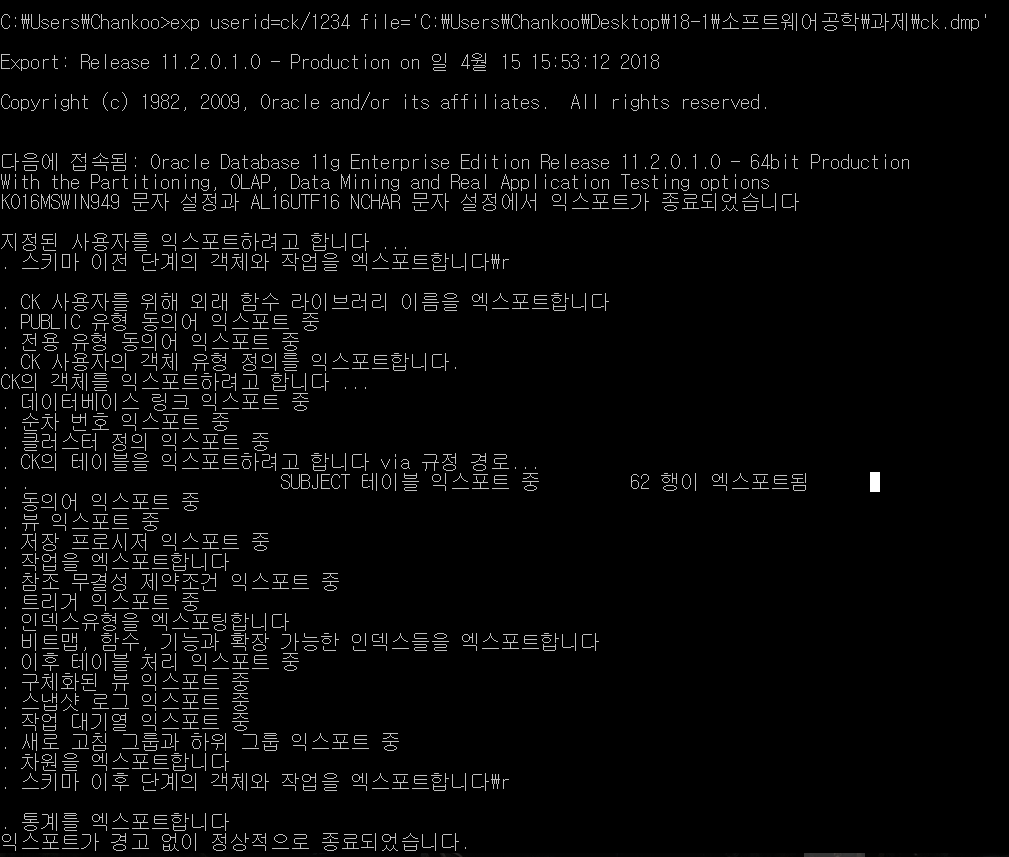
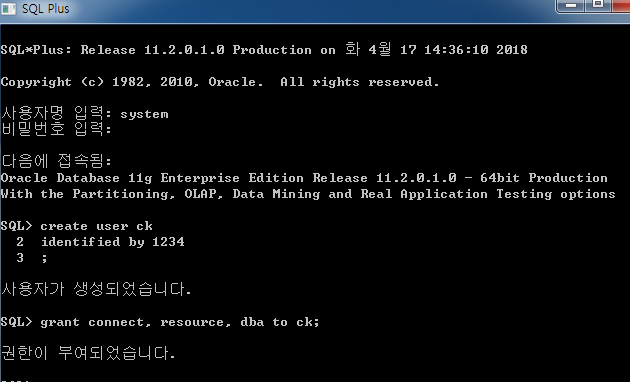
<실행 방법>

파일은 SE\_hw1.jar, ck.dmp, 그리고 doc인 SE\_hw1.docx로 구성됩니다.

