제목:

전북대학교 18학번 컴퓨터공학부 졸업관리 시스템 프로젝트

보고서

목차

1. 프로젝트의 목적 및 필요성

2. 프로젝트의 배경

3. 프로젝트의 수행과정

4. 프로젝트의 결과

5. 참고문헌

1. 프로젝트의 목적 및 필요성

이 프로젝트의 목적은 전북대학교 컴퓨터공학부 18학번 학생들이 자신의 졸업요건들 중에서 누락된 것이 없는지 확인하여, 졸업에 대한 문제를 사전에 줄이기 위함이다.

프로젝트의 필요성: 졸업 직전에 몇 개의 졸업요건을 알지 못하고 졸업요건을 만족시키지 못한 학생도 있을 수 있다.

이러한 학생들의 경우, 사전에 요건을 충족시켜서 졸업요건 때문에 졸업이 미뤄지는 경우를 방지할 수 있을 것이다.

졸업이 늦어지는 것은 자신이 세운 진로의 계획에 지장을 줄 수도 있기 때문에, 졸업관리는 학생들에게 필요하다고 할 수 있다.

또, 이렇게 학생들이 졸업을 스스로 관리하면 조교님들께서도 일의 부담이 줄어들 것으로 기대된다.

2. 프로젝트의 배경

이 프로젝트를 생각하게 된 동기가 크게 두 가지가 있다.

첫 번째는 졸업요건 중 취업지원과 상담이라는 것이였다.

나는 상담의 경우는 평생지도교수 상담이라는 항목만 충족하면 되는 줄 알고있었다.

그런데 학과 게시판에 올라온 졸업사정을 보니 3학년 때 취업지원과 상담이라는 것을 해야했다.

현재 나는 3학년 2학기인데 학교 재학중에 한 번도 들어본 적이 없는 내용이였다. 이것을 할 시간이 이제 약 2달 밖에 남지 않았다는 것에 당황했다.

두 번째 이유는 졸업 요건 중 모험과창의 활동에서 교과목 이수 2개가 있었다. 나는 이를 통해 졸업 요건을 충족시키려고 조교님께 여쭤보니, 조교님께서 이는 사라질 항목이니 다른 항목을 통해서 충족을 시켜야 한다고 하셨다.

이 역시 내가 확인하지 않았다면 졸업직전에 알아차려서 졸업이 늦어졌을지도 모른다.

위와 같은 경험을 통해서, 학생이 스스로 자신의 졸업관리를 할 수 있는 시스템이 있으면 좋겠다고 생각하였다.

3. 프로젝트의 수행과정

3-1. 계획단계:

주어진 제안 요청서가

제안 요청서

1. 학생들이 직접 졸업가능여부를 알 수 있는 기능이 필요하다.

2. 학생들의 정보를 입력하고 졸업요건과 비교해야 한다.

3. 학생들이 자신이 현재 졸업할 수 있는지 알 수 있어야한다.

4. 모든 요건이 충족이 안된다면 무엇이 부족한지 알려주어야 한다.

5. 학년이 올라가면 정보가 수정되므로 정보 수정기능이 필요하다.

6. 가장 중요한 것은 가능한 구체적인 부분까지 확인할 수 있도록 하는 것이다. 최소한 조교님들께서 봐주시는 정도와 비슷하게 구현해야 한다.

7. 12월 10일안에 프로젝트가 끝나야 한다.

이므로, 이에 대한 문제 설명서를 만들었다.

문제 설명서

졸업관리 시스템의 목표는 학생들이 사전에 졸업을 준비할 수 있게 하고, 조교님 또한 학생들 스스로의 졸업관리로 업무가 줄어들도록 하는 것이다.

이 시스템의 사용자인 학생은 현재 자신의 상태를 입력하여 학점, 자격증 및 다른 요건들의 만족상태를 확인할 수 있다.

졸업요건은 부전공/복수전공이나 조기졸업여부와 같은 학생정보에 따라 달라지므로, 상위관계에 있는 항목을 먼저 입력한 후 하위항목을 입력한다.

만약 졸업요건 충족이 안되었다면, 부족한 요건에 대해서 무엇이 부족한지 메시지를 준다. 그 다음 결과적으로 졸업이 가능한지 아닌지 알려준다.

그 후에 충족이 안된 항목의 대체방안이 있다면 대체방안을 알려준다.

