

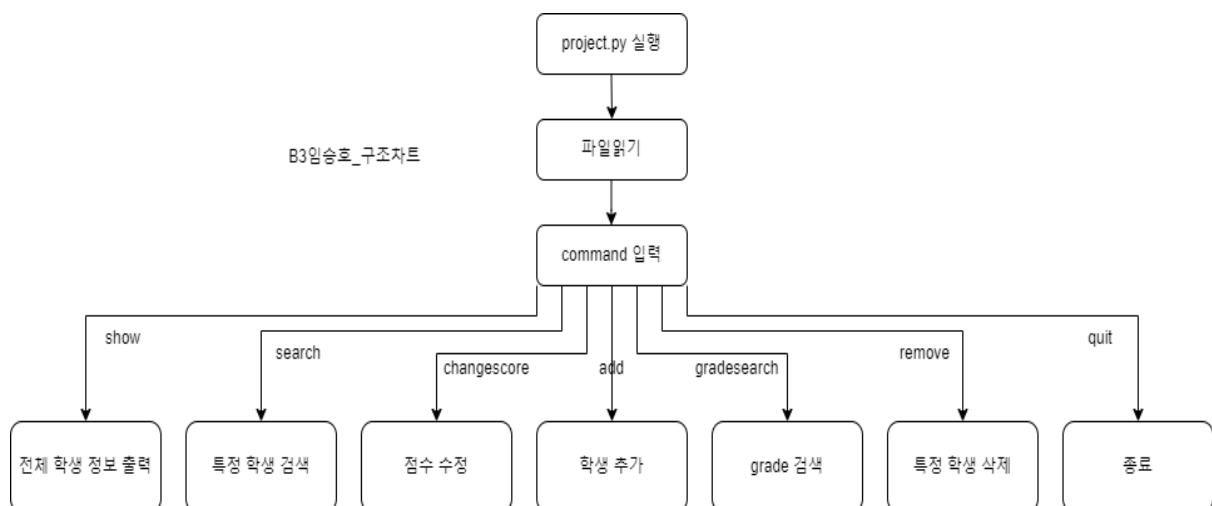
Problem 1: 성적 관리 프로그램

1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- 파일로부터 데이터를 읽어서 성적 목록을 만들어 관리하는 성적 관리 프로그램을 작성한다.
- 이 프로그램은 사용자로부터 7 개의 명령어(show, search, changescore, searchgrade, add, remove, quit)를 입력 받아 각 기능을 수행하게 된다.
- 최소한 각 명령어 별로 함수를 정의하여 사용하며, 명령어 외에 필요한 함수는 추가로 정의하여 사용할 수 있다.
- 사용자는 7 개의 명령어(show, search, changescore, searchgrade, add, remove, quit)를 사용할 수 있으며, 명령어를 입력하였을 때만 기능이 실행된다.
- 사용자가 명령어 입력 시, 대소문자를 구분하지 않고 동일한 명령어의 기능을 수행하도록 작성한다.
- 명시된 7 가지 명령어(show, search, changescore, searchgrade, add, remove, quit) 외의 명령어가 입력될 경우 무시하고 다시 명령어 입력을 대기한다.

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



- 1) 학생 정보리스트 파일 불러오기
- 2) 프로그램 실행
- 3) 사용자로부터 입력 받은 명령어에 따라 해당 기능 함수 실행
 - show: 학생 정보 출력
 - search: 학생 정보 검색
 - changescore: 학생 정보 변경
 - add: 학생 정보 추가
 - searchgrade: 학점으로 검색
 - quit: 프로그램 종료
- 4) 프로그램 종료

