



1 주 차 과 이 션 프로젝트



14기 A조 양진미



CONTENTS

문제의 개요	3
알고리즘	3
프로그램 구조 및 설명	4
프로그램 실행 방법 및 예제	4
토론	11
결론 및 개선방향	11



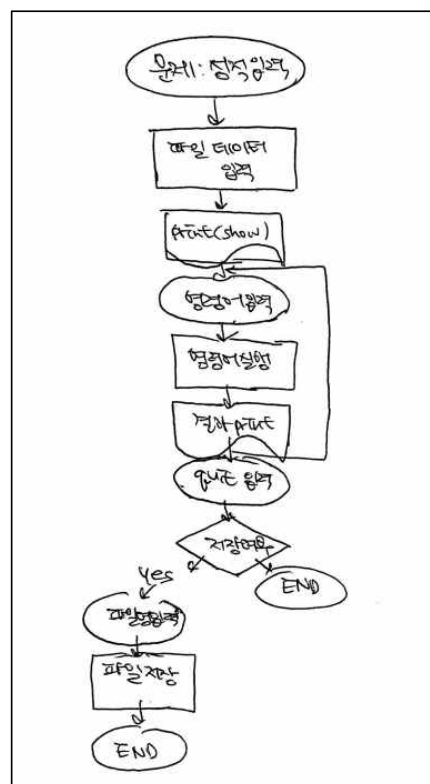
Problem : 성적 관리 프로그램

1. 문제의 개요

본 프로그램을 간단히 설명하자면 다음과 같다.

- 사용자가 작성해둔 성적 목록 파일의 데이터를 읽으며 학번, 이름, 중간고사 성적, 기말고사 성적을 입력 받는다.
- 프로그램을 실행함과 동시에 파일에 있는 성적 목록을 평균과 학점까지 계산하여 table로 바로 도출한다.
- 사용자가 원하는 명령어를 입력할 시, 명령어에 따라 프로그램을 작동하며, 기존 데이터를 수정 혹은 읽어온다.
- 마지막에는 사용자가 수정한 데이터를 파일에 저장 가능하게 한다.
- 입력부 : 사용자가 작성해둔 성적 목록 파일의 데이터를 읽고 저장한다.
사용자가 원하는 명령어를 입력 받고 저장한다.
- 처리부 : 사용자가 입력한 명령어를 받고 실행하여 결과를 반환한다.
- 출력부 : 사용자가 요청하여 변환이 완료된 데이터 값을 처음 받은 데이터의 형태로 저장한다.

2. 알고리즘





3. 프로그램 구조 및 설명

- a) 파일을 불러오기
 - 프로그램을 실행 할 때, open('파일명', 'r') 로 텍스트 파일을 읽어오게 한다.
 - f.readlines()을 매 행마다 반복문을 통해 실행하여, 빈 딕셔너리와 리스트를 만들고 실행 할 때마다 딕셔너리에 저장한다.
- b) 데이터의 유무 확인하기
 - if를 사용하여 데이터가 불러온 데이터 안에 있는 지 확인하는 과정을 계속 거친다.
- c) 데이터 값에 따라 성적 산출하기
 - if를 사용하여 데이터 평균 값에 따라 성적 산출
- d) 함수 정의하여 사용하기
 - def를 사용하여 함수를 정의한다.
- e) 데이터 저장하기
 - open('파일명', 'a')로 데이터를 저장하게 한다. 파일명이 생성되어 있지 않으면, 파일을 새로 생성한다.