학년이 올라가는 경우에는 상태가 변하므로 정보 조회 및 수정기능도 필요하다.

목표를 파악하였으므로 개발 일정 계획을 세워야 한다.

먼저 해야 할 일의 범위를 정하고, 이들을 작업단위로 쪼개서, 각각의 작업들에 대해 일정계획을 세운다.

시스템의 기능:

1. 학생은 졸업요건 항목에 대한 자신의 정보를 입력한다.

2. 정보를 입력하고, 졸업이 가능한지 확인하는 기능을 선택했을 때, 충족된 졸업요건과 충족되지 않은 졸업요건을 보여주어야 한다.

3. 만약 졸업요건을 충족하지 못하면 결과로 충족되지 않음을 말해준다.

4. 어떤 졸업요건은 대체방안이 존재한다. 만약 대체방안이 존재하는 졸업요건이 충족되지 않으면, 그 부분에 대해서는 따로 대체방안을 제시해준다.

5. 졸업가능 확인기능 이외에 학생은 자신의 정보 확인 및 수정을 할 수 있어야한다.

프로젝트의 범위는 위의 기능을 구현하는 정도이다.

구현을 위해 해야 할 작업들을 작업 패키지로 분류하면:

A.1: 학교자체의 졸업요건을 위해 학교홈페이지의 졸업요건을 본다.

A.2: 학과의 졸업요건을 위해 학과 홈페이지의 졸업요건을 본다. 이 때 학번마다 졸업요건이 다르므로 18학번의 졸업요건을 본다.

A.3: 요건을 만족 못했을 때의 대체 방안과 같은 구체적인 경우에 대해서 조사를 한다.

A.4: 각각 경우들의 관계들을 파악한다. (예: 조기졸업인지 아닌지에 따라 전평점 4.0이상 졸업요건의 유무가 달라지므로, 조기졸업인지를 먼저 질문해야 한다.)

A.5: 조사한 내용들을 모두 항목으로 정리한다.

A.6: 정리한 내용을 기반으로 테스트 세트를 만든다. 졸업 요건을 모두 만족한 경우 1가지, 만족하지 못한 경우 2가지를 준비한다.

A.7: 수집하고 정리한 내용들을 가지고 구현할 프로그래밍 언어를 선별한다.

A.8: 나눈 경우들의 관계를 기반으로, 의존도가 낮은 항목부터 차례대로 구현해 나간다. 이 때 구현을 하면서 사용자의 요구대로 결과가 나오는지 중간에 점검을 한다.

A.9: 구현을 하고나서 준비한 테스트 세트로 테스트를 한다.

A.10: 테스트를 해서 결과가 제대로 나오지 않는다면 7번으로 가서 다시 시작한다. 결과가 제대로 나올 때까지 반복한다.

A.11: 프로그램 완성 후, 필요한 추가사항이나 다른 오류발견을 위해 계속해서 점검을 한다.

프로젝트 일정의 세분화:

A.1~A.2: 11/01~11/05

A.3: 11/07 ~ 11/13

A.4~A.5: 11/15~11/18

A.6: 11/19

A.7: 11/21~11/23

A.8: 11/25

A.9: 11/28~12/01

A.10: 12/03~12/08

위의 내용들을 바탕으로 일정 계획을 정리하면,

1. 전체적인 졸업요건 조사는 11월 1일 ~ 11월 13일 안에 끝내야 한다. 이 프로젝트는 가능한 구체적인 경우까지 고려해서 졸업가능여부를 판단하는 것이 가장 중요하므로 조사기간을 많이 주었다.

2. 수집한 졸업요건들의 관계파악 및 정리하는 작업은 11월 15일 ~ 11월 19일 안에 끝내야 한다. 이것도 졸업요건 조사와 마찬가지로 조기졸업인데 전평점 4.0과 같은 조건이 누락되면 시스템의 신뢰도가 매우 떨어지게 된다.

따라서 우선순위가 높은 구체적인 경우파악을 위해 정리기간을 많이 주었다.

3. 정리한 내용을 기반으로 프로그램을 구현하는 작업은 11월 21일 ~ 11월 23일 안에 끝내야한다. 이는 A.3단계에서 이미 정리를 해놓았으므로 기계적인 코딩이 예상된다. 따라서 개발기간에 많은 비중을 차지하지 않게 설정하였다. 하지만 기본적으로 코딩도중 나오는 에러, 중간점검과 같은 과정을 고려하여, 3일을 주었다.