2. 알고리즘

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

Pseudo-algorithm for dot-line distance

//프로그램에 필요한 변수들은 미리 선언해 놓은 것으로 가정한다.

```
1  def start():
2      print( " Student Name Midterm Final Average Grade")
3      print("-----") #구분선
4
5  def show_function(slist):
6      start()
7      slist.sort(key=lambda x:x[4],reverse=True) #학생 평균을 내림차순 정렬
8      for( for(print(slist)) ) #이중for문을 통해 학생데이터 출력
9
```

```
10 def search_function(slist)
11     slist의 학번 = input(입력 받은 학번)
12     found = False로 할당
13     if slist의 학번 == 입력 받은 학번
14         start()
15         for(찾은 학생의 데이터 반복)
16             print()
17     if not found: print("NO SUCH PERSON")
18
19 def changescore_function(slist)
20     start()
21     found = False로 할당
22     for(slist):
23         if slist의 학번 == 입력 받은 학번
24             found = True
25         'mid' 또는 'final' 외의 값이 입력된 경우 실행되지 않음
26         새로운 점수 입력 받음
27         0~100 외의 값이 입력된 경우 실행되지 않음
28         for( for(print(slist)) ) #이중for문을 통해 학생데이터 출력
29         if 바꿀 점수 == 'mid':
30             중간고사 인덱스에 새로운 점수 바꿈
31         if 바꿀 점수 == 'final':
32             기말고사 인덱스에 새로운 점수 바꿈
33
```

```
34 def add_function(slist):
35     slist의 학번 = input(입력 받은 학번)
36     found=False로 할당
37     if slist의 학번 == 입력 받은 학번
38         found=True
39         print("ALREADY EXISTS.")
40     이름 입력 받기
41     중간고사 점수 입력 받기
50     중간고사 점수에 0~100외의 값이 입력된경우 실행되지않음
51     기말고사도 동일
52     평균과 학점을 계산해 slist에 추가
53
54 def searchgrade_function(slist):
55     grade를 입력 받음
56     grade not in ['A', 'B', 'C', 'D', 'F'] 인 경우 실행되지 않음
57     if slist의 학점 == 입력 받은 학점
58         for( for(print(slist) ) ) #이중for문을 통해 학생데이터 출력
59
60 def remove_function(slist):
70     if 빈리스트면
71         print("List is empty.")
72     elif slist의 학번 == 입력 받은 학번
73         해당학생을 리스트에서 삭제
74
```

```
75 def quit_function(slist):
76     answer = input(save data)
77     if answer == 'yes'
78         파일저장
79     elif answer == 'no'
80         pass
81     파일 읽기
82
83     while True:
84         if command == "show":
85             show_function
86         elif command == "search":
87             search_function
88         elif command == "changescore":
89             changescore_function
90         elif command == "add":
91             add_function
92         elif command == "searchgrade":
93             searchgrade_function
94         elif command == 'remove':
95             remove_function
96         elif command == "quit":
97             quit_function
```

3. 프로그램 구조 및 설명

a) 출력을 위한 틀 (start())

- 학생의 데이터를 출력할 때, 호출하여 사용하는 함수로 데이터의 인덱스와 구분선을 출력해준다.

b) 학생 출력 (show_function())

- show 명령어를 입력하면 실행되는 함수이다.
- 출력을 위한 기본 인덱스와 구분선을 start() 함수 호출을 통하여 출력한다.
- 학생 평균(내림차순)으로 출력하기 위해 sort(key=lambda x: x[4], reverse=True)를 사용했다.
- 정렬을 마친 후, 이중 for 문을 사용하여 각학생들의 데이터를 출력한다.

c) 학생 검색 (search_function())

- search 명령어를 입력하면 실행되는 함수이다.
- 초깃값으로 found 는 False 로 할당한 다음 s_id 가 일치하는 경우 found = True 로 변경한 다음 예외처리를 한다
- s_id 와 일치하게 되면 해당학생의 데이터를 출력한다.
- 검색결과가 없으면 "NO SUCH PERSON"를 출력하게 된다.

d) 학점 변경 (changescore_function())

- changescore 명령어를 입력하면 실행되는 함수이다.
- 초깃값으로 found 는 False 로 할당한 다음 s_id 가 일치하는 경우 found = True 로 변경한 다음 예외처리를 한다
- Mid/Final? 을 입력 받은 후, mid 와 final 이 아닌 경우 실행되지 않는다.
- new_score 를 입력 받은 후, 0~100 외의 값이 입력된 경우 실행되지 않는다.
- 전에 있던 학생을 출력한 후, mid 인 경우 중간고사 인덱스를 변경하고 final 인 경우 기말고사 인덱스를 변경하고 평균과 학점을 계산해 출력한다
- 검색 결과가 없으면 "NO SUCH PERSON"를 출력하게 된다.

e) 학생 추가 (add_function())

- add 명령어를 입력하면 실행되는 함수이다.
- 초깃값으로 found 는 False 로 할당한 다음 s_id 가 일치하는 경우 found = True 로 변경한 다음 ALREADY EXISTS 출력한다.
- name, mid, final 을 입력 받은 후 중간고사, 기말고사 점수에 0~100 외의 값이 입력된 경우 실행되지 않는다.
- 평균과 학점 계산 후, slist 에 추가하고 "Student added."를 출력한다.

f) 학점 출력 (searchgrade_function())

