

## 목차

1. PK, FK, ER모델이란?
2. 참조 무결성이란?
3. RDBMS란?
4. DB에서의 Commit와 Rollback 이란?
5. DBMS에 대해 설명해보시오.
6. DB정규화란(개요)?
7. DB정규화의 목적?
8. 데이터베이스의 정규화에 대해 설명해보시오.(상세설명)
9. SQL에 대해 설명해보시오.
10. TCP/IP에 대해 설명해보시오
  
11. OSI(Open Systems Interconnection)계층을 설명해 보세요.
12. TCP/UDP의 차이점에 대해 답변해 보세요.
13. 임베디드 시스템이란?
14. 리눅스와 유닉스의 차이점은?
15. C 와 C++ 언어의 차이점
16. C와 자바의 차이점은?
17. 공학과 과학의 차이는 무엇입니까?
18. 객체지향 모델의 종류와 내용에 대해 말해보시오.
19. 객체지향 언어의 특징에 대해 답변해 보세요.
20. OOP란?
  
21. OOP의 장점
22. 자바의 특징은?
23. UML 이란?
24. 스레드(Thread) 정의
25. 스레드(Thread) 특징
26. 스레드(Thread)의 장점 / 단점
27. 스레드와 프로세스의 차이점은?
28. JVM이란?
29. SI란 무엇인가?
30. 객체에 대해 설명 해보시오.
  
31. 클래스 파일에 대해 설명해보시오.
32. 인터넷과 인트라넷의 차이점은?
33. Overloading과 Overriding의 차이점은?
34. JDBC 코딩 단계는?
35. JDBC에서 Transaction 관리는 어떻게 하나요?
36. JDBC에서 Transaction을 처리할 때, setAutoCommit란?

- 37. setAutoCommit(true);인 경우와 setAutoCommit(false);인 경우의 차이점은?
- 38. statement와 preparestatement의 차이점은?
- 39. Servlet과 JSP의 차이점에 답변해 보세요.
- 40. 엔터프라이즈 자바빈즈란?

- 41. JDBC란?
- 42. Get과 Post의 차이점은?
- 43. Session과 Cookie의 차이점과 사용용도?
- 44. 구구단 출력 프로그램을 코딩할 수 있나요?
- 45. String과 StringBuffer의 차이점은?
- 46. 모델 1,2 MVC패턴에 대해 설명해보세요.
- 47. 서블릿의 라이프 사이클에 대해 설명하세요.
- 48. EJB에 대해 설명해보시오.
- 49. XML 이란?
- 50. ODBC와 JDBC에 대해 설명해보시오.

- 51. CS 기반과 웹기반의 차이점은?
- 52. Interface란?
- 53. Abstract 란?
- 54. Interface vs. Abstract Class 용도는?
- 55. overloading과 overriding의 차이점
- 56. Inner Class란? 용도는?
- 57. Reflection이란?
- 58. J2ME vs. J2SE/J2EE 차이점 및 특징?
- 59. MVC model이란?
- 60. Call by reference, Call by value

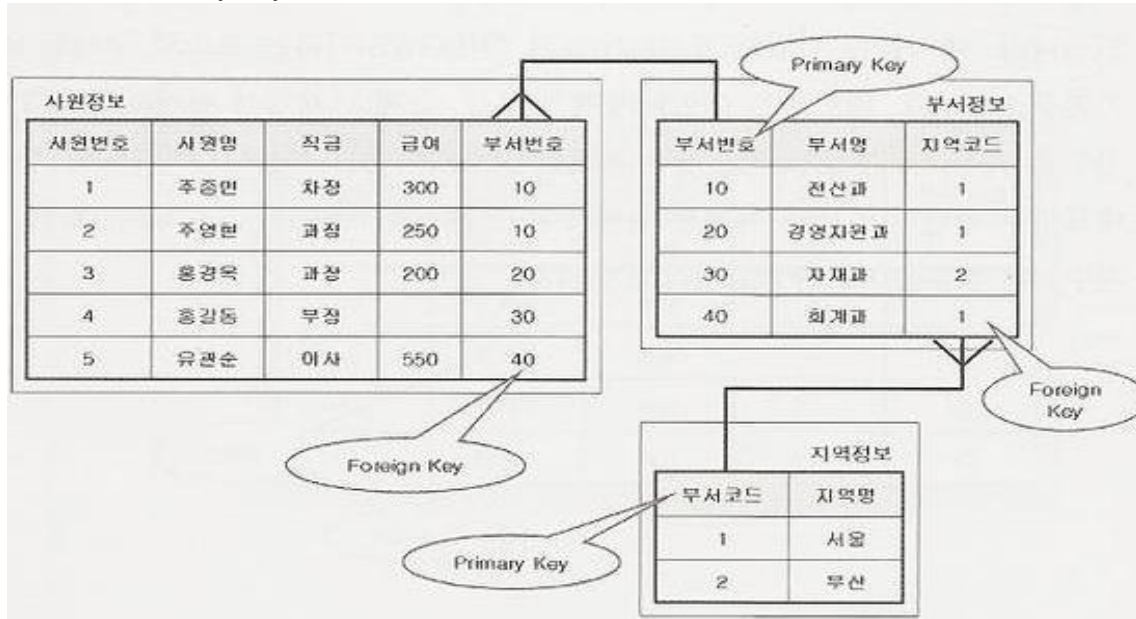
- 61. Static의 의미는?
- 62. Heap이란?
- 63. Stack 이란?
- 64. 프레임워크란?
- 65. SPRING FRAMWORK

## ※실무면접준비

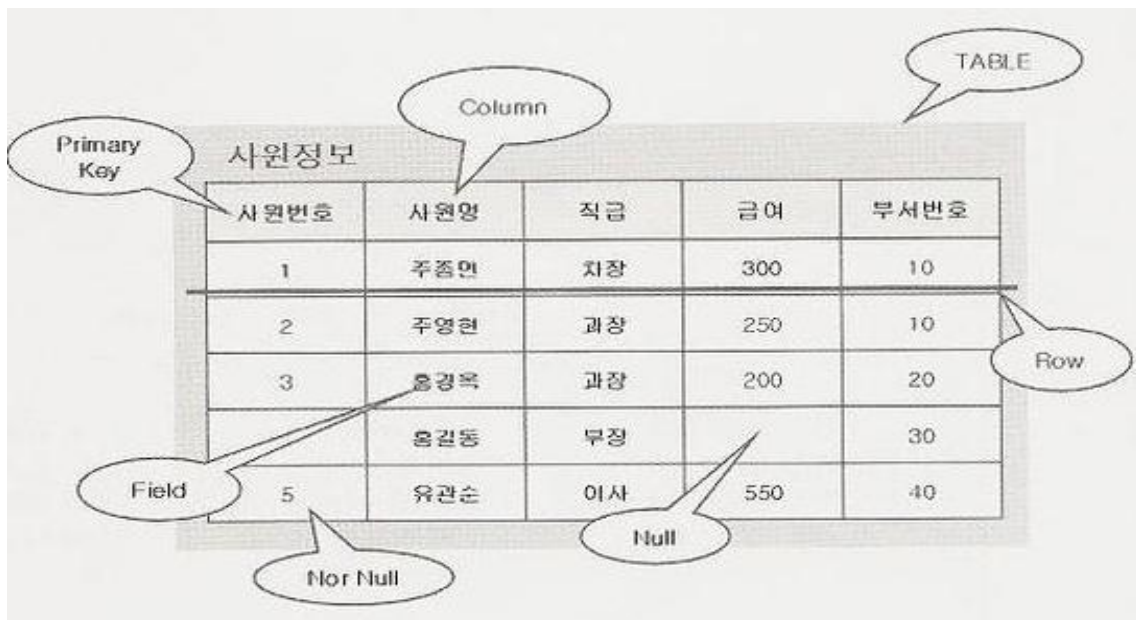
### 1. PK, FK, ER모델이란?

-Primary Key: 테이블에서 각 Row를 유일하게 구분하는 Column-Key.

-Foreign Key: 하나의 테이블에 있는 Column으로는 그 의미를 표현할 수 없는 경우, 다른 테이블의 Primary-Key Column의 값을 반드시 참조하여 표현해야 하는 Column.



-Field: Table에서 Row와 Column이 교차하는 데이터.



## 2. 참조 무결성이란?

-참조무결성이란 관련된 테이블의 레코드 간의 관계를 유효하게 하는 규칙으로 사용자의 실수로 관련 데이터가 삭제되거나 수정되는 것을 막아준다. 참조 무결성을 설정할 수 있는 조건은 기본 테이블에서 일치하는 필드가 기본 키이거나 고유 인덱스를 갖고 있거나 관련 필드의 데이터 형식이 같아야 한다.