4. 프로그램 실행방법 및 예제

```
piat@piat-Precision-7920-Tower: ~/바탕화면/testtt
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
(base) piat@piat-Precision-7920-Tower:~$ cd 바탕화면
(base) piat@piat-Precision-7920-Tower:~/바탕화면$ cd testtt
(base) piat@piat-Precision-7920-Tower:~/바탕화면/testtt$ python project1.py students.txt
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
#
```



- 프로그램을 실행하면 다음과 같은 화면으로, students.txt에 저장되어 있는 데이터를 읽어오고, 그에 따른 평균과 grade를 산출하여 table화 시키고, 바로 보여준다.
- 명령어를 입력하는 input 부분에 '#'을 표시하여 입력을 대기한다는 것을 알린다.

```
piai@piai-Precision-7920-Tower: ~/바탕화면/testtt
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
(base) pi ai@piai-Precision-7920-Tower:~$ cd 바탕화면
(base) pi ai@piai-Precision-7920-Tower:~/바탕화면$ cd testtt
(base) pi ai@piai-Precision-7920-Tower:~/바탕화면/testtt$ python project1.py students.txt
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
#
```

- show를 입력하면, 이제까지 입력되어 (수정되어) 저장된 데이터 값을 table화 하여 출력한다.

```
piai@piai-Precision-7920-Tower: ~/바탕화면/testtt
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
#add
Student ID: 20180001
ALREADY EXISTS.
#add
Student ID: 20180021
Name: jinmi yang
Midterm Score: 93
Final Score: 95
Student added.
#add
Student ID: 20180006
Name: lee hyori
Midterm Score: 77
Final Score: 66
Student added.
#show
Student          name      Midterm  Final   Average  Grade
-----
20180021         jinmi yang   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 hyori    77       66     71.5      C
20180007         Kim Cheolsu  75       62     68.5      D
20180011         Ha Donghun   58       68     63.0      D
#
```



- add를 입력하면, 다음과 같이 학생의 학번을 입력하게 하고, 그 이후엔 학생의 이름, 학생의 중간고사 성적, 기말고사 성적을 입력하게 한다.
- 이 때, 이미 데이터에 존재하는 학번을 입력할 경우, 이미 학생이 있다는 'already exists.' 멘트를 출력한다.
- 또한, 성적 값을 100 초과 값의 값을 입력하면, 성적을 100 이하로 입력 해야 한다는 'please enter a score below 100'을 출력한다.

```
piai@piai-Precision-7920-Tower: ~/바탕화면/testtt
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
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
#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 명령어 입력 후, 학번을 받아서 학번에 해당하는 학생의 정보를 table 화 하여 출력한다.

```
piai@piai-Precision-7920-Tower: ~/바탕화면/testtt
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
(base) piat@piai-Precision-7920-Tower:~$ cd 바탕화면
(base) piat@piai-Precision-7920-Tower:~/바탕화면$ cd testtt
(base) piat@piai-Precision-7920-Tower:~/바탕화면/testtt$ python project1.py students.txt
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
#search
Student ID: 20180050
NO SUCH PERSON.
#
```




- 이 때, 학번에 해당하는 학생의 정보가 없는 경우, 'no such pearson'이라는 문구를 출력한다.

```
#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
20180007        Kim Cheolsu  75       62     68.5      D
#remove
Student ID: 20180030
NO SUCH PERSON.
#remove
Student ID: 20180011
NO SUCH PERSON.
#remove
Student ID: 20180007
Student removed.
#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
#
```

- remove를 입력하여, 학번을 입력받아 학생의 정보를 저장된 데이터에서 지운다.
- 학번에 해당하는 데이터가 없다면, 'no such person'이라는 문구를 출력한다.
- 학번에 해당하는 학생의 데이터를 지움에 성공했다면, 'student removed.'이라는 문구를 출력하고, show 명령어를 입력하여 이를 확인 가능하다.

```
#searchgrade
Grade to search: E
Please enter another grade.
#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
#
```

- searchgrade로 학점을 입력하면, 그 학점에 해당하는 학생들을 table로 보여준다.
- 학점을 갖고 있는 학생이 존재하지 않는다면, 'no results' 멘트를 출력한다.



```
#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.
20180007         Kim Cheolsu    75      62     68.5     D
#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
20180007         Kim Cheolsu   75      62     68.5     D
20180011         Ha Donghun    58      68     63.0     D
#
```

- changescore 명령어를 입력하여, 학번을 입력 받아, 바꾸고 싶은 성적의 종류를 선택하고, 그에 따른 바꿀 성적을 입력한다.
- 성적을 바꾸는 것에 성공하면, 그 바꾸는 것에 성적이었던 전 데이터와 바꾸고 난 후의 데이터를 함께 출력하여, 성적 변환에 성공했다는 알림을 같이 출력한다.
- 이후, show 명령어를 통해 저장된 데이터가 바뀌어 있음을 확인 가능하다.

