2022년도 청년 AI·Big Data 아카데미 19기

파이썬 기초 – 프로젝트

Assignment #1

담당교수 윤은영

반 A반

이름 이무동

Project Name : 성적 처리 프로그램

1. 문제의 개요

파일로부터 데이터를 읽어서(students.txt 성적 목록을 만들어 관리하는 성적 관리 프로그램을 작성한다. 이 프로그램은 사용자로부터 7개의 명령어 (show, search, changescore, searchgrade, add, remove, quit)를 입력 받아 각 기능을 수행 하게 된다. 최소한 각 명령어 별로 함수를 정의하여 사용한다. 명령어 외에 필요한 함수는 추가로 정의하여 사용할 수 있다.

● show (전체 학생 정보 출력) : show 입력 시, 저장되어 있는 전체 목록을 아래와 같이 평균 점수를 기준으로 내림차순으로 출력한다. 평균 점수는 소수점 이하 첫째 자리까지만 표시한다.

● search (특정 학생 검색) : search 입력 시, 아래와 같이 검색하고자 하는 학생의 학번을 요구해 입력 받아 학번, 이름, 중간고사 점수, 기말고사 점수, 평균, 학점을 출력한다.

● changescore (점수 수정) : 목록에 저장된 학생 중 1명의 중간고사(mid) 혹은 기말고사(final)의 점수를 수정한다. changescore 입력 시, 수정하고자 하는 학생의 학번, 수정하고자 하는 점수가 중간고사인지 기말고사인지와 수정하고자 하는 점수를 순서대로 입력 받아 해당 학생의 점수를 수정한다. 점수가 바뀜에 따라 Grade도 다시 계산하여 수정한다.

● add (학생 추가) : add 입력 시, 아래와 같이 학생의 학번, 이름, 중간고사 점수, 기말고사 점수를 차례로 요구해 입력 받는다. 추가되면, 메시지 'Student added'를 아래 예제와 같이 출력한다. Average와 Grade는 중간고사 점수와 기말고사 점수를 사용하여 계산하여 저장한다. 학생 추가 후 show 명령어를 사용하면 평균을 기준으로 내림차순으로 출력한다.

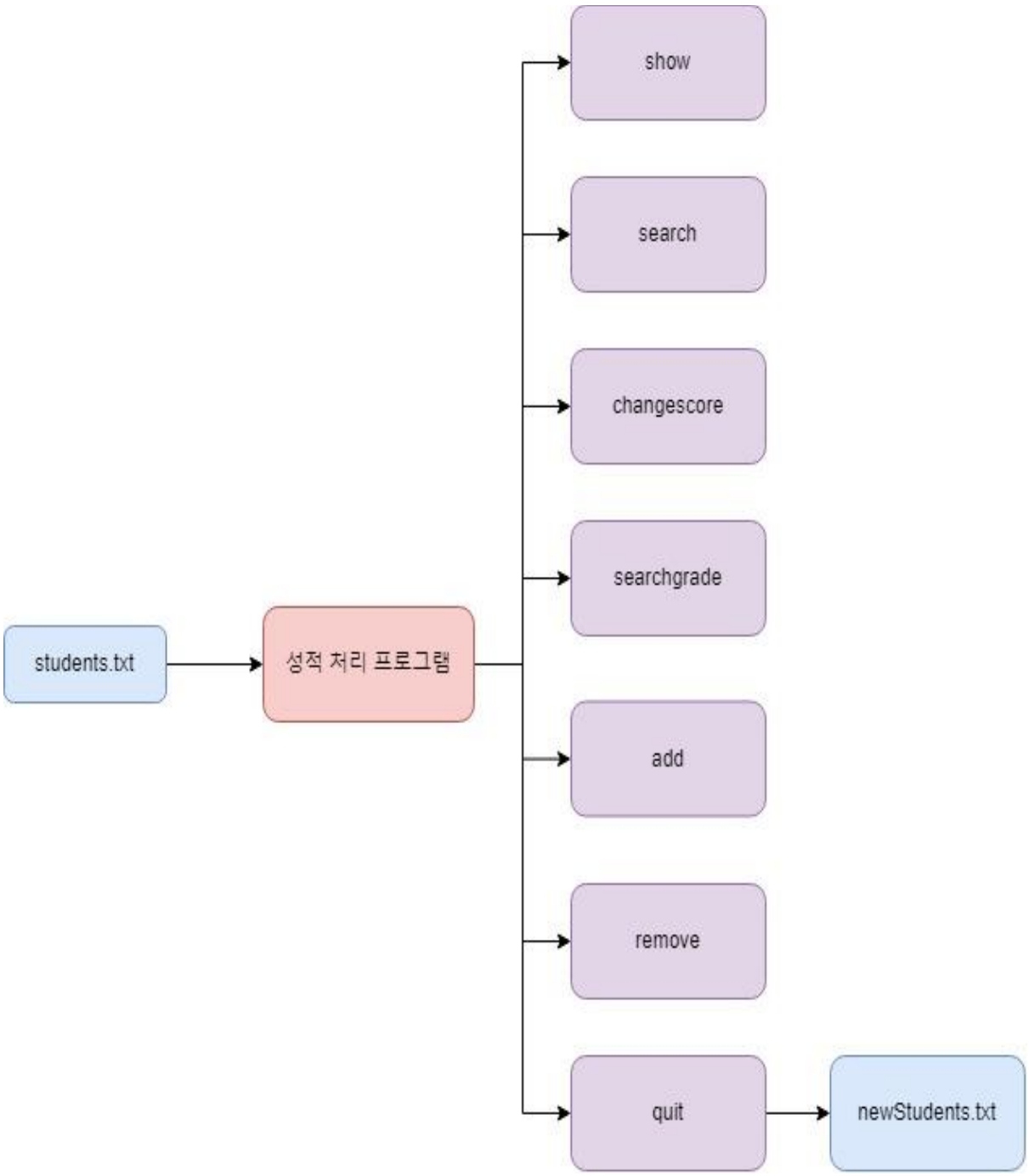
● searchgrade (Grade 검색) : searchgrade 입력 시, 특정 grade를 입력 받아 그 grade에 해당하는 학생을 모두 출력한다.