## 3. RDBMS란?

- 모든 데이터를 2차원 테이블로 표현.
- 테이블은 row(record, tuple)과 column(field, item)으로 이루어진 기본 데이터 저장 단위.
- 상호관련성을 가진 테이블(table)의 집합.
- 만들거나 이용하기도 비교적 쉽지만, 무엇보다도 확장이 용이하다는 장점을 가짐.
- 데이터베이스의 설계도를 ER(Entity Relationship) 모델.
- ER모델에 따라, 데이터베이스가 만들어지며, 데이터베이스는 하나 이상의 테이블로 구성됨. ER모델에서 엔티티를 기반으로 테이블이 만들어짐.
- SQL(Structured Query Language)은 오라클 데이터베이스와 통신하기 위한 언어.

## 4. DB에서의 Commit와 Rollback 이란?

- Commit는 쿼리문에서 Update, Delete, Insert을 수행했을 때, 그 쿼리문 수행결과에 대해서 확정을 짓는다는 것이다. Commit하기 전까지는 실제로 적용이 되지 않는다. Rollback은 사용자가 Update, Delete, Insert 등을 실수로 수행했을 때, 데이터를 원복시킬 경우에 사용한다.
- Commit: 수행 쿼리문을 확정.
- Rollback: Commit하기 전에 수행함으로써 수행 쿼리문에 대해서 원상복귀 시킴.
- 테이블에 데이터가 1개 있고 추가로 사용자가 테이블에 데이터를 2개 Insert했다고 한다면, 이때 Commit을 하게 될 때, 해당 테이블에 3개 데이터가 확정이 되고, Commit하기 전에 Rollback을 하게 되면 3개의 데이터가 아닌 1개의 데이터가 그대로 보인다는 것이다.

## 5. DBMS에 대해 설명해보시오.

DBMS는 database management system의 약자로 데이터베이스 관리시스템이라고도 한다. 다수의 컴퓨터 사용자들이 컴퓨터에 수록된 수 많은 자료들을 쉽고 빠르게 추가, 수정, 삭제할 수 있도록 해주는 소프트웨어이다. 데이터베이스 내의 정보를 검색하거나 데이터베이스에 정보를 저장하기 편리하고 효율적인 환경을 제공하는 것이 목적이다. DBMS의 기능은 크게 세가지 기능으로 나눌 수 있다. 구성기능과 조작기능, 제어기능이다. 먼저 구성기능은 데이터베이스에 저장될 자료의 구조와 응용 프로그램이 그것을 이용하는 방식을 정의하는 것으로 레코드 구조의 정의, 데이터 보형의 정의, 물리적 구조의 정의등을 포함한다. 그 다음으로 조작기능은 사용자의 도구에 따라 데이터베이스에 저장된 자료를 검색, 갱신, 삽입, 삭제할 수 있도록 하는 기능이다. 마지막으로, 제어기능은 데이터베이스의 내용이 일관성을 유지하고 중복되지 않도록 하며, 허가되지 않은 접근을 제한하여 자료를 보호하는 기능이다. 최근에는 관계형 DBMS가 DBMS의 주류를 이루고 있는데 DBMS는 계층형, 네트워크형, 관계형 세가지로 나눌 수 있다. 계층형에서는 이름과 같이 계층구조로 데이터를 보존유지하게

되는데 데이터를 대분류, 중분류, 소분류 등으로 분류 및 정리할 수 있을 경우에 계층형 DBMS가 적용된다. 네트워크형에서는 데이터 끼리의 상호관계를 네트워크로 나타내는데 대규모 데이터베이스에서 많이 사용되고 있으며, 최근엔 객체지향 기술을 사용한 DBMS도 제 품화 되어 있다. 관계형에서는 DBMS가 정보계 시스템용으로 업계표준이 되어 있는데 최근엔 트랜잭션(transaction) 처리를 목적으로 하는 업무계의 DBMS로도 사용되게 됐다. 정보계 시스템용 DBMS는 기간 시스템에서 축적한 데이터를 사용자가 자유롭게 검색 및 가공하도록 하기 위한 시스템으로 영업의 기획 영업, 기획 등 여러면에서 클라이언트/서버 시스템에 대응하기 쉽도록 돼 있는 것이 그 특징이다.

## 6. DB정규화란(개요)?

자료의 손실이나 불필요한 정보의 도입 없이 데이터의 일관성, 데이터 중복을 최소화하고 최대의 데이터 안정성 확보를 위한 안정적 자료 구조로 변환하기 위해서 하나의 테이블을 둘 이상을 분리하는 작업

## 7. DB정규화의 목적?

자료 저장에 필요한 저장 공간을 최소화

자료의 삽입, 갱신 및 삭제에 따른 이상 현상 제거 (데이터 무결성 유지)

자료 구조의 안정성 최대화

## 8. 데이터베이스의 정규화에 대해 설명해보시오.(상세설명)

데이터베이스 정규화(Database Normalization)는 논리적 데이터베이스 설계에 있어서 테이블 들을 구조화하는 기법 중 하나다. 어느 테이블이든 어느 정도는 정규화 될 수 있는데 데이터베이스 이론에서는 테이블의 정규화된 정도를 정규형(Normal Form) 이라는 용어를 써서 표현합니다. 보통 많이 쓰는 정규형을 순서로 나열하면 다음과 같다.

제 1정규형(First Normal Form, 1NF)

-테이블의 각 셀의 값은 단일값을 가진다.

제2정규형(2NF)

-주키가 합성키며 부분종속(Partial Dependence)이 존재할 경우 2차 정규형의 대상이 된다.

제 3정규형

-비주키 속성간에 발생하는 함수적 종속(이전종속)이 발생하면 3차 정규형의 대상이 된다.

이전 종속(Transitive Dependence)이 존재하지 않는 관계를 말한다.

정규형 차수가 높을수록 데이터베이스의 무결성을 저해할 수 있는 다양한 종류의 논리적 모순이 존재할 가능성과 데이터가 중복으로 존재할 가능성을 줄여 준다. 높은 차수의 정규화는 보통 테이블의 개수를 많아지게 하고 많은 수의 JOIN을 창출합니다. 이것은 데이터베이스 성능을 저해할 수도 있습니다. 그 결과 차수가 높게 정규화된 테이블들은 고립 시킬 수 있는 트랜잭션들을 많이 쓰는 데이터베이스 (예 :자동입출금기기 시스템)에 많이 쓰이고, 차수가 낮게 정규화된 테이블들은 읽기 동작이 대부분인 데이터베이스(예:리포트)에 많이 쓰입니다.

## 9. SQL에 대해 설명해보시오.

먼저 SQL은 structured query language의 약자로 구조화 질의어라고 한다. 데이터 정의어(DDL)와 데이터 조작어(DML)를 포함한 데이터베이스용 질의 언어(query language)의 일종으로 데이터베이스를 사용 할 때 데이터베이스에 접근할 수 있는 데이터베이스 하부 언어를 말한다.

MS SQL은 각종 자료를 저장하는 데이터베이스(DB) 서버를 관리하는데 쓰이는 언어를 뜻하는 것으로 마이크로소프트(MS)에서 개발한 프로그래밍 언어이다. MS SQL에 기반해 MS가 개발한 데이터베이스서버다 시스템 운영시 혹은 웹사이트 운영시 필요한 수천만건의 데이터가 저장돼 있으며 데이터 사이의 관계를 관리하고 이용자의 인터넷 검색 기능을 지원한다. 1992년 윈도우NT에서 운용되는 SQL 서버 1.0판을 시장에 내놓은 것을 시작으로 매년 이를 업데이트한 것이 발표되고 있는데 소스코드를 공개하지 않고 있어 문제 발생시 대응이 어렵다는 지적이 나오고 있다.

MY SQL은 SQL을 사용하는 개발 소스의 관계형 데이터베이스 관리 시스템(RDMS)이다 사용하기 쉬운뿐만 아니라 매우 빠르고 유연한 것이 특징이다. 다중 사용자, 다중 쓰레드를 지원하고 C, C++, 자바, PHP, Python 스크립트 등을 위한 응용 프로그래머 인터페이스를 제공한다. 윈도우 운영 체제, 리눅스나 유닉스 등에서 사용가능하다. 홈페이지나 쇼핑몰 등 일반적인 웹 개발에 널리 이용되고 있다. LAMP 즉 리눅스 운영 체제와 Apache 서버 프로그램, MySQL, PHP 스크립트 언어 구성은 상호 연동이 잘되면서도 오픈 소스로 개발되는 무료 프로그램이기 때문이다. MY SQL은 오픈 소스라이선스를 다루기는 하나 상업적으로 사용할 때에는 라이선스를 구입해야한다.

## 10.TCP/IP에 대해 설명해보시오

( 요약 :

-인터넷에서, 서로 다른 시스템을 가진 컴퓨터들을 서로 연결하고, 데이터를 전송하는 데 사용하는 통신 프로토콜.

