**2019년 AI Big Data Academy 7기**

**Assignment**

**성적 관리 프로그램**

**담당교수: 윤은영**

**이름: 최소현**

**이메일: sohyun4283@naver.com**

|  |
| --- |
| **명예서약(Honor code)**  **“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”** |

**Problem : 사용자 명령을 받아 성적을 처리하는 프로그램 만들기**

1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

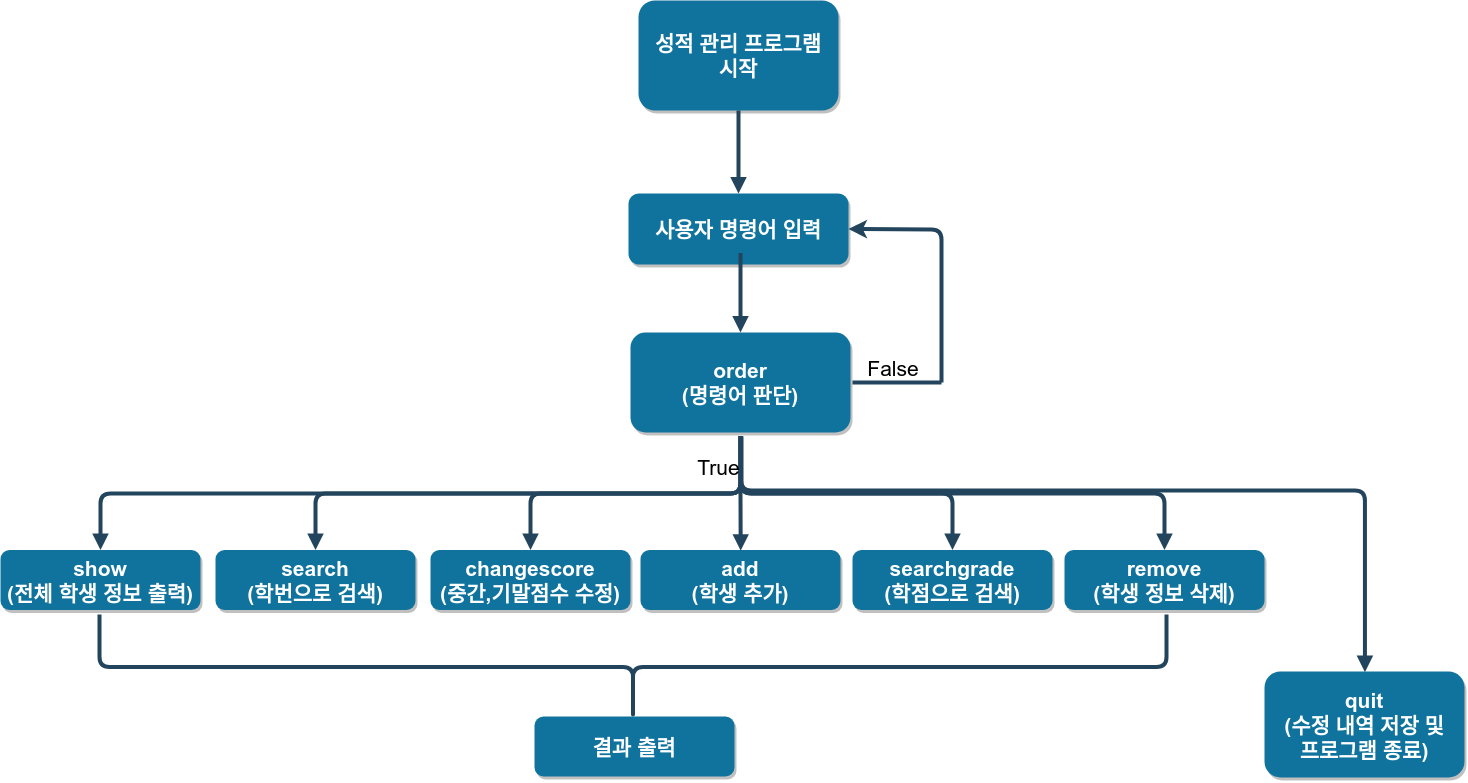
* 기본 성적 파일을 연다.
* 함수를 실행하기 위해 문자열을 입력 받는다.
* 올바른 명령어(7가지 명령어 중 1개)를 받으면 각각의 명령에 대한 함수를 실행한다.

