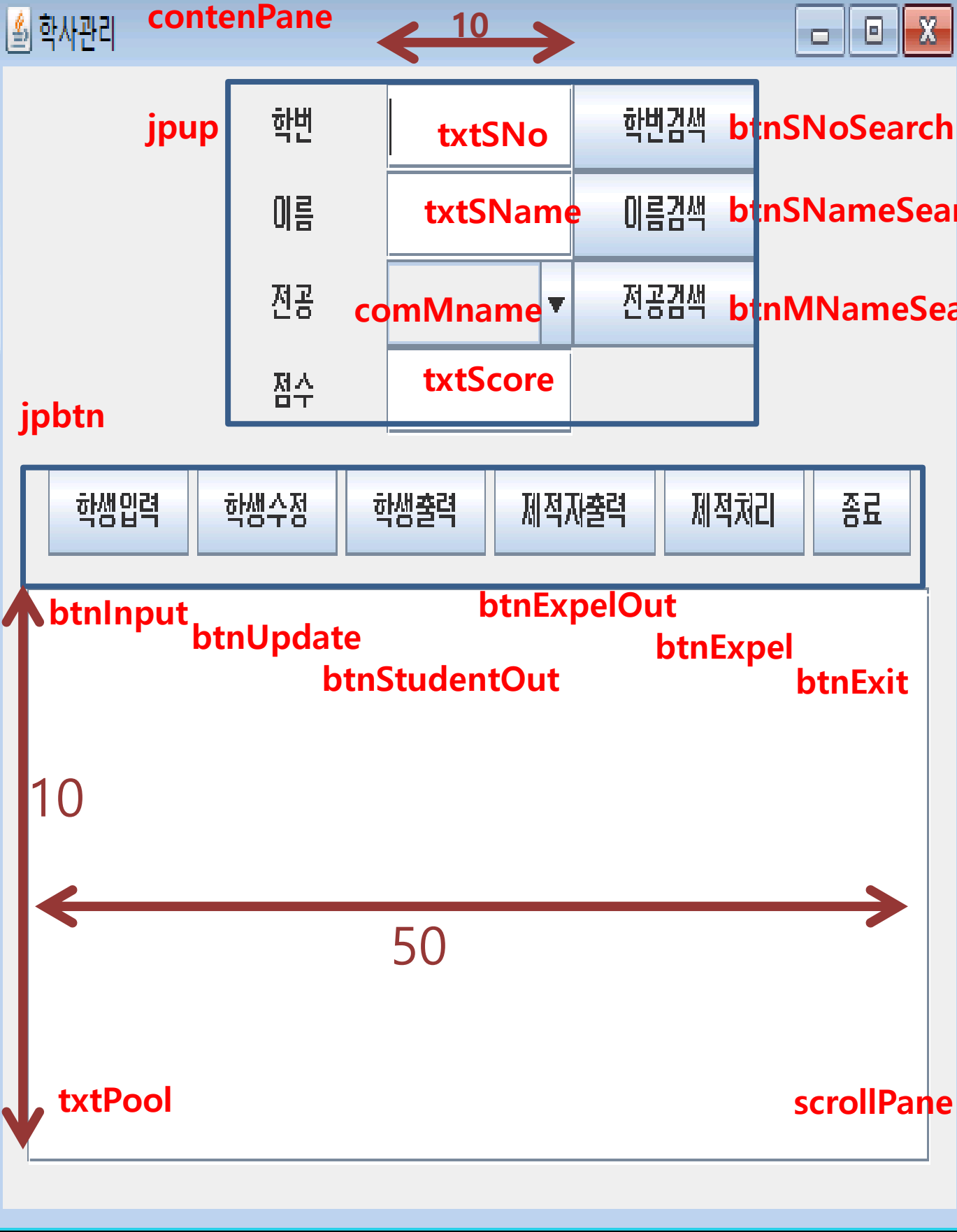


응용SW 문제

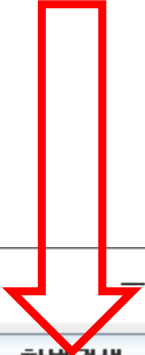
- 아래의 파일을 하나로 압축하여 제출합니다
 1. exERD 파일
 2. exERD 캡처
 3. Student.sql
 4. DTO, DAO (자바파일)
 5. Swing java program(자바파일)
 6. 실행화면 캡처
 - ✓ 학번검색, 이름검색, 전공검색
 - ✓ 입력, 수정
 - ✓ 제적자출력
 - ✓ 학생출력



