

성적관리 프로그램 Assignment

담당교수: 윤은영

소속: C 반 4 조

이름: 김민정

이메일: miinn81@gmail.com

Problem: 파일로부터 데이터를 읽어서 성적 목록을 만들어 관리하는 성적 관리 프로그램을 작성한다.

1. 본 프로그램의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- 학번, 이름, 중간고사 성적, 기말고사 성적 데이터가 저장된 파일을 입력 받는다.
- 입력 받은 데이터에 의거하여 평균과 학점을 계산하여 기존 데이터에 추가하여 저장한다.

* 평균(Average)항목은 중간고사 점수와 기말고사 점수의 평균을 계산하여 저장한다.

* 학점(Grade)항목의 기준은 아래와 같다.

- A: 평균이 90 점 이상
- B: 평균이 80 점 이상, 90 점 미만
- C: 평균이 70 점 이상, 80 점 미만
- D: 평균이 60 점 이상, 70 점 미만
- F: 평균이 60 점 미만

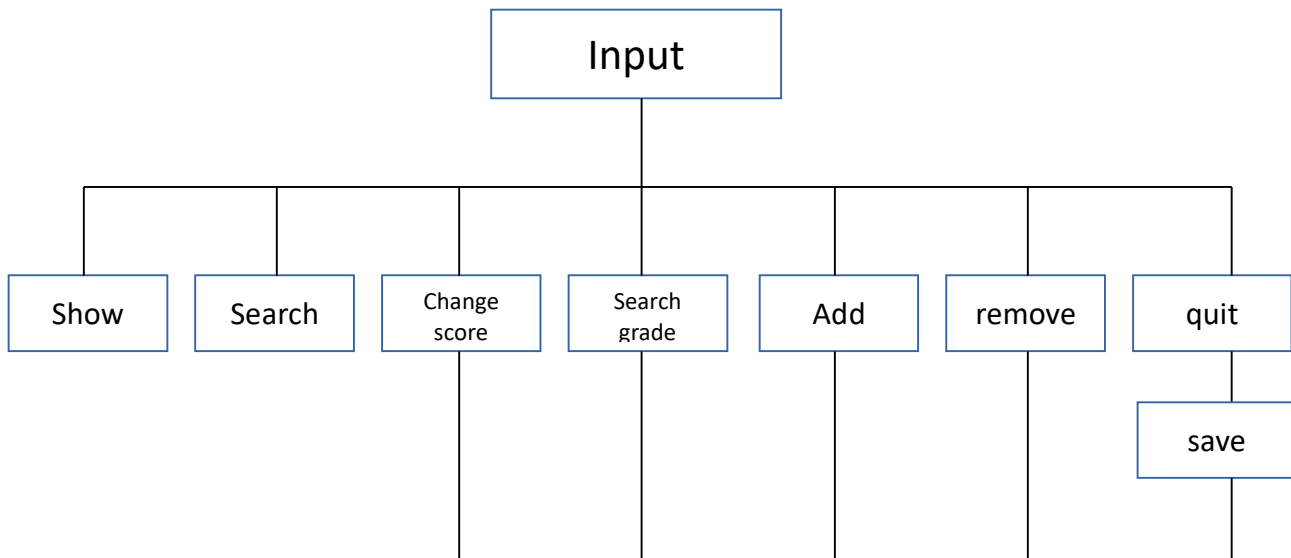
- 사용자로부터 7 개의 명령어(show, search, changescore, searchgrade, add, remove, quit)를 입력 받아 해당 명령어에 따른 함수를 실행한다.

- 입력 받은 명령어를 통해 산출된 결과 값은 기존 데이터에 업데이트한다.

- 업데이트 된 데이터는 사용자가 원한다면 새로운 파일에 출력하여 저장할 수 있다.

