|  |
| --- |
| 객체지향프로그래밍  실습과제보고서 #3 |
|  |

|  |
| --- |
| 2019년 3월 25일  김예원  소프트웨어학과 |

**실습과제 #3 성적처리 프로그램 #2**

: 각종 함수를 이용해 성적처리 프로그램#1을 개선하는 과제였다. 구조체를 레퍼런스와 포인터를 이용해 함수 인자를 전달한다.

**함수를 사용하여 성적처리프로그램#1 개선**

1. **메뉴 출력 함수**

|  |
| --- |
| void PrintMenu() {  cout << "=====메뉴=====" << "\n";  cout << "1. 학생 성적 입력" << "\n";  cout << "2. 전체 학생 성적 보기" << "\n";  cout << "3. 학생 이름 검색" << "\n";  cout << "4. 프로그램 종료" << "\n";  } |

성적처리 프로그램#1에 없던 3번 학생 이름 검색 기능이 추가되었다

**(2)평점 계산 함수**

|  |
| --- |
| float CalcGPA(Subject& Sub) { //학점 평점을 계산  if (Sub.Grade[0] == 'A') {  if (Sub.Grade[1] == '0') {  Sub.GPA=Sub.Hakjum \* 4.0;  }  Sub.GPA=Sub.Hakjum \* 4.5;  return Sub.GPA;  }  else if (Sub.Grade[0] == 'B') {  if (Sub.Grade[1] == '0') {  Sub.GPA = Sub.Hakjum \* 3.0;  }  Sub.GPA = Sub.Hakjum \* 3.5;  return Sub.GPA;  }  else if (Sub.Grade[0] == 'C') {  if (Sub.Grade[1] == '0') {  Sub.GPA = Sub.Hakjum \* 2.0;  }  Sub.GPA = Sub.Hakjum \* 2.5;  return Sub.GPA;  }  else if (Sub.Grade[0] == 'D') {  if (Sub.Grade[1] == '0') {  Sub.GPA = Sub.Hakjum \* 1.0;  }  Sub.GPA = Sub.Hakjum \* 1.5;  return Sub.GPA;  }  else {  Sub.GPA = Sub.Hakjum \* 0.0;  return Sub.GPA;  }  } |

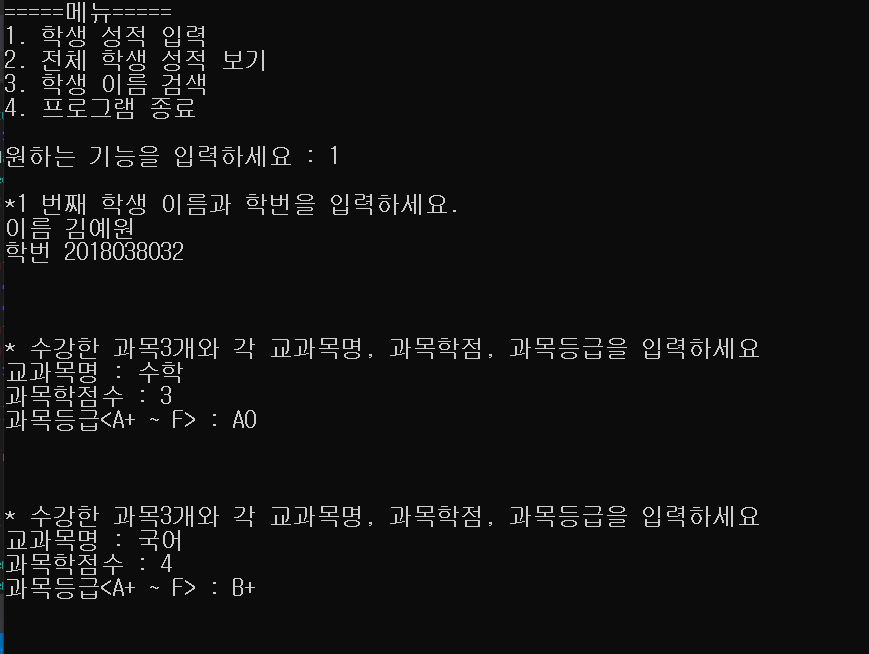
배열을 이용해 알파벳과 0을 구별했고, 그에 맞춰 학점을 계산하도록 했다. 계산한 값은 Sub.GPA에 저장해 반환하도록 했다.