요구사항

- 학번검색 : 학번을 입력하고 학번검색버튼을 클릭하면 해당 학번의 학생 정보가 txtSNo, txtSName, comMName, txtScore에 뜬다. 해당 학생이 없거나 학번을 입력하지 않을 시, txtPool에 메시지를 뿌린다
- **이름검색 : 이름을 입력하고 이름검색버튼을 클릭하면 해당 이름의 학생 정보가 txtSNo, txtSName, comMajor, txtScore와 txtPool에 뜬다. 동명이인이 있을 시 txtPool엔 리스트가 출력되고, textField에는 마지막 학생이 출력된다**
- 전공검색 : 콤보박스에서 선택된 전공학생(제적자 포함)을 txtPool에 출력
- 학생입력
- 학생수정
- 학생출력(제적자를 제외하고 txtPool에) 출력
- 제적처리
- 제적자출력

학번 검색시 학번은 입력하고



학사관리

학번	<input type="text"/>	학번검색
이름	<input type="text"/>	이름검색
전공	<input type="text"/>	전공검색
점수	<input type="text"/>	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

학번검색

학사관리

학번	2021	학번검색
이름		이름검색
전공		전공검색
점수		

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

없는 학번을 검색하면 없다고 출력

학사관리

학번	2021	학번검색
이름		이름검색
전공		전공검색
점수		

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

2021은 유효하지 않는 학번입니다

학번검색

학사관리

학번	2021001	학번검색
이름		이름검색
전공		전공검색
점수		

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

있는 학번을 해당 학번의 정보로 setting

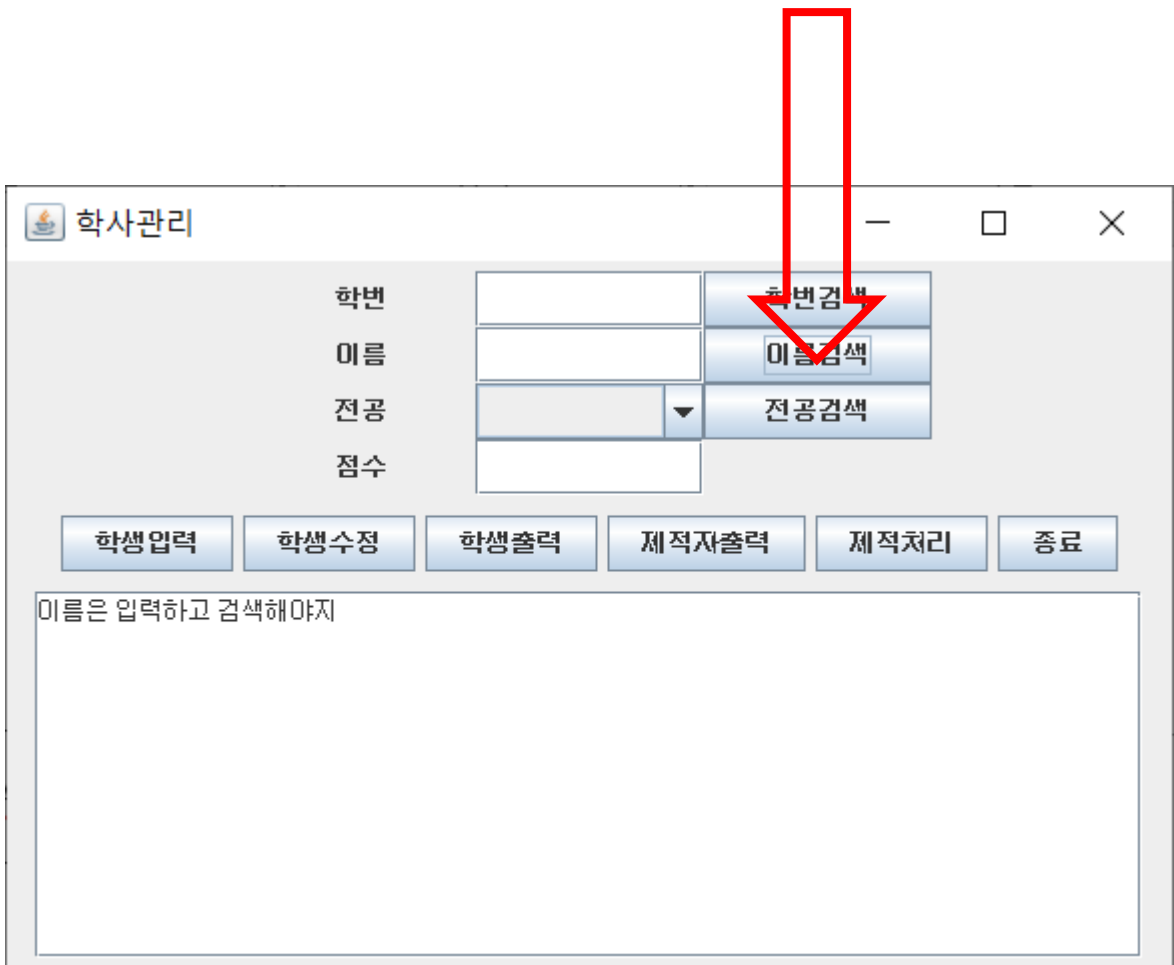
학사관리

학번	2021001	학번검색
이름	정우성	이름검색
전공	빅데이터학	전공검색
점수	90	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

2021001 검색 완료

이름 검색시 이름은 입력해야
한다. Space나 입력하지 않을
시 "이름은 입력하고 검색해
야지"라고 txtPool에 출력



학사관리

학번	<input type="text"/>	학번검색
이름	<input type="text"/>	이름검색
전공	<input type="text"/>	전공검색
점수	<input type="text"/>	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

이름은 입력하고 검색해야지

이름검색

학사관리

학번	<input type="text"/>	학번검색
이름	자바양	이름검색
전공	<input type="text"/>	전공검색