4. 구현한 다음, 준비한 테스트 세트를 가지고 테스트를 하고 결과가 예상한 것과 일치하는지, 고객의 구대로 결과가 잘 나왔는지 검사한다. 만약 검사에서 오류가 나면 다시 수정을 한다.

이 과정은 11월 28일 ~ 12월 1일 안에 끝나야한다.

5. 오류가 나지 않아도 추가 요구사항이나 유지보수를 위해서 계속해서 점검한다. 이 과정은 계속해서 지속된다. 이 과정은 12월 3일부터 이 프로젝트가 끝날 때까지 한다.

3-2. 요구사항 분석단계:

이 프로젝트에서 가장 중요한 사항은 학생 개개인 상황에 따라 졸업요건을 달라질 수도 있으므로 모든 졸업요건을 정리하고 흐름을 파악하는 것이 가장 중요하다.

이를 위해서는 요건을 만족 못했을 때 대체방안, 조기졸업 여부에 따른 졸업요건들의 차이 등등 각각의 경우에 따른 사건의 흐름을 파악할 필요가 있다. 따라서 이 프로젝트에 가장 적합한 모델링 관점은 동적 모델링이 되고, 동적 모델링을 통해서 상태의 흐름을 파악하는 것이 중요하다.

고객이 필요로 하는 것은:

1. 학생의 졸업관련 정보입력

2. 졸업 가능 불가능 여부 알려주기

3. 졸업 불가능이면 만족되지 않은 요건 알려주기

4. 대체방안이 있는 항목이면 대체방안 알려주기

5. 학년이 올라가서 정보를 수정해야하는 경우에 정보를 수정하는 기능 필요

먼저 수집한 졸업요건들을 정리하면:

정규졸업의 경우: 총 130학점 이상 이수, 영어 성적, 자격증, 8학기이수, 교양필수 최소 35, 전공필수 18학점, 진로개발및상담 모든학기 이수, 3학년 2학기에 취업진로 전문가 1회상담 이수, 포트폴리오 제출

조기졸업의 경우: 정규졸업의 경우 + 전평점 4.0 이상

교직이수: 140학점 이상

복수전공: 전공선택 24학점 이상, 복수전공 필수과목, 복수전공 과목 42학점 이상

부전공: 전공선택 24학점 이상, 부전공 필수과목, 부전공 과목 21학점 이상

전공심화: 전공선택 45학점

영어성적: 토익-630, CBT-173, PBT-500, IBT-70, TEPS-503, TOEIC Speaking Level 5, OPIc-IM1, G-TELP(LEVEL 2) 54  中 선택 1

자격증: 정보처리기사, 정보보안기사, 네트워크관리사, 게임프로그램밍전문가, 멀티미디어콘텐츠제작전문가, CCNA, CCNP, OCA, OCP, 학부장이 인정하는 이와 유사한 수준의 자격증 中 선택 1

모험활동: 모험 프로젝트 수행, 창업교육프로그램 참가자 중 70점 이상의 활동점수를 획득하여 '창업교육인증서' 취득, 국내 15주 이상의 산학협력현장실습을 이수 中 선택 1

교양필수과목: 글쓰기, 실용영어, 모험과 창의, 인문고전읽기, 수학1, 수학2, 일반물리학1, 인문사고의 기반 영역 택3학점, 사회이해의기반 영역 택3학점

전공필수과목: 소프트웨어공학, 알고리즘, 운영체제, 자료구조, 컴퓨터구조, 프로그래밍 언어론

영어항목 대체방안: 교양 중 영어로 진행하는 과목을 2과목 이상 이수하고 성적이 각각 B0 이상인 경우,

전공 과목 중 영어로 진행하는 2과목 이수하고 성적이 각각 B+이상인 경우(전공필수과목 제외),

언어교육부에서 영어 관련 이수 프로그램을 120시간 이상 이수한 경우,

해외 오프캠퍼스 4주 이상 참여한 경우,

해외 교류대학에서 교환학생으로 1학기 이상 이수,

모의 TOEIC 700점 이상

자격증 항목 대체방안: 교과과정 내 프로그래밍 과목 중 2과목 이상 B학점 이수, 학부 자체 프로그램 테스트 합격한 경우, 산업체 현장실습 3학점 이상 취득

동적 모델링을 사용하므로 일단 시나리오를 먼저 작성한다.