-TCP는 L4의 전송 프로토콜이고, IP는 L3의 통신 프로토콜. )

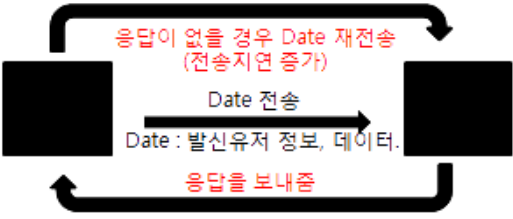
TCP/IP(Transmission Control Protocol/Internet Protocol)는 광역 통신망(WAN)에서 사용하도록 설계된 산업 표준 프로토콜군을 말한다. 컴퓨터 간의 통신을 위해 개발한 통신으로 TCP와 IP를 조합한 프로토콜이다. TCP/IP는 통신 프로토콜로 현재 인터넷에서 사용되고 있다. 현재 사용하고 있는 프로토콜로서 생산되고 있는 거의 모든 운영체제에서 작동하며 통신 프로토콜이 통일되면서 세계 전 지역의 어떤 기종과도 정보를 교환할 수 있는 것이다. RFC(인터넷 연구와 개발 공동체의 작업 문서) 형태로 공개되고 있고 거의 모든 운영체제에서 구현되어 널리 보급되어 있다.

## 11. OSI(Open Systems Interconnection)계층을 설명해 보세요.

→통신 접속에서 완료까지의 과정을 7단계로 정의한 국제 통신표준규약으로 다음과 같이 분류된다.

- ①L1: 물리계층-전송하는 데 필요한 기능 제공. 장비로는 통신 케이블, 허브가 있음.
- ②L2: 데이터링크계층-송수신을 확인. MAC Address를 가지고 통신. 장비로는 브릿지와 스위치가 있음.
- ③L3: 네트워크계층-패킷을 네트워크 간의 IP를 통하여 데이터를 전달. 장비로는 라우팅이 있음.
- ④L4: 전송계층-두 호스트 시스템으로부터 발생하는 데이터의 흐름을 제공.
- ⑤L5: 세션계층-통신 시스템 사용자간의 연결을 유지, 설정.
- ⑥L6: 표현계층-세션 계층 간의 주고받는 인터페이스를 일괄성 있게 제공.
- ⑦L7: 응용계층-사용자가 네트워크에 접근할 수 있도록 서비스를 제공.

## 12. TCP/UDP의 차이점에 대해 답변해 보세요.

TCP	UDP
<u>연결형</u> 프로토콜(흐름제어)	<u>비연결형</u> 프로토콜(단순전송)
신뢰적	비신뢰적
<u>패킷</u> 오류 복구를 위한 재전송	오류 제어 <u>안함</u>
데이터 순서 보장	데이터 순서 보장 <u>안함</u>
	빠른 전송에 적합
중요 정보 <u>전송시</u> 사용	자체전송확인 매커니즘 필요 (오류 처리, <u>패킷</u> 유실여부, 수신순서, 발신유저 확인)
	캐릭터의 위치 <u>행동들</u> 수시로 변경되는 곳에 사용

### 13. 임베디드 시스템이란?

임베디드 시스템(Embedded System, 내장형 시스템)이란 특정 제품이나 솔루션에서 특정 작업을 수행하도록 추가로 탑재된 솔루션을 말한다. 본래 임베디드(Embedded)가 '내장된'이라는 의미이므로 임베디드 시스템 역시 어떠한 시스템에 내장된 혹은 끼워 넣어진 시스템'이라고 볼 수도 있다. 이 시스템은 특정 기능만을 수행하며 가지고 있는 요구사항 또한 특정하다. 예로는 첨단 기능의 컴퓨터, PDA, 스마트폰, 홈네트워크, 서버 등의 각종 네트워크 장비 등을 들 수 있으며, 이와 같이 임베디드 시스템은 다른 부분과 결합하여 부수적으로 기능을 수행하는 것을 의미한다.

### 14. 리눅스와 유닉스의 차이점은?

리눅스(Linux)와 유닉스(UNIX)는 window와 같은 서버 OS의 한 일종이다. 동시 처리가 가능한 시스템이기 때문에 윈도우보다 월등한 성능을 자랑하고 있다. 유닉스는 1960, 70년대 개발된 운영체제로 다양한 시스템 사이에서 이식이 가능하다. 다중 사용자를 지원하고 있기 때문에 시스템에 직접 로그인 가능하고 원격 로그인 역시 가능한 네트워크 기능을 기본적으로 포함하고 있으며 멀티 태스킹이 가능하도록 만들어 졌다. 운영체제에 여러 가지 하드웨어 플랫폼을 최대한 맞출 수 있도록 자체 드라이버와 유틸리티를 내포하고 있기 때문에 기본적으로 운영체제로서의 기능과 유틸리티의 총체적인 집합으로 볼 수도 있다. 유닉스의 상표권은 오픈 그룹이 가지고 있으며 유닉스 소스 코드의 저작권은 노벨이 소유하고 있다.

리눅스는 유닉스를 기반으로 1991년 핀란드의 대학생인 리누스 토르발즈가 개발, 재생산된 공개 운영체제이다. 유닉스와 유사한 환경에 덧붙여, 저렴한 비용과 같은 특징을 중심으로 급속히 확대되어 사용되고 있다. 소스 코드가 무료로 공개된 오픈 소스이기 때문에 전세계적으로 프로그램 개발자 그룹이 형성되었고 이들에 의해 지속적인 업그레이드가 이루어지고 있다.

### 15. C 와 C++ 언어의 차이점

C언어는 인간의 명령어를 컴퓨터가 수용할 수 있는 기계적 언어로 변환 시켜주는 역할을 하는 프로그램이며, 1970년대 초 벨 연구소(AT&T Bell Laboratories)의 데니스리치(Dennis Ritchie)에 의해 개발되었다. C언어는 전세계적으로 많이 쓰이고 있는 언어이자 가장 기초적인 입문 언어이다. 그러나 몇가지 한계에 부딪히게 되었고 고도로 복잡한 프로그램을 관리하기 위해 C++를 탄생 시키게 되었다.

C++는 C언어의 확장판으로 만들어진 객체지향형 프로그래밍 언어이다. 복잡한 대규모 응용 프로그램을 만드는데 최적의 객체지향 프로그래밍 언어로 간주된다. C언어를 모태로 탄생하게 된 C++은 C의 대부분의 기능을 흡수하고 추가된 몇가지 기능을 가지게 되었다. 같은 이름의 함수가 존재할 수 없었던 C와 달리 C++은 동일한 이름을 가진 함수가 존재할 수 있게 되었다. C에서는 함수 머리말에서만 변수 선언이 가능했지만 C++에서는 중간에 변수 선언을 할 수 있게 되었다는 장점도 있다.



## 16. C와 자바의 차이점은?

C언어란 인간의 명령어를 컴퓨터가 수용할 수 있는 기계적 언어로 변환시켜주는 역할을 하는 프로그램이다. 시스템 간 호환 및 이식성이 좋고 고급언어와 저급 언어간의 인터페이스가 용이하다는 특징이 있다. C언어는 게임이나 운영체제의 개발을 위해 사용되었기 때문에 시스템 프로그램을 작성하는데 필요한 기능 외에 문서 및 파일 처리 프로그램이나 수식계산 프로그램, 데이터베이스 프로그램 등 대부분의 범용 프로그램 작성에 있어서 결코 다른 프로그래밍 언어에 뒤지지 않는 편의성과 다양한 기능을 제공한다.

JAVA(자바)란 인터넷의 분산 환경에서 사용되도록 설계된 프로그래밍 언어이다. 많은 소프트웨어들이 자바언어를 적극적으로 지원하고 있기 때문에 자바 자체의 기술 뿐만 아니라 다른 소프트웨어와 직접적인 연관성을 가지고 있기 때문에 모든 산업분야 개발에 참여할 수 있는 언어로 꼽힌다. 기존 C언어와 같은 프로그래밍 언어들은 주로 자신의 PC에서 프로그램을 만들고 수행하는 것에 주안점을 두었지만 자바는 인터넷 환경 기반의 프로그램을 만들고 수행 시킬 수 있는 응용 프로그램을 만들 수 있다.

## 17. 공학과 과학의 차이는 무엇입니까?

공학이란 전자, 전기, 기계, 항공, 토목 등 여러 분야의 공업 이론으로 기술, 생산 등을 체계적으로 연구하는 학문을 뜻한다. 이에 반해 과학은 이론적 기술보다는 보편적인 진리나 법칙의 발견을 목적으로 하는 체계적인 지식이라는 점에서 공학과 차이를 보인다고 할 수 있다.

## 18. 객체지향 모델의 종류와 내용에 대해 말해보시오.

객체 지향(Object-oriented)이란 실제 세계를 모델링하여 소프트웨어를 개발하는 방법이다. 여기서 객체란 우리 주변에 실제로 존재하는 사람, 컴퓨터, TV등의 사물을 말한다. 객체는 프로그래밍을 통해 클래스에 정의된 내용대로 메모리에 생성된다. 클래스는 객체의 설계도라고 볼 수 있는데, 이 클래스로부터 만들어지는 각각의 객체들을 인스턴스(instance)라고 부른다. 객체지향에는 캡슐화(객체 내부에 구현된 정보를 외부에 노출하지 않는다), 정보은닉(객체 내부의 데이터와 구현 세부 사항을 외부 세계로부터 감춘다), 상속(부모클래스에 정의 되어 있는 코드를 자식 클래스가 이어 받아 사용 할 수 있다.) 다형성(객체가 취하는 동작은 상황에 따라 달라진다.)의 네가지 특징을 가지고 있다.