● remove(특정 학생 삭제) : remove 입력 시, 아래와 같이 삭제하고자 하는 학생의 학번을 입력 받은 후, 학생이 목록에 있는 경우 삭제한다. 삭제하면, 메시지 'Student removed'를 아래와 같이 출력한다.

● quit(종료) : quit 입력 시, 프로그램을 종료한다. 해당 명령어를 실행할 경우, 현재까지 편집한 내용의 저장 여부를 묻고, 저장을 선택 (yes 입력)할 경우 파일명을 입력 받아서

저장하도록 한다. 앞서 본 'student.txt'와 같이 내용을 구성한다. 저장할 때 목록의 순서는 평균을 기준으로 내림차순으로 한다. 파일 이름에는 공백이 없다고 가정한다.

2. 알고리즘 Flow Chart



3) 프로그램 실행방법 및 출력 예시

0) 메인함수

```python
# coding: utf-8
import os.path

# 메인함수 (show(), command 받기)
def main():
    # 평균과 학점계산해서 scoredict에 저장
    command = input("# ").lower()

    if command == 'show':
        show(scoredict)
    elif command == 'search':
        search(scoredict)
    elif command == 'changescore':
        changescore(scoredict)
    elif command == 'add':
        rawscoredict = add(scoredict)
        return rawscoredict
    elif command == 'searchgrade':
        searchgrade(scoredict)
    elif command == 'remove':
        rawscoredict = remove(scoredict)
        return rawscoredict
    elif command == 'quit':
        quit(scoredict)
```

```python
# 부가적인 함수

def setting():
    # student 표의 머릿말 부분
    print(index)
    print(dot)
def stu(id,scoredict):
    # student id와 딕셔너리를 넣으면 한줄로 values를 예쁘게 뽑아줌
    id = str(id)
    # print(id, scoredict[id][0], scoredict[id][1],scoredict[id][2],scoredict[id][3],scoredict[id][4], sep='\t')
    print("%s %18s %10s %10s %10s %10s"%(id, scoredict[id][0], scoredict[id][1],scoredict[id][2],scoredict[id][3],scoredict[id][4]))

def dictsave(rawscoredict):
    # 평균과 학점을 딕셔너리로 저장
    scoredict = {}
    for stuID in rawscoredict:
        li = rawscoredict[stuID][:]
        avg = (float(li[1])+float(li[2]))/2
        if avg >= 90:
            grade = 'A'
        elif avg >= 80:
            grade = 'B'
        elif avg >= 70:
            grade = 'C'
        elif avg >= 60:
            grade = 'D'
        else:
            grade = 'F'
        li.append(avg) ; li.append(grade)
        scoredict[stuID] = list(map(str, li))
    return scoredict
```

```
# 프로그램시작(파일열기, 파일을 간단한 딕셔너리로 저장, 필요한 변수 선언)
if __name__ == '__main__':
    # 라이브러리 import
    import sys
    # 파일열기
    try:
        file = open(sys.argv[1], 'r')
    except:
        file = open("students.txt", 'r')

    # rawscoredict에 저장
    rawscoredict = {}
    for line in file:
        li = []
        li = line.strip().split('\t')
        id = li.pop(0)
        rawscoredict[id] = li
    # 평균과 학점계산해서 scoredict에 저장
    scoredict = dictsave(rawscoredict)
    # 변수선언
    index = ' Student            Name       Midterm     Final     Average      Grade'
    dot = '-------------------------------------------------------------------------'
    # show,main 함수 호출
    show(scoredict)
    while True:
        main()
        scoredict = dictsave(rawscoredict)
```

