성적 관리 프로그램

성명 : 김민호(B반)

이메일 : wdg2358@naver.com

1. **문제의 개요**

본 프로그램을 간략히 설명하면 다음과 같다.

* 프로그램의 목적은 학생 네 명의 성적을 관리하기 위함이다.
* 데이터에는 학생들의 학번과 이름, 중간고사 및 기말고사 성적, 평균점수, 학점이 포함되어 있다.
* 학생들의 성적내역 출력, 검색, 학점 검색, 성적 추가 및 삭제, 점수 수정이 가능하도록 한다.

이 문제를 해결하기 위해 다음과 같이 구조차트를 표현해 볼 수 있다.

수정된 성적표

newStudents

1. **알고리즘**

* 학생 성적 처리 프로그램의 알고리즘을 의사 코드로 나타내 보면 다음과 같다.

Stu\_ list

Open students. txt

For I in range

Stu\_list[i][j]

If avg>=90 : A

Elif avg>=80 :B

Elif avg>=70 : C

Elif avg>=60 : D

Else : F

Stu\_list, sort, lambda, reverse.

Def show

Print( student name mideterm final average grade)

Print(---------------------)

For :

Print(i[0] i[1] i[2] i[3] i[4])..

Def search

Input (student ID)

For I in range(5)

Print (---------------------)

Print (i[0] i[1] i[2] i[3] i[4])..)

Def changescore

A = Input studentID

For I in range(5):

If a not in stu\_list

No such person

Else

Mid/final 🡪 mid

Input new score

If score>100 or <0

Print(“”)

Else : print student id, name, mid, final, avg, grade

Mid/final 🡪 final

Input new score

If score>100 or <0

Print(“”)

Else : print student id, name, mid, final, avg, grade

Def add

Liste = []

Input student ID

If id in liste : already exists

Else : append id, name, mid, fin, avg, grade

Def searchgrade

List = [a,b,c,d,f]

A = Input grade to search

List2 = []

For I in range()

List2. append

If A not in list : return

If a in list : print stu\_list[i][0] ~ stu\_list[i][5]

Def remove

Delist = []

Delnum = input(“student id”)

For I in range()

If delnum == stu\_list[i][0]

Append

If m ==1 :

Remove

If len(stu\_list) = 0:

Print(“list is empty”)

Def quit

Input (“Save data? / yes or no”)

If yes : input file name

F = open(file\_name,”w”)

Else :

Return 0

1. **프로그램 구조 및 설명**
2. 학생 점수 리스트를 불러옴

- 프로그램 실행 전 main 함수를 정의하여 학생들의 점수를 불러온다.

- 학생들의 점수에 따라 평균과 학점을 산출하고 2차원 리스트에 저장한다.

2) 함수 정의 : 다음에 해당하는 함수를 각각 정의하여 명령어를 입력하면 실행될 수 있도록 한다.

- show : for문을 사용하여 학생들의 학번, 이름, 중간과 기말 성적, 평균, 학점에 접근하여 출력한다.

- search : input으로 학번을 입력받은 후 임의의 변수 n=0으로 설정한다. for문을 통해서 학생들의 리스트(stu\_list)를 순회한다. 입력받은 학번이 stu\_list에 있다면 출력을 하는 동시에 n에 1을 더해준다. 이를 통해 n이 0이라면 stu\_list에 입력받은 학번을 가지는 학생이 존재하지 않는다는 것이므로 “NO SUCH PERSON”을 출력하게 된다.

- add : 빈 리스트를 생성하고, 학번을 입력한다. for문을 활용하여 stu\_list의 길이만큼 범위를 정하여 순회하고 빈 리스트에 학번을 넣는다. 그 후 if문을 활용하여 입력한 학번이 리스트에 들어있는 학번과 일치하면 ALREADY EXISTS란 메시지를 출력한다. 그렇지 않으면 새로운 이름, 중간기말 성적을 입력받아 새로 입력된 학번과 함께 저장한다.

- searchgrade : 학점은 A,B,C,D,F 5개 중 하나로 결정된다. 따라서 A,B,C,D,F가 들어있는 리스트를 한 개 생성한다. 그 후 찾고자 하는 학점을 입력한다. 만약 A,B,C,D,F 외에 다른 것을 입력하면 공백이 출력되고 다시 입력을 받는다. A,B,C,D,F중 하나를 입력하였지만, 학생 리스트에 해당 학점을 받은 학생이 존재하지 않는다면 No REUSLT가 출력된다. 만약 리스트에 해당 학점을 받은 학생이 있다면, 2차원 리스트의 인덱스를 이용해 접근하여 모두 출력하도록 한다.

