

파이썬 프로젝트

C반 4조

이름 : 고다영

이메일 : kkoda327@naver.com

명예서약(Honor code)

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

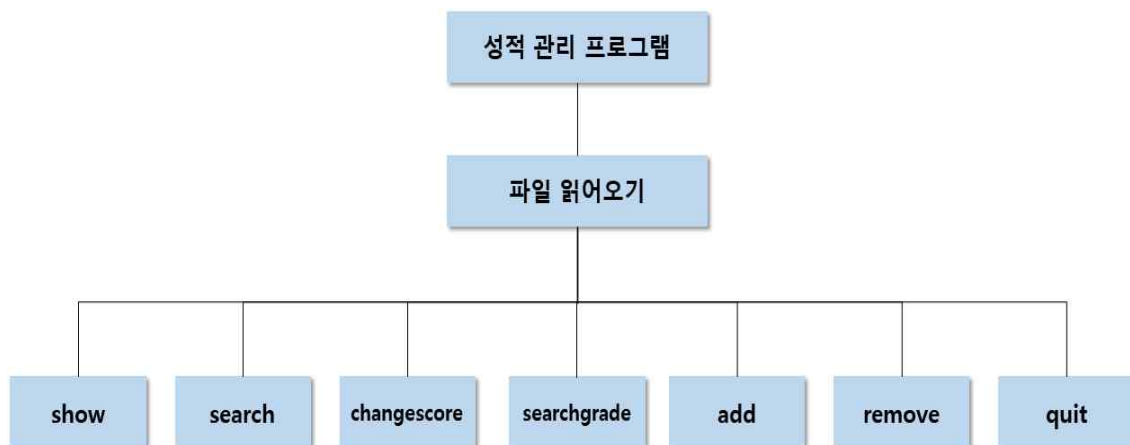
Problem : 성적 관리 프로그램

1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- 프로그램 실행 시 파일로부터 학생들의 성적 목록을 읽어온다.
- 읽어 온 성적 목록으로부터 각 학생마다의 평균 점수와 학점을 계산한다.
- 사용자가 성적을 관리하기 위한 명령어를 입력하였을 때 아래와 같은 기능을 수행하여 성적을 관리한다.
 - show 입력 시, 전체 성적 목록을 평균 점수를 기준으로 내림차순으로 출력한다.
 - search 입력 시, 찾고자 하는 학번과 일치하는 학생의 성적 정보를 출력한다.
 - changescore 입력 시, 목록에 저장된 학생 중 1명의 중간고사(mid) 혹은 기말고사(final)의 점수를 수정한다.
 - add 입력 시, 새로운 학생의 성적 정보를 입력받는다.
 - searchgrade 입력 시, 특정 grade를 입력 받아 그 grade에 해당하는 학생을 모두 출력한다.
 - remove 입력 시, 특정 학생의 성적 정보를 삭제한다.
 - quit 입력 시, 프로그램을 종료한다.

이 때 사용되는 성적 관리 프로그램의 구조 차트(structure chart)는 아래와 같다.