점수	<input type="text"/>	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

없는 이름을 검색하면 없다고 출력

학사관리

학번	<input type="text"/>	학번검색
이름	자바양	이름검색
전공	<input type="text"/>	전공검색
점수	<input type="text"/>	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

해당 이름의 학생이 없습니다

이름검색

학사관리

학번	<input type="text"/>	학번검색
이름	박세영	이름검색
전공	<input type="text"/>	전공검색
점수	<input type="text"/>	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

동명이인이 없는 이름을 검색하면
해당 정보를 textFile에 setting

학사관리

학번	2021002	학번검색
이름	박세영	이름검색
전공	빅데이터학	전공검색
점수	80	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

이름검색

학사관리

학번	<input type="text"/>	학번검색
이름	홍길동	이름검색
전공	<input type="text"/>	전공검색
점수	<input type="text"/>	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

동명이인이 있는 이름을 검색할 경우
아래에 list로 출력

학사관리

학번	<input type="text"/>	학번검색
이름	홍길동	이름검색
전공	<input type="text"/>	전공검색
점수	<input type="text"/>	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

학번	이름	학과명	점수
2021008	홍길동	인공지능학	99
2021009	홍길동	빅데이터학	88

전공검색시 전공은 선택하고

학사관리

학번	<input type="text"/>	학번검색
이름	<input type="text"/>	이름검색
전공	<input type="text"/>	전공검색
점수	<input type="text"/>	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

전공은 선택하고 검색해야지

전공검색

해당학과의 학생이 없는 경우

학사관리

학번 학번검색

이름 이름검색

전공 경영정보학 ▼ 전공검색

점수

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

해당 학과의 학생이 없습니다

전공검색

해당학과의 학생이 있는 경우
점수 내림차순으로 출력

학사관리

학번

이름

전공

점수

빅데이터학

학번검색

이름검색

전공검색

학생입력

학생수정

학생출력

제적자출력

제적처리

종료

등수	이름	학과명	점수
1등	정우성(2021001)	빅데이터학(1)	90
2등	홍길동(2021009)	빅데이터학(1)	88
3등	박세영(2021002)	빅데이터학(1)	80