함수에 따라 다른 정보를 입력 받고, 입력 받은 값을 활용하여 연산을 수행한다.

연산 수행 결과 또는 해당되는 학생 정보를 출력한다.

* 올바르지 않은 명령어를 받으면 올바른 명령어를 계속해서 기다린다.
* 그동안의 수정 내역을 저장하고 프로그램을 종료한다.

이 때 사용되는 구상 가능한 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.



* 입력: 사용자는 지정된 명령어 7개 중 1개의 명령어를 입력한다.
* 처리: 예외 처리를 통해 예외 상황인 경우 메시지를 반환한다.

예외 처리에 해당되지 않을 경우 적절한 처리 값을 반환한다.

* 출력: 입력한 함수에 해당하는 연산 및 학생 정보를 출력한다.
* 저장: 프로그램 종료 시, 편집한 내용의 저장 여부를 묻고 사용자 지정 파일에 새롭게 저장한다.

2. 알고리즘

본 프로그램 작성을 위한 알고리즘을 Pseudo 코드 형태로 나타내면 다음과 같다.

본 프로그램에는 총 10개의 함수가 사용되었고, 이 중 7개의 함수는 사용자의 명령에 의해 실행되는 사용자 명령어 함수이다. 나머지 3개의 함수는 avg\_grade, order, base\_prt이다.

- avg\_grade: 학생의 중간, 기말 점수의 평균과 학점을 계산하는 함수이다. 파일 읽기, 성적 수정, 학생 정보 추가 시 사용된다.

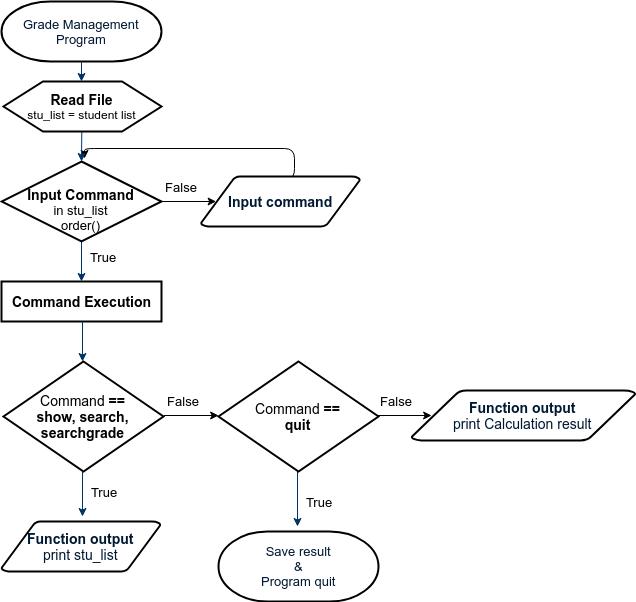
- order: 명령어 입력 받고 판단해 함수를 실행하게 하는 함수이다. Main 함수에서 처음으로 명령어를 받기 위해 사용한다. 함수를 수행한 후에도 명령을 받기 위해 각 사용자 명령어 함수 마지막 부분에 호출한다.

- base\_prt : 학생 내역 출력 시 기본 포멧 함수이다. 학생 정보를 출력할 때 복잡한 코드를 줄이기 위해 생성하였다.