2. 알고리즘

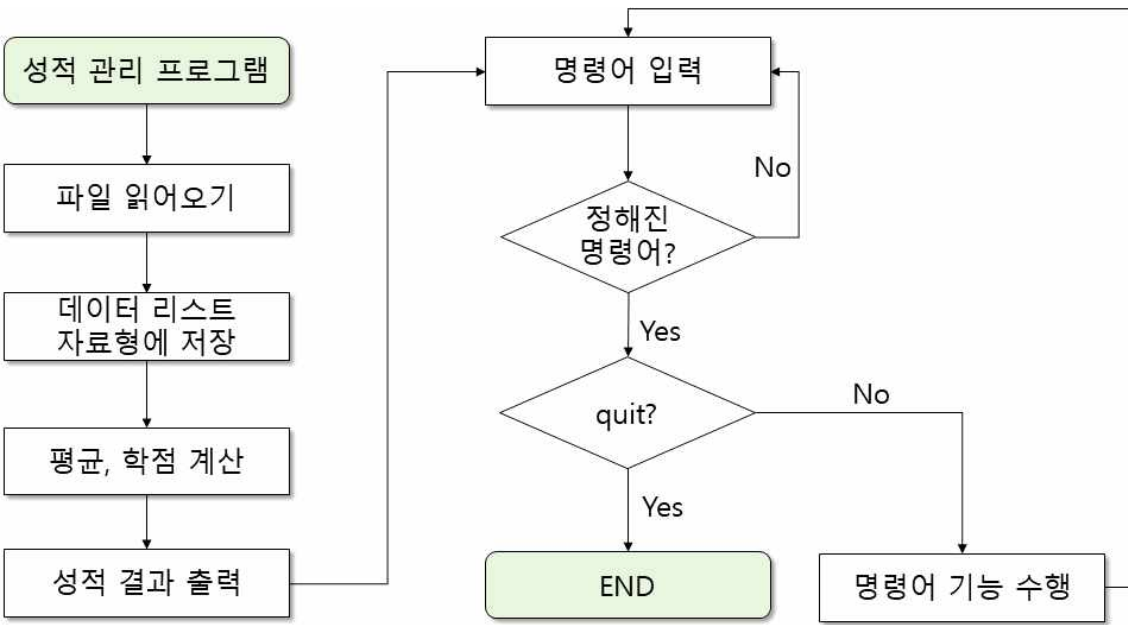
본 프로그램 작성을 위한 알고리즘을 Pseudo 코드 형태로 나타내면 다음과 같다.

Pseudo-algorithm for 성적 관리 프로그램

//프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

- 1 프로그램 실행 시 입력된 파일명을 받아온다. 파일명을 입력하지 않았을 경우, 'students.txt'를 기본으로 한다.
- 2 해당 파일명을 가진 파일을 읽기모드로 불러와서 파일에 저장되어 있는 데이터를 읽어온다.
- 3 각 학생의 데이터를 리스트 자료형으로 저장한다.
- 4 각 학생의 평균 점수를 계산하고 각 평균 점수가 해당하는 학점을 계산하여 평균 점수와 학점 모두 각 학생의 데이터에 추가한다.
- 5 저장된 성적 목록 결과를 평균 점수를 기준으로 내림차순 정렬하여 화면에 출력한다. 동일한 평균 점수를 가진 학생들이 있는 경우 순서는 상관없다.
- 6 # 뒤에 명령어를 입력받는다. 명령어의 대소문자를 구별하지 않기 위해 lower()를 사용하여 명령어는 모두 소문자로 전달한다.
- 7 입력된 명령어가 show, search, changescore, searchgrade, add, remove, quit 중 하나일 때 해당 명령어에 맞는 기능을 수행한다. 위의 명령어가 아닌 잘못된 명령어가 입력될 시 에러 메시지 없이 다시 # 뒤에 명령어를 입력받도록 한다.
- 8 위 7~8 과정을 무한반복문으로 반복한다.
- 9 quit 명령어가 입력될 경우, 파일의 저장 여부를 묻은 이후 무한 반복문을 끝내고 프로그램을 종료한다.

위의 의사 알고리즘을 Flowchart를 통해 표현하면 아래와 같다.



추가적으로 아래는 각 명령어를 수행하기 위한 알고리즘을 Pseudo 코드 형태로 나타낸 것이다.

Pseudo-algorithm for 성적 관리 프로그램 - show

// 프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

- 1 저장된 성적 목록을 평균 점수를 기준으로 내림차순 정렬한다.
- 2 내림차순으로 정렬된 성적 목록을 학번, 이름, 중간고사, 기말고사, 평균, 학점 순서로 학생별로 한 줄씩 출력한다.

