

성적 관리 프로그램

포스코 청년 청년 AI-Big Data 아카데미 22기

B반 1조

김수환

s_0000000@nate.com

1. 알고리즘 (Pseudo Code)

- (1) 학생의 정보를 받을 빈 리스트를 생성한다. `sys.argv` 리스트의 길이가 1이라면, 리스트의 끝에 'students.txt'를 추가한다. 파일을 읽기모드로 열고 `rstrip()`, `split()` 함수를 사용하여 줄 바꿈 문자를 제거하고 문자열을 공백을 기준으로 분리해 리스트로 만든다. 변형된 리스트를 설정해놓은 'student_list'에 추가한 후 파일을 닫는다.
- (2) 입력 받을 7개의 명령어(show, search, changescore, searchgrade, add, remove, quit)를 설정하고 사용자로부터 명령어를 입력 받게 한다. 해당 명령어를 입력하면 각 기능을 수행하게 한다.
- (3) 명령어가 'show'라면 현재 저장된 학생 정보 목록을 성적 순으로 정렬하고 출력한다.
- (4) 명령어가 'search'라면 학생 ID를 입력하도록 요청한다. 학생 ID가 일치 하는지 차례로 확인 후, 해당 학생의 정보를 출력한다. 검색된 학생 정보가 없으면 'NO SUCH PERSON' 메시지를 출력한다.
- (5) 입력한 명령어가 'changescore' 이라면 사용자로부터 학생의 ID를 입력 받는다. 해당 정보를 가진 학생이 존재하는지 확인 후, 존재하지 않는다면 'NO SUCH PERSON'을 출력하고 다시 명령어 입력을 받는다. 존재한다면, Mid/Final을 입력 받아 중간고사 성적을 수정 할 것인지 기말고사 성적을 수정 할 것인지 결정한다(입력 받은 값이 'mid'나 'final'이 아니라면, 다시 명령어 입력을 받는다. 결정 후, 새로운 점수를 입력 받고 그 값이 0과 100사이인지 확인한다. 범위에 들지 않는 경우 다시 점수를 입력 받도록 한다. 0에서 100 사이라면 해당 학생의 점수를 새로 받은 값으로 변경한다.
- (6) 만약 명령어가 'searchgrade'라면, 검색 대상 등급을 입력 받는다. 정보를 저장 할 새로운 student 리스트를 생성하고 student_list에서 검색 등급과 일치하는 학생을 찾아서 student 리스트에 추가 후, 학생의 평균 점수를 계산하여 학생 정보에 추가한다. 검색한 등급이 리스트에 없을 경우 "NO RESULTS"라는 메시지를 출력한다. student 리스트에 해당 등급 학생의 정보가 있을 경우 그 학생의 정보만 출력한다.
- (7) 명령어에 'add'가 입력 될 경우 학생 ID를 입력 받는다. student_list에 동일한 ID를 가진 학생이 있는지 검사한다. 같은 ID를 가진 학생이 있다면 에러 메시지를 출력하고 다시 입력 받도록 한다. 새로운 ID를 입력 받는 경우 학생이름, 중간고사 점수, 기말고사 점수를 입력 받는다. 점수의 평균값을 구하고 평균값에 대한 등급을 계산해 학생 정보를 리스트 형태로 student_list에 추가한다.
- (8) 'remove' 입력 시 입력 받은 학생의 정보를 삭제시키는 역할을 수행하게 한다. 만약 student_list가 비어있다면 "List is empty"를 출력한다. 삭제하고 싶은 학생의 ID를 입력하면 리스트에서 해당 학생의 정보가 완전히 삭제된다.
- (9) 사용자가 'quit' 명령어를 입력하면, 먼저 저장 여부를 묻는다. 'yes'를 입력할 시 파일 이름을 입력 받아 학생들의 성적을 기준으로 내림차순 정렬하여 파일에 저장한다. 저장되는 파일의 정보는 학생ID, 이름, 중간고사 및 기말고사 점수, 평균 점수, 학점이다. 사용자가 'no'를 입력하면 저장하지 않고 종료한다.