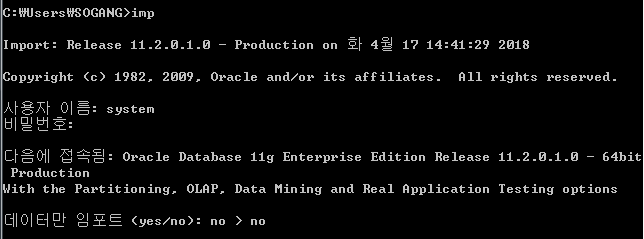
dmp파일은 아래와 같이 개설교과목정보 테이블인 SUBJECT를 유저 ck 수준에서 export한 파일입니다.



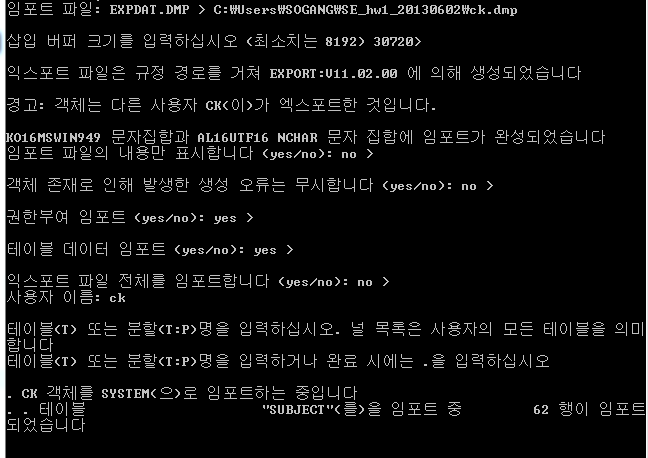
임포트를 위해서는 기존 DB에 사용자 계정이 생성 되어 있어야 합니다. DB생성 후에 계정이름 ck, 비밀번호 1234인 계정 ck를 생성합니다.



그리고 유저 ck에 권한을 부여합니다.



그런 후에 imp라는 프로그램을 실행하여 DB에 접속하고 각종 옵션을 설정한 뒤 임포트 파일의 경로를 입력합니다. 아래와 같이 ck.dmp가 저장된 경로를 넣어줍니다. (혹시 오류 나는 경우 imp userid=ck/1234 file=’~\ck.dmp’ full=y 명령어로 시도 바랍니다)



ck 유저의 테이블을 받아오기에 사용자 이름을 ck라 입력합니다. 임포트가 완료되면 jar 파일을 실행합니다. java -jar SE\_hw1.jar 명령어로 실행합니다. 실행 예시를 따로 첨부하였습니다.

<구현 과정>

학생정보 클래스인 StudentInfo를 중심으로 수강신청이 이루어집니다.

멤버변수는 학번인 id, 이름인 name, 과거수강정보인 pastSubject, 현재신청정보 presentSubject 입니다.

메소드는 saveInfo(), doRegit(), savePresentSbj(), showMySbj() 입니다.

\*개설 교과목 조회

학생객체와 상관없이 조회가능 하므로 ShowSbj 클래스를 따로 구현하여 showSbj() static메소드를 이용합니다.

\*수강신청

StudentInfo 클래스의 doRegit() 메소드를 이용합니다.

\*신청과목조회

StudentInfo 클래스의 showMySbj() 메소드를 이용합니다.

이밖에 db와의 연동위한 OraJava 클래스 구현하여 필요시마다 JDBC 드라이버를 로드하고 클로즈할 수 있게 하였습니다.

main을 보면

우선 StudentInfo의 객체를 생성합니다.

그리고 saveInfo() 메소드로 생성된 객체정보를 db에 저장합니다.

이 과정에서 해당 학번으로 table을 생성해 속성으로 과거수강과목과 현재신청과목을 저장합니다.

테이블에 기본키가 없고, 정규화 안되어있지만 현재 시스템을 구현하기에는 문제가 없습니다.

다음으로 개설교과목을 조회하는데 showSbj() 메소드를 이용합니다.

개설교과목은 간단한 구현을 위해 18-1학기 융합소프트웨어와 경제 전공과목으로 구성했습니다.

그리고 본격적으로 수강신청을 합니다. 이 과정에서 무한루프를 돌며

doRegit() 으로 신청과목코드와 분반을 입력 받아 임시배열 tempSbj에 저장합니다.

tempSbj에 저장된 과목을 db에서 조회하여 해당 과목의 강의시간과 선수과목을 tempTime, tempPrerequisite 임시배열에 각각 저장합니다.