Pseudo 코드에서는 7개의 사용자 명령어 함수에 대해서는 자세히 표현하지 않았다.

|  |  |
| --- | --- |
| **Pseudo-algorithm for Grade Management Program** | |
| 1  2  3  4  5  6  7 | *# 프로그램에 필요한 변수들은 미리 선언해 놓은 것으로 가정한다.*  Import sys  read file # 파일 읽기  명령어 input  유효한 명령어 = [ add, search, changescore, add, searchgrade, remove, quit ] # 7개의 명령어  if input in 유효한 명령어:  입력한 명령어 함수로 실행  print(학생 정보) or print(연산 결과)  명령어 input  if input == ‘quit’:  수정 정보 저장  프로그램 종료  else:  명령어 input |

위의 Pseudo 알고리즘을 Flowchart로 표현하면 아래와 같다.



3. 프로그램 구조 및 설명

1) 파일을 읽어 학생 정보를 리스트로 저장한다.

2) 명령어를 입력 받아 올바른 명령어인지 판단한다.

a) 올바르지 않은 명령어이면 계속해서 명령어를 입력 받는다.

b) 올바른 명령어이면 해당 함수를 실행한다.

3) 명령어가 학생 정보를 출력하는 함수(show, search, searchgrade)이면 연산 작업은 따로 진행되지 않으므로 해당하는 학생 정보를 알맞게 출력한다.

4) 명령어가 종료 함수가 아니면 연산하고 학생 정보를 변경 시키는 함수(changescore, add, remove)이므로 연산 결과(성공, 실패 등)를 출력한다.

5) 명령어가 종료 함수이면 그동안 수정한 결과를 저장하고 프로그램을 종료한다.

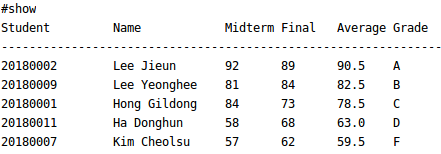
4. 프로그램 실행 방법 및 예제

• 터미널에서 프로그램 코드가 있는 파일로 이동 후 python project1.py 또는 python project1.py students.txt 명령을 입력한다.

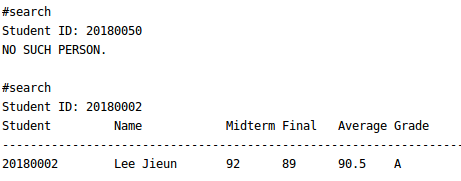
# 단, 다른 이름의 성적 txt 파일이 있다면 해당 파일의 이름을 입력한다.



• 사용자가 show라는 명령어를 입력하면, 학번, 이름, 중간고사 점수, 기말고사 점수, 평균, 학점 순서대로 표처럼 출력한다.

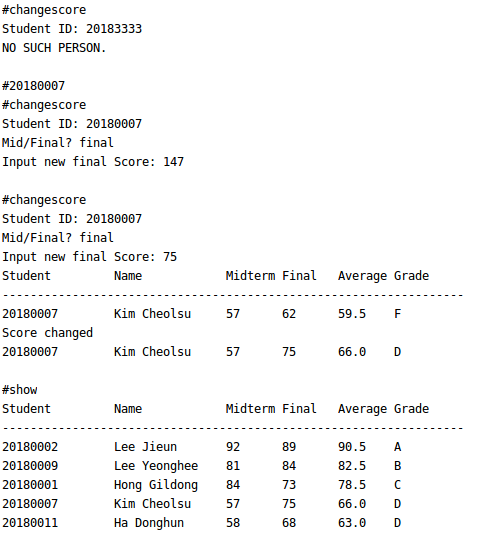


• 사용자가 search라는 명령어를 입력하면 학번을 묻는다. 학생 정보에 없는 학번을 입력하면 없다는 출력, 학생 정보에 있는 학번을 입력하면 해당하는 학생을 출력한다.