- searchgrade 명령어를 입력하면 실행되는 함수이다.
- 학점을 입력 받아 A,B,C,D,F 가 아니면 함수를 종료한다.
- 초깃값으로 found 는 False 로 할당한 다음 학점이 일치하는 경우 found = True 로 변경한 다음 학생정보를 출력한다.
- 검색 결과가 없는 경우, "NO SUCH PERSON." 를 출력한다.

g) 학생 삭제(Remove_function())

- Remove 명령어를 입력하면 실행되는 함수이다.
- 만약 빈 리스트라면, "List is empty." 를 출력한다.
- 학번을 입력받아 s_id 와 일치한 경우 해당 학생을 slist 에서 삭제한다
- 삭제하고 "Student removed" 를 출력한다
- 검색 결과가 없는 경우, "NO SUCH PERSON." 를 출력한다.

h) 프로그램 종료(quit_function())

- quit 명령어를 입력하면 실행되는 함수이다.
- 데이터의 저장유무를 입력 받는다.
- 'yes'를 입력했을 시 저장할 파일 이름을 입력 받고, 학생의 데이터를 평균을 기준으로 내림차순 저장되며 해당기능을 종료하게 된다.
- 'no'를 입력했을 시 해당 프로그램이 명령어 입력 부분으로 가지 않고 바로 종료된다.

4. 프로그램 실행방법 및 예제

1) 초기 실행 및 7개의 명령어 외의 잘못된 명령어 입력 시

```
(base) PS C:\Users\#82108> python project.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

```
# find
#
```

현재 윈도우 환경(노트북)에서 실행했으며, python project.py students.txt 를 입력할 경우, 현재 student.txt 에 있는 학생정보들을 바로 출력합니다

그리고 7개의 명령어 이외의 다른 명령어(find)를 입력 시, 에러 메시지 없이 다시 명령어를 입력 받을 준비를 합니다

2) show

```
# 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 입력시 전체학생 정보를 확인할 수 있습니다.

3) search

```
# 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 후 바로 학번을 입력 받습니다. 하지만 20180050이라는 학번이 존재하지 않아 'NO SUCH PERSON'을 출력했습니다.

이후엔 데이터가 있는 20180002를 제대로 출력한 것을 볼 수 있습니다

4) changescore

```
# 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. 20180007	Kim Cheolsu	75	62	68.5	D

```
#
```

changescore 입력 후 바로 학번을 입력 받습니다. 하지만 20180050이라는 학번이 존재하지 않아 'NO SUCH PERSON'을 출력했습니다.

이후 mid가 잘못 입력되거나, 0~100사이가 아닌 값을 입력 받을 때도 바로 다음 명령어를 받을 준비를 합니다.

이후 제대로 입력했을 경우 제대로 점수수정이 된 것을 볼 수 있습니다.

5) add

```
# 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 후 바로 학번을 입력 받습니다. 하지만 20180001이라는 학번은 데이터가 존재하여 'ALREADY EXISTS'를 출력했습니다.

그리고 결과를 보시면 Lee Hyori 학생과 Lee Sangsun 학생의 중간고사점수, 기말고사점수가 잘 입력된 것을 볼 수 있습니다.

6) searchgrade

```
# searchgrade
Grade to search: e
# searchgrade
Grade to search: f
```

Student	Name	Midterm	Final	Average	Grade
20180007	Kim Cheolsu	57	62	59.5	F

```
# searchgrade
Grade to search: d
```

Student	Name	Midterm	Final	Average	Grade
20180011	Ha Donghun	58	68	63.0	D

```
#
```

Searchgrade 입력 후 바로 grade를 받습니다. 하지만 e는 a,b,c,d,f 외의 값이기 때문에 실행되지 않습니다

그리고 학점들을 입력했을 때, 제대로 입력된 것을 볼 수 있습니다.

또한 현재 students.txt에는 grade가 A,B,C,D,F가 모두 존재하여 'NO RESULTS'출력이 캡처본에는 없지만 아주 잘 작동합니다.

7) remove

```
# remove
Student ID: 20180030
NO SUCH PERSON.