```
pi@pi-Precision-7920-Tower: ~/바탕화면/testt
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
20180001         Hong Gildong  84      73     78.5     C
20180011         Ha Donghun    58      68     63.0     D
20180007         Kim Cheolsu   57      62     59.5     F
#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
#changescore
Student ID: 20180050
NO SUCH PERSON.
#changescore
Student ID: 20180007
mid/final? mid
Please enter 'mid' or 'final'
#changescore
Student ID: 20180007
mid/final? mid
Input new score: 147
Please enter a score below 100.
#
```

- changescore 명령어 입력 후, 학생의 학번을 입력 했는데, 해당 학번의 데이터가 없다면, 'no such person'을



출력한다.

- 변경 할 성적 종류를 'mid' 혹은 'final' 외의 다른 값을 입력하면, 중간고사, 기말고사 성적만 입력 해야 한다는 문구를 출력한다.
- 바뀌야 할 성적을 100점 이상으로 입력하면, 성적을 100점 이하로 입력하라는 문구를 출력한다.

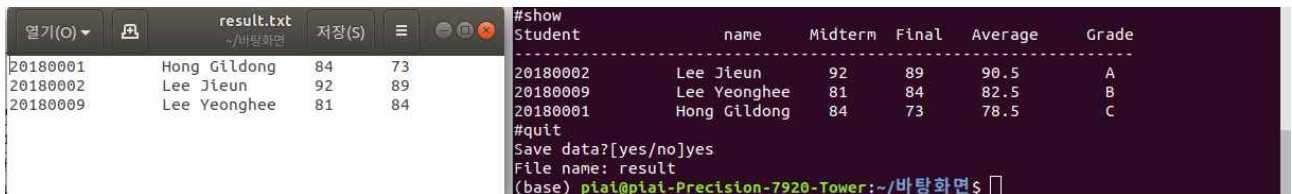
```
#show
Student          name      Midterm  Final   Average   Grade
-----
20180002         Lee Jieun    92       89     90.5       A
20180009         Lee Yeonghee 81       84     82.5       B
20180001         Hong Gildong 84       73     78.5       C
20180011         Ha Donghun   58       68     63.0       D
20180007         Kim Cheolsu  57       62     59.5       F
#add
Student ID: 20202020
Name: jinmi yang
Midterm Score: 99
Final Score: 100
Student added.
#search
Student ID: 20202020
Student          name      Midterm  Final   Average   Grade
-----
20202020         jinmi yang    99       100    99.5       A
#SearchGrade
Grade to search: A
Student          name      Midterm  Final   Average   Grade
-----
20180002         Lee Jieun    92       89     90.5       A
20202020         jinmi yang    99       100    99.5       A
#CHangeScore
Student ID: 20202020
mid/final? mid
Input new score: 78
Student          name      Midterm  Final   Average   Grade
-----
20202020         jinmi yang    99       100    99.5       A
Score changed.
20202020         jinmi yang    78       100    89.0       B
#show
Student          name      Midterm  Final   Average   Grade
-----
20180002         Lee Jieun    92       89     90.5       A
20202020         jinmi yang    78       100    89.0       B
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
#
```

- 이 때, 명령어를 대문자와 소문자를 구별하지 않고, 섞어서 입력해도, 명령어를 수행 가능하도록, upper() 혹은 lower() 함수를 이용하여, 실행시킨다.