무한루프 탈출조건(신청과목저장 조건)은 다음과 같습니다.

1) tempSbj의 사이즈가 3보다 크고 8보다 작아야 한다 - db에 저장된 신청가능교과목이 모두 3학점이기에 9학점이상 24학점이하라는 수강신청조건을 반영

2)tempTime에 String으로 저장된 강의시간 중 같은 것이 존재하지 않을 때

3)모든 신청과목의 선수과목이 pastSbj에 존재할 경우

1) 2) 3) 세조건을 모두 만족시킬때 break하며,

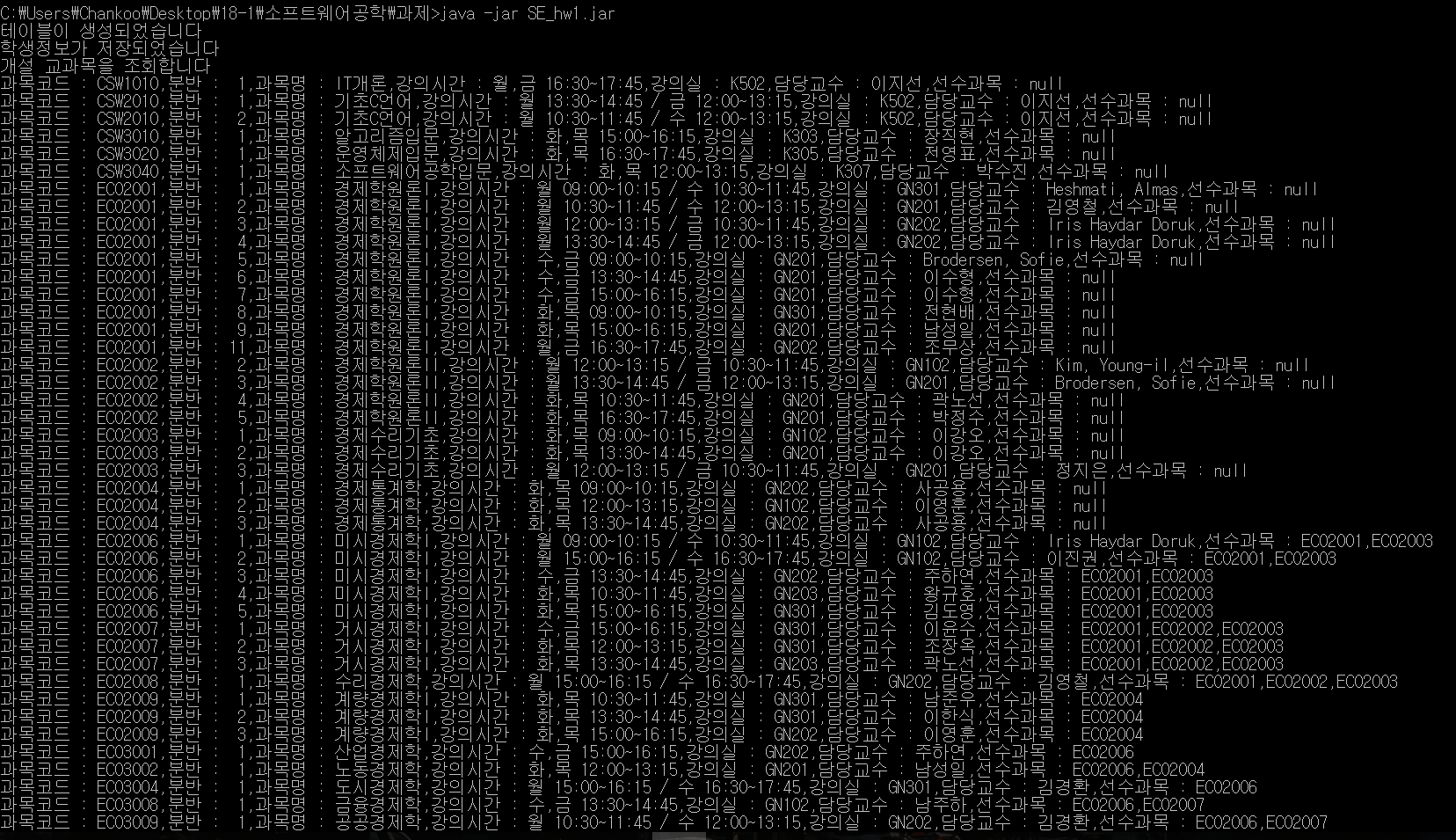
이후 StudentInfo 객체의 presentSubject에 임시신청과목 배열을 할당합니다.