## 19. 객체지향 언어의 특징에 대해 답변해 보세요.

### ① 추상화(abstraction)

추상화(abstraction)라는 것은 미술의 '추상화'라는 것과 비슷한 개념이다. 미술에서 추상화라는 것은 그 핵심 요지만 살리고 나머지는 생략하거나 변형하는 방법으로 그린 그림을 뜻하는 것이다. 마찬가지로 클래스를 구현함에 있어 모든 속성과 기능을 구현할 필요는 없다. 예를 들어, '회사원 관리'를 위한 '사람' 클래스에서는 '성별'이라는 속성이 반드시 필요하지만 '남자목욕탕 회원 관리'를 위한 '사람' 클래스에서는 굳이 성별이라는 속성을 표시할 필요가 없다. 어차피 모든 구성원이 '남자' 일테니까 말이다. 이처럼 굳이 구현할 필요가 없는 부분을 과감히 생략하여 필요한 부분만을 나타내는 것을 추상화라고 한다.

### ② 캡슐화(encapsulation)와 데이터 은닉(hidden)

캡슐화(encapsulation)라는 것은 은닉화(hidden)라는 것과 비슷하지만 조금 차이가 있다. 캡슐이라는 것은 약국에서 파는 알약 중에 '캡슐'에 들어가 있는 약을 생각하면 된다. 우리는 이 약을 복용할 때 그냥 캡슐채로 삼킬 뿐 그 안에 들어가 있는 알갱이 숫자를 세거나 구분해서 복용하지 않는다. 그냥 캡슐의 겉데기만 보고 무슨 약이라는 것을 알고 사용할 뿐이다. 마찬가지로 어떤 클래스를 사용함에 있어 그 클래스가 제공하는 메소드의 기능만을 알고 그 기능만을 사용할 뿐 실제로 그 메소드가 어떻게 움직이는지는 굳이 알 필요가 없다. `System.out.println();`이라는 메소드를 쓰면 괄호 안의 내용이 출력된다는 사실만을 알면 되지 그것이 어떤 원리로 나오는지 굳이 알 필요가 없는 것과 마찬가지다.

그리고 은닉화라는 것은 이 캡슐화 때문에 나오는 것인데 클래스의 속성들을 `private`으로 만들어 클래스 밖에서 함부로 건드리지 못하게 하는 것을 말한다. 클래스를 사용함에 있어 속성들에 직접 접근하는 것은 데이터 무결성 오류 등에 치명적일 수 있기 때문에 이들을 접근하지 못하게 하고 대신 '게터(getter)'와 '세터(setter)'라고 불리는 메소드를 통해서만 접근 가능케 하는 것을 말한다. 이렇게 하면 초기에는 번거로워 보이지만 들어와서는 안될 데이터가 들어오는 것을 막을 수 있을 뿐 아니라, '객체 지향'이라는 의미에 보다 가깝게 접근하게 할 수 있다.

### ③ 상속(inheritance)

상속이라는 것은 말그대로 부모의 유산을 자식이 이어받듯이 부모클래스의 속성과 메소드를 자식클래스가 이어받는 것을 말한다. 자식클래스는 `extends`라는 키워드를 통해 상속받은 부모클래스의 모든 멤버변수와 메소드를 승계하여 자신의 것처럼 쓸 수 있다. 단, 이때 부모클래스에서 `private`으로 선언했다면 가지고는 있지만 사용은 불가능하다. 자바는 단일 상속만을 지원하기 때문에 한 클래스가 여러 클래스로부터 상속을 받는 것은 불가능하며 이러한 다중 상속의 장점을 구현하기 위해 '인터페이스(interface)'를 사용한다.

### ④ 객체지향의 특징인 추상화, 캡슐화 및 은닉화, 그리고 상속의 의미 요약

#### -추상화

클래스를 구현함에 있어 필요한 부분만 구현할 수 있다.

#### -캡슐화 및 은닉화

캡슐화: 어떤 클래스를 이용함에 있어 내용(내부적인 데이터 흐름)은 알 필요 없이 사용해도 된다.

은닉화: 클래스의 내용을 함부로 고치지 못하도록 하여 무결성을 보장한다.

-상속

어떤 클래스(부모 클래스)의 속성과 메소드를 다른 클래스(자식 클래스)가 이어 받아 사용할 수 있는 것.

## 20. OOP란?

OOP란 Object- Orient Programming 의 약어로서 객체지향형 프로그래밍을 의미한다. 모든 데이터를 객체로 취급하여 프로그램에 반영한 것이며 순차적으로 프로그램이 동작하는 기존의 것들과는 다르게 객체와 객체의 상호작용을 통해 프로그램이 동작하는 것을 말한다.

객체 지향형 프로그래밍 기법은 코드의 재사용이 가능하다는 장점이 있다. 또한 오류 발생 가능성이 적고 안정성은 높은 것도 장점이라고 할 수 있다.

## 21. OOP의 장점

a.변경용이 b.직관적인 코드분석 c.개발속도향상 d.상속을 통한 장점 극대화

## 22. 자바의 특징은?

자바(JAVA)는 네트워크상에서 쓸 수 있도록 미국의 선 마이크로 시스템스가 개발한 객체 지향 프로그래밍언어이다.

자바의 특징으로는

1. 자바 가상 머신(JVM)만 설치하면 컴퓨터의 운영체제에 상관없이 작동한다. 즉, 운영체제에 독립적이라는 것이다. 구조 중립적이라고도 표현하며, 구조 중립적인 자바는 높은 이식성을 제공한다.
2. 기본 자료형을 제외하고는 모든 요소들이 객체로 표현되면 객체 지향 개념의 특징인 캡슐화, 상속, 다형성등이 잘 적용된 객체지향언어이다.
3. 보안성이 뛰어나다.
4. Garbage Collector가 자동적으로 메모리를 관리해주는 자동 메모리 관리(Garbage Collection)이다
5. 멀티 쓰레드(Multi-thread)를 지원한다.
- 6.미리 설치되지 않은 프로그램이 서버로부터 동적으로 다운로드 되어 웹 브라우저에서 실행할 수 있도록 동적인 성능 확장을 제공한다.
7. 분산 환경에 적합하여 네트워크 수행 능력이 뛰어나다.

자바의 가장 대표적인 단점이 속도인데 바이트 코드를 하드웨어의 기계어로 바로 변환해주는 JIT컴파일러와 Hotspot과 같은 신기술의 도입으로 JVM의 기능이 향상됨으로써 속도문제가 상당히 개선되었다.

## 23 .UML 이란?

UML은 Unified Modeling Language의 약어로 '객체모델링 언어' 또는 '통합모델링 언어'를 뜻한다. 시스템 설계, 요구분석, 시스템 구현등의 과정에서 사용되는 모델링 언어로 표기법의 표준화를 목적으로 한 것이다. 기존에 사용되어 오던 표기체계를 통합하여 모델링이 필요한 모든 분야에 적용 가능한 범용성이 높은 언어다. UML은 시스템에 대해 동일한 의미를 공유할 수 있게 하여 언어를 가시화 시킬 수 있고 시스템 구조와 모든 상세 내역에 대해 문서화하여 모델링하는 기능 등을 제공한다. UML의 구성은 다양한 모델링 도구로서의 다이어그램 들로 이루어져 있다. 주요 다이어그램으로는 다음을 들 수 있다.

-클래스 다이어그램(Class Diagram) : 가장 많이 이용되는 다이어그램. 클래스의 내부구조와 클래스 사이의 관계를 모델링 할 때 사용.

-객체 다이어그램(Object Diagram) : 클래스 다이어그램을 보충하기 위하여 사용.

-컴포넌트 다이어그램(Component Diagram) : 소프트웨어 컴포넌트 사이의 정렬과 의존을 나타냄.

## 24. 스레드(Thread) 정의

: 프로세스내에서 동시에 실행되는 독립적인 실행 단위를 말함, 장점으로는 자원을 많이 사용하지 않고 구현이 쉬우며 범용성이 높다

## 25. 스레드(Thread) 특징

- 어플리케이션의 성능을 향상시킬수 있고, 구현이 쉽다. 즉 작은 투자로 큰 효과를 기대.
- 스레드를 활용한 대표적인 예) 웹 브라우저
- 스레드를 쓰지 않고 프로세스나 다른 방법을 통하여 비슷하게 구현 할 수 있다.

반면 대부분의 자원을 많이 소모하여 구현이 복잡해지고 범용성이 떨어지는 등의 문제점이 있다. 스레드를 사용함으로써 이 문제들을 해결할 수 있다.

스레드는 각자의 스택 메모리영역을 가지고 있다. 프로세스와 달리 스레드는 동일한 프로세스내의 다른 스레드들과 전역메모리를 공유한다. 이에 반해 프로세스는 자기 자신의 메모리 영역을 가진다.