일반적인 경우:

1. 프로그램을 작동시킨다.

2. 메인메뉴가 나온다.

3. 학생 정보를 프로그램에 입력한다

4. 메인메뉴가 나온다.

5. 졸업가능여부 확인을 선택한다.

6. 학생정보와 졸업요건을 비교해서 졸업가능이라는 메시지를 출력한다.

7. 메인메뉴로 돌아온다.

8. 프로그램을 종료한다.

예외적인 경우:

1. 프로그램을 작동시킨다.

2. 메인메뉴가 나온다.

3. 학생 정보를 프로그램에 입력한다. 이 때 자격증항목이 만족되지 않도록 입력한다.

4. 메인메뉴가 나온다.

5. 졸업가능여부 확인을 선택한다.

6. 학생정보와 졸업요건을 비교해서 “졸업 불가능”이라는 메시지를 출력한다.

7. “자격증 항목 만족 못함”과 대체방안을 출력한다.

8. 메인메뉴로 돌아온다.

9. 프로그램을 종료한다.

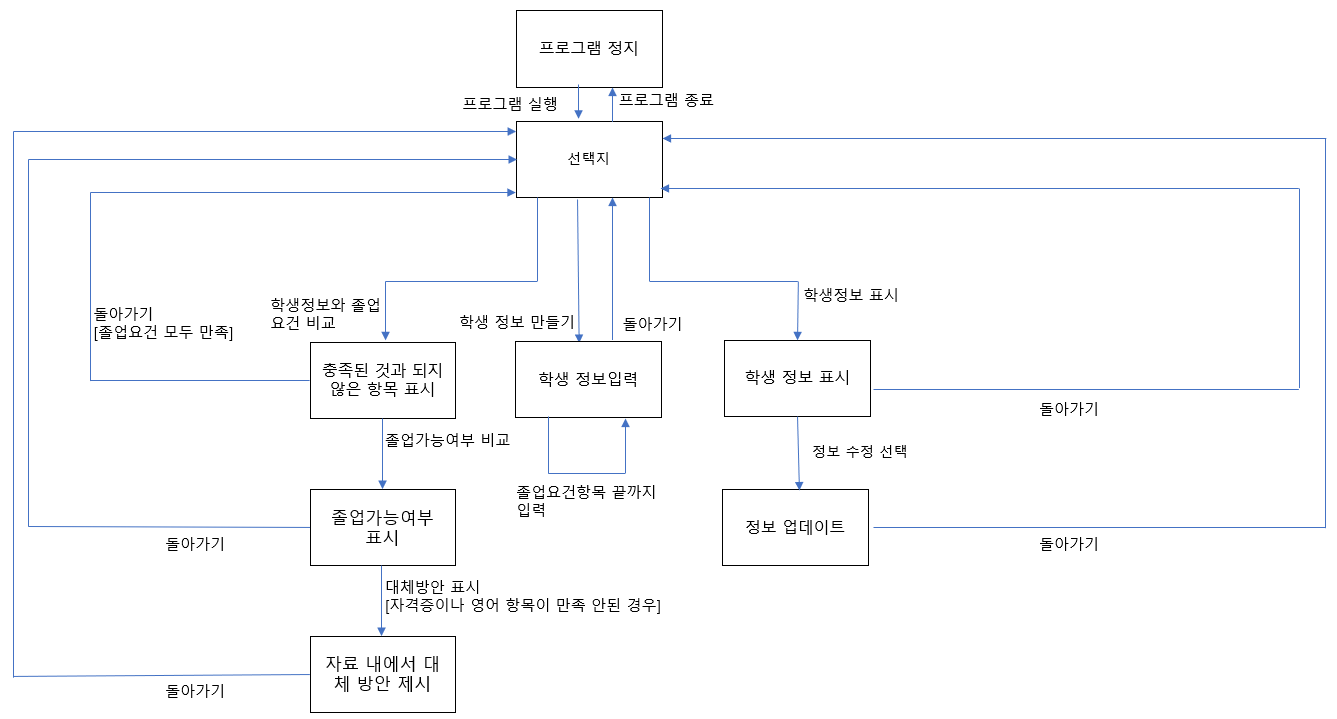
시나리오를 작성한 후에는 사건 추적을 한다.

