

2020 전공 기초 프로젝트 설계서 수정판

수강신청 프로그램

B분반 2조

201711366 허세봉

201711303 박성진

201512265 박인우

201511193 박경민

201611222 이종우

Contents

1. 설계 대상.....	2p.
2. 전역.....	3p.
3. modify.cpp.....	3p.
3-1. modify 함수.....	3p.
3-2. classlistchange 함수.....	4p.
3-3. stateout 함수.....	4p.
3-4. classfilecheck 함수.....	5p.
3-5. accountfilecheck 함수.....	7p.
3-6. divisioncheck 함수.....	7p.
3-7. finalcheck 함수.....	9p.
3-8. timecheck 함수.....	10p.
4. table.cpp.....	11p.
5. class_list.txt.....	11p.

수정 사항: '3-2. classlistchange 함수' 부분에서 수강신청 과목 한도 확인 부분이
수강신청 철회 기능 부분의 뒤쪽으로 옮겨짐

1. 설계 대상

1-1. 전역 변수

- int classdivision : struct classinfo에 분반 번호 저장
- struct vecarr : 분반 처리에 관련된 구조체로 내부에 분반 번호를 의미하는 int num과 분반 번호의 개수를 의미하는 int cnt가 있음

1-2. modify 함수 내부에 입력값에 한글이 있는지 여부를 확인하는 루프 설치

- 입력값에 한글이 있을 경우 입력값의 길이(int chksize)를 -1로 변경
- chksize가 -1일 경우 해당 오류코드 (void stateout(int swt)) 출력

1-3. classlistchange 함수 내부에 과목신청 철회 기능과 같은 분반인 과목 신청시 오류 메시지 출력 기능 추가

- 최대 신청 가능 과목 수보다 신청한 과목 수가 적을 때 입력한 과목이 이미 신청한 과목일 경우 과목신청을 철회
- 입력한 과목의 분반번호가 0이 아닐 때 이미 신청한 과목의 분반번호와 입력한 과목의 분반번호가 일치하면 오류코드 (int bcel)을 분반 중복 오류에 해당하는 값으로 변경

1-4. stateout 함수 내부에 오류 메시지와 과목신청 철회 완료 메시지 추가

- “과목번호의 의미규칙 중 '분반 중복' 오류입니다” 메시지 추가
- “과목신청 철회가 정상적으로 완료되었습니다” 메시지 추가

1-5. classfilecheck 함수 내부에 divisioncheck 함수와 연동하여 분반 관련 오류검사 부분 추가

- 분반번호가 0 이상의 정수인지 확인 후 아닐 시 오류메시지 출력
- 0이 아닌 분반번호를 가진 과목들이 짝을 이루는지 확인 후 아닐 시 오류메시지 출력
- 분반번호가 입력되지 않은 과목이 있는지 확인 후 있을 시 오류메시지 출력

1-6. accountfilecheck 함수 내부에 finalcheck 함수와 연동하여 형식 관련 오류검사 부분 추가

- 신청한 과목 수가 입력되지 않은 계정이 있는지 확인 후 있을 시 오류메시지 출력

1-7. divisioncheck 함수로 분반 관련 오류검사 부분 추가

- 같은 분반 사이에 교수명이 동일하고 강의시간이 겹치는 과목이 있는지 확인 후 있을 시 오류메시지 출력

1-8. finalcheck 함수로 class_list.txt파일과 account_list.txt파일의 형식에 대한 추가적인 오류검사 부분 추가

- class_list.txt 검사 시 과목번호 중복, 과목명 중복, 교실 및 시간 중복 검사 후 오류 발견 시 오류메시지 출력
- account_list.txt 검사 시 각 계정 별 과목번호 중복 검사 후 오류 발견 시 오류메시지 출력