## 26. 스레드(Thread)의 장점 / 단점

장점 : ①빠른 프로세스 생성 ② 적은 메모리 사용 ③ 쉬운 정보 공유

단점 : 교착상태에 빠질 수 있다.

-스레드는 각자의 스택 메모리영역을 가지고 있다. 프로세스와 달리 스레드는 동일한 프로세스내의 다른 스레드들과 전역메모리를 공유한다. 이에 반해 프로세스는 자기 자신의 메모리 영역을 가진다.

①빠른 프로세스 생성 ② 적은 메모리 사용 ③ 쉬운 정보 공유

-스레드는 작업 디렉토리, 파일 지시자들, 대부분의 전역변수와 데이터, UID와 GID 같은 자원들을 서로간에 공유하고 있기 때문에 자원을 프로세스 복사(copy-on-write)방식으로 자식 프로세스에게 복사하여 사용하는 프로세스(fork) 방식에 비해 더 빠른 수행능력을 보여

주고 있다. 하지만 다중스레드 방식의 프로그래밍에서는 매우 정교한 실패나 의도하지 않은 변수의 공유로 인해 야기되는 실패가 발생할 가능성이 매우 높다. 그 대표적인 예)교착 상태

-교착상태 : 다중프로그래밍 체제에서 하나 또는 그 이상의 프로세스가 수행 할 수 없는 어떤 특정시간을 기다리고 있는 상태. 망에서는 사용 가능한 버퍼가 없어 일단의 노드들이 패킷을 전송할 수 없는 상태이다. 데이터 통신망에서 송신기 및 수신기간에 전송할 정보는 있지만 송수신에 필요한 자원이 서로 맞물려 있어 송수신이 이루어질수 없는 현상. 교착상태에는 직접, 간접 및 재조립 교착상태가 있다.

## 27. 스레드와 프로세스의 차이점은?

운영체제에서 실행중인 하나의 프로그램을 프로세스(Process)라고 한다. 여러 분야에서 '과정' 또는 '처리'라는 뜻으로 사용되는 용어로 컴퓨터 분야에서는 '실행중인 프로그램'이라는 뜻으로 쓰인다. 이 프로세스 내에서 실행되는 각각의 일을 스레드라고 한다. 프로세스 내에서 실행되는 세부 작업 단위로 여러 개의 스레드가 하나의 프로세스를 이루게 되는 것이다.

## 28. JVM이란?

-자바 프로그램은 자바 컴파일러에 의해 바이트 코드로 번역됨. 이 번역된 바이트 코드는 인터프리터에 의해 해석되고 실행됨. 여기서 자바 바이트 코드를 기계어로 사용하는 컴퓨터를 자바 가상기계라고 함. 즉, 자바의 바이트 코드를 해석하여 실행하는 해석기가 설치된 컴퓨터는 모두 자바 가상 기계가 될 수 있음.

## 29. SI란 무엇인가?

-SI란 System Integration의 약자로 시스템 통합을 말한다. 기업이 필요로 하는 정보 시스템에 관한 기획과 개발, 나아가 운영에 이르기까지 모든 서비스를 제공하는 것이다. 시스템 통합은 기업 내의 각종 어플리케이션이나 전산 자원들을 통합·연계하여 전체를 하나의 시스템처럼 유기적으로 조정할 수 있도록 하기 위함이다.

## 30. 객체에 대해 설명 해보시오.

객체(Object)는 객체 지향 프로그래밍(OOP)이나 설계에서 데이터(실체)와 그 데이터에 관련되는 동작 즉 절차, 방법, 기능을 모두 포함한 개념이다. 기차역에서 승차권을 발매하는 경우를 예로 들면 실체인 '손님'과 동작인 '승차권 주문'은 하나의 객체이며, 실체인 '역무원'과 동작인 '승차권 발매'도 하나의 객체이다. 같은 성질 즉 같은 구조와 형태를 가지는 객체는 등급으로 정의하고 같은 등급에 속하는 객체는 그 등급의 인스턴스(instance)라고 한다.

## 31. 클래스 파일에 대해 설명해보시오.

JAVA(이하 자바)는 인터프리터 언어와 컴파일 언어의 특성을 모두 가지고 있는 언어이다. 자바 언어를 컴파일 했을 때 나오는 결과물이 클래스 파일(class file)이다. 클래스 파일은 온

전한 실행 파일이 아니기 때문에 자바 버추얼머신(JVM)이 한번 더 컴파일을 했을 때에야 제대로 된 실행을 할 수 있다. 즉, 클래스 파일은 소스파일과 실행파일의 중간단계로 볼 수 있다. 두 번 컴파일을 해야하기 때문에 속도가 느리다는 단점이 있지만, JVM만 있다면 플랫폼을 가리지 않고 실행할 수 있다는 장점을 가지고 있다.

### 32. 인터넷과 인트라넷의 차이점은?

**인터넷(internet):** 전 세계를 연결하는 거대한 컴퓨터 통신망을 말한다. 알파넷(ARPANET)에서 시작되어 현재 세계 최대 규모의 통신망으로 발전하였다. 통신량이 급속도로 증가하고 있으며 정보의 기반으로 자리잡아 다양한 서비스와 정보를 지원한다. 정보 검색, 전자우편, 뉴스, 게임 등 새로운 서비스가 차례로 개발되어 이용이 가능해지며 사용자의 편의를 도모하고 있다.

**인트라넷(intranet):** 인터넷을 이용하여 일정 지역 내에서 정보를 교환하거나 제공하는 시스템을 말한다. 인터넷망 접속보다는 기업 내부의 네트워크에 기술을 이식하여 사용하기 때문에 별도의 통신망을 구축하지 않아도 어디에서나 조직의 정보시스템을 확인 할 수 있다. 서로 다른 속성을 가진 정보시스템이라도 통합이 용이하며 적은 비용으로 큰 성과를 얻을 수 있다. 기술표준을 이용하기 때문에 폭넓은 적용이 가능하지만 정보 보안에 대한 문제가 단점으로 제기된다.

### 33. Overloading과 Overriding의 차이점은?

- Overloading: 같은 이름의 메소드를 여러 개 정의하는 것.
  - 단, 매개변수의 타입이 다르거나 개수가 달라야 한다.
  - return type과 접근 제어자는 영향을 주지 않는다.
- Overriding: 상속에서 나온 개념.
  - 상위 클래스의 메소드를 하위 클래스에서 재정의.

### 34. JDBC 코딩 단계는?

-DriverLoading>>Connection>>SQL작성>>Statement>>ResultSet>>Data get>>close

### 35. JDBC에서 Transaction 관리는 어떻게 하나요?

- conn.setAutoCommit(false);
- 정상 실행: conn.commit();
- exception 발생: conn.rollback();

### 36. JDBC에서 Transaction을 처리할 때, setAutoCommit란?

→Transaction이란 일련의 작업 단위이다. 프로그램 내에서 쿼리문을 몇 번 실행하던지, 그 안에 몇 개의 쿼리나 몇 개의 프로시저가 있더라도 Commit이나 Rollback을 만날 때까지 하나의 작업 단위가 된다. 이렇게 보면 Transaction은 Select를 제외하고, Insert, Update, Delete문을 실행하고 Commit이나 Rollback을 할 때까지의 작업 단위로 볼 수 있다.

-conn=커넥션 생성;

```
conn.setAutoCommit(false); //AutoCommit을 false로 설정
```

```
try {
```

```
    insert 실행
```

```
    update 실행
```

```
    delete 실행
```

```
    conn.commit(); //Commit 실행
```

```
} catch(Exception e) {
```

```
    try {
```

```
        conn.rollback(); //Exception 발생시 Rollback 처리
```

```
    } catch (Exception e1) {}
```

```
}
```

커넥션 해제;

### 37. setAutoCommit(true);인 경우와 setAutoCommit(false);인 경우의 차이점은?

-setAutoCommit(true);인 경우 쿼리문을 실행할 때마다 Commit처리가 된다. 비정상적으로 종료되기 전까지는 실행한 쿼리문이 모두 반영되어 있다는 뜻이다.

-setAutoCommit(false);이면, 코드상에서 직접 Commit이나 Rollback을 해주어야만 실행한 쿼리문이 반영된다. Rollback 실행 후에 추가적으로 Commit을 할 필요는 없다.

### 38. statement와 preparestatement의 차이점은?

-동적으로 쿼리문이 생성된다면, statement를 사용.

-쿼리문이 변하지 않는다면, preparedStatement를 사용.