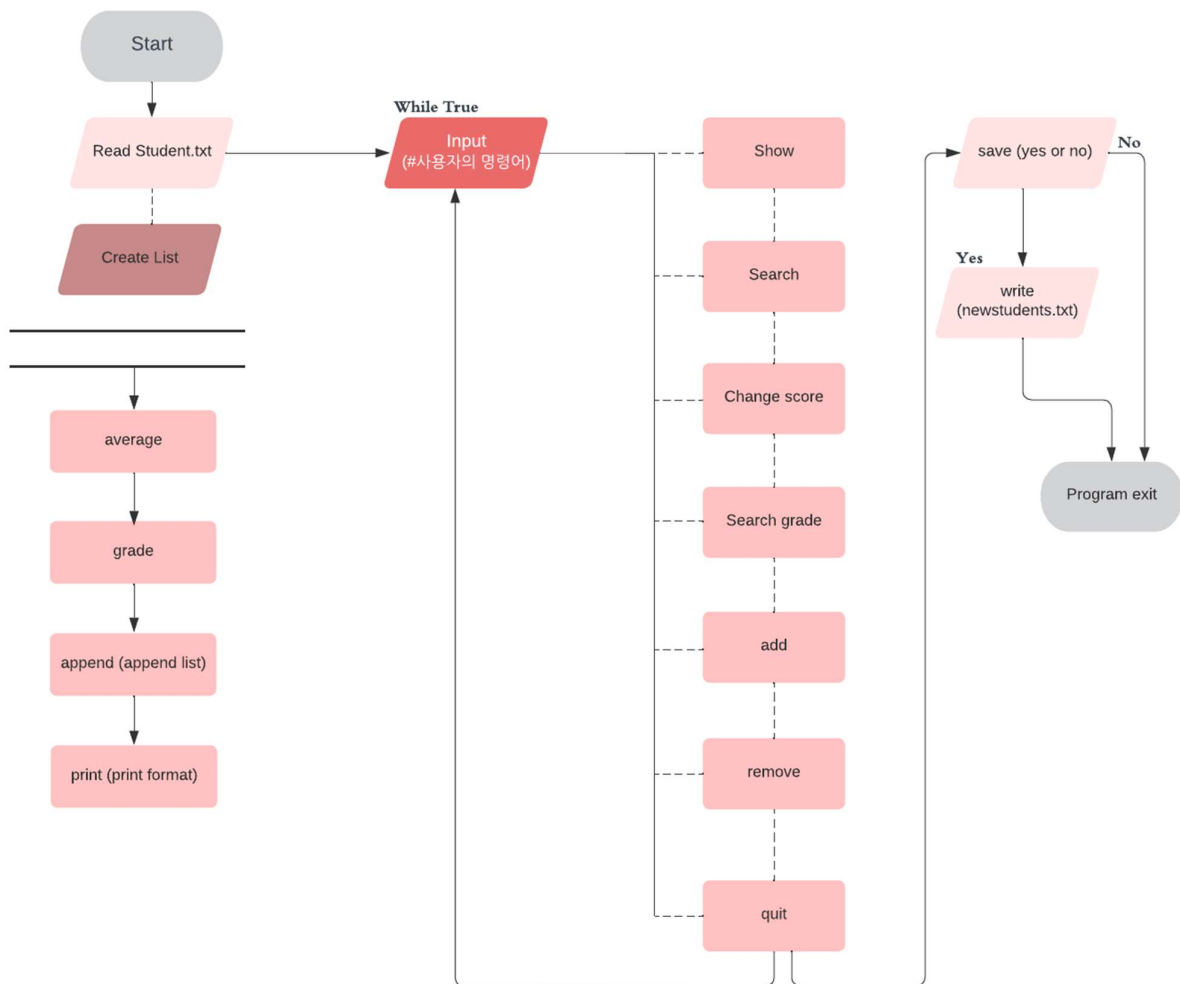
이 때 사용되는 구상 가능한 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.

- 입력부 : 사용자가 원하는 명령어를 입력한다.
- 처리부 : 해당 명령어를 받아 각 기능을 수행한다.
- 출력부 : 계산이 완료된 결과를 print 함수에 맞추어 출력하고, 업데이트된 데이터는 사용자의 의중에 따라 저장한다.



2. 알고리즘

본 프로그램 작성을 위한 알고리즘을 Flow chart를 통해 표현하면 아래와 같다.



3. 프로그램 구조 및 설명

- 기능을 수행할 명령어 입력

- 프로그램을 실행시키면 텍스트 파일로부터 데이터를 읽고 평균과 학점을 추가하여 목록을 리스트(list) 자료형으로 저장한다.

- 사용자로부터 명령어를 입력 받으면 입력된 명령어와 연결된 함수가 실행되며 기능 수행을 위한 input 값을 사용자에게 요청한다.

- 7 개의 기능 별 함수

- show (): 전체 학생 정보를 출력하는 기능으로 show 입력 시, 저장되어 있는 전체 목록을 평균 점수를 기준으로 내림차순으로 출력한다. 평균 점수는 소수점 이하 첫째 자리까지만 표시한다.