```
(base) pi1ai@pi1ai-Precision-7920-Tower:~/바탕화면$ python project1.py students.tx
t
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
#quit
Save data?[yes/no]no
(base) pi1ai@pi1ai-Precision-7920-Tower:~/바탕화면$
```

- quit 명령어를 입력하면, 데이터를 저장할 것인지 묻는 문구를 출력하고, yes 또는 no를 입력하라는 명령어를 받을 준비를 한다.
- 이 때, 입력을 yes 또는 no를 하지 않은 경우, 'please enter 'yes' or 'no''를 출력한다.
- no를 입력한 경우, 프로그램에서 빠져나오게 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
#quit
Save data?[yes/no]yes
File name: result
(base) pi1ai@pi1ai-Precision-7920-Tower:~/바탕화면$
```

- yes를 입력할 경우, 저장할 파일명을 적도록 문구를 출력하고, 받은 파일명을 따라, 저장공간에 파일을 저장시킨다.
- 이 때, 기존 파일이 존재할 경우, 파일을 이어서 덧붙일 수 있도록 'a'로 파일을 open 하고, 만약 파일이 없다면, 파일을 새로 생성하여 저장하게 한다.

```
(base) pi1ai@pi1ai-Precision-7920-Tower:~/바탕화면$ python project1.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
#
```

- default 값으로, students.txt를 실행하지 않고 그냥 프로그램을 실행할 때, 자동으로 students.txt를 불러오게 한다.
- students.txt 가 아닌, 다른 파일명을 쓸 경우, import한 sys를 이용하여, 입력한 파일을 읽어서 프로그램을 실행한다.



5. 토론

- 프로그램 코드를 짤 때, changescore 함수 정의에서, 질문을 Mid/Final? 이라고 출력하라고 적혀 있었습니다. 그러나, 값은 mid 또는 final로 받는데, Mid/Final?로 묻는 것이 말이 되지 않아, mid/final?로 문구를 수정하였습니다.
- 또한, 100 이상의 값이 입력되면, 그냥 그 함수가 멈춰 버리고 끝이 나는데, 정확한 오류를 알아야, 다음에 다시 입력 하더라도, 수정하여 왜 오류가 났는지 알 수 있어야 한다고 생각했기 때문에, 'please enter a score below 100' 와, 'mid' 또는 'final' 외의 다른 문구를 입력 했을 때도 멈춰서 다시 돌아가기 때문에, 'please enter 'mid' or 'final' '문구를 출력하게 했고, searchgrade 함수에서도 지정되어 있지 않은 성적을 입력 할 때, 'please enter another grade.', quit 함수에서도 'yes' 또는 'no'외의 다른 문구를 입력할 때, 'please enter 'yes' or 'no'를 출력하게 했다.
- 명시되어 있지 않은 세부사항까지 결정하여 함께 코드를 짰다.

6. 결론 및 개선방향

이번 숙제를 통해, 이런 프로그램 하나하나를 다 검사해보실 조교님이 너무 힘드실 것 같은 생각이 들었다..808..

이번 숙제를 하면서, sys를 import하는 것을 생각하지 못한 탓에, 친구에게 조언을 구하여 다시 생각해보기도 했고, 특히 코드가 조잡하지만, 결국 프로그램 하나를 만들어냈다는 것에 내 자신에게 뿌듯함을 많이 느꼈던 과제였다.

내가 배웠던 아무 도움도 되지 않을 것 같았던 기초 지식들이 모여서, 이렇게 하나의 큰 프로그램을 이루니 그 시스템이 너무 대단했다!

이 프로그램에서는 따로 수정하지 못했지만, 중간에 지정되어 있지 않은 문구를 입력 할 때, 아예 그 명령어에서 빠져나가는 것이 너무 번거롭게 느껴질 것 같다. 하여, 잘못된 문구를 입력하여도, 바로 다시 제대로 입력하게 하거나, 입력어를 입력하다가, 빠져나가고 싶을 때, 빠져나갈 방법이 없으므로, 빠져나가게 만들 수 있도록 while 문을 전체 뿐만 아니라, 하나의 함수마다 걸어서, quit 입력어를 하나의 함수마다 걸어, break를 할 수 있도록 하면 더 좋은 프로그램이 될 것 같다. 또한, show에서도 다른 기준으로 정렬하여 보여줄 수 있는, 함수를 정의하면 더 사용자에게 다양한 편의성을 제공할 것 같다.