시 모든 서비스를 객체로 보고 구현**

**플랫폼**

1. **OS: 윈도우,유닉스,리눅스,맥킨토시,안드로이드... 컴퓨터의 자원을 운영할 수있도록 서비스 하는 것**

**고급언어 : 인간이 생각하는 것과 가장 가깝게 구현하도록 돕는 프로그래밍 (java,python,c,c++)**

**저급언어 : 기계(컴퓨터자원)를 직접적으로 제어하는 언어 (어셈블리어)**

**컴파일 언어 : 사람의 언어로 된 코드를 기계가 읽기 쉽게 번역하는 것**

1. **오류가 있는 코드를 배포하지 않는다**
2. **스크립트 언어보다 빠르다.**
3. **스크립트 언어보다 생산성이 떨어진다.**

**스크립트(인터프리터) 언어 : 사람의 언어로 된 코드를 바로실행하는 언어, 이때 바로 실행하는 엔진을 인터프리터라 부른다. (실행과 동시에 컴파일이되는 엔진이 많다.)**

1. **오류가 있는 코드를 배포할 수있다.(위험)**
   1. **개발툴(IDE)이 오류를 방지**
2. **컴파일 언어보다 실행속도가 느리다.**
   1. **실행과 동시에 컴파일해서 어느정도는 따라갈 수 있다.**
   2. **인기가 많은 언어는 많은 개발자들이 속도를 올리기 위해 노력한다.**
3. **컴파일 언어보다 생산성이 좋다.**

**자바**

1. **객체지향언어로 객체지향 프로그래밍을 할 수 있다. (class)**
2. **컴파일되는 언어로 \*.java 로 작성한 문서를 javac(컴파일러)로 \*.class 파일로 컴파일 후 실행**
3. **CLASS 문서: Java Virtual Machine (JVM) 이 실행하는 문서로 프로그램된 코드( main이 작성되 어 있으면 어플, 모듈(도구,lib))**
4. **JVM : 자바로 작성된 프로그램을 플랫폼에 독립적이게 만들어주는 엔진(프로그램을 실행하는 엔진)**
5. **JavaRuntimeEnv : jvm을 포함하는 실행도구 (java)**
6. **JavaDevelopementKit : jre 와 javac, lib를 포함하는 자바 개발자 도구**

**환경변수 :??**

**Integrated Development Env : 개발자 툴 (자동완성, 자동컴파일,테스트,DB연결, 버전관리 GUI 제공 .....)**

**Oracle : db회사 , Oracle, mysql, java**

**gradle : intellj 라는 ide 회사가 JAVA가 업데이트가 안되서 jvm 기반으로 새롭게 만든 언어**

**L01 DATA**

1. **객체와 객체지향 프로그래밍**
2. **클래스** 
   1. **객체를 추상화해서 타입(설계도)으로 만든것**
   2. **속성(전역변수,필드)**
   3. **메소드**
   4. **생성자(클래스명과 똑같은 함수) : 객체의 필드를 초기화할 때 사용 (지금은 몰라도 된다!!)**
3. **데이터타입**
   1. **자료형**
      1. **객체**
      2. **new 연산자로 생성자로 호출**
      3. **힙(동적)메모리 (GC)**
      4. **인스턴스, 객체라 부른다**
   2. **기본형**
      1. **수로된 데이터**
      2. **타입은 모두 소문자**
      3. **정수 : byte,short,int,long,**
      4. **실수 : float,double**
      5. **논리형 : boolean (true:1,false:0 1bit DATA 지만 1Byte를 사용)**
      6. **문자: char (문자표의 번호)**