- changescore : 학번을 입력받은 뒤, if문을 사용하여 중간/기말점수를 선택하도록 하거나 입력이잘못되었으면 다시 입력하도록 한다. 그 후 다시 if문을 사용하여 점수가 0점에서 100점 사이이면, 점수를 바꿀 수 있도록 한다.

- remove : 삭제할 학생의 리스트를 생성한다. 그 후 삭제할 학생의 학번을 입력한 다음 for문과 if문을 통해 삭제 리스트에 학생을 추가한다.

- quit : yes/no에 따라 파일 이름을 newstudent로 저장한 뒤 f.write를 이용하여 파일을 생성하거나, 저장하지 않고 그냥 나간다.

1. **프로그램 실행방법 및 예제**

스크린샷이(가) 표시된 사진

자동 생성된 설명

* 최초에 프로그램을 실행했을 때의 화면
* Terminal에서 python project1.py를 입력하면 성적관리 프로그램이 실행된다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

* Show 명령어를 실행하면 다음과 같이 학생들의 성적 정보가 출력된다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

* Search 명령어를 실행하면 다음과 같이 학번이 입력된 학생의 성적을 볼 수 있다. 만약 입력한 학번에 해당하는 학생이 없다면 “NO SUCH PERSON”이라는 메시지가 뜬다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

* Changescore 명령어를 입력하면 점수를 바꾸고자 하는 학생의 학번을 입력받아 중간 또는 기말 점수를 바꿀 수 있다. 만약 입력한 학번에 해당하는 학생이 존재하지 않는다면 “NO SUCH PERSON.”이라는 메시지가 뜨게 된다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

* Add 명령어를 실행할 경우 학생들의 학번과 이름, 중간 및 기말 성적을 입력할 수 있으며, 평균과 학점까지 산출할 수 있다. 이미 기재된 학생의 학번을 입력할 경우 “ALREADY EXISTS.”란 메시지가 뜬다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

* Searchgrade 명령어를 입력할 경우 입력한 학점에 해당하는 학생들의 이름과 중간기말 성적 및 평균을 확인할 수 있다.
* A,B,C,D,F 외의 알파벳을 입력할 경우 다시 명령어를 입력해야 한다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

* Remove 명령어를 실행할 경우 입력한 학번에 해당하는 학생의 성적 정보가 삭제된다. 만약 입력한 학번에 해당하는 학생 정보가 없을 경우에는 “NO SUCH PERSON.”이라는 메시지가 뜬다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

* Quit 명령어를 실행할 경우 파일을 저장하고 종료할 것인지, 혹은 그냥 종료할 것인지를 선택하게 된다. 파일을 저장하고 종료할 경우 파일명을 입력해준다.

1. **토론**

* Lambda 함수의 사용법 : 함수를 간단하게 사용할 수 있도록 해주는 함수. 특히 map 함수와 함께 사용되는 경우가 많다.

예시)

>>> def add(a,b):

return a+b

>>> add(1,2)

3

>>> (lambda a,b : a+b) (1,2)

3

이번 과제에서는 “stu\_list.sort(key = lambda e : e[4], reverse = True) “ 와 같이 리스트에서 평균점수를 기준으로 내림차순 정렬을 하기 위해 lambda함수를 사용하였다.

1. **결론**

* 이 과제를 수행하기 위해서 함수의 정의, for문과 if문, 파일 입출력, 1차원 또는 2차원 리스트의 생성과 접근, 수정 등 다양한 기능들을 활용하였다. 이러한 기능을 활용하여 파이썬의 기본적인 활용 능력을 익힐 수 있었다. 또한 파이썬의 기능을 어떠한 방식으로 실생활에 응용하여 문제를 해결할 수 있을지 고민해 보는 계기가 되었다.

1. **개선 방향**

* 리스트 및 for문을 활용한 알고리즘을 구상할 때 다소 어려움을 겪었다. 반복문, 제어문, 자료형의 활용은 파이썬의 기본적이면서도 널리 활용되는 요소인데, 이와 같은 프로그래밍 언어의 기본기를 꾸준히 개발해야 할 필요성을 느꼈다.