1-9. timecheck 함수로 강의시간이 겹치는지 검사하는 부분 추가

- 비교하는 두 과목의 강의시간이 겹치는지 검사 후 겹치면 오류메시지 출력

1-10. table 함수 내부에 string str의 크기 조정

1-11. class_list.txt에 분반코드 추가

2. 전역

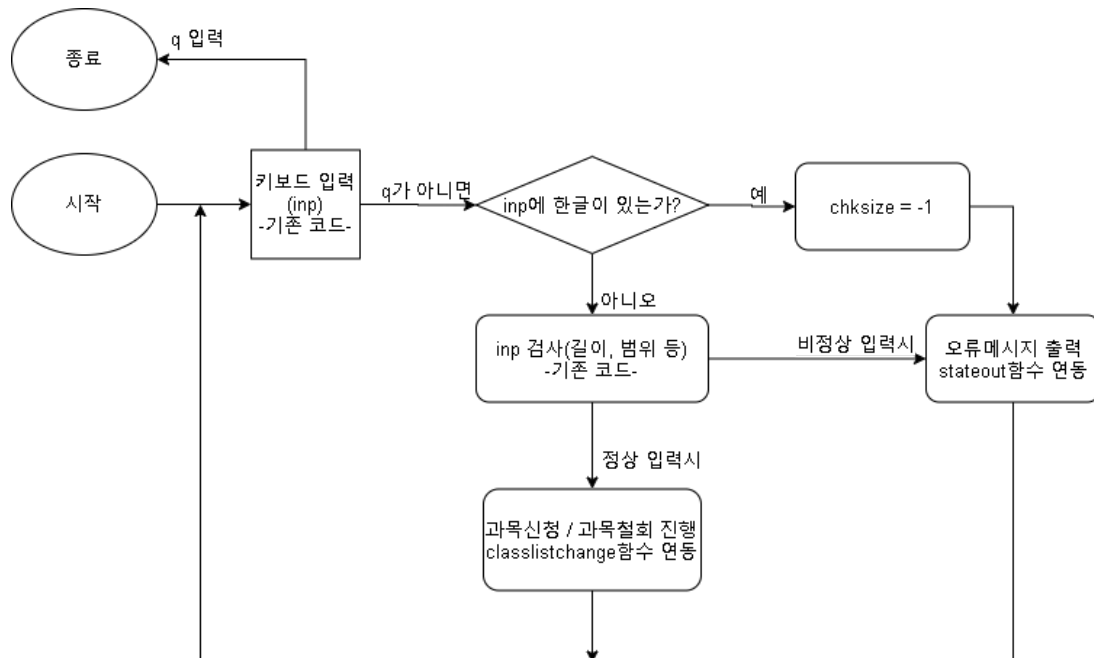
modify.h(전역)

- `#include<vector>` : vector 라이브러리 사용을 위해 추가
- `int classdivision` : struct classinfo에서 분반 번호를 저장(0 이상의 정수)
- `struct vecarr` : 분반 처리에 사용되는 벡터 배열을 위한 구조체로 내부에 `int num`과 `int cnt`이 있음
- `int num` : 분반 번호를 저장
- `int cnt` : `int num`에 해당하는 분반 번호를 가진 과목의 수를 저장

3. modify.cpp

3-1. modify 함수

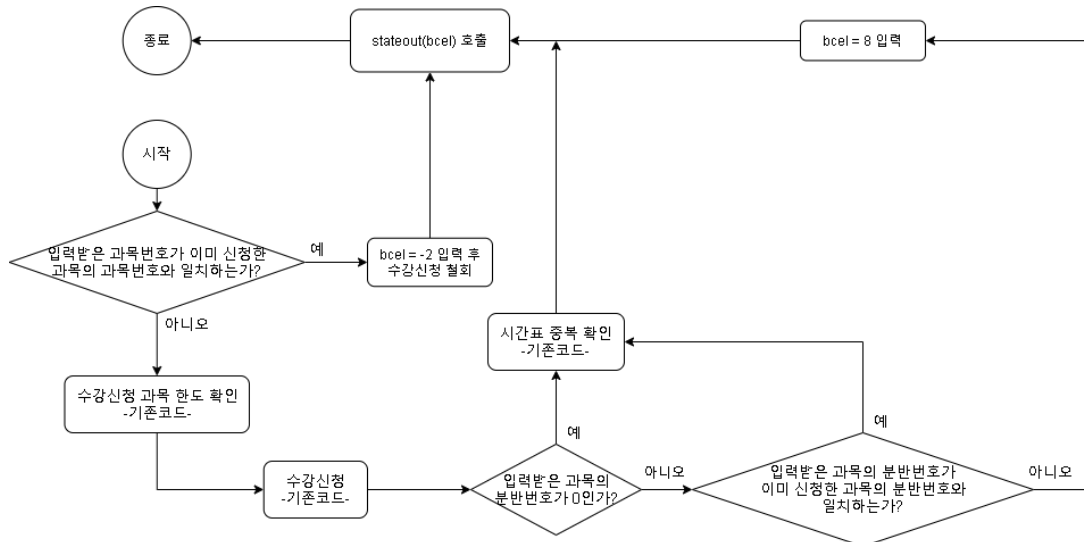
3-1-1. 추가 루틴



- `string inp`의 길이(`int chksize`)를 이용하여 for문으로 `string inp`에 한글이 있는지 확인 후 있을시 `chksize`를 -1로 변경 후 `break`.
- 이후 `chksize`가 -1이면 `stateout(1)` 실행 (“과목번호의 문법규칙 중 '범위 조건' 오류입니다” 출력)