2. 프로그램 구조 및 설명(실행 결과)

- 성적 관리 프로그램 사용자는 7개의 명령어 (show, search, changescore, searchgrade, add, remove, quit)를 사용할 수 있으며, 사용하고 싶은 각 명령어를 입력하면 기능이 실행된다.

```
(base) plai@plai-Precision-7920-Tower:~/다운로드$ python3 project.py students.txt
Student      Name      Midterm Final  Average Grade
-----
20180001     Hong Gildong    84      73      78.5      C
20180002     Lee Jieun      92      89      90.5      A
20180007     Kim Cheolsu    57      62      59.5      F
20180009     Lee Yeonghee   81      84      82.5      B
20180011     Ha Donghun     58      68      63.0      D
# ^Z
[2]+  정지됨                  python3 project.py students.txt
(base) plai@plai-Precision-7920-Tower:~/다운로드$ python3 project.py
Student      Name      Midterm Final  Average Grade
-----
20180001     Hong Gildong    84      73      78.5      C
20180002     Lee Jieun      92      89      90.5      A
20180007     Kim Cheolsu    57      62      59.5      F
20180009     Lee Yeonghee   81      84      82.5      B
20180011     Ha Donghun     58      68      63.0      D
#
```

- 파일명을 입력하지 않을 경우, default로 “students.txt”로부터 데이터를 읽는다. 파일명이 입력되는 경우와 입력되지 않는 경우 모두 프로그램이 실행된다.

```
# find
# asd
# qwe
# zxc
#
```

- 7개 외 잘못된 명령어 입력 시, 에러 메시지 없이 다시 명령어를 입력 받을 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
#
```

- show 명령어 : show 입력 시 , 텍스트 파일로부터 데이터를 읽어 리스트 형태로 값을 저장하고 전체 목록을 평균 기준 내림차순 정렬하여 출력한다.

```
# search
Student ID : 20181133
NO SUCH PERSON.

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

- search 명령어 : search 입력 시, 학번이 입력된 학생의 이름, 점수, 평균, 학점을 출력한다.

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

- 리스트에 없는 학번인 경우 에러 메시지 출력.

```
# changescore
Student ID: 20180009
Mid/Final? mid

# changescore
Student ID: 20180009
Mid/Final? mid
input new score :150
```

- mid/final 외의 값이 입력된 경우에는 실행되지 않고 다시 명령어를 받는다. 수정 하는 점수가 0~100 이외의 값이 입력된 경우 실행되지 않고 명령어를 받는 창으로 다시 돌아간다.

```
# changescore
Student ID: 20180009
Mid/Final? mid
input new score :99
Score changed.
# show
Student          Name          Midterm Final  Average Grade
-----
20180009         Lee Yeonghee      99      84      91.5    A
20180002         Lee Jieun         92      89      90.5    A
20180001         Hong Gildong      84      73      78.5    C
20180011         Ha Donghun        58      68      63.0    D
20180007         Kim Cheolsu       57      62      59.5    F
#
```

- 학생 ID와 중간/기말 점수가 제대로 입력된 경우, 입력 받은 값으로 점수와 평균 점수가 변경되고 평균 점수에 따라 학점도 다시 계산하여 수정 후 출력한다.

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

- add 명령어 : add 입력 시 먼저 학생의 학번을 입력 받는다. 이미 존재할 경우 에러 메시지와 함께 다시 명령어를 받는 화면으로 돌아간다.

```
# add
Student ID: 20180021
Name: Lee Hyori
Midterm Score: 98
Final Score: 78
Student added.

# add
Student ID: 20180006
Name: Lee Sangsun
Midterm Score: 78
Final Score: 66
Student added.
```

- 제대로 된 입력 값을 받으면 학생의 학번, 이름 중간/기말고사 점수를 차례로 요구해 입력 받는다. 입력해 정보를 성공적으로 추가하면 Student added 메시지를 출력한다.

```
# show
Student      Name      Midterm Final  Average Grade
-----
20180002     Lee Jieun    92      89     90.5    A
20180021     Lee Hyori    98      78     88.0    B
20180009     Lee Yeonghee 81      84     82.5    B
20180001     Hong Gildong 84      73     78.5    C
20180006     Lee Sangsun  78      66     72.0    C
20180011     Ha Donghun   58      68     63.0    D
20180007     Kim Cheolsu  57      62     59.5    F

#
```

- 추가 시킨 후, show 명령어를 입력하면 추가된 학생의 정보를 평균 기준 내림차순으로 정렬해 출력한다.

```
Grade to search : E
NO RESULTS.

# searchgrade
Grade to search : B
Student      Name      Midterm Final  Average Grade
-----
20180009     Lee Yeonghee 81      84     82.5    B
20180021     Lee Hyori    98      78     88.0    B
# searchgrade
Grade to search : C
Student      Name      Midterm Final  Average Grade
-----
20180001     Hong Gildong 84      73     78.5    C
20180006     Lee Sangsun  78      66     72.0    C
```

- searchgrade 명령어 : 없는 값을 입력 받으면 NO RESULTS 라는 에러 메세지가 출력 되고, 특정 학점 입력 시 그 학점에 해당하는 학생을 모두 출력한다.

```
# remove
Student ID: 20180001
Student removed.

# remove
Student ID: 20180002
Student removed.

# remove
Student ID: 20180007
Student removed.

# remove
Student ID: 20180009
Student removed.

# remove
Student ID: 20180011
Student removed.

# remove
List is empty.

#
```