- search (): 특정 학생을 검색하는 기능으로 검색하고자 하는 학생의 학번을 요구해 입력 받아 학번, 이름, 중간고사 점수, 기말고사 점수, 평균, 학점을 출력한다.

- changescore (): 점수를 수정하는 기능으로 목록에 저장된 학생 중 1 명의 중간고사 혹은 기말고사의 점수를 수정한다. 점수가 바뀔 때 따라 Grade 도 다시 계산하여 수정한다.

- add (): 학생을 추가하는 기능으로 학생의 학번, 이름, 중간고사 점수, 기말고사 점수를 차례로 입력 받는다. Average 와 Grade 는 중간고사 점수와 기말고사 점수를 사용하여 계산하여 저장한다. 학생 추가 후 show 명령어를 사용하면 평균을 기준으로 내림차순으로 출력한다.

- searchgrade(): Grade 를 검색하는 기능으로 특정 grade 를 입력 받아 그 grade 에 해당하는 학생을 모두 출력한다.

- remove(): 특정 학생을 삭제하는 기능으로 삭제하고자 하는 학생의 학번을 입력 받아 해당 학생을 삭제한다.

- quit: 프로그램을 종료하는 기능으로 해당 명령어를 실행할 경우, 현재까지 편집한 내용의 저장 여부를 묻고, 저장을 선택(yes 입력)할 경우 파일명을 입력 받아서 저장하도록 한다.

- 추가 함수

- avg (): 중간고사 점수와 기말고사 점수를 입력 받아 평균 점수를 출력한다.

- grade (): 평균 점수를 기반으로 학점을 출력한다.

- append (): 리스트에 새로 추가하고 싶은 요소가 있을 때 사용하여 추가한다.

- 새로 지정한 파일에 성적 출력

- 출력 파일에는 프로그램 실행 시 출력 되었던 구성과 동일하게 평균과 학점정보는 제외하여 저장한다.

- 출력 파일에는 마지막으로 편집된 내용의 데이터 중 평균을 기준으로 내림차순 정렬되어 보여진다.

4. 프로그램 실행방법 및 예제

#실행

```
(base) piat@piat-Precision-7920-Tower:~$ ls
Anaconda3-2022.05-Linux-x86_64.sh  공개      '알고리즘 스터디'
Untitled.ipynb                     다운로드 음악
anaconda3                          문서      템플릿
ddd.pptx                           바탕화면 파이썬
examples.desktop                   비디오   함수모음.ipynb
snap                               빅데이터
test1                              사진
(base) piat@piat-Precision-7920-Tower:~$ cd 다운로드
(base) piat@piat-Precision-7920-Tower:~/다운로드$ python project.py
#
```

- 리눅스 환경에서 project.py 파일이 있는 위치를 찾아 실행시킨다.

#다른 명령어 입력

```
# find
#
```

- 7 개의 명령어 이외의 다른 명령어 입력 시 별도 에러 메시지 없이 아래와 같이 다시 명령어를 입력 받을 준비를 한다.

#show

```
(base) piat@piat-Precision-7920-Tower:~/다운로드$ python project.py
# 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 입력 시 저장되어 있는 전체 목록을 아래와 같이 평균 점수를 기준으로 내림차순으로 출력한다.
평균 점수는 소수점 이하 첫째 자리까지만 표시한다.

#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 시, 아래와 같이 검색하고자 하는 학생의 학번을 요구해 입력 받아 학번, 이름, 중간고사 점수, 기말고사 점수, 평균, 학점을 출력한다.

- 찾고자 하는 학생이 목록에 없는 경우에는 "NO SUCH PERSON." 이라는 에러 메시지를 출력한다.

```
# changescore
Student ID: 20180050
NO SUCH PERSON.
# changescore
Student ID: 20180007
Mid/Final: mid
# changescore
Student ID: 20180007
Mid/Final: mid
new score : 147
# changescore
Student ID: 20180007
Mid/Final: mid
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 입력 시, 수정하고자 하는 학생의 학번, 수정하고자 하는 점수가 중간고사인지 기말고사인지 확인한다.
- 이후 수정하고자 하는 점수를 순서대로 입력 받아 해당 학생의 점수를 수정한다.
- 점수가 바뀔때 따라 Grade도 다시 계산하여 수정한다.
- 학번이 목록에 없는 경우에는 "NO SUCH PERSON."이라는 에러 메시지를 출력한다.
- "mid" 또는 "final" 외의 값이 입력된 경우에는 실행되지 않는다.
- 점수에 0~100 외의 값이 입력된 경우에는 실행되지 않는다.