3-2. classlistchange 함수

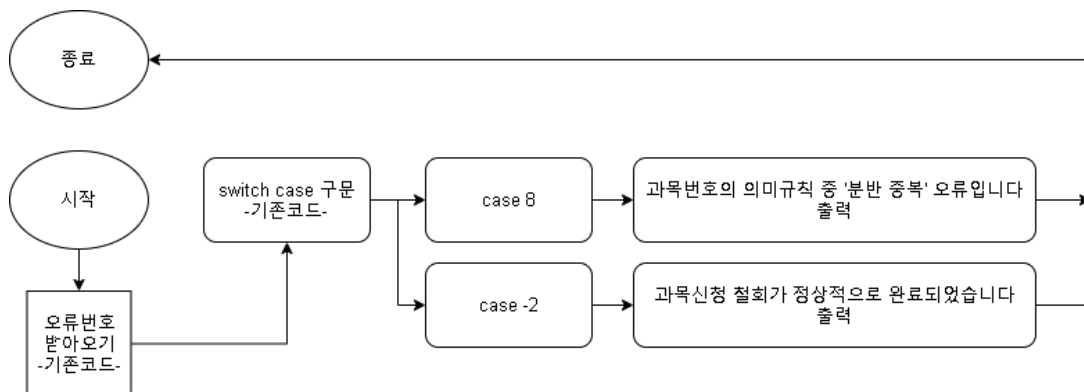
3-2-1. 추가 루틴



- 입력받은 과목번호와 현재 신청한 과목들의 과목번호를 비교하여 일치하는 경우 오류코드(int bcel)을 -2로 변경하고 해당 과목 이후에 있는 과목번호들을 배열에서 한 칸씩 앞으로 옮기고 마지막 칸은 null값을 채운 후 해당 계정의 수강 신청 과목 수(logstudent.classlistnum)를 1 줄임
- 입력받은 과목의 분반번호가 0이 아닐 때 이미 신청한 과목들 중 분반번호가 일치하는 과목이 있으면 오류코드(int bcel)를 8로 변경

