**2024 Spring OOP Assignment Report**

과제 번호 : 2

학번 : 20230499

이름 : 김재환

Povis ID : carotinoid

**명예서약 (Honor Code)**

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

1. **프로그램 개요**

* OOP Assignment #2 – 문제 1번 (배점 35점)

학생들의 점수와 정보를 관리하는 프로그램을 작성한다. 이 프로그램이 지원하는 기능은 다음이 있다. 1. 학생정보 추가. 2. 학생 점수 수정. 3. 평균 및 표준편차 출력. 4. 학생 점수 리스트 출력. 5. 학생 등급 리스트 출력. 메뉴 화면에서 번호를 입력받아 해당 번호에 해당하는 기능을 수행하며, 6을 입력받을시 종료된다.

1. **프로그램의 구조 및 알고리즘**

* 학생 정보를 구현하기 위해 구조체를 사용하였다. 구조체의 이름은 student이며, int형 변수 id, midterm\_exam\_score, final\_exam\_score와 char타입의 배열 name[20], bool 타입 변수 retake를 갖는다.
* student 구조체 배열 stu\_list[105]를 선언하여 학생 정보를 저장하였고, numofstu는 학생 수를 저장하며, 메뉴에서 selection변수에 선택을 입력받는다. while문으로 전체를 무한반복하면서 branch문을 통해 selection에 해당하는 기능을 실행한다. selection이 6인 경우 exit(0); 코드를 통해 프로그램을 종료한다.
* 다음은 프로그램의 각 기능에 대한 설명이다.

1. selection이 1일 때 실행된다. 학생의 정보를 입력받고 stu\_list[105]리스트에 해당 정보를 저장한 뒤 numofstu를 증가시킨다. 이때 올바르지 않은 형식으로 학생의 정보를 입력받을 시 그에 맞는 에러 로그를 출력한 후 메뉴로 돌아간다. 학생의 이름의 경우 입력을 받은 char타입 배열에서 구조체의 char배열로 옮기기 위해 strcpy\_s()함수를 사용하였다.
2. selection이 2일 때 실행된다. select변수에 1 또는 2를 입력받고, stu\_id변수에 학번을, score변수에 점수를 입력받아 select가 1인 경우 입력한 학번의 중간고사 점수를 score로, select가 2인 경우 입력한 학번의 기말고사 점수를 score로 수정한다. 이 과정을 수행하기 위해 입력받은 학번이 존재하는지, 어디있는지 찾기 위해 thereisid, findidx 변수가 사용되었다. 유효하지 않은 선택이나 학번, 점수가 입력될 경우 그에 해당하는 에러 로그를 출력하고 메뉴로 돌아간다.
3. selection이 3일 때 실행된다. 추가된 학생 전체의 평균과 표준편차를 출력하며, 학생 수가 없을 경우 평균과 표준편차 모두 0으로 출력한다. 평균과 표준편차를 계산하는 과정에서 double형 변수인 평균: mean, 표준편차: standard, 중간과정에서 사용되는 변수 sqaurediffmean등을 선언하여 사용하였다.
4. selection이 4일 때 실행된다. 학생의 중간고사 점수와 기말고사 점수의 합을 기준으로 내림차순 정렬하여 학번, 이름, 중간고사 점수, 기말고사 점수, 두 점수의 총합 순서로 모든 학생의 점수를 출력한다. 점수가 같은 경우 학번 오름차순으로 정렬한다. 문제에서 요구된 Bubble sort를 통해 정렬을 구현하였으며, 정렬 과정에서 사용된 pre, next변수는 각각 한 버블에서 이전과 다음 학생의 두 점수 합이며, 둘을 비교하여 pre가 next보다 크면 둘을 바꾼다. 둘이 같다면 둘의 학번을 비교하여, 학번이 오름차순이 되도록 한다. 정렬 이후에는 numofstu만큼 반복하며 학생의 정보를 출력한다.
5. selection이 5일 때 실행된다. 등급을 판별하는 기준 중 하나인 상위 ~%를 쉽게 판단하기 위해 4번에서 사용한 정렬 알고리즘을 다시 사용하였다. 학생 등급을 계산하는 과정에서 학생 수 만큼 반복문을 진행하는데, 이때 내림차순으로 정렬되어 있으므로 for문 내에서 선언된 변수 i에 1을 더하여(0-base -> 1-base) 학생수로 나눔으로써 상위 얼마인지 계산할 수 있다. 이때 두 변수는 모두 int형이므로 double형으로 명시적으로 형변환하였다. 이러한 방법으로, 제시된 조건에 따라 grade변수에 값을 저장한다. A등급 조건에 만족하면 1, B등급은 2, C등급은 3, D등급은 4를 저장한다. 이후 재수강 여부에 따라 grade변수에 1을 더한 후 실제 등급으로 변환하여 출력한다.
6. **토론 및 개선**
   * + 실제 저장하는 정수 최댓값과 입력될 수 있는 최대값에 유의하여 자료형을 선택하여야 한다. 이번 프로그램에서는 학생의 학번을 저장하는 값은 8자리 정수이므로 int형 변수에 저장할 수 있지만, 입력 받을 때에는 10자리 정수가 가능하므로 맨 앞자리가 3 이상으로 시작한다면 int형 범위를 벗어나므로 오버플로우를 일으킬 수 있으므로 입력용 변수는 long long 타입을 이용하였다.
     + 과제에서 Branch문을 줄여 속도를 개선하라는 조건이 제시되었다. 반복문은 반복이 돌아가는 만큼 실제 연산이 많아지므로 속도가 느려지지만, 분기문은 다른 방식으로 실행 속도를 저하시킨다. 연산을 수행하는 CPU는 분기를 빠르게 실행하기 위해서 이전 결과를 저장하여 미리 다음 조건문의 결과(참, 거짓)을 예측한다. 이를 Branch Prediction이라고 하는데, 이 예측이 맞다면 더 효율적으로 동작하지만, 에측이 틀리면 예측값을 모두 다시 연산해야하므로 그만큼 패널티를 얻는다. 주어지는 데이터 상태에 따라 정확도가 달라지는데, 예측이 틀리는 횟수가 증가할수록 실행속도는 더 느려지게 된다.
     + 이번 과제에서는 반복문과 조건문을 이용하여 연속된 데이터를 쉽게 수정하고 적절히 예외처리하는 방법을 배웠다. 또한 프로그램을 작성하면서 다양한 타입을 사용하고, 오버플로우 또한 직접 경험해 보면서 조건에 해당하는 타입을 선택하는 것이 중요함을 알게 되었다.
7. **참고 문헌**

* Prata, S. (2013). C Primer Plus (6th ed.).
* Johnny's Software Lab, December 26. 2023. Unexpected Ways Memory Subsystem Interacts with Branch Prediction. From. <https://johnnysswlab.com/unexpected-ways-memory-subsystem-interacts-with-branch-prediction/>