

17. JDBC(Java Database Connectivity)

//src: ch20_JDBC 참고

1) JDBC(Java Database Connectivity) 프로그래밍

- (1) 개념: 자바 패키지의 일부로 자바 프로그램이 데이터베이스(DB)와 연결되어 데이터를 주고 받을 수 있게 해주는 프로그래밍 인터페이스. '자바 데이터베이스 프로그래밍 API'
- (2) JDBC 드라이버: %JAVA_HOME%\JRE\LIB\EXT 폴더에 다음 Oracle(or MySQL) 드라이버 복사
- (3) 이클립스: Windows > preferences > Java > installed JREs > 클릭 후 edit > Add External JARs > 드라이버(ojdbc6.JAR) 등록 > [Restore Default] finish > Apply and Close
 //MySQL: mysql-connector-java-8.0.22.jar (MySQL버전에 따라 드라이버 파일 다를 수 있음)

2) JDBC프로그래밍 단계와 사용 클래스 //(1)→(7)순서대로 진행

String driver = "oracle.jdbc.driver.OracleDriver"; //드라이버 버전(8.0)에 따라 주소 다름
 String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe"; //@127.0.0.1=@localhost

- (1) **JDBC 드라이버 로드**: Class.forName(driver);
- (2) **DB연결**: DriverManager.getConnection(url, "id", "password"); 이용, Connection 객체 생성
- (3) **Statement 생성**: Statement or PreparedStatement API 이용, SQL문 수행할 객체 생성
- (4) **SQL문 전송**: DB로 전달해 처리
 - 조회(SELECT): executeQuery(); 반환 값은 ResultSet으로 받음: (4)+(5)
 - 입력(INSERT), 수정(UPDATE), 삭제(DELETE): executeUpdate(); //작업 성공 횟수(int) 반환
- (5) **결과 받기**: SQL 실행 결과를 받는 값이 있다면(=SELECT문일 경우) ResultSet 객체로 받음
- (6) 결과 받아서 **로직 수행**
- (7) **연결 해제**(DB연결 해제): close(); 닫는 순서는 진행 순서(=객체 생성 순서) 반대로 닫음

※테이블은 칼럼(속성, 필드)과 로우(entity)로 이루어져 있으며 서로 연관된 데이터들이 표로 구성