일반적인 경우에 대해서 사건 추적을 하면



여기서 객체는 사용자와 프로그램이다.

이들을 기반으로 작성한 상태 변화도는



가 된다.

앞에서 문제설명서를 기반으로 본 내용과 동적 모델링을 기반으로 만든 요구사항 분석의 최종 결과물인 요구사항 명세서는

이다.

요구사항 명세서

1. 이 프로젝트는 전북대학교 컴퓨터공학부 18학번을 대상으로 하는 프로젝트이다.

2. 학생은 프로그램을 통해 졸업이 가능한지 아닌지 알 수 있어야 한다.

3. 학생은 자신의 정보를 조회할 수 있어야한다.

4. 학년이 올라가면 정보를 수정할 수 있어야한다.

5. 대체방안이 있는 경우는 대체방안을 제시해야한다.

6. 정보를 입력히고 수정하는 것이 대부분이므로 인터페이스는 값을 입력하는 형식으로 구현한다.

7. 최소한 조교님께서 봐주시는 정도와 비슷하게 졸업확인을 할 수 있어야한다.

8. 이 프로젝트의 가장 높은 우선순위는 가능한 많고 구체적인 경우에 대해서 다루는 것이다.

9. 이 프로젝트는 비용이 필요없는 프로젝트로, 프로젝트에서 비용과 관련한 시간소비는 없어야한다.

10. 프로젝트는 12월 10일안에 끝나야한다.

3-3. 설계

설계는 기본 설계단계와 상세 설계단계로 나눈다.

기본 설계:

1. 먼저 시스템을 서브시스템으로 나눈다:

시스템을 서브시스템으로 나누면 학생 정보 입력, 학생 정보 조회, 학생 정보 수정, 졸업가능 여부 확인, 졸업요건 대체방안 제시가 있다.

이 기능들을 구현하려면 수 많은 함수와 변수를 필요로 할 것이다.

만약 이것을 학생이라는 하나의 클래스에 저장하면, 구현하는데에 코드가 길어져도 정리하고 한 눈에 알아보기가 쉽다.

뿐만 아니라 학생과 졸업요건은 학점, 자격증, 영어성적 등등 서로 같은 종류의 변수를 가진다. 따라서 구현하는데에 걸리는 시간도 클래스로 지정하고 객체로 만들면 작업하는 시간이 최소 ½로 줄어들 것이므로 프로그래밍 언어는 클래스를 지원하는 언어를 선택한다. C++과 JAVA가 있는데, 개발자는 C++이 더 익숙하므로 C++로 구현한다.

앞으로는 모든 졸업요건을 변수 a라고한다.

먼저 추상화를 통해서 시스템의 기능과 구현방향을 간단하게 본다.

Class Grad{

Private:

a;

public:

input\_a() { a = temp;}

show\_a() { printf(a);}

return\_a() {return a;} }

Grad Gildong; // 학생객체 생성

void main{

printf(메인메뉴);

menu = scanf(메인메뉴에서 고르고싶은 선택지)

if(menu == 학생정보입력) {Gildong.a = 학생정보}

else if(menu == 졸업가능여부 확인){

모든 항목을 학생정보와 비교

if(졸업가능이 아닐 때){

대체방안이 있다면 보여줌}}

else if(menu == 학생정보 확인){

printf(학생정보);