3-3. stateout 함수

3-3-1. 추가 루틴



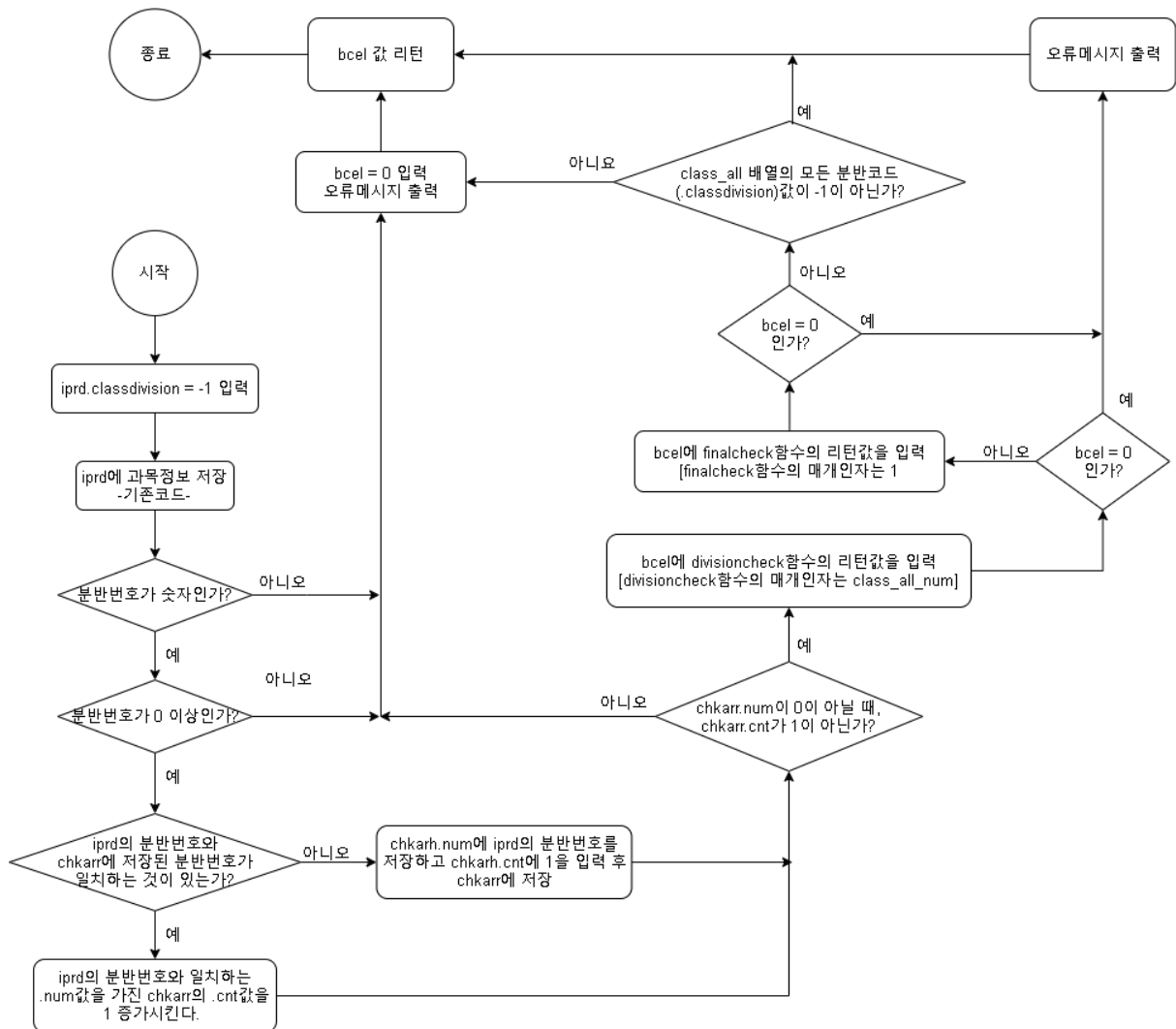
- case 8의 경우 “과목번호의 의미규칙 중 '분반 중복' 오류입니다” 출력
- case -2의 경우 “과목신청 철회가 정상적으로 완료되었습니다” 출력

3-4. classfilecheck 함수

3-4-1. 지역 변수

- vecarr chkarr : 벡터 배열인 chkarr에 넣을 값을 임시저장
- vector<vecarr> chkarr : class_list.txt에 저장되어있는 분반 번호(.num)와 그 번호가 나온 횟수(.cnt)를 저장