학생입력

학사관리

학번	<input type="text"/>	학번검색
이름	모범생	이름검색
전공	빅데이터학 ▼	전공검색
점수	100	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

학번은 입력하지 않고 자동 셋팅
년도4자리+순차번호3자리
ex. 2022001

학사관리

학번	<input type="text"/>	학번검색
이름	<input type="text"/>	이름검색
전공	▼	전공검색
점수	<input type="text"/>	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

모범생 학생 입력 성공

학생수정

학사관리

학번	2021010	학번검색
이름	모범생	이름검색
전공	빅데이터학	전공검색
점수	90	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

학사관리

학번	2021010	학번검색
이름	모범생	이름검색
전공	빅데이터학	전공검색
점수	90	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

모범생 학생 입력 수정

학생출력

학사관리

학번

이름

전공

점수

학번검색

이름검색

전공검색

학생입력

학생수정

학생출력

제적자출력

제적처리

종료

등수	이름	학과명	점수
1등	홍길동(2021006)	인공지능학(5)	99
2등	정우성(2021001)	빅데이터학(1)	90
3등	홍길동(2021007)	빅데이터학(1)	88
4등	박세영(2021002)	빅데이터학(1)	80
5등	신길동(2021005)	경영정보학(2)	40
6등	배수지(2021003)	컴퓨터공학(3)	20
7등	김제적(2021004)	컴퓨터공학(3)	10

제적자를 제외하고 모든 학생들을 점수 내림차순으로 출력

제적처리

학사관리

학번	2021004	학번검색
이름	김제적	이름검색
전공	컴퓨터공학 ▼	전공검색
점수	10	

학생입력 학생수정 학생출력 제적자출력 제적처리 종료

2021004검색 완료

학사관리

학번	2021004	학번검색
이름	김제적	이름검색
전공	컴퓨터공학 ▼	전공검색
점수	10	

학생제적 학생수정 학생출력 제적자출력 제적처리 종료

2021004 학생 제적 처리 완료

제적자출력

학사관리

학번

이름

전공

점수

학번검색

이름검색

전공검색

학생입력

학생수정

학생출력

제적자출력

제적처리

종료

등수	이름	학과명	점수
1등	못난이(2021008)	빅데이터학(1)	11
2등	김제적(2021004)	컴퓨터공학(3)	10

제적자들 점수 내림차순으로 출력

-- SwingStudentMng에서 필요한 Query

-- 0. 첫화면에 전공이름들 콤보박스에 추가(mName) : DAO에서 public Vector<String> getMNameList()

-- 1. 학번검색 : DAO에서 public StudentDto sNogetStudent(String sNo)

-- 출력 sNo, sName, mName, score

-- 2. 이름검색 : DAO에서 public ArrayList<StudentDto> sNamegetStudent(String sName)

-- 출력 sNo, sName, mName, score

-- 3. 전공검색 : DAO에서 public ArrayList<StudentDto> mNamegetStudent(String mName)

--- 출력 : rank, sName(sNo포함), mName(mNo포함), score / ex. 1 정우성(2021001) 빅데이터학(1) 90

-- 4. 학생입력 : DAO에서 public int insertStudent(StudentDto dto)

-- 5. 학생수정 : DAO에서 public int updateStudent(StudentDto dto)

-- 6. 학생출력 : DAO에서 public ArrayList<StudentDto> getStudents()

-- 출력 : rank, sName(sNo포함), mName(mNo포함), score / ex. 1 정우성(2021001) 빅데이터학(1) 90

-- 7. 제적자출력 () : DAO에서 public ArrayList<StudentDto> getStudentsExpel()

-- 출력 : rank, sName(sNo포함), mName(mNo포함), score / ex. 1 김제적(2021004) 컴퓨터공학(3) 10

-- 8. 제적처리 : DAO에서 public int sNoExpel(String sNo)

--0. 첫화면에 전공이름들 콤보박스에 추가(mName) : public Vector<String> getMNameList()

SELECT MNAME FROM MAJOR;

--1. 학번검색 (sNo, sName, mName, score) : public StudentDto sNogetStudent(String sNo)

SELECT SNO, SNAME, MNAME, SCORE

FROM STUDENT S, MAJOR M

WHERE S.MNO=M.MNO AND SNO=2022001;