#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 시, 아래와 같이 학생의 학번, 이름, 중간고사 점수, 기말고사 점수를 차례로 요구해 입력 받는다. 추가되면, 메시지 "Student added"를 출력한다.
- Average와 Grade는 중간고사 점수와 기말고사 점수를 사용하여 계산하여 저장한다.
- 학생 추가 후 show 명령어를 사용하면 평균을 기준으로 내림차순으로 출력한다.
- 목록에 있는 학생의 학번을 입력 시, 'ALREADY EXISTS.' 이라는 에러 메시지 출력한다.

#searchgrade

```
# searchgrade
Grade to search: E
# searchgrade
Grade to search: F
No Result
# search grade
wrong input!
# D
wrong input!
# 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
```

- searchgrade 입력 시, 특정 grade를 입력 받아 그 grade에 해당하는 학생을 모두 출력한다.
- A, B, C, D, F 외의 값이 입력된 경우 실행되지 않는다.
- 해당 grade 의 학생이 없는 경우 아래와 같이 메시지 "NO RESULTS." 출력한다.

#remove

```
# 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
```

- remove 입력 시, 아래와 같이 삭제하고자 하는 학생의 학번을 입력 받은 후, 학생이 목록에 있는 경우 삭제한다. 삭제하면, 메시지 "Student removed."라는 문구를 출력한다.
- 학생이 목록에 없는 경우에는 "NO SUCH PERSON."이라는 에러 메시지를 출력한다.

```
# remove
List is empty.
```

- 목록에 아무도 없을 경우 "List is empty"라는 메시지를 출력한다.

```
# quit
Save data?[yes/no]: no
(base) pi.ai@pi.ai-Precision-7920-Tower:~/다운로드$
```

```
# quit
Save data?[yes/no]: yes
File name: newStudents.txt
(base) pi.ai@pi.ai-Precision-7920-Tower:~/다운로드$
```

- quit 입력 시, 프로그램을 종료한다.
- 해당 명령어를 실행할 경우, 현재까지 편집한 내용의 저장 여부를 묻고, 저장을 선택(yes입력)할 경우 파일명을 입력 받아서 저장하도록 한다.
- 저장할 때 목록의 순서는 평균을 기준으로 내림차순으로 한다.

#newstudents.txt

20180021	Lee Hyori	93	95
20180002	Lee Jieun	92	89
20180009	Lee Yeonghee	81	84
20180001	Hong Gildong	84	73
20180006	Lee Sangsun	77	66
20180007	Kim Cheolsu	75	62

5. 토론

- 파일을 읽어서 평균과 학점을 계산하여 추가하는 작업에서 이중 리스트로 구현할 때 일반 리스트와 혼동했다. 이러한 문제로 append 진행 시 추가가 되지 않는 예러가 있었다.

- search 함수 생성 시 입력된 ID가 stu_list에 없을 때 "No Such person"이라는 문구를 출력할 때, if-else 문에 대한 이해가 부족하여 구현이 어려웠다. 그러나, k라는 변수를 두어 stu_list 안에 있는 요소들을 하나씩 확인할 때 마다 k라는 변수가 1 씩 더해지는 코드로 구현할 수 있었다.

- changescore 함수 생성 시 100 보다 큰 점수가 입력 되었을때 어떻게 함수를 실행시키지 않을 수 있을까 고민하다가 pass 입력으로 원하는 때에만 함수를 실행시킬 수 있음을 알게 되었다.

- 추가로 리스트에 추가 요소를 append 할 수 있는 함수를 구현할 때 가변 매개변수 사용에 대해서 알게 되었다. 처음에는 튜플로 출력이 되어 이 부분에 고민한 결과, list화하는 작업을 추가로 진행하여 해당 튜플의 원하는 인덱스 값을 출력하는 방법으로 구현할 수 있었다.

6. 결론

- 해당 과제를 수행하며 전반적으로 어떻게 프로그램을 생성하는지에 대해 학습할 수 있었다. 이 과정에서 파일 입/출력에 대해 완벽히 이해할 수 있었다.
- 또한 7 개 이상의 함수를 정의해보며, 각 명령어에서 요구하는 기능을 구현하는 경험을 할 수 있었다. 특히, 자주 사용되는 코드에 대해서는 함수(모듈화)를 통하여 효율적인 프로그래밍을 할 수 있음을 배웠다.
- 이번 과정을 통해 순차적으로 코드를 구현 하는법과 다양한 파이썬 기능들을 익힐 수 있는 기회였다.