3-4-2. 추가 루틴

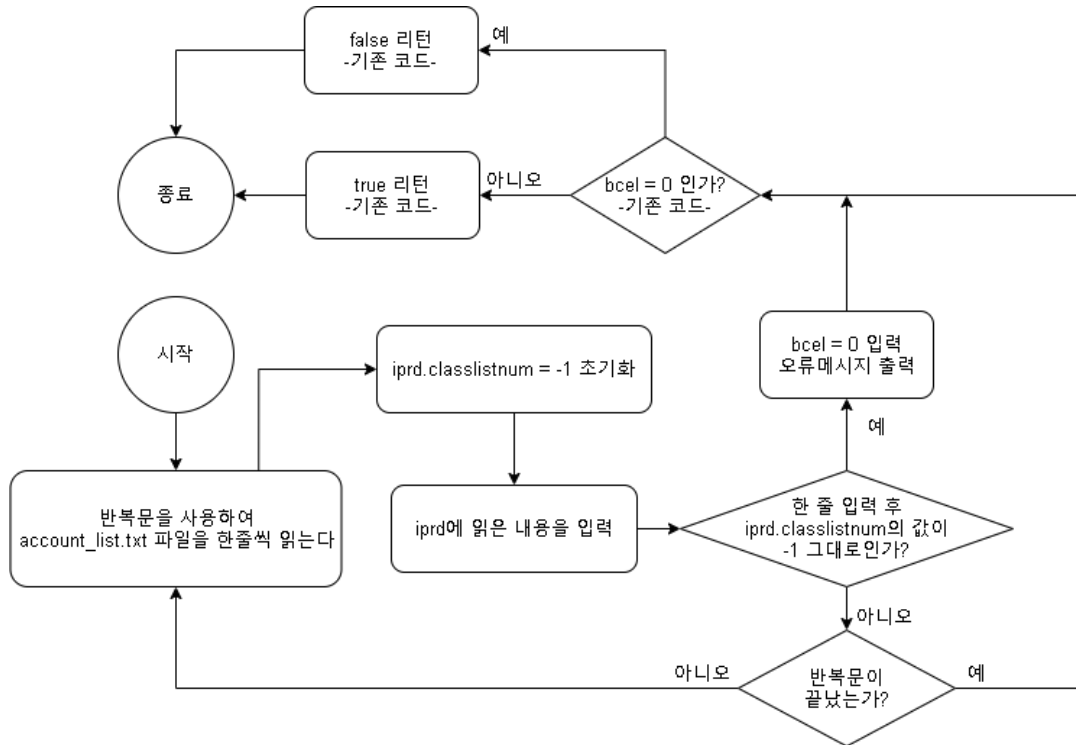


- classinfo iprd의 분반번호(classdivision)에 -1을 넣은 후 class_list.txt의 한 줄을 입력받아 토큰나이징 시켜 classinfo iprd에 각각의 정보를 입력
- class_list.txt에 분반번호가 추가됨에 따라 분반번호를 토큰나이징 해주어야 하므로 그 부분을 추가함

- 분반번호가 숫자인지, 0 이상인지 확인 후 만족하지 못 할 경우 오류코드(int bcel) 0으로 변경
- 분반번호가 chkarr에 저장되어 있는 분반 번호(.num)중 동일한 것이 있는지 검사
 1. 동일한 것이 있다면 해당 위치의 chkarr.cnt 값을 1 증가시켜서 chkarrh에 저장 후 chkarrh를 chkarr에 입력하고 기존 chkarr 값을 제거
 2. 동일한 것이 없다면 chkarrh.num은 해당 분반번호로, chkarrh.cnt는 1로 입력 후 이를 chkarr에 입력
- 더 이상 class_list.txt에 읽어올 값이 없으면 오류코드(int bcel)가 0이 아니면 분반 번호와 횟수가 저장된 chkarr벡터를 검사하여 분반인데(.num 이 0이 아님) 분반쌍이 없는 경우(.cnt가 1임)를 검사해서 만약 분반쌍이 없다면 int bcel에 0을 입력
- 오류코드(int bcel)가 0이 아니면 bcel에 divisioncheck(class_all_num)의 리턴값을 저장
- 오류코드(int bcel)가 0이 아니면 bcel에 finalcheck(1)의 리턴값을 저장
- class_all 배열에 저장된 모든 과목에 대해 분반코드(.classdivision)의 값이 -1인 과목이 존재하면 오류코드(int bcel)에 0을 입력

3-5. accountfilecheck 함수

3-5-1. 추가 루틴



- student iprd에 학생 정보를 입력 받기 전 신청한 과목 개수(.classlistnum)에 -1을 넣은 후 account_list.txt의 한 줄을 입력받아 토큰라이징 시켜 student iprd에 각각의 정보를 입력
- 한 줄 입력이 끝난 후 student iprd의 신청한 과목 개수(.classlistnum)가 -1이면 오류코드(int bcel) 0으로 변경
- account_list.txt의 정보를 읽어오는 루프가 끝난 후 오류코드(int bcel)가 0이 아니면 bcel에 finalcheck(2)의 리턴값을 저장