if(정보 수정을 원하면){ Gildong.a = temp2;}

else if(menu == 프로그램 종료){

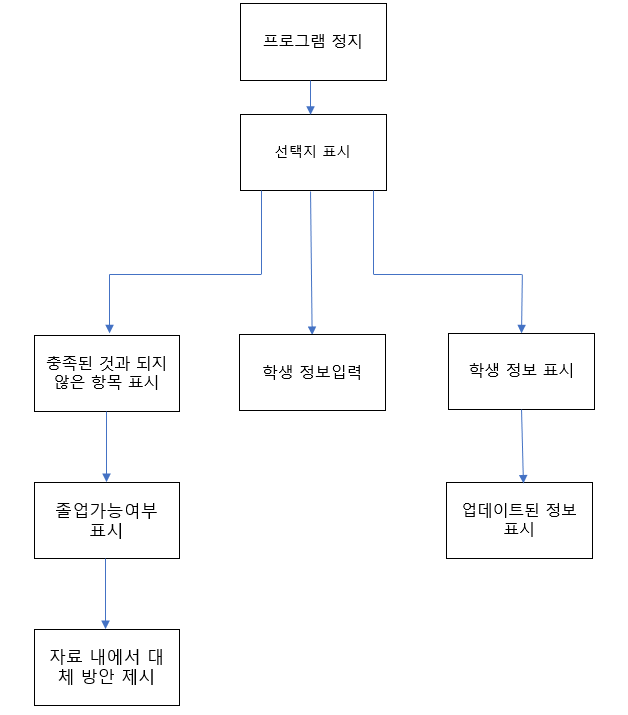
프로그램 종료}}}

정보은닉:

이 프로그램에서 각각의 졸업항목들은 가질 수 있는 값이 있다. 예를 들어 학점을 나타내는 변수인 GPA는 0이상이면서 4.5 이하여야 한다.

이렇게 변수들은 함부로 접근되면 안되므로 private로 설정해서 정보은닉을 하고, input\_a라는 함수를 통해서 수정을 하되, 실제로 입력될 수 없는 값은 입력할 수 없도록 해준다.

그 후에 프로그램의 구조를 모듈들간의 관계 그림으로 나타내면

이다.

이들을 바탕으로한 기본설계 문서는,

기본설계 문서:

기본설계 문서

1. 구현은 C++언어로 한다.

2. Class를 기반으로 학생객체를 만들어서 main부분에서 구현을 단순화한다.

3. 정보은닉을 위해 Class에서 학생의 정보를 private로 설정한다.

4. 클래스에서 포함해야할 내용은 졸업요건항목 변수, 정보 은닉을 위한 변수 대입함수, 졸업요건과의 비교를 위한 변수 값 반환함수, 변수 출력함수, 정보 수정을 위한 edit() 함수 이다.

5. 기능을 다 사용하고 다른 기능을 원할 수 있으므로, 어떤 기능이든 끝에 다시 선택지 상태로 되돌아오도록 한다.

6. 모듈간 관계 그림은 복잡도가 낮지는 않지만, 클래스 내부에서 함수들의 관계는 수평적이기 때문에 유지보수에 큰 문제가 되지는 않을 것이다.

이다.

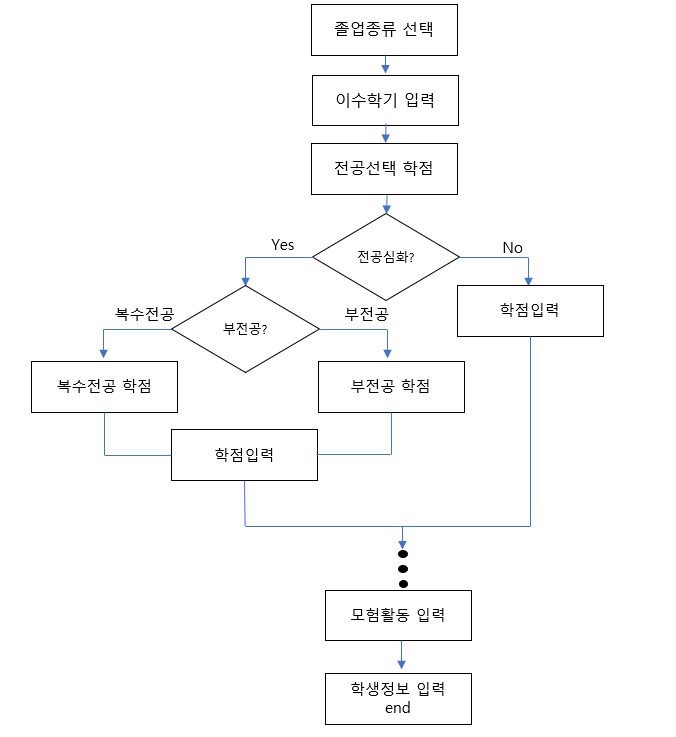
상세설계:

프로그램의 전체적인 과정은 기본설계를 따라가고, 기본설계에서 언급되지 않은 구체적인 부분만 확인한다.

구체적으로 모듈의 동작방식을 볼 때, 모든 모듈의 동작방식은 비슷하므로, 학생 정보입력 모듈만 살펴본다.