그리고 savePresentSbj()으로 db의 해당 학번 테이블에 현재신청과목을 저장합니다.

마지막으로 showMysbj()으로 db에 저장된 신청과목 조회하고 시스템을 종료합니다.

<실행 예시>

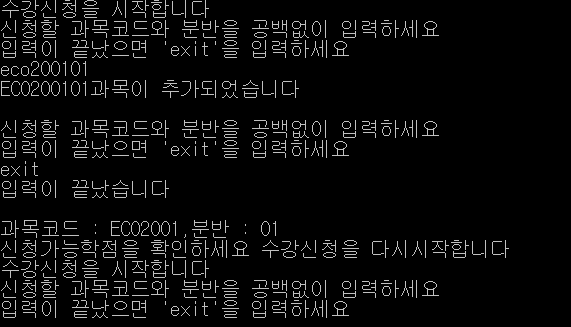
본 예시는 제 노트북에서 실행한 결과이나 다른 환경에서도 실행됨을 확인하였습니다.



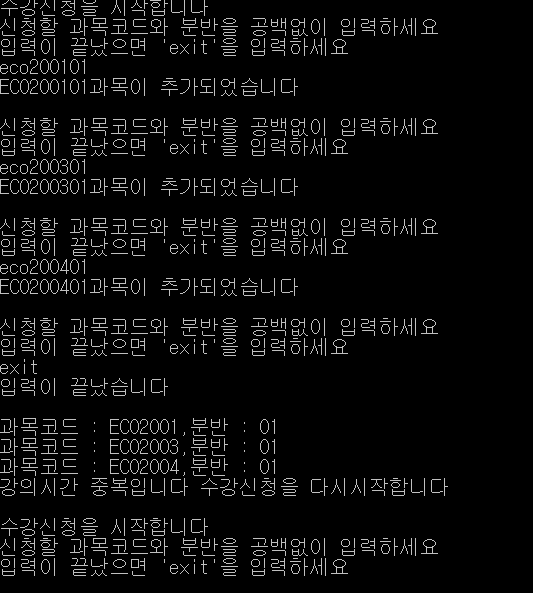


존재하지 않는 과목의 입력에 대해 ‘신청과목을 바르게 입력하세요’ 출력하며 예외 처리한다.

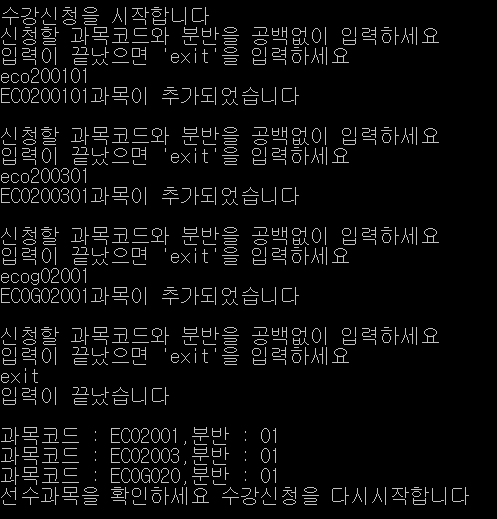
이후 수강신청이 정상적으로 완료되었다.



신청가능학점 조건을 만족하지 못한 경우 수강신청을 다시 시작한다.

ㄴ

강의시간 중복에 대해 수강신청을 다시 시작한다.



선수과목을 이수 못한 경우 수강신청 다시 시작한다.