### 39. Servlet과 JSP의 차이점에 답변해 보세요.

-Servlet: container가 이해할 수 있게 구성된 순수 자바 코드로만 이루어진 것(html in java)

-JSP: html기반에 자바코드를 블록화하여 삽입한 것(java in html)

#### 40. 엔터프라이즈 자바빈즈란?

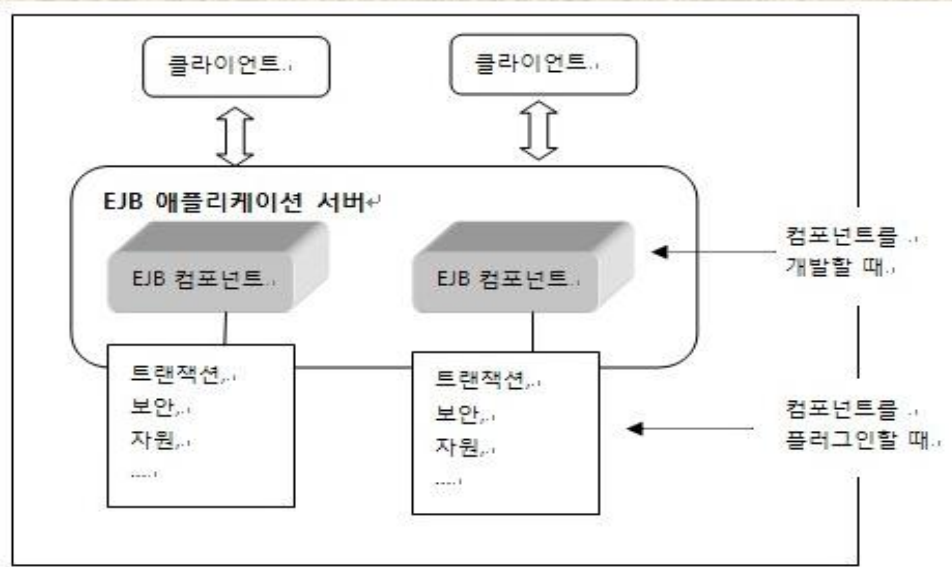
-엔터프라이즈 자바빈즈는 기업 환경의 시스템을 구현하기 위한 서버측 컴포넌트 모델이다. 즉, EJB는 애플리케이션의 업무 로직을 가지고 있는 서버 애플리케이션이다. 주로 웹시스템에서 JSP는 화면 로직을 처리하고, EJB는 업무 로직을 처리하는 역할을 한다.

자바언어로 분산 객체 형태의 비즈니스 애플리케이션을 개발하기 위한 표준 컴포넌트 아키텍처이다.

자바로 분산 객체 형태의 엔터프라이즈용 비즈니스 애플리케이션을 쉽게 만들기 위해서 EJB 아키텍처가 만들어졌다.

일반적으로 비즈니스 애플리케이션은 크게 프로세스와 데이터 부분으로 나뉜다. 프로세스라는 것은 비즈니스 업무의 단위이며 데이터는 업무를 수행하기 위해 필요한 데이터 및 정보들을 의미하며, EJB 아키텍처는 비즈니스 애플리케이션의 "프로세스"와 "데이터"를 안정적으로 처리하기 위한 모델을 제시한다. EJB 아키텍처에서 "프로세스"는 "세션 빈(Session Bean)", "데이터"는 "엔티티 빈(Entity Bean)"이라는 EJB 컴포넌트들이 담당하게 된다.

EJB 애플리케이션 서버의 구조



#### 41. JDBC란?

-JDBC(Java Data Base Connection)의 약자로 Java Language에서 데이터 베이스에 접근할 수 있는 프로그래밍을 의미한다.



## 42. Get과 Post의 차이점은?

### ①GET과 POST의 차이점

GET은 주로 웹 브라우저가 웹서버에 데이터를 요청할 때 사용되고, POST는 웹브라우저가 웹 서버에 데이터를 전달하기 위해서 사용된다. GET을 사용하면 웹브라우저에서 웹서버로 전달되는 데이터가 인코딩되어 URL에 붙는다. 반면에 POST방식은 전달되는 데이터가 보이지 않는다. 프로그래밍에서 GET방식을 사용하는 경우에는 전달되는 데이터가 255개의 문자를 초과하면 문제가 발생할 수 있다. 따라서 웹서버에 많은 데이터를 전달하기 위해서는 POST 메소드를 사용하는 것이 바람직하다.

### ②Get방식의 특징

- 클라이언트에서 서버로 데이터를 전달할 때, 주소 뒤에 "이름"과 "값"이 결합된 스트링 형태로 전달된다.
- 주소창에 쿼리 스트링이 그대로 보여지기 때문에 보안을 요하는 전송에는 부적합하다.
- 길이에 제한이 있다.(전송 스트링이 너무 길면, 255글자 이상은 전송이 안된다.)
- post방식보다 상대적으로 속도가 빠르다.
- HTML FORM에서 method="get/post"를 지정하지 않으면 기본으로 GET방식으로 넘어간다.

### ③POST방식의 특징

- 일정 크기 이상의 데이터를 보내야 할때 사용한다.(255글자 이상의 긴 정보를 보낼 때 사용한다.)
- 서버로 보내기 전에 인코딩(일종의 암호화)하고, 전송 후 서버에서는 다시 디코딩(원상복귀) 작업을 한다.
- 속도가 GET 방식보다 느리고, 주소창에 보이지 않으니, 좀 더 안전하다.
- 쿼리스트링(문자열) 말고, "라디오 버튼", "텍스트 박스", "텍스트 에어리어"같은 객체들의 값도 보낼 수 있다.

## 43. Session과 Cookie의 차이점과 사용용도?

-사용이유: 현재 우리가 인터넷에서 사용하고 있는 HTTP프로토콜은 연결 지향적인 성격을 버렸기 때문에 새로운 페이지를 요청할 때마다 새로운 접속이 이루어지며 이전 페이지와 현재 페이지 간의 관계가 지속되지 않는다. 이에 따라 HTTP프로토콜을 이용하게 되는 웹 사이트에서는 웹페이지에 특정 방문자가 머무르고 있는 동안에 그 방문자의 상태를 지속시키기 위해 쿠키와 세션을 이용한다.

①Session: 특정 웹사이트에서 사용자가 머무르는 기간 또는 한 명의 사용자의 한번의 방문을 의미한다. Cookie와는 다르게 개인 아이디와 비밀번호와 같은 웹사이트의 데이터는 Client가 아닌 Server에 저장되거나 웹브라우저의 캐시에 저장되어 브라우저가 닫히거나 서버에서 삭제시 사라진다.

- ②Cookie: 사용자 정보를 유지할 수 없다는 HTTP고유의 한계를 극복할 수 있는 방법으로 인터넷 웹사이트의 방문 기록을 남겨 사용자와 웹사이트 사이를 매개해 주는 정보이다. 이러한 Cookie는 인터넷 사용자가 특정 웹서버에 접속할 때, 생성되는 개인 아이디와 비밀번호, 방문한 사이트의 정보를 담은 임시 파일로써, Server가 아닌 Client에 텍스트 파일로 저장되어 다음에 해당 웹서버를 찾을 경우 웹서버에서는 그가 누구인지 어떤 정보를 주로 찾았는지 등을 파악할 때 사용된다.

#### 44. 구구단 출력 프로그램을 코딩할 수 있나요?

```
①class Gugudan1 {
    public static void main(String[] args) {
        int dan = 3;
        System.out.println("***" + dan + "단***");
        for(int i=1;i<10;i++){
            System.out.println(dan + "*" + i + "=" + dan*i);
        }
    }
}

②class Gugudan2 {
    public static void main(String[] args) {
        System.out.println("***구구단1***");
        for (int dan=2;dan<10;dan++){
            for(int i=1;i<10;i++){
                System.out.print(dan + "*" + i + "=" + dan*i + "\t");
                if(i==9)
                    System.out.println("");
            }
        }
        System.out.println("\n\n***구구단2***");
        for (int i=1;i<10;i++){
            for(int dan=2;dan<10;dan++){
                System.out.print(dan + "*" + i + "=" + dan*i + "\t");
                if(dan==9)
                    System.out.println("");
            }
        }
    }
}
```

#### 45. String과 StringBuffer의 차이점은?

```
public class TimeTest {
    public static void main(String[] args){
        String str1 = "abc";
```

```

String str2 = "def";
long t1 = System.currentTimeMillis();
for(int i=0;i<10000;i++){
    str1 += str2;
}
System.out.println(str1);
System.out.println("time == " + (System.currentTimeMillis()-t1));

StringBuffer str1 = new StringBuffer("abc");
long t1 = System.currentTimeMillis();
for(int i=0;i<10000;i++){
    str1.append("def");
}
System.out.println(str1.toString());
System.out.println("time == " + (System.currentTimeMillis()-t1));
}
}

```

## String 클래스

### 1. 성격

- final 클래스이므로 상속시킬 수 없다.
- 내부 문자열 데이터의 수정이 불가능하다.

```

public final class 클래스이름 {
} // 상속 불가

```

### 2. 특징

- 문자열은 메모리에 한 번 생성되면 같은 문자열 객체를 공유한다.

```

String str1 = "TEST";
String str2 = "TEST";
String str3 = new String("TEST");

// 힙영역에서의 주소(위치) 비교
if(str1==str2)
    System.out.println("동일한 객체!"); // 출력 될
else
    System.out.println("다른 객체!");

// 힙영역에서의 주소(위치) 비교
if(str1==str3)
    System.out.println("동일한 객체!");
else
    System.out.println("다른 객체!"); // 출력 될
// str1과 str2는 서로 같은 주소를 가진다.
// 그러나 str3는 new 키워드를 사용해 새로 생성했기 때문에 다른 주소를 가진다.

// 문자열 값을 비교
if(str1.equals(str3))
    System.out.println("동일한 문자열!"); // 출력 될
else
    System.out.println("다른 문자열!");

```

## StringBuffer 클래스

### 1. 특징

- 문자열을 처리하는 클래스이다.
- String 클래스에서는 내부 문자열 데이터의 수정이 불가능했다.
- StringBuffer 클래스에서는 문자열을 사용할 때, 문자열을 실행시간에 변경할 수 있다.
- 가변적인 길이를 가지므로 객체를 생성하는 시점에서 미리 그 크기나 값을 지정할 수 있다.
- 또한, 실행시간에 버퍼의 크기를 바꾸는 기능도 제공한다.

