# 성적 관리 프로그램

**A4 - 노 태 윤**

이메일 : nty0725@naver.com

**2022년 8월 5일**

1. 문제의 개요
파일로부터 데이터를 읽어서 성적 목록을 만들어 관리하는 성적 관리 프로그램을 작성한다.

2. 의사코드(pseudo code)

2-1 main
(1) 리스트 생성
(2) 딕셔너리 생성
(3) 공백의 수를 저장하는 리스트 생성
(3) 파일 읽고 딕셔너리에 key: value 형태로 저장
(여기서 key는 학번이고 나머지는 value에 리스트 형태로 저장)
(4) 평균함수 및 등급 함수 호출
(중간점수와 기말 점수를 이용해 평균을 구하고 딕셔너리에 저장, 그 후 그 평균을 이용해 성적 등급을 구한후 딕셔너리에 저장)
(5) 입력창에 show를 입력하면 show함수 호출
(6) 입력창에 search를 입력하면 search함수 호출
(7) 입력창에 changescore를 입력하면 changescore함수 호출
(7) 입력창에 add를 입력하면 add함수 호출
(8) 입력창에 remove를 입력하면 remove함수 호출
(9) 입력창에 searchgrade를 입력하면 searchgrade함수 호출
(10) 입력창에 remove를 입력하면 remove함수 호출
(11) 입력창에 quit를 입력하면 quit함수 호출

2-2 show()
(1) 평균과 등급함수 실행
(2) 평균을 기준으로 내림차순
(3) 출력

2-3 search()
(1) 검색하고 싶은 학생 아이디를 입력받음
(2) 검색후 있으면 출력 없으면 'no such person'출력

2-4 changescore()
(1) 점수를 바꾸고 싶은 학생 아이디를 입력받음
(2) 조회결과가 없다면 'no such person; 출력
(3) 있다면 바꾸고 싶은 점수가 중간인지 기말인지 입력받고 점수도 입력받음
(4) 중간을 입력받으면 중간 점수 바꿈, 기말을 입력받으면 기말 점수를 바꿈

2-5 add()
(1) 학생 아이디를 입력받는데, 이미 있는 학생 아이디이면 'already exists.' 출력
(2) 없다면, 이름, 중간점수, 기말점수를 입력받고 딕셔너리에 저장

2-6 searchgrade()
(1) 등급을 입력받음
(2) 만약 입력받은 등급이 A, B, C, D, F 중에 없다면 종료
(3) 있다면 그 등급에 해당하는 학생의 정보 출력
(4) 등급은 있는데 조회결과가 없다면, NO RESULTS. 출력

2-7 remove()
(1) 딕셔너리가 비어있으면 'list is empty' 출력하고 return
(2) 학생 아이디를 입력 받음
(3) 만약 딕셔너리에 입력 받은 학생 아이디가 없다면 'no such person' 출력
(4) 있다면 딕셔너리에서 제거 후 'student removed' 출력

2-8 quit()
(1) 파일 이름 입력 받음
(2) 파일 열기
(3) 중간고사 점수로 정렬
(4) 파일에 쓰기
(5) 파일 닫기

3. 프로그램 실행 예제:
(1) show(전체 학생 정보 출력)
show 입력 시, 저장되어 있는 전체 목록을 아래와 같이 평균 점수를 기준으로 내림차순으로 출력합니다. 평균 점수는 소수점 이하 첫째 자리까지만 표시합니다.

```
# 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 입력 시, 아래와 같이 검색하고자 하는 학생의 학번을 요구해 입력 받아 학번, 이름, 중간고사 점수, 기말고사 점수, 평균, 학점을 출력합니다.
찾고자 하는 학생이 목록에 없는 경우에는 "NO SUCH PERSON." 이라는 에러 메시지를 출력합니다.

```
# 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)

목록에 저장된 학생 중 1명의 중간고사(mid) 혹은 기말고사(final)의 점수를 수정합니다. changescore 입력 시, 수정하고자 하는 학생의 학번, 수정하고자 하는 점수가 중간고사인지 기말고사인지와 수정하고자 하는 점수를 순서대로 입력 받아 해당 학생의 점수를 수정합니다.

점수가 바뀜에 따라 Grade도 다시 계산하여 수정한다.

학번이 목록에 없는 경우에는 "NO SUCH PERSON."이라는 에러 메시지를 출력합니다. "mid" 또는 "final" 외의 값이 입력된 경우에는 실행되지 않습니다. 점수에 0~100 외의 값이 입력된 경우에는 실행되지 않습니다.

```
# changescore
Student ID: 20180050
NO SUCH PERSON.
# changescore
Student ID: 20180007
Mid/Final? miid
# 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)

add 입력 시, 아래와 같이 학생의 학번, 이름, 중간고사 점수, 기말고사 점수를 차례로 요구해 입력 받습니다. 추가되면, 메시지 "Student added."를 아래 예제와 같이 출력합니다.

Average와 Grade는 중간고사 점수와 기말고사 점수를 사용하여 계산하여 저장합니다.

학생 추가 후 show 명령어를 사용하면 평균을 기준으로 내림차순으로 출력된다.

목록에 있는 학생의 학번을 입력 시, 'ALREADY EXISTS.' 이라는 에러 메시지 출력 합니다.

