**로고학습일지**

**kt ds University 자바 기반의 데이터 사이언티스트 양성과정**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 07. 31 (화) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 | 자바, 데이터베이스 | | | | |
| 작 성 자 | 유재헌 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | **01. 예외처리**  **02. Oracle DB 설치**  **03. DB** |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| **01. 예외처리**  - 예외처리 : 프로그램이 동작하는 과정 중 잘못된 입력값이 전달되거나, 프로그램이 정상적으로 동작되지 않을 때,  “예외가 발생했다.” 혹은 “예외가 던저졌다.”라고 표현한다.  - Java로 작성된 프로그램에서 예외가 발생되었을 때는 JVM에 의해 프로그램이 종료된다  .  - Java 프로그램에서 예외가 발생하면 JVM에서 Exception을 보내게 되고, JVM은 Java 프로그램을 죽인다.  - 이때, Java 프로그램에서 JVM에게 Exception 보내는 것을 막는 역할이 예외처리이다.  - 예외처리는 크게 2가지로 나눠진다.  1) RuntimeException  2) Exception  - RuntimeException은 코드 복구가 가능하며 두 가지 선택을 할 수 있다.  1) 개발자에게 예외를 처리하라고 책임을 전가시킨다.  2) 예외를 JVM에게 떠넘긴다.  🡪 throws로 예외를 호출자에게 던진다.  🡪 try ~ catch ~ finally 구문을 통해 예외 소지가 있는 코드를 try구문에 작성하고  예외 발생은 catch 문에서 잡는다.  - Exception는 코드 복구가 불가능하고 예외 처리에 대해 JVM에게 책임을 전가시킨다.  즉, 개발자에게 예외에 대한 책임을 전가시킬 수 없다는 뜻이다.  - 여기서 RuntimeException이나 Exception에서 예외를 JVM에게 떠넘길 수 있는데, 2가지 방식으로 나워진다.    1) 메소드명 옆에 throws 예외명을 기입해서 전달하면 된다. (Add throws declaration)    - FileReader 클래스를 Add throws declaration을 선택하게 되면 실행하는 메소드 명 옆에 throws FileNotFoundException구문이 추가된다. 하지만 run() 메소드를 호출한 main 메소드에서 예외를 받았기 때문에 여기서도 예외를 JVM에게 예외를 던져야 한다. Main() 메소드를 호출하는 곳은 JVM이기 때문이다.  - 하지만 이런식으로 예외를 던지게 되면 프로그램 실행 도중 언젠가는 프로그램이 죽게 될 것이다.  - 그래서 try ~ catch문을 사용해야한다.      2) Surround with try/catch      - try{} 은 문제가 발생할 소지가 있는 코드를 작성하면 구문이다.  - catch문은 try문에서 문제가 발생했을 때 예외를 처리해주는 구문이다.  - 여기서 개발자가 예외 가능성이 있는 코드를 if문을 사용하여 예외를 방지해주고 개발자에게 해결못할 예외들은 try ~ catch문을 사용하여 코드를 작성한다..    - finally 구문은 try ~ catch 구문에 상관없이 무조건 실행한다. 즉, 예외가 발생하던 그렇지 않던 마지막에 반드시 실행되어야하는 코드가 있다면 finally를 이용한다.  - DB를 연결을 끊고 닫아줄 때 finally를 사용할 수 있다.  - 예외 중 절대 발생시키면 안되는 Case  1) NullPointerException : 주소값을 null로 가리킬 때 발생하는 예외  2) NumberFormatException : 문자를 숫자로 바꾸려고 할때 발생하는 예외  3) IndexOutofBoundsException : 리스트에서 인덱스의 범위가 넘어갔을 때 (잘못된 인덱스 참조했을 때)발생하는 예외  4) ArrayIndexOutofBoundsException : 배열에서 인덱스의 범위가 넘어갔을 때 (잘못된 인덱스를 참조했을 때) 발생하는 예외  **02. Oracle DB 설치**  < 준비 단계 >  1) Oracle 11g Expression  2) Tomcat 8.5  3) Toad for Oracle  - 먼저, [www.oracle.com](http://www.oracle.com) 사이트에 계정으로 로그인한 후 Oracle 11g Expression 설치 파일을 다운로드한다.  - tomcat.apache.org 사이트에 접속하여 Tomcat 8.5 64bit ZIP파일을 다운로드 한다.  - 구글에 Toad for Oracle을 검색하여 사이트에 접속한 후, Downloads -> Toad for Oracle 13.0.0 select -> LIMITED FREEWARE를 다운 받는다.  < 설치 단계 >  - tomcat 폴더 -> bin 폴더 -> startup.bat 실행한다.  - Oracle 설치 시,  TNS Port 1521  MTS Port 2030  HTTP Port 9999 🡺 Tomcat의 포트번호가 8080이기 때문에 충돌나서 Http 포트번호를 9999로 변경한다,  - Toad for Oracle을 설치한다.  **03. DB**  - RDBMS(관계형 데이터베이스) : 두 개 이상의 자료을 집합처럼 관계를 만드는 형태의 데이터베이스  - TABLE : 데이터 저장될 형태  - DBMS : 데이터베이스를 관리하는 시스템  - COLUMN : 관계형 데이터베이스에서 행(레코드)을 분류하는 기준  - 데이터베이스는 한 개 이상의 테이블로 이루어지고, 테이블은 한 개 이상의 컬럼을 가지는데, 컬럼은 반드시 PK가 있어야 한다.  - PRIMARY KEY(기본키)  🡪PK는 다른 항목과 절대로 중복되어 나타낼 수 없는 단일 값(unique)을 가진다.  🡪PK는 절대 NULL(아무런 값이 없는 상태) 값을 가질 수 없다.  예를 들어 주민등록번호 같은 개념이다. 동일한 이름을 가진 사람은 많을 수 있고, 동일한 날에 동일한 이름을 가진 사람도 존재할 수 있지만, 결국 그 사람들이 만나 서로 만나 서로의 민증을 대조해 보면 결국 다른 번호로 구분된다.  🡪 PK는 하나 이상의 컬럼이 그룹화 되어 기본키로도 쓰일 수도 있다.  🡪 PK의 값은 변경할 수 없다.  🡪 PK는 기본적으로 오름차순 정렬이다.  🡪 즉, 기본키는 각 레코드를 식별하는데 사용하고, 각 레코드를 다른 레코드와 구분하는 열이다.  - FORIGN KEY(외래키, 참조키)  🡪 FK는 테이블의 한열로 다른 테이블의 PK를 가리킨다.  🡪 두 테이블간으로 연결을 설정하고 강제 적용한다.(개체간의 무결성 유지 때문)  🡪 부모테이블이 먼저 만들어지고 자식테이블을 작성한다.  🡪 FK는 부모의 PK, UNIQUE만 설정할 수 있다.  🡪 참조제약조건이 설정된 경우 부모테이블에 없는 참조키값은 자식 테이블에 추가하거나 수정할 수 없다.  🡪 참조제약조건이 설정된 자식테이블에 존재하는 값을 부모테이블에서 제거할 수 없다.  🡪 PK는 NULL일 수 없지만, FK 값은 NULL일 수 있다.  🡪 FK 값이 NULL이라고 부모 테이블에 연결되는 PK가 없다는 것이 아니다.  - DISTINCT : 중복을 제거하는 키워드  🡪 한 ROW에서 중복되는 ROW가 있으면 중복 제거 해준다.  - ORDER BY[ASC | DESC ] : 데이터를 정렬해주는 함수이다.  🡪 QUERY문의 맨 뒤에 온다.  🡪 디폴트는 ASC이다. 즉, ASC, DESC 키워드를 입력하지 않으면 자동으로 오름차순 정렬된다.  🡪 참고로 오름차순 정렬은 : ASC, 내림차순 정렬은 DESC이다.  🡪 문자열은 ASC이면 A,B,C순으로 ASC코드 값이 작은 것부터이며,  DESC이면 C,B,A순으로 큰 것부터 정렬된다.  🡪 정수나 실수는 ASC이면 전체 수 중 작은 숫자부터이며,  DESC이면 전체수 중 큰 숫자부터 정렬된다.  🡪 날짜는 ASC이면 오래된 날짜부터 정렬되며  DESC이면 최근의 날짜부터 정렬된다.  🡪 한글은 ASC이면 가,나,다 순으로 정렬되며, DESC이면 다,나,가 역순으로 졍렬된다.  - BETWEEN 연산자는 특정 구간에 속한 모든 값을 출력한다.  🡪 예를 들어 급여가 1000~2000 사이의 값을 출력한다면 BETWEEN을 이용하여 출력할 수 있다.  🡪 하지만 실무에서 잘 사용하지 않는다.  🡪 그 이유가 BETWEEN 연산자는 속도가 느린 연산자에 속하므로 특정 범위에 속하는 값을 출력하는 경우 비교 연산자를 이용하는 것을 추천한다. |
|  |

|  |  |
| --- | --- |
| 특이사항 |  |