학생정보 입력의 작동은 각각의 항목을 순서대로 표시하고 입력하는 것을 끝날 때까지 반복하는 방식으로 진행된다.

이를 순서도로 표현하면:

이다.

3-4. 구현

구현하는 코드는 700줄을 가까이 되므로, 졸업요건 중 하나인 GPA를 기준으로 본다. 구현은 C++언어로 하였다.

class Grad\_cond{

private:

double GPA;

public:

void GPA\_in(){

while(true){

printf("\n전평점을 입력해주세요 \n");

double aa,a;

a = scanf("%lf",&aa);

a=aa;

if(a>=0 && a<=4.5){

GPA=a;

break;} else printf("잘못된 입력입니다.\n");

}

}

double GPA\_val(){

return GPA;}

void GPA\_show(){

printf("\n총 평점: %f", GPA);}

void edit()

}

와 같이 클래스내부 대부분의 졸업요건은 항목당 변수 1개와 함수 3개, 그리고 마지막으로 학생정보 수정을 위한 edit()함수로 이루어져 있다.

그 후 main 함수 밖에서

Grad\_cond GilDong; 를 선언한다.

int main()

{

printf("\n\n프로그램 시작상태\n");

int i=0;

int studen\_ID=0;

printf("\n1. 내 졸업요건 항목 작성 \n2. 졸업가능 확인 \n3. 내 정보 조회\n4. 프로그램 종료\n:");

while(true){

int u = scanf("%d",&i);

u = i;

if(u==1){

printf("\n\n내 졸업요건 항목 작성을 선택하셨습니다. \n");

break;

} else if(u==2){

printf("\n\n졸업가능 확인을 입력하셨습니다.\n ");

break;

} else if(u==3){

printf("\n\n내 정보 조회를 입력하셨습니다.\n ");

break;

} else if(u==4){

printf("\n\n프로그램 종료를 입력하셨습니다.\n ");

break;

} else {

printf("\n잘못된 입력입니다. 다시 입력해주세요.");

}

}

여기서 1. 학생정보 입력 2. 졸업가능 확인 3. 내 정보 조회 4. 프로그램 종료를 하나씩 구현하면,

if(i==1){

GilDong.GPA\_in();

}

else if(i==2){

if(GilDong.semest\_val()<8 && GilDong.semest\_val()>=6 && GilDong.GPA\_val()>=4){

d=1;

printf("\n조기졸업 학기이수 및 학점 4.0이상 만족\n");

} else if(GilDong.semest\_val()>=8 && GilDong.GPA\_val()>=4) {

printf("\n조기졸업 학기이수 만족못함, 학점 4.0이상 만족\n");

} else if(GilDong.semest\_val()<8 && GilDong.semest\_val()>=6 && GilDong.GPA\_val()<4) {

printf("\n조기졸업 학기이수 만족, 학점 4.0이상 만족못함\n");

} else { a=1;

printf("\n조기졸업 학기이수 및 학점 4.0이상 만족못함\n");

}

}

else If(i==3){

GilDong.GPA\_show() ;

printf("\n정보를 수정하시겠습니까? 1.Yes | 2.No\n");

int edit=0;

scanf("%d",&edit);

if(edit==1){

GilDong.edit();

}

}

else if(i==4){

break;

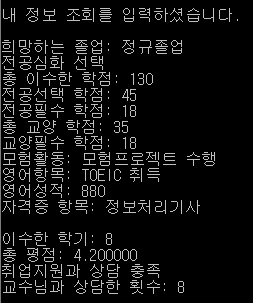
}

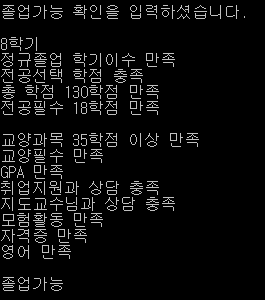
이다.

3-5. 결과

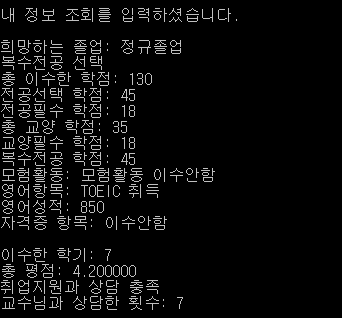
프로그램 구현 결과를 사전에 준비해둔 졸업가능한 경우 1경우와 졸업요건을 만족하지 못한 경우 2경우의 테스트 데이터로 확인한다. 입력정보 조회와 졸업비교 결과를 나타낸다.