**main 반복문에서의 함수 호출**

|  |
| --- |
| for (j = 0; j < 3; j++) { //학생 당 3개의 과목 정보를 입력 받는다  cout << "\n\n\n" << "\* 수강한 과목3개와 각 교과목명, 과목학점, 과목등급을 입력하세요\n";  cout << "교과목명 : ";  cin >> s[i].Sub[j].Subname;  cout << "과목학점수 : ";  cin >> s[i].Sub[j].Hakjum;  cout << "과목등급<A+ ~ F> : ";  cin >> s[i].Sub[j].Grade;  s[i].Sub[j].GPA = CalcGPA(s[i].Sub[j]); //즉시 초기화  } |

학생 한명 당 과제를 입력받으며 함수를 바로 호출했고, s[i].Sub[j].GPA에 바로 초기화했다.

**실행 중인 화면**

****

**(3) 교과목 평점 계산 함수**

|  |
| --- |
| float CalcAveGPA(Subject\* Sub) { //교과목 평점 계산 함수  int sum=0;  for (int i = 0; i < SubjectNum; i++) {  sum += Sub[i].Hakjum;  }  return sum / SubjectNum;  } |

포인터 변수를 넘겨 받아 학점을 모두 더한 뒤 나눈 값을 return 한다.

**main에서 CalcAveGPA 호출**

|  |
| --- |
| while (i < 2) { //학생 2명의 데이터를 입력 받는다  cout << "\n";  cout << "\*" << i + 1 << " 번째 학생 이름과 학번을 입력하세요." << "\n";  cout << "이름 ";  cin >> s[i].StdName;  cout << "학번 ";  cin >> s[i].Hakbun;  for (j = 0; j < 3; j++) {  //학생 당 3개의 과목 정보를 입력 받는다    …*(2)의 내용과 동일하므로 생략…*  }  s[i].AveGPA = CalcAveGPA(s[i].Sub);  i++;  } |

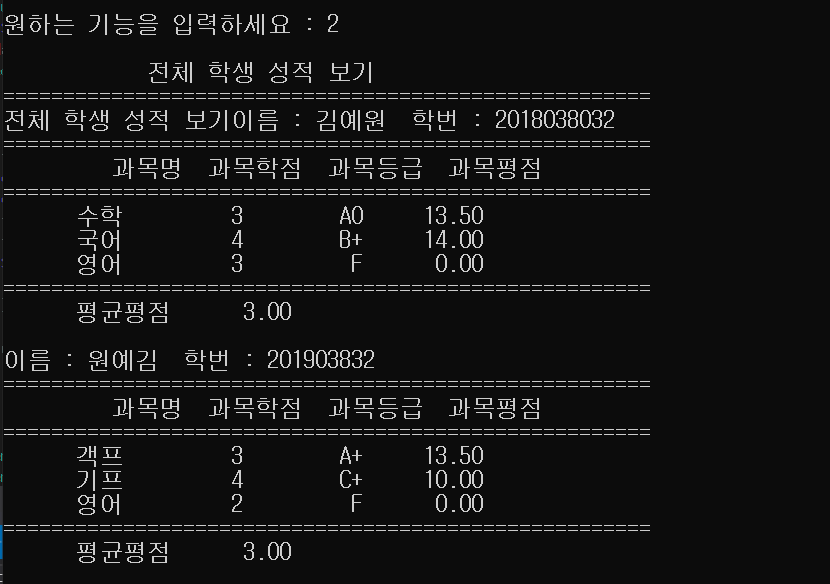
s[i].AveGPA를 즉시 초기화 한다.

**(4) 전체 학생 정보 출력 함수**

|  |
| --- |
| void PrintAllData(const Student\* pSt) { //전체 학생들의 성적 정보를 출력한다  cout << "전체 학생 성적 보기";  for (int i = 0; i < StudentNum; i++) {  PrintOneData(pSt[i]);  }  } |

학생 한 명의 데이터를 출력하는 void PrintOneData(const Student& rSt); 함수를 반복문을 이용해 호출한다. StudentNum은 2로 고정 되어있다.

**실행화면**

****

**(5) 개인 학생 정보 출력 함수**

|  |
| --- |
| void PrintOneData(const Student& s) {//학생 한명의 성적 정보를 출력한다  cout << "이름 : " << s.StdName << " 학번 : " << s.Hakbun << "\n";  cout << "======================================================" << "\n";  cout << " 과목명 과목학점 과목등급 과목평점" << "\n";  cout << "======================================================" << "\n";  for (int i = 0; i < 3; i++) { //학생 한명의 과목3개  cout.width(10);  cout << s.Sub[i].Subname; //과목명 출력  cout.width(10);  cout << s.Sub[i].Hakjum; //과목학점출력  cout.width(10);  cout << s.Sub[i].Grade; //과목등급 출력  cout.precision(2);  cout << fixed;  cout.width(10);  cout << s.Sub[i].GPA << "\n"; //과목평점 출력  }  cout << "======================================================" << "\n";  cout.width(20);  cout << "평균평점 " << s.AveGPA << "\n\n"; //평균평점을 출력  } |

학생 이름, 학번, 과목명, 과목학점, 과목등급과 과목평점을 출력하고 평균평점을 출력한다.