Pseudo-algorithm for 성적 관리 프로그램 - search

// 프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

- 1 찾을 학번을 입력받는다.
- 2 입력받은 학번에 해당하는 학생이 없다면 'NO SUCH PERSON'을 출력하도록 한다.
- 3 입력받은 학번에 해당하는 학생이 있다면 해당 학번 학생의 성적 정보를 학번 이름, 중간고사, 기말고사, 평균 학점 순서로 한 줄에 출력한다.

Pseudo-algorithm for 성적 관리 프로그램 - changescore

//프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

- 1 성적을 바꿀 학번을 입력받는다.
- 2 입력받은 학번에 해당하는 학생이 없다면 'NO SUCH PERSON'을 출력한다.
- 3 입력받은 학번에 해당하는 학생이 있다면 중간고사 성적을 바꿀지 기말고사 성적을 바꿀지 입력받는다.
- 4 중간고사, 기말고사 둘 중 하나를 올바르게 입력했다면 바꿔 줄 새로운 성적을 입력받는다.
- 5 만약 입력한 새로운 성적이 0점부터 100점 사이라면 바뀐 결과를 성적 목록을 저장해놓은 리스트에 저장하고 바뀌기 전과 후 결과를 출력한다.

Pseudo-algorithm for 성적 관리 프로그램 - add

//프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

- 1 추가할 학번을 입력받는다.
- 2 입력받은 학번에 해당하는 학생이 이미 있다면 'ALREADY EXISTS.'를 출력한다.
- 3 입력받은 학번에 해당하는 학생이 없다면 순서대로 이름, 중간고사 성적, 기말고사 성적을 입력받고 평균과 학점을 계산한다.
- 4 3에서 입력받은 값과 계산한 결과를 성적 목록을 저장해놓은 리스트에 추가한다.

Pseudo-algorithm for 성적 관리 프로그램 - searchgrade

//프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

- 1 검색할 학점을 입력받는다.
 - 2 검색할 학점을 가진 학생이 없는 경우 'NO RESULTS.'를 출력한다.
 - 3 검색할 학점을 가진 학생이 있는 경우 해당 학생들을 학번 이름, 중간고사, 기말고사, 평균 학점 순서로 한 줄에 각각 출력한다.
-

Pseudo-algorithm for 성적 관리 프로그램 - remove

//프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

- 1 성적 목록이 비어있는 경우 'List is empty'를 출력한다.
- 2 성적 목록이 비어있지 않은 경우 제거할 학생의 학번을 입력받는다.
- 3 입력받은 학번을 가진 학생이 없는 경우 'NO SUCH PERSON.'을 출력한다.
- 4 입력받은 학번을 가진 학생이 있는 경우 성적 목록 리스트에서 해당 학생에 관련한 정보를 삭제한다.

Pseudo-algorithm for 성적 관리 프로그램 - quit

//프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

- 1 성적 목록 데이터를 저장할지 묻는다.
 - 2 저장하지 않을 경우 프로그램을 종료한다.
 - 3 저장할 경우 저장할 파일명을 입력받는다.
 - 4 평균 점수를 기준으로 내림차순으로 정렬한 이후 해당 파일명으로 파일을 저장한다.
 - 4 저장한 이후 프로그램을 종료한다.
-

3. 프로그램 실행 방법 및 예제

```
C:\python\AI\Python Programming>python project.py students.txt
```

```
C:\python\AI\Python Programming>python project.py
```

| Student | Name | Midterm | Final | Average | Grade |
|----------|--------------|---------|-------|---------|-------|
| 20180002 | Lee Jieun | 92 | 89 | 90.5 | A |
| 20180009 | Lee Yeonghee | 81 | 84 | 82.5 | B |
| 20180001 | Hong Gildong | 84 | 73 | 78.5 | C |
| 20180011 | Ha Donghun | 58 | 68 | 63.0 | D |
| 20180007 | Kim Cheolsu | 57 | 62 | 59.5 | F |

```
#_
```