- remove 명령어 : 학생이 목록에 없는 경우에는 빈 리스트라는 에러 메세지를 출력한다.

```
# remove
Student ID: 20180002
Student removed.

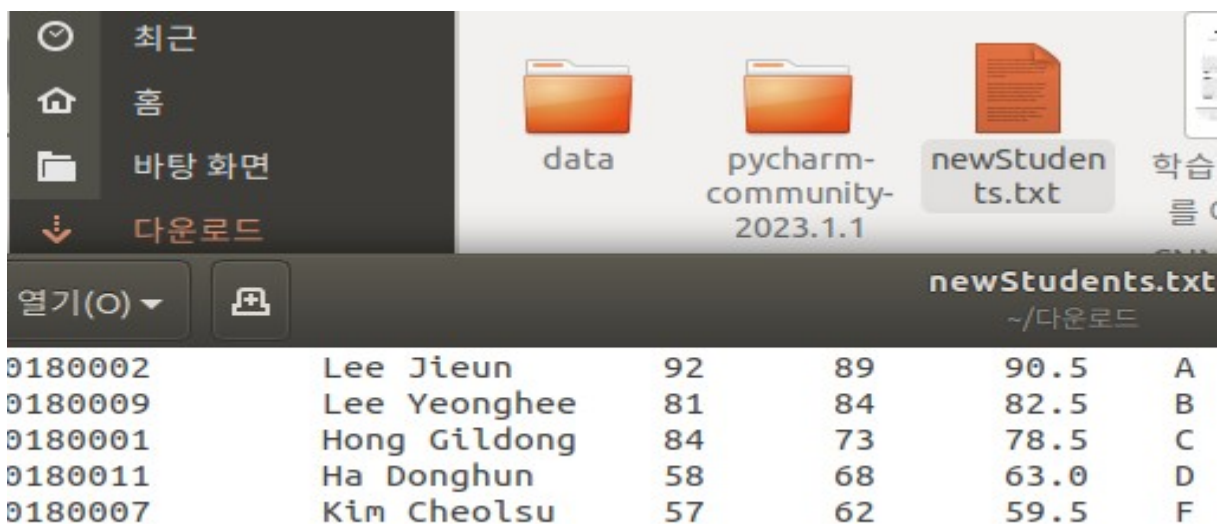
# show
Student      Name      Midterm Final  Average Grade
-----
20180009    Lee Yeonghee   81      84      82.5    B
20180001    Hong Gildong   84      73      78.5    C
20180011    Ha Donghun     58      68      63.0    D
20180007    Kim Cheolsu    57      62      59.5    F

#
```

- 삭제하고자 하는 학생의 학번을 입력 받은 후, 학생이 목록에 있는 경우 삭제한다. Show 명령어를 입력하면 삭제된 학생의 정보가 보이지 않는다.

```
# quit
Save data?[yes/no] no
(base) piyai@piyai-Precision-7920-Tower:~/다운로드$ python3 project.py
Student      Name      Midterm Final  Average Grade
-----
20180001      Hong Gildong      84      73      78.5      C
20180002      Lee Jieun      92      89      90.5      A
20180007      Kim Cheolsu      57      62      59.5      F
20180009      Lee Yeonghee      81      84      82.5      B
20180011      Ha Donghun      58      68      63.0      D
# quit
Save data?[yes/no] yes
Enter filename: newStudents.txt
```

- quit 명령어 : quit를 입력받으면 프로그램을 종료시킨다. 내용의 저장 여부를 묻고, no를 입력하면 저장하지 않고 종료시키고, yes를 입력할 경우 새로운 파일명을 입력 받아서 저장하도록 한다.



- 새로 입력 받은 파일이 저장된다.

3. 결론 및 느낀점

이번 프로젝트를 수행하면서 함수 작성 및 구현 기술(동작 과정)에 대해 학습할 수 있었고 for문을 작성하고 리스트에 값을 추가하는 코드를 작성하면서 for문의 동작에 대해 더 자세히 알 수 있었다. 또한 한번도 다루어 보지 않았던 파일 입출력 및 입력인수 전달하는 방법에 대해 학습할 수 있었다. 파이썬 지식을 잘 몰라서 프로젝트를 수행하는 과정에서 시간도 많이 소요되고 어려움이 있었지만 동료 교육생들과 함께 이 프로젝트에 필요한 파이썬 지식들을 인터넷에 서칭해가면서 실력도 향상되었고 리스트, 딕셔너리 등 파이썬 자료형의 특성에 대해 배울 수 있었다.