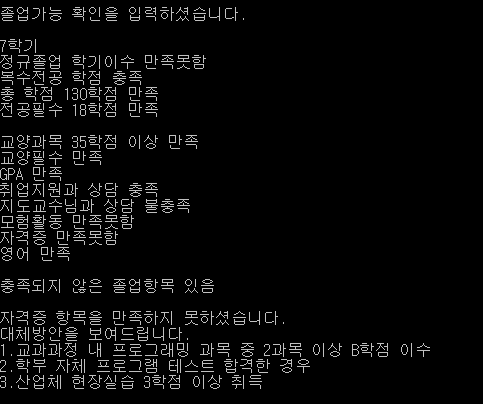
졸업요건을 만족한 경우:



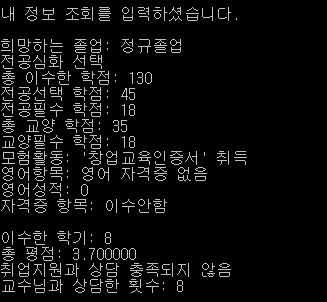


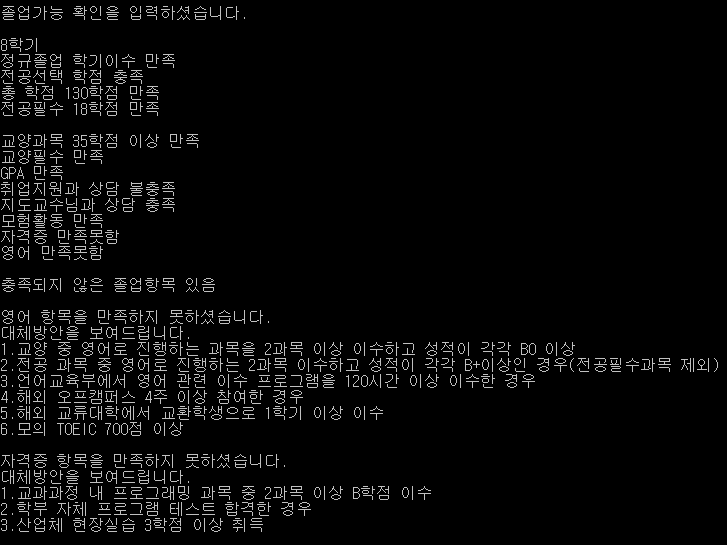
졸업요건을 만족하지 못한 경우 1:





졸업요건을 만족하지 못한 경우 2:





4. 프로젝트의 결과

요구사항 명세서와 비교를 하였을 때, 요구한 정보 조회, 정보 수정, 졸업가능여부 확인을 모두 구현하였다. 또한 인터페이스역시 학생이 정보를 입력한다는 점에서 입력형식의 인터페이스로 만들었다.

실제 프로그램이 문제없이 작동하는지는 미리 준비한 테스트 셋을 통해서 검토하였다.

이를 통해 고객이 요구한 기능들을 모두 구현하였음을 확인할 수 있다.

클래스를 기반으로 구현하였으므로 나중에 유지보수가 어렵지 않을 것이다.

만약 여기서 고객이 요구한 것 이상으로 기능을 확장한다면, 18학번보다 많은 학번으로 넓히고, 컴퓨터공학과뿐만 아니라 여러 학과로 넓히고, 학교까지 넓히는 것을 생각해볼 수 있다.

참고 문헌

개발 방향: 이해하기 쉬운 소프트웨어 공학(윤청 저), 114p, 폭포수 모델

프로젝트 결과 형식: 이해하기 쉬운 소프트웨어 공학(윤청 저), 197p

프로젝트 일정 관리 계획: 이해하기 쉬운 소프트웨어 공학(윤청 저) 228p

문세설명서 양식: 이해하기 쉬운 소프트웨어 공학(윤청 저), 300p

동적모델링의 시나리오: 이해하기 쉬운 소프트웨어 공학(윤청 저), 319p

동적모델링의 사건 추적: 이해하기 쉬운 소프트웨어 공학(윤청 저), 321p

동적모델링의 상태 변화도: 이해하기 쉬운 소프트웨어 공학(윤청 저), 322p