- command 창에서 위 명령어를 통해 프로그램을 실행한다. 파일명 입력 시 해당 파일명에 저장된 성적 정보를 불러오고 파일명을 입력하지 않을 경우 students.txt 를 default로 불러온다.
- 불러온 전체 성적 목록을 평균 점수를 기준으로 내림차순으로 정렬하여 출력한다. 평균 점수는 소수점 아래 한자리까지 출력된다.

```
#find
#_
```

- 명령어 입력 시 7개의 명령어(show, search, changescore, searchgrade, add, remove, quit)를 제외한 명령어가 입력될 경우, 에러 메시지 없이 다시 명령어를 입력받을 준비를 한다.

```
#show
Student      Name      Midterm   Final    Average  Grade
-----
20180002     Lee Jieun    92        89       90.5     A
20180009     Lee Yeonghee 81        84       82.5     B
20180001     Hong Gildong 84        73       78.5     C
20180011     Ha Donghun   58        68       63.0     D
20180007     Kim Cheolsu  57        62       59.5     F

#ShoW
Student      Name      Midterm   Final    Average  Grade
-----
20180002     Lee Jieun    92        89       90.5     A
20180009     Lee Yeonghee 81        84       82.5     B
20180001     Hong Gildong 84        73       78.5     C
20180011     Ha Donghun   58        68       63.0     D
20180007     Kim Cheolsu  57        62       59.5     F
```

- 사용자가 명령어 입력 시, 대소문자를 구분하지 않고 동일한 명령어의 기능을

수행하도록 한다.

- 명령어 show 입력 시, 전체 학생의 목록을 위와 같이 평균 점수를 기준으로 내림차순으로 출력한다.
- 평균 점수는 소수점 아래 첫째 자리까지 출력한다.
- 하나의 명령어가 끝나면 다음 명령어를 입력받을 준비를 한다.

```
#search
Student ID: 20180050
NO SUCH PERSON.

#search
Student ID: 20180002
Student          Name      Midterm   Final   Average  Grade
-----
20180002         Lee Jieun    92        89      90.5     A
```

- search 입력 시 검색할 학번을 입력받고 해당 학번을 가진 학생의 성적 정보를 출력한다. 해당 학번을 가진 학생이 없을 경우 에러 메시지를 출력한다.

```
#changescore
Student ID: 20180050
NO SUCH PERSON.

#changescore
Student ID: 20180007
Mid/Final? miid

#changescore
Student ID: 20180007
Mid/Final? mid
Input new score: 147

#changescore
Student ID: 20180007
Mid/Final? mid
Input new score: 75
Student          Name      Midterm   Final   Average  Grade
-----
20180007         Kim Cheolsu    57        62      59.5     F
Score changed.
Student          Name      Midterm   Final   Average  Grade
-----
20180007         Kim Cheolsu    75        62      68.5     D
```

- changescore 입력 시, 성적을 바꿀 학번, 중간고사(mid), 기말고사(final) 중 어떤 성적을 바꿀 것인지, 바꿀 성적을 차례대로 입력받는다. 이후 바뀌기 전과 후의 결과를

출력한다. 예외적으로 해당 학번의 학생이 없을 경우, mid, final 이외의 값이 입력될 경우, 0부터 100 범위를 초과하는 성적이 입력된 경우 실행되지 않고 다음 명령어를 입력받는다.

```
#add
Student ID: 20180001
ALREADY EXISTS.
```

```
#add
Student ID: 20180021
Name: Lee Hyori
Midterm Score: 93
Final Score: 95
Student added.
```

```
#add
Student ID: 20180006
Name: Lee Sangsun
Midterm Score: 77
Final Score: 66
Student added.
```

```
#show
Student      Name      Midterm   Final   Average   Grade
-----
20180021      Lee Hyori    93       95      94.0      A
20180002      Lee Jieun    92       89      90.5      A
20180009      Lee Yeonghee 81       84      82.5      B
20180001      Hong Gildong 84       73      78.5      C
20180006      Lee Sangsun  77       66      71.5      C
20180007      Kim Cheolsu  75       62      68.5      D
20180011      Ha Donghun   58       68      63.0      D
```