# remove
Student ID: 20180002
Student removed
# remove
Student ID: 20180009
Student removed
# remove
Student ID: 20180001
Student removed
# remove
Student ID: 20180011
Student removed
# remove
Student ID: 20180007
Student removed
# remove
List is empty.
# show
Student          Name          Midterm      Final        Average      Grade
-----
#
```

remove 입력 후 바로 학번을 입력 받습니다. 하지만 20180030이라는 학번이 존재하지 않아 'NO SUCH PERSON'을 출력했습니다.

students.txt 에 존재하는 학생들은 모두 잘 지워지는 것을 볼 수 있습니다.

그리고 모든 학생이 삭제된 경우 'List is empty'라고 나오는 것을 볼 수 있습니다.

8) quit

```
# quit
Save data?[yes/no] yes
File name: newstudents.txt
(base) PS C:\Users\82108>
```

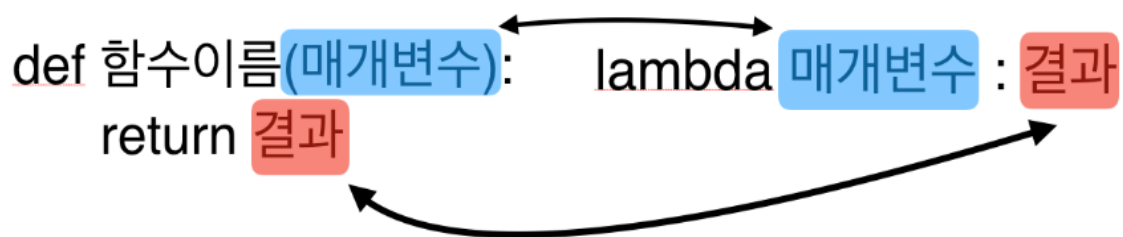
파일 저장이 잘 되는 것을 볼 수 있습니다.

```
# quit
Save data?[yes/no] no
(base) PS C:\Users\82108>
```

프로그램이 종료되는 것을 잘 볼 수 있습니다.

5. 토론

- 처음에 출력을 위한 틀인 start()함수를 만들면서 일일이 tab 을 하나하나 추가해가면서 만든 것이 조금 논리적이진 못했지만, 표가 이쁘게 나와서 만족했습니다.
- 초깃값을 False 로 할당한 다음, 조건문 뒤에 변수를 True 로 변경하는 예외처리를 조원들에게 배워 유용하게 사용했습니다.
- 교수님께서 올려 주신 hint 에서 lambda 함수를 사용해, 기말고사를 기준으로 오름차순 정렬하는 것을 배웠습니다
- 여기서 lambda 함수란 쉽게 말하면 이름이 없는 함수입니다. 람다함수의 장점은 코드의 간결함 메모리의 절약이라고 생각할 수 있습니다. 예를 들어 이렇게 볼 수 있습니다.



- 제가 파일을 읽어올 때 공백을 단위로 분리하기위해, split 함수를 이용하는 것은 알고 있었습니다. 하지만 문장 뒤의 \n 을 없애는 방법을 몰랐습니다. 이 때 이용한 것이 strip()함수입니다.
- strip() 함수는 문자열의 시작과 끝에서 공백을 제거하고 공백 없이 동일한 문자열을 반환합니다.
- 또 프롬프트에서도 출력하기 위해서 sys 라는 모듈을 사용했습니다.
- sys 모듈은 파이썬에서 제공하는 모듈입니다. os 모듈안에 있으며, 파이썬 인터프리터가 제공하는 변수와 함수를 직접 제어할 수 있게 해주는 모듈입니다.
- sys.argv 는 파이썬 스크립트로 넘어온 argument 들의 리스트입니다.

6. 결론

- 본 과제에서 found = False/ found = True 를 사용하는 방법을 익히는데 유용했으며, 예외처리를 쉽게 하는 방법을 알 수 있어 좋았습니다.
- 새로 배운 함수(strip() 함수 등)을 알게 되어 더욱 좋았습니다.
- sys 모듈을 통해 프롬프트를 사용하는 법도 조금 알게 되었습니다.
- 프로그래밍언어의 기본인 조건문, 반복문이 정말 중요함을 다시 한번 느끼는 계기가 되었습니다.
- 만약 나중에 코딩테스트를 준비할 때도 이번 과제를 떠올리며 열심히 공부해 보겠습니다.

7. 개선방안

- changescore 함수나 add 함수에서, 평균이나 학점을 계산하는 부분을 함수로 빼지 않고 모두 안에서 선언했던 부분이 가독성이 떨어지는 것 같습니다.
- 이를 밖에서 함수로 선언했으면 가독성이 더욱 좋았을 것 같습니다.
- 또한 출력을 위한 틀을 만들 때, 저는 일일이 tab 을 쳤지만 다른 조원들은 왼쪽 정렬, 오른쪽 정렬 등을 사용해서 가독성 좋게 만든 것이 개선방안 같습니다.