3-6. divisioncheck 함수

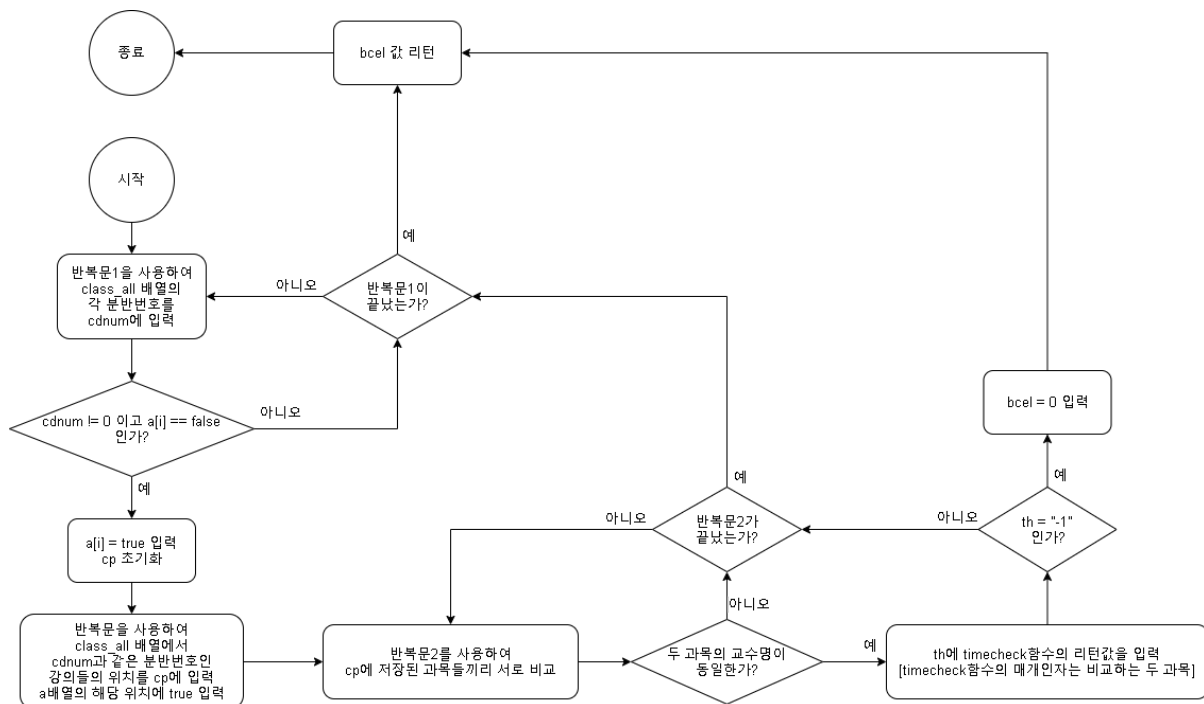
3-6-1. 인터페이스 및 액션

- 매개변수 : int many (chkarr의 크기)
- 참조하는 전역변수 :
modify.h 전역변수 중
extern classinfo class_all[100] / extern int class_all_num
- 리턴값: bcel (오류 여부)
- 액션: 강의정보가 저장된 class_all에서 분반번호가 같고, 교수명이 동일하며, 강의시간이 겹치는 경우가 있는지 검사

3-6-2. 지역 변수

- int bcel : 오류가 없을 때는 1, 오류가 있을 때는 0을 입력
- int cdnum : 분반번호를 임시저장
- bool* a : 한번 검사한 분반번호를 중복으로 검사하는 경우를 막는 용도
- string th : timecheck함수의 리턴값을 저장하는 변수
- vector<int> cp: 비교할 과목이 class_all 배열의 몇 번째 값인지 저장

3-6-3. 루틴



- int cdnum에 class_all 배열에 있는 각 과목의 분반번호를 반복문을 사용하여 하나씩 순서대로 불러옴
- 분반번호(cdnum)가 0이 아니고 분반 중복이 확인되지 않은 과목일 때(a[i]가 false일 때) a[i]을 true로 바꾸고 cp에 할당되어 있던 메모리를 해제함
- class_all 배열 전체에서 cdnum과 같은 분반번호를 가진 모든 강의들의 위치 (class_all 배열의 몇번째 칸인지)를 cp에 입력하고 a배열의 해당 위치를 true로 변경함
- cp에 저장된 과목들끼리 서로 같은 교수명이 있는지 비교해서 같다면 해당 두 과목을 timecheck함수에 대입하여 리턴값을 th에 저장
- th가 -1이 아니라면 (int bcel) 0으로 변경 후 반복문 탈출
- 반복문이 끝난 후 a와 cp에 할당되어 있던 메모리를 해제한 다음 bcel값 리턴