### 2. String과 StringBuffer의 차이점

```
// String class
// 메모리 상에서 "대한" 문자열 생성
String str = "대한";
// 메모리 상에서 "대한민국" 문자열 생성. 기존의 "대한"은 가비지컬렉션 대상.
// 즉, 문자열의 잦은 변경시 많은 가비지(쓰레기)가 생성된다.
str += "민국";
System.out.println(str);

// StringBuffer class
// 메모리 상에서 "대한" 문자열 생성
StringBuffer sb = new StringBuffer("대한");
// 메모리 상에서 "대한" 문자열에 "민국" 문자열을 추가.
sb.append("민국");
System.out.println(sb.toString());
```

## 46. 모델 1,2 MVC패턴에 대해 설명해보세요.

- ① 객체지향프로그래밍에서, MVC란 사용자 인터페이스를 성공적이며 효과적으로 데이터 모형에 관련 시키기 위한 방법론 또는 설계 방식중 하나이다. MVC방식은 자바, Smalltalk, C 및 C++ 등과 같은 프로그래밍 언어를 쓰는 개발 환경에서 널리 사용된다. MVC 형식은 목적 코드의 재사용에 유용한 것은 물론, 사용자 인터페이스와 응용프로그램 개발에 소요되는 시간을 현저하게 줄여주는 형식이라고 많은 개발자들이 평가하고 있다.
- ② MVC 형식은 소프트웨어 개발에 사용될 세 가지 구성요소 또는 객체를 제안한다.
- Model: 소프트웨어 응용과 그와 관련된 고급 클래스 내의 논리적 데이터 기반 구조를 표현. 이 목적 모형은 사용자 인터페이스에 관한 어떠한 정보도 가지고 있지 않다.
  - View: 사용자 인터페이스 내의 구성요소들을 표현하는 클래스들의 집합.  
(누름단추, 표시 상자 등과 같이 사용자가 화면상에서 보고 응답할 수 있는 모든 것들)
  - Controller: 모형과 뷰를 연결하고 있는 클래스들을 대표하며, 모형과 뷰 내의 클래스들 간에 통신하는데 사용됨.

## 47. 서블릿의 라이프 사이클에 대해 설명하세요.

- ① 서버를 실행시키는 순간 서블릿의 생성자가 호출된다.
- ② 생성자를 호출해서 객체가 생성되면 초기화 작업을 한다.(init() 메소드가 최초 한번만 자동적으로 호출된다.)
- ③ 초기화 작업 후에는 Service메소드가 호출이 된다.(req, res를 통해 data get, logic 구현

등의 작업이 일어난다. -> Service메소드는 클라이언트가 요청할 때마다 호출이 된다.)

- ④ 마지막으로 서버가 종료되는 시점에 destroy()메소드가 호출된다.(혹은 서버 재시작 시에 자원 반납을 위해서 호출된다.)

#### 48. EJB에 대해 설명해보시오.

EJB란 Enterpris java Beans(기업자바빈)의 약자로 애플리케이션의 업무 로직을 가지고 있는 서버 애플리케이션이다. 이것은 시스템의 유연성을 높이고 수월한 객체 환경을 구축하기 위해 만들어진 기술이며, 신뢰성 있는 N-tier 시스템을 쉽게 구축하기 위해 개발되었다. 그리고 이것은 컴포넌트 기반으로 작성되기 때문에 자바 RMI를 사용한다. EJB 아키텍처를 적용한 프로그램은 모든 소스 코드를 이해할 필요 없이 해당 컴포넌트에 대한 비즈니스 메소드의 기능만 이해하면 된다는 장점을 가진다. EJB의 종류에는 세션 빈(Session Bean) 엔티티 빈(Entity Bean) 메시지 구동 빈(Message-driven Bean)등이 있다.

#### 49. XML이란?

XML(extensible markup language)이란 구조화된 문서를 웹상에서 구현할 수 있는 인터넷 프로그래밍 언어를 말한다. HTML은 웹 상에 문서정보를 구현하는데 많은 역할을 해왔으나 날로 늘어가는 정보들을 좀 더 구조화 된 형태로 표현하기에는 한계가 있었다. 반면 문서의 구조적인 정보를 갖고 있는 XML 문서는 웹상에서 검색의 용이함과 체계적이고 일관적인 접근, 재가공의 용이함 때문에 HTML과 인터넷 표준 문서 포맷이 되었다. XML은 구조적인 정보만을 갖고 있기 때문에 HTML처럼 편집기를 통해 간단하게 조작하기가 어려워 일반인들이 만들기는 어려운 점이 많다. XML은 현재 기업 간 전자상거래(B2B)의 문서 양식으로 쓰이고 있으며 소규모의 데이터베이스를 대체하는데 쓰이기도 하고 있다.

#### 50. ODBC와 JDBC에 대해 설명해보시오.

ODBC란 윈도우즈 프로그램에서 다양한 데이터베이스 관리 시스템(DBMS)에 접근하여 사용할 수 있도록 개발한 표준 개방형 응용 프로그램 인터페이스(API)규격을 말한다. ODBC는 서로 다른 데이터베이스에서 구축된 데이터를 마이크로 소프트 윈도우 환경에서 상호간에 공유할 수 있게 하며, 비록 데이터베이스가 교체되더라도 응용 시스템은 그대로 사용할 수 있게 해준다. 반면 JDBC는 자바 프로그램 내에서 데이터베이스를 연결하여 SQL(구조화 조회 언어)로 실행할 수 있도록 하는 자바 API(application programming interface)이다.

자바와 JDBC의 결합은 기능을 확장하여 하나의 프로그램이 어디에서나 실행되게 하며 응용 업무를 만드는데 최적의 언어라 할 수 있는 자바와 다양한 데이터베이스를 연결하는데 사용되는 도구가 JDBC인 것이다.

## 51. CS 기반과 웹기반의 차이점은?

클라이언트 서버 시스템(Client/Server System) 즉 C/S 시스템을 네트워크 아키텍처를 말한다. 이것은 이름 그대로 클라이언트와 서버로 나눌 수 있는 시스템인데 클라이언트(Client)는 '고객'의 뜻을 가진 단어에서 가져온 것으로 서비스를 사용하는 사람이나 사용자의 단말기를 의미하며, 서버(Server)는 그러한 서비스를 제공하는 컴퓨터를 의미한다. 이 C/S 시스템은 웹 시스템 이전의 시스템이다. 한편, 웹 시스템은 다른 매체와는 비교할 수 없을 정도로 강력한 툴이다. 인터넷이라는 곳은 공간을 초월하여 여러 정보를 전달, 공유할 수 있기 때문인데 이러한 이유로 기존 C/S 시스템에서 웹 기반으로 바뀌는 움직임이 많았다.

C/S 기반과 웹기반의 차이점을 비교하면 다음과 같다. 먼저 GUI(Graphical user interface)면에서 C/S는 문자나 숫자 자료만 가능한 제약이 있고 화면 구성도 단순한 반면 웹에서는 HTML기반의 문자나 이미지 영상 등 다양한 자료를 이용할 수 있다. UI처리에 있어서도 브라우저에 의해 통일성 있게 할 수 있으며 구성 또한 멀티미디어 자료에 의해 다양하게 하는 것이 가능하다. 그리고 C/S의 개발도구는 표준화 되지 않은 툴로 다양한 개발도구가 지원되지만 웹기반에서는 다양성이 부족하다. 또한 C/S의 사용자는 LAN을 기반으로 하는 조직 내부의 사용자들에 한하지만 웹기반에서는 다양성이 부족하다. 또한 C/S의 사용자는 LAN을 기반으로 하는 조직 내부의 사용자들에 한하지만 웹기반에서는 인터넷을 사용하는 사람이라면 어느 곳의 사용자나 가능하다. 유지면에서 C/S는 클라이언트 모듈도, 관리도 별도로 필요하지만 웹기반에서는 그렇지 않다. 그리고 C/S는 이식성이 약한데 웹기반은 이식성이 강하고 C/S는 사용자에게 사용될 때마다 별도로 교육을 해야하지만 웹기반은 웹교육만이 필요하다.