```
# 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
```

(5) searchgrade (Grade 검색)

searchgrade입력 시, 특정 grade를 입력 받아 그 grade에 해당하는 학생을 모두 출력합니다.

A, B, C, D, F 외의 값이 입력된 경우 실행되지 않습니다. 해당 grade 의 학생이 없는 경우 아래와 같이 메시지 "NO RESULTS." 출력합니다.

```
# searchgrade
Grade to search: E
# 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
```

(6) REMOVE (특정 학생 삭제)

remove 입력 시, 아래와 같이 삭제하고자 하는 학생의 학번을 입력 받은 후, 학생이

목록에 있는 경우 삭제합니다. 삭제하면, 메시지 "Student removed."를 아래와 같이 출력합니다.

목록에 아무도 없을 경우 아래의 예제와 같이 "List is empty." 메시지를 출력합니다.

```
# remove
Student ID: 20180002
Student removed.
# remove
Student ID: 20180007
Student removed.
# remove
Student ID: 20180001
Student removed.
# show
   Student             Name    Midterm     Final       Average     Grade
------------------------------------------------------------------------
  20180009     Lee Yeonghee       81         84          82.5         B
  20180011       Ha Donghun       58         68          63.0         D
# remove
Student ID: 20180011
Student removed.
# remove
Student ID: 20180009
Student removed.
# remove
List is empty
```

학생이 목록에 없는 경우에는 "NO SUCH PERSON."이라는 에러 메시지를 출력합니다.

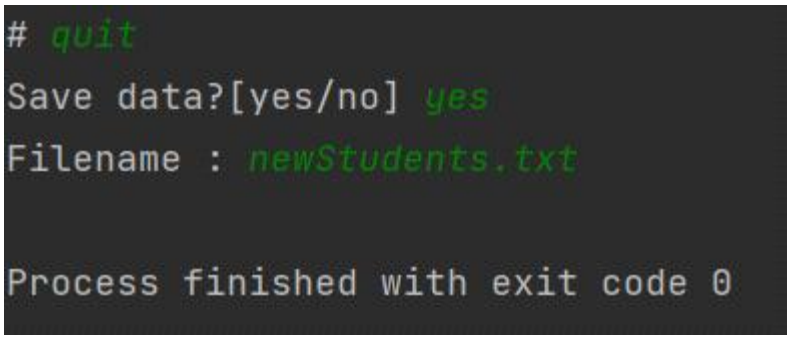
```
# remove
Student ID: 29189939
NO SUCH PERSON.
# remove
Student ID: 20180011
Student removed.
# show
   Student             Name    Midterm     Final       Average     Grade
------------------------------------------------------------------------
  20180021        Lee Hyori       93         95          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)

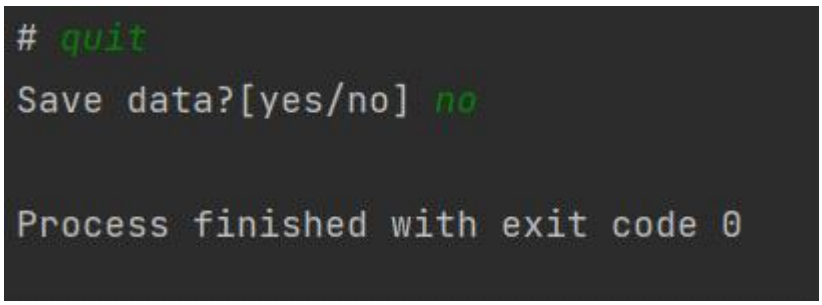
quit 입력 시, 프로그램을 종료합니다. 해당 명령어를 실행할 경우, 현재까지 편집한 내용의 저장 여부를 묻고, 저장을 선택(yes 입력)할 경우 파일명을 입력 받아서 저장하도록 합니다. 앞서 본 "students.txt"와 같이 내용을 구성합니다.

[Student number][₩t][Name][₩t][Midterm][₩t][Final][₩n]

저장할 때 목록의 순서는 평균을 기준으로 내림차순으로 합니다. 파일 이름에는 공백이 없다고 가정합니다.








4. 토론:

- 플로우 차트를 먼저 작성하지 않고 프로그램 코드를 먼저 작성했습니다. 그 결과 중간에 알고리즘이 순서가 뒤죽박죽 섞이는 결과가 발생했습니다. 이러한 문제를 해결하기 위해 플로우 차트를 그리면서 다시 설계를 해보았습니다.

5. 결론:

- 이과제를 통해 알고리즘 구현 능력을 익힐 수 있었습니다. 그리고 딕셔너리에 익숙하지 않아서 평소에 많은 어려움이 있었지만, 이번 과제를 통해 많이 사용해봄으로써 딕셔너리

에 대해 익숙해졌습니다.