3-7. finalcheck 함수

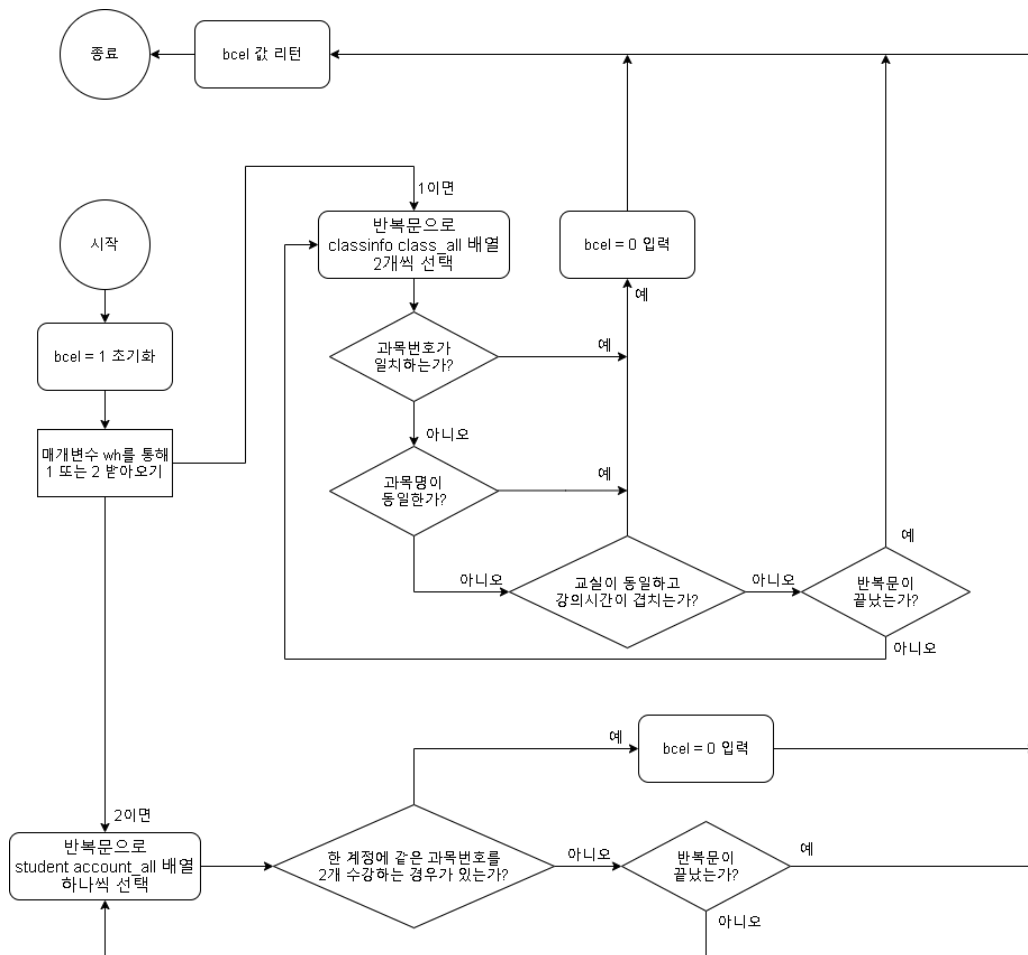
3-7-1. 인터페이스 및 액션

- 매개변수 : int wh (검사하는 파일이 account_list.txt인지 class_list.txt인지 구분)
- 참조하는 전역변수 :
modify.h 전역변수 중
- extern classinfo class_all[100] / extern int class_all_num
extern student account_all[100] / extern int account_all_num
- 리턴값 : bcel (오류 여부)
- 액션 : int wh가 1일 땐 class_list.txt의 과목번호 중복, 과목명 중복, 교실 및 시간 중복 검사. int wh가 2일 땐 account_list.txt의 과목번호 중복 검사

3-7-2. 지역 변수

- int bcel : 오류가 없을 때는 1, 오류가 있을 때는 0을 입력
- string th : 교실과 시간이 겹치는 과목이 없을 경우 -1을 입력
- student aci : account_list.txt의 한 줄을 struct student 형식에 맞게 입력

3-7-3. 루틴



- int wh가 1일 때
extern classinfo class_all에 있는 과목에서 2개를 고르는 모든 조합에 대하여 과목번호가 같거나, 과목명이 같거나, 교실이 같고 시간이 겹치는지 확인 후 셋 중 하나라도 해당되면 bcel에 0을 대입하고 반복문 탈출 후 bcel값 리턴
- int wh가 2일 때
extern student account_all에 있는 모든 계정에 대하여 한 계정이 같은 과목번호를 2개 수강하는 경우 bcel에 0을 대입하고 반복문 탈출 후 bcel값 리턴