#

- add 입력 시 위와 같이 학생의 학번, 이름, 중간고사 점수, 기말고사 점수를 차례로 입력받고 성적 정보 list에 추가한다. 예외적으로 해당 학번을 가진 학생이 이미 있는 경우 에러 메시지를 출력한다.

```
#searchgrade
Grade to search: E
```

```
#searchgrade
Grade to search: F
NO RESULTS.
```

```
#searchgrade
Grade to search: D
Student      Name      Midterm   Final   Average   Grade
-----
20180007      Kim Cheolsu  75       62      68.5      D
20180011      Ha Donghun   58       68      63.0      D
```

- searchgrade 입력 시, 검색할 학번을 입력받고 해당 학번을 가진 학생들의 성적 정보를 출력한다. 예외적으로 A, B, C, D, F 외의 학번이 입력된 경우 실행되지 않고 다음 명령어를 입력받으며 해당 학번을 가진 학생이 없는 경우 에러 메시지를 출력한다.

```
#remove
List is empty.
#remove
Student ID: 20180030
NO SUCH PERSON.

#remove
Student ID: 20180011
Student removed.

#show
Student      Name      Midterm   Final   Average   Grade
-----
20180021      Lee Hyori    93        95      94.0      A
20180002      Lee Jieun    92        89      90.5      A
20180009      Lee Yeonghee 81        84      82.5      B
20180001      Hong Gildong 84        73      78.5      C
20180006      Lee Sangsun  77        66      71.5      C
20180007      Kim Cheolsu  75        62      68.5      D
```

- remove 입력 시, 제거할 학생의 학번을 입력받고 해당 학생의 성적 정보를 제거한다.
show를 통해 remove가 성공적으로 실행되었음을 확인할 수 있다. 이 때 성적 목록이 비어있는 경우와 해당 학번을 가진 학생이 없던 경우 에러 메시지를 출력함을 확인할 수 있다.

```
#quit
Save data?[yes/no] no

C:\python\AI\Python Programming>

#quit
Save data?[yes/no] yes
File name: newStudents.txt

C:\python\AI\Python Programming>■
```

- quit 입력 시, 현재까지 편집한 내용의 저장 여부를 묻고 저장을 선택할 경우(yes) 파일명을 입력받아 성적 정보를 저장하고 프로그램을 종료한다. 저장하지 않는다고 선택할 경우(no) 파일명을 묻지 않고 프로그램은 바로 종료된다.

4. 토론

- 사용자로부터 명령어를 입력받은 이후, 입력받은 명령어를 lower()를 통해 모두 소문자로 변환하여 기능을 수행하도록 하면 대소문자를 구분하지 않을 수 있다.
- 기본적으로 모든 성적관리는 학번을 기준으로 이루어져 있다. 따라서 성적 정보를 저장한 list 외에 학번만을 모아 list로 저장해두면 해당 학번을 가진 학생이 존재하는지 아닌지 쉽게 확인할 수 있다.
- 학생들의 성적 정보를 출력할 때, 그리고 quit 입력 이후 학생들의 성적 정보를 저장할 때 모두 평균 점수를 기준으로 내림차순으로 하게 된다. 이 때 lambda 함수를 사용하면 별도의 함수를 선언하지 않고도 쉽게 정렬한 결과로 수행할 수 있다.

5. 결론

- 본 과제를 통해 수행하고자 하는 기능을 갖춘 함수를 선언하고 작성하는 방법을 학습할 수 있었다. 특히 모든 과정을 리스트를 통해 수행함으로써 리스트를 통한 데이터 처리 방법을 완벽하게 학습할 수 있었다.