## 52. Interface란?

일종의 추상클래스이다. 오직 추상메소드와 상수만을 멤버로 가진다. Implements 키워드 사용한다. 상속의 관계가 없는 클래스간에 서로 공통되는 로직을 구현하여 쓸 수 있도록함. Extends는 하나의 클래스만 상속가능하나 인터페이스는 다중상속 가능하다.

## 53. Abstract 란?

추상메소드를 하나 이상 가진 클래스. 자신의 생성자로 객체 생성 불가능. 하위 클래스를 참조하여 상위 클래스의 객체를 생성. 하위 클래스를 제어하기 위해 사용

## 54. Interface vs. Abstract Class 용도는?

공통점 : new로 인스턴스 생성이 안됨. 프로토타입만 있는 메소드를 가짐. 사용하기 위해서는 하위클래스에서 확장/구현 해야 한다.

차이점 : 사용하는 키워드가 다르다. Abstract는 일반 메소드를 사용할 수 있지만 interface는 메소드 선언만 가능하다. Interface는 다중상속이 가능하다.

### 55. overloading과 overriding의 차이점

overloading : 같은 이름의 메소드를 여러개 정의 하는것. 단, 매개변수의 타입이 다르거나 갯수가 달라야 한다. return type과 접근 제어자는 영향을 주지 않는다.

overriding : 상속에서 나온개념. 클래스의 메소드를 하위 클래스에서 재정의.

### 56. Inner Class란? 용도는?

하나의 클래스안에 다른 클래스를 정의. 클래스의 모든 멤버 참조 가능(private도 가능). 내부 클래스를 생성하기 위해서는 먼저 외부 클래스로 객체를 생성해야 한다.

서로 독립적인 관계가 아닌 하나의 멤버처럼 사용될 수 있는 경우. 대표적으로 GUI와 GUI 이벤트를 구현할 때 사용되어질 수 있다.

### 57. Reflection이란?

클래스의 이름으로부터 클래스의 정보를 취득하고 그 인스턴스를 생성하는 방법. jdbc드라이버 로드 방법도 리플렉션 패키지를 사용한 방법임. 그 외 Struts의 다이내믹 액션폼, 스프링에서 쓰임.

### 58. J2ME vs. J2SE/J2EE 차이점 및 특징?

Java 2 Standard Edition – 기본적인 자바의 개발/실행 환경으로 어플리케이션, 애플릿, 컴포넌트 등을 개발하고 실행할 수 있는 환경을 제공하는 플랫폼

Java 2 Enterprise Edition – 전사적 차원에서 필요로 하는 웹 어플리케이션 관련 기술 등으로 개발을 할 수 있는 라이브러리들이 포함되어 있다.

Java 2 Micro Edition – 가전제품이나 휴대폰 등 임베디드 디바이스의 제한된 환경에서 Java를 탑재하기 위한 기술을 제공한다.

### 59. MVC model이란?

객체지향프로그래밍에서, MVC란 사용자 인터페이스를 성공적이며 효과적으로 데이터 모형에 관련 시키기 위한 방법론 또는 설계 방식중 하나이다. MVC 방식은 자바, Smalltalk, C 및 C++ 등과 같은 프로그래밍 언어를 쓰는 개발 환경에서 널리 사용된다. MVC 형식은 목적 코드의 재사용에 유용한 것은 물론, 사용자 인터페이스와 응용프로그램 개발에 소요되는 시간을 현저하게 줄여주는 형식이라고 많은 개발자들이 평가하고 있다.

MVC 형식은 소프트웨어 개발에 사용될 세 가지 구성요소 또는 객체를 제안한다.

model : 소프트웨어 응용과 그와 관련된 고급 클래스 내의 논리적 데이터 기반 구조를 표현. 이 목적 모형은 사용자 인터페이스에 관한 어떠한 정보도 가지고 있지 않다.

view : 사용자 인터페이스 내의 구성요소들을 표현하는 클래스들의 집합

(누름단추, 표시 상자 등과 같이 사용자가 화면상에서 보고 응답할 수 있는 모든 것들)

controller : 모형과 뷰를 연결하고 있는 클래스들을 대표하며, 모형과 뷰 내의 클래스들 간에 통신하는데 사용됨

## 60. Call by reference, Call by value

Call by Value – 인수로 기본 데이터형을 사용. 주어진 값을 복사하여 처리하는 방식. 메소드 내의 처리 결과는 메소드 밖의 변수에는 영향을 미치지 않는다.

Call by Reference – 매개 변수의 원래 주소에 값을 저장하는 방식. 클래스 객체를 인수로 전달한 경우.

## 61. Static의 의미는?

정의 : 클래스가 로딩될 때 메모리 공간을 할당하는데 처음 설정된 메모리 공간이 변하지 않음을 의미

객체를 아무리 많이 만들어도 해당 변수는 하나만 존재. 객체와 무관한 키워드!

- static 멤버변수의 메모리는 한 번 만들어지므로 전체 객체에서 공유 할 수 있음.

## 62. Heap이란?

실행되는 동안 동적으로 할당 되는 메모리 영역. 객체나 배열 new 키워드로 할당된 객체들이 저장된다.

모든 객체들이 저장되는 공간이다. (인스턴스와 배열)

JVM의 모든 스레드들이 공유하며, JVM이 시작될 때 생성된다.

## 63. Stack 이란?

메서드가 호출될 때마다, 스택 프레임(stack frame)이라고 불리는 새로운 데이터 영역이 생성된다. 스택 프레임이 모여 자바 스택(java stack)을 생성한다.

스택의 최상단에 존재하는 스택 프레임은 활성화된 스택 프레임(active stack frame)이라고 부른다.

스택 프레임의 구조

- 피연산자 스택(operand stack)
- 지역 변수(로컬 변수)들을 저장하는 배열
- 현재 실행중인 명령어를 나타내는 포인터.

실행 순서

- a. 프로그램이 수행될 때는 활성화된 프레임의 피연산자 스택과 지역 변수 배열만 사용된다.
- b. 메서드가 호출되면, 새로운 자바 스택 프레임이 생성되고, 새로 생성된 프레임이 자바 스택의 최상단으로 오게 된다.
- c. 이전의 자바 스택 프레임에서는 모든 변수들과 프로그램 카운터를 저장한다.
- d. 새로운 자바 스택 프레임은 자기 자신의 프로그램 카운터를 생성하고, 이 카운터는 호출된 메서드의 시작 부분을 가리키게 된다
- e. 호출된 메서드가 종료되면 최상단 스택 프레임은 없어지며, 아래쪽에 있는 스택 프레임이 다시 활성화된 스택 프레임이 된다.
- f. 이 스택 프레임에서 프로그램 카운터는 메서드를 호출한 명령어의 다음 명령어를 가리키고 있을 것이고, 프로그램은 메서드를 호출한 부분의 다음부터 수행한다



## 64. 프레임워크란?

\* 특정 형태의 소프트웨어 문제를 해결하기 위해 상호 협력하는 클래스프레임과 인터페이스 프레임의 집합

\* 특정한 틀을 만들어놓고 거기에 살을 붙여 놓음으로써 프로그램을 만들어 작업시간을 줄여주는 것이다.

스켈레톤 코드라고도 하는데, 뼈대가 이미 만들어져 있어서 거기에 살(기능)만, 덧붙이면 완성이 되도록 공통된 함수 또는 클래스를 미리 만들어 놓는 것을 이야기한다.

-. 프레임워크는 특정 개념들의 추상화를 제공하는 여러 클래스나 컴포넌트로 구성된다.

-. 프레임워크는 이렇게 추상적인 개념들이 문제를 해결하기 위해 같이 작업하는 방법을 정의한다.

-. 프레임워크 컴포넌트 들은 재사용이 가능하다.

-. 프레임워크는 좀 더 높은 수준에서 패턴을 조작한다.

\* 프레임워크가 중요한 이유는 객체지향 개발을 하게 되면서 개발자의 취향에 따라 다양한 프로그램이 나오게 되었다. 프로그램 개발에 투입되는 개발자도 점점 늘어남에 따라 전체 시스템의 통합성, 일관성이 부족하게 되었기 때문이다. 그래서 개발자의 자유를 제한하기 위해 프레임워크를 도입했다.

-. 프레임워크가 가져야할 특징

a. 개발자들이 따라야할 가이드라인을 가진다.

b. 개발할 수 있는 범위가 정해져 있다.

c. 개발자를 위한 다양한 도구들이 지원된다.

\* 프레임워크의 장/단점

-. 장점 : 개발 시간을 줄일 수 있고 오류로부터 자유로울 수 있다.

-. 단점 : 프레임워크에 너무 의존하면 개발 능력이 떨어져서 프레임워크 없이 개발하는 것이 불가능해지는 점이다.

## 65. SPRING FRAMEWORK

EJB기반으로 개발을 하지 않고 POJO기반으로 개발을 하더라도 가볍고, 제어가 가능한 상호 관련이 적은 AOP를 지원한다. 컨테이너를 통해 라이프사이클을 관리하고 XML기반으로 컴포넌트를 개발할 수 있도록 지원하는 프레임워크