3-8. timecheck함수

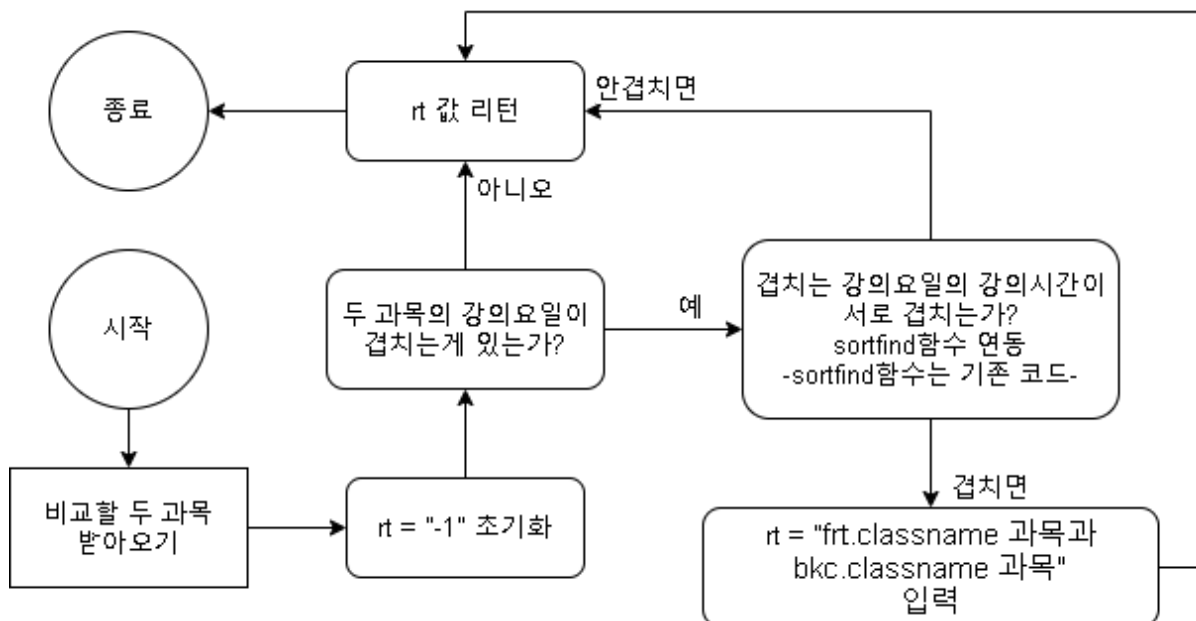
3-8-1. 인터페이스 및 액션

- 매개변수 : classinfo frt, classinfo bkc (시간 중복 여부를 확인하려는 두 과목)
- 참조하는 전역변수 : 없음
- 리턴값 : string rt (시간 중복이 아닐 시 -1 입력)
- 액션 : 두 과목의 시간 중복 여부를 확인

3-8-2. 지역 변수

- string rt : 요일과 시간이 일치하지 않으면 -1입력, 일치하면 "frt.classname 과목과 bkc.classname 과목" 입력

3-8-3. 루틴



- classinfo frt와 classinfo bkc의 각 시간표에 대해 요일이 일치하고 강의시간이 겹치면(sortfind가 true) rt에 "frt.classname 과목과 bkc.classname 과목" 입력하고 반복문 탈출 후 rt값 리턴

4.table.cpp

4-1. table함수

class_list.txt의 양식변경에 따라 class_list.txt의 한 줄을 토큰나이징하여 저장하는 배열을 string str[9]에서 string str[10]으로 변경

5. class_list.txt

5-1. 파일양식

class_list.txt 내부 과목정보들의 양식을 [과목번호 과목명 교수명 강의실명 과목구분 학점 주수업횟수 수업시간(주 수업횟수 만큼) 분반코드] 으로 변경. 과목번호, 과목명, 교수명, 강의실명, 이수구분, 학점, 주 수업횟수, 수업시간(주 수업횟수에 따라 1개 혹은 2개), 분반코드(분반이 아닐 경우 0이며, 분반코드는 0이상의 정수)가 순서대로 나열. 모든 과목 목록에서 과목번호나 과목명이 겹치는 과목이 존재해서는 안되고, 교수님이 같은 과목 간에는 시간표가 겹쳐서는 안되며, 같은 분반이더라도 교수님이 다를 수 있음. 같은 교수님이란 이름이 동일한 교수님을 뜻하며, 동명이인은 고려하지 않음. 그리고 분반코드는 최소 한쌍을 이루어야함.