**(6) 학생 이름 검색 함수**

|  |
| --- |
| Student\* StdSearch(Student \*pSt) {  char stdname[20];  cout << "검색할 이름을 입력하시오: ";  cin >> stdname;  for (int i = 0; i < StudentNum; i++) {  if (strcmp(stdname, pSt[i].StdName) == 0) {  return (pSt + i);  }    }  return NULL;  } |

strcmp함수를 이용해, 넘겨받은 구조체의 문자열과 입력받은 문자열을 비교한다. 일치하

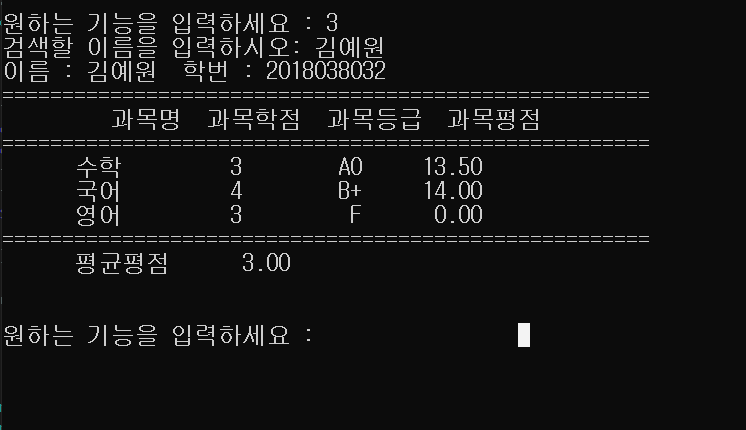
면 일치한 이름의 주소값을 반환하고 일치하지 않으면 NULL값을 반환한다.

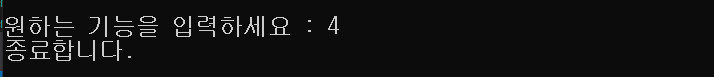
|  |
| --- |
| else if (func == 3) {  SearchStd = StdSearch(s);  if (SearchStd != NULL) {  PrintOneData(\*SearchStd);  }  else  cout << "일치하는 학생이 없습니다." << endl;  } |

**main 반목문에서의 함수 호출**

주소값을 넘겨 받으면 학생의 성적 정보를 출력하고 NULL 값을 넘겨받으면 일치하는 학생이 없다는 말을 출력한다.

**실행화면**

****

****

**소감문**

이번 과제를 하며 무엇보다 힘들었던 것은, 소스코드가 너무 길어서 변수와 함수인자, 매개변수가 헷갈렸다는 것이다. 함수를 여러 개를 사용하면서 변수가 여러가지가 되었고 레퍼런스, 포인터 등등의 것들을 구별하기도 힘들었었다. 소스코드를 작성하는 내내 느꼈던 것 중 하나는 함수가 다르다고 해서 변수 이름을 다르게 할 것이 아니라 동일하게 짓는 것이 훨씬 낫겠다는 생각을 했다. cout이 모호하다며 뜨지 않던 오류들이 너무 많이 생기기도 했다. 다른 오류들을 고치니까 cout 오류는 자연스럽게 사라졌는데, 꽤나 당황스러웠다. 또한 const 지정된 인자에서 const를 발견하지 못하고 에러가 뜨는 것을 보며 왜 에러가 뜨는지 발견하지 못했었는데, 오랫동안 고민하고 찾아 헤맨 뒤 에서야 const가 지정되었다는 것을 발견했다.

매번 과제 할 때마다 스스로에 대해 돌아보게 되는 것 같다. 평소에도 덤벙거리는 성향이었는데, 프로그래밍을 하는 중에도 이런 덤벙스러움이 비쳐지는 듯했다. 이런 성향을 고치려면 어떻게 해야할까, 어떻게 해야 작은 실수없이 프로그래밍을 해낼 수 있을까 계속 고민하게 된다. 비주얼 스튜디오로 코딩을 하다가 눈이 너무 아프고 헷갈려서 종이로 출력해서 틀린 부분을 하나하나 찾아내 고쳐낼 수 있었다. 이번에도 마찬가지로 괄호 같이 사소한 것들 때문에 뜬 오류들이 있었다. 종이에 출력해서 틀린 것을 찾아내는 습관이 좋은 습관인지는 모르겠지만, 나만의 방법을 찾아낸 것 같은 기분이 든다.