1) show (전체 학생 정보 출력)

```
# show (전체 학생 정보 출력)
def show(scoredict):
    setting()
    sortscoredict = sorted(scoredict.items(), key=lambda x: x[1][3], reverse=True)
    for i in sortscoredict:
        stu(i[0], scoredict)
```

```
# 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
```

2) search (특정 학생 검색)

```
# search (특정 학생 검색)
def search(scoredict):
    id = input('Student ID: ')
    if id in scoredict:
        setting()
        stu(id,scoredict)
    else:
        print("NO SUCH PERSON")
```

```
# 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
```

3) changescore (점수 수정)

```
# changescore (점수 수정)
def changescore(scoredict):
    id = input("Student ID: ")
    if id in scoredict:
        term = input("Mid/Final?").lower()
        if term == 'mid':
            newScore = int(input("Input new score: "))
            if 0<= newScore <=100 :
                setting()
                stu(id,scoredict)
                rawscoredict[id][1] = newScore
                scoredict = dictsave(rawscoredict)
                print('Score changed.')
                stu(id, scoredict)
                return rawscoredict
        elif term == 'final':
            newScore = int(input("Input new score: "))
            if 0 <= newScore <= 100 :
                setting()
                stu(id,scoredict)
                rawscoredict[id][2] = newScore
                scoredict = dictsave(rawscoredict)
                print('Score changed.')
                stu(id, scoredict)
                return rawscoredict
    else:
        print("NO SUCH PERSON")
```

```
# 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
```

4) add (학생 추가)

```python
# add (학생 추가)
def add(scoredict):
    stuId = input("Student ID: ")
    if stuId in rawscoredict:
        print("ALREADY EXIST")
        main()
    else :
        name = input("Name: ")
        mid = input("Midterm Score: ")
        final = input("Final Score: ")
        rawscoredict[stuId] = [name, mid, final]
        print("Student added.")
    return rawscoredict
```

```
# add
Student ID: 20180001
ALREADY EXIST
# 20180021
# 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.
```

5) searchgrade (Grade 검색)

```
# searchgrade (Grade 검색)
def searchgrade(scoredict):
    grade = input("Grade to search: ")
    if grade in ('A','B','C','D','F'):
        count = 0
        li = []
        for i in scoredict:
            if scoredict[i][-1] == grade:
                li.append(i)
                count += 1
        if count == 0:
            print("NO REUSLTS")
        else:
            setting()
            for i in li:
                stu(i,scoredict)
```

```
# searchgrade
Grade to search: E
# searchgrade
Grade to search: F
NO REUSLTS
# 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

6) remove (특정 학생 삭제)

```
# Remove (특정 학생 삭제)
def remove(scoredict):
    if scoredict == {}:
        print("List is empty.")
    else:
        id = input("Student ID: ")
        if id not in scoredict:
            print("NO SUCH PERSON.")
        else:
            del rawscoredict[id]
            print("Student removed.")
    return rawscoredict
```

```
# 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
```

7) quit (종료)

```python
# quit (종료)
def quit(scoredict):
    yesno = input('Save data?[yes/no]').lower()
    if yesno == 'yes':
        filename = input('File name : ')
        newfile = open(filename, 'w')
        for id in scoredict:
            line = id + '\t' + scoredict[id][0] + '\t' + scoredict[id][1] + '\t' + scoredict[id][2] +'\n'
            newfile.write(line)
        newfile.close()
    file.close()
    sys.exit()
```

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

4) 토론

람다란, 프로그래밍 언어에서 사용되는 개념으로 익명의 함수, 이름 없는 함수를 지칭하는 용어이다. 익명함수란 말 그대로 함수의 이름이 없는 함수다. 보통 파이썬에서 def 키워드를 사용하여 함수를 선언하고 기능을 정의 하는 것과는 달리, 람다는 함수를 하나의 식으로 정의한다. 파이썬의 람다 기능은 익명 함수(Anonymous), 즉 이름이 없는 함수를 정의하기 위한 용도로 사용된다. 참고로 컴퓨터 분야에서 정의는 없던 것에 대한 구체적인 생성을 의미하며, 선언은 일단 이름만 붙여두고 구체적인 생성은 다른 곳에서 대신하는 것을 의미한다. 예를 들어, 인자를 두개 받아 받은 인자값을 합해 반환하는 람다 함수는 **'func = lambda a,b : a+b'** 로 표현한다.

5) 결론

파이썬 기초 프로젝트를 통하여 'Posco AI/Big Data 아카데미' 1주차 Python 교육 과정동안 익힌 Python 내용을 복습할 수 있었고, 실제로 구동되는 프로그램을 만들어보는 값진 경험을 할 수 있게 되었다. 앞으로 AI 및 Big Data 관련 분야의 직종에서 다양한 Project를 진행할 수 있다는 자신감을 얻게 되었다.