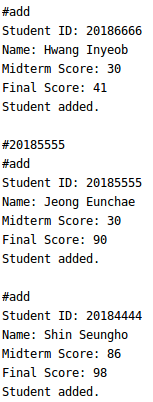
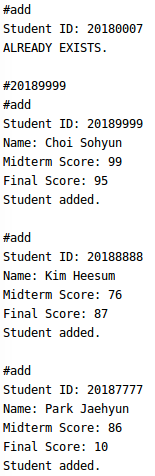
• 사용자가 changescore라는 명령어를 입력하면 학번을 묻는다. 해당하는 학생이 없으면 없다는 출력 문구가 나온다. 해당하는 학생이 있으면 학생의 중간과 기말 점수 중 어느 점수를 수정할 것인지 묻는다. 그 후 변경할 점수를 입력하라는 문구가 나오는데, 이 때 0점 이상 100점 이하의 값만 입력해야 점수 수정이 성공적으로 진행된다. 이 외의 값을 입력하면 무시된다.

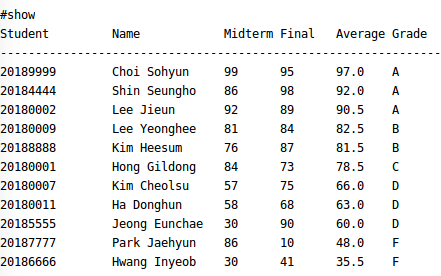
• 점수 수정 후 show 명령어를 통해 확인하면 평균과 학점, 학생 출력 순서까지 함께 변경된 것을 볼 수 있다.



• 사용자가 add라는 명령어를 입력하면 학번을 묻는다. 학생 리스트에 있는 학번이면 이미 존재한다는 문구를 출력한다. 학생 리스트에 없는 학번이면 이름, 중간고사 점수, 기말고사 점수를 입력 받아 리스트로 저장한다.

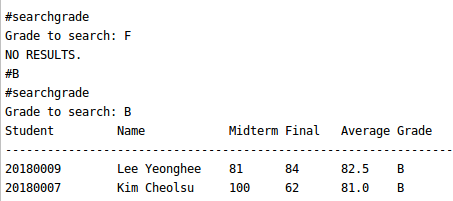
• 학생 추가 이후 show 명령어를 입력해 확인해 보면 추가한 학생들이 모두 목록에 정렬되고, 평균과 학점을 계산되어 있는 것을 확인할 수 있다.



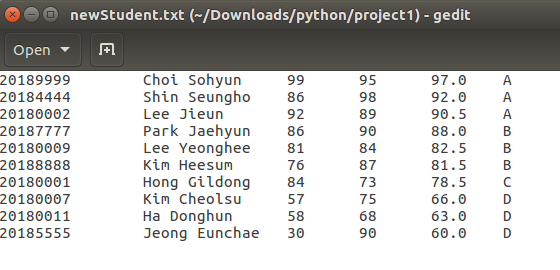


• 사용자가 searchgrade라는 명령어를 입력하면 검색하고 싶은 학점을 입력한다. 지정한 학점인 A,B,C,D,F가 아니면 더 이상 실행되지 않고 명령어를 벗어난다. 또, 예를 들어 학생 리스트 안에 F학점이 없는데 검색하려고 하면 검색 결과가 없다는 출력을 한다. 학생 리스트에 있는 학점을 검색하고 출력 결과가 여러 개이면 평균 점수를 기준으로 정렬한 결과를 출력한다.





• 사용자가 quit라는 명령어를 입력하면 데이터 저장 여부를 묻는다. no라고 입력하면 변경 내용을 저장하지 않고 프로그램을 종료한다. yes라고 입력하면 새로 저장할 파일 명을 입력 받고 해당 파일에 학번부터 학점까지 정렬되어 저장되는 것을 확인할 수 있다.