--2. 이름검색 (sNo, sName, mName, score) : public ArrayList<StudentDto>

sNamegetStudent(String sName)

SELECT SNO, SNAME, MNAME, SCORE

FROM STUDENT S, MAJOR M

WHERE S.MNO=M.MNO AND SNAME='홍길동';

--3. 전공검색 (rank, sName(sNo포함), mName(mNo포함), score)

--: public ArrayList<StudentDto> mNamegetStudent(String mName)

SELECT ROWNUM RANK, SNAME, MNAME, SCORE

FROM (SELECT SNAME||'('||SNO||')' SNAME, MNAME||'('||M.MNO||')' MNAME, SCORE

FROM STUDENT S, MAJOR M

WHERE S.MNO=M.MNO AND MNAME='컴퓨터공학'

ORDER BY SCORE DESC); -- DAO에 들어갈 SQL

--4. 학생입력 : public int insertStudent(StudentDto dto)

INSERT INTO STUDENT (SNO, SNAME, MNO, SCORE)

VALUES (EXTRACT(YEAR FROM SYSDATE)||LPAD(STUDENT_SEQ.NEXTVAL, 3, '0'), '신길동',

(SELECT MNO FROM MAJOR WHERE MNAME='경영경제학'), 99);

--5. 학생수정 : public int updateStudent(StudentDto dto)

```
UPDATE STUDENT SET SNAME='신수지',  
  
    MNO=(SELECT MNO FROM MAJOR WHERE MNAME='원예산업학'),  
  
    SCORE = 98,  
  
    SEXPEL = (SELECT 1 FROM STUDENT WHERE SNO=2022003 AND 98>60)  
  
WHERE SNO=2022003;
```

-- 수정된 학생점수(SCORE)가 80이상이고, 제적자일 경우 제적해지를 추가하고자 할 때

```
UPDATE STUDENT SET SEXPEL = 1  
  
WHERE SNO=2022003 AND SCORE>80;
```

--6. 학생출력 (rank, sName(sNo포함), mName(mNo포함), score) : public ArrayList<StudentDto> getStudents()

```
SELECT ROWNUM RANK, SNAME, MNAME, SCORE  
  
FROM (SELECT SNAME||'('||SNO||')' SNAME, MNAME||'('||M.MNO||')' MNAME, SCORE  
  
FROM STUDENT S, MAJOR M  
  
WHERE S.MNO=M.MNO AND SEXPEL=0  
  
ORDER BY SCORE DESC);
```

--7. 제적자출력 (rank, sName(sNo포함), mName(mNo포함), score)

-- : public ArrayList<StudentDto> getStudentsExpel()

```
SELECT ROWNUM RANK, SNAME, MNAME, SCORE  
  
FROM (SELECT SNAME||'('||SNO||')' SNAME, MNAME||'('||M.MNO||')' MNAME, SCORE  
  
FROM STUDENT S, MAJOR M  
  
WHERE S.MNO=M.MNO AND SEXPEL=1  
  
ORDER BY SCORE DESC);
```

--8. 제적처리 : public int sNoExpel(String sNo)

```
UPDATE STUDENT SET SEXPEL = 1 WHERE SNO=2022003;  
  
ROLLBACK;
```

DAO

0. 첫화면에 전공이름들 콤보박스에 추가(mName)

: public Vector<String> getMNamelist()

1. 학번검색 (sNo, sName, mName, score)

: public StudentDto sNogetStudent(String sNo)

2. 이름검색 (sNo, sName, mName, score)

: public ArrayList<StudentDto> sNamegetStudent(String sName)

3. 전공검색 (rank, sName(sNo포함), mName(mNo포함), score)

: public ArrayList<StudentDto> mNamegetStudent(String mName)

4. 학생입력

: public int insertStudent(StudentDto dto)

5. 학생수정

: public int updateStudent(StudentDto dto)

6. 학생출력 (rank, sName(sNo포함), mName(mNo포함), score)

: public ArrayList<StudentDto> getStudents()

7. 제적자출력 (rank, sName(sNo포함), mName(mNo포함), score)

: public ArrayList<StudentDto> getStudentsExpel()

8. 제적처리 : public int sNoExpel(String sNo)