5. 토론

* 프로그램을 만들 때 학생 정보 파일을 중첩 리스트에 저장했는데, 리스트의 내용을 수정하는 함수를 구현하려고 할 때 call by value/call by object reference와 관련된 문제가 발생했다. 함수 내에서는 리스트의 내용이 잘 수정됐지만, 해당 함수를 벗어나면 원본 파일의 형태로 되돌아갔기 때문이다. 하지만 Mutable, immutable한 변수에 대한 변수에 대해서도 알아봤지만, 리스트는 mutable한 변수형이었다. 오랫동안 고민한 결과 최종적으로 문제가 발생한 이유는 파일을 읽어 리스트에 저장하는 함수를 따로 만들었기 때문이었다. 함수를 따로 만들면서 함수를 부를 때마다 원본 파일을 읽게 되면서 저장이 되지 않았던 것이다. 함수화로 인한 변수들의 충돌 문제가 존재할 수 있다는 것을 프로그램 제작을 통해 알게 되었다.
* 위와 비슷한 개념으로, 초안 함수에서는 파일을 읽고 저장하는 것에 대한 정의가 스스로 내려지지 못해 명령어 함수마다 파일을 계속해서 읽고 저장, 수정까지 되도록 하는 실수를 하기도 했다. 프로그램을 실행하는 동안은 하나의 리스트에 계속 수정되다가 종료 함수를 쓸 때만 저장이 된다는 것을 나중에 이해했기 때문이다. 초안 함수에 이해한 내용을 바탕으로 수정 작업을 거쳐 프로그램 실행이 성공적으로 될 수 있도록 진행했다.
* tab(\t) 문제도 발생했다. 처음 txt file을 생성할 때 Tab 키를 사용해 항목마다 tab을 입력했지만 split이 되지 않고 오류가 발생했다. 이 문제를 거의 마지막까지 해결하지 못해 문자열로 \t를 입력하고 r’\t’로 split을 진행했다. 마지막에 종료 함수(quit)를 정의하면서 새롭게 저장한 파일을 \t으로 split을 해 보니 오류가 없는 것을 확인하고 처음 파일과 비교해 보았다. 비교 결과, 처음 작성한 파일에는 tab이 띄어쓰기 4번으로 삽입되어 있었고, 마지막에 저장한 파일은 tab 하나로 삽입되어 있었다. 코딩을 할 때는 문자 하나가 다르게 읽히면 까지 세세하게 확인해야 한다는 것을 깨달았다.
* 코드의 길이를 줄이는 것도 중요한 부분인 것을 알게 되었다. 초반 코딩을 할 때, 반복되는 함수나 작업을 계속 복사해서 붙여 넣었는데 한 함수의 코드 길이가 너무 길어져 가독성이 떨어졌다. 중복되는 부분을 빨리 찾아 함수로 묶는 작업이나 필요한 코드만 입력하는 것이 가독성과 코드의 효율을 높인다는 것을 알았다.

6. 결론 및 개선 방법

* 본 과제에서는 중첩 리스트를 사용해 파일을 읽고, 수정하고 저장까지 하는 프로그램을 구현했다. 구현 과정에서 코드를 효율적으로 작성하는 방법을 익혔다. 특히, 생각하고 있던 코드보다 훨씬 짧은 코드로 구현이 가능한 것을 경험했다. 예를 들어, 중첩 리스트에서 일부 열만 따로 저장하려고 할 때 오류가 많이 났었는데, 한 줄로 일부 열만 저장할 수 있는 방법이 있었다.
* 코드에 반복문이 다수 존재하는데, 반복문이 많이 있는 것은 코드의 효율성을 떨어트린다. 학생 데이터 수가 많아질 수록 효율성이 떨어질 것이므로 반복문을 최대한 쓰지 않는 방법의 코드를 고민해 볼 필요가 있다.
* 코드를 작성하기 시작하는 시점에서 주석도 함께 작성해야 가독성을 늘릴 수 있다. 코드를 완성한 후에 주석을 쓰게 되면 변수나 코드의 의미를 다르게 작성할 수도 있다.