# 2022년 HuStar 4기

Python Programming

**1주차 과제**

**담당교수: 윤은영**

**HuStar 4기**

**이름: 김민호**

**메일: alsgh1125@gmail.com**

**명예서약(Honor code)**

**“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”**

# Problem: 성적 관리 프로그램

## 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

* + 학생들의 성적을 관리하는 프로그램이다.
  + 사용자로부터 7개의 명령어 (show, search, changescore, searchgrade, add, remove, quit)를 입력 받아 각 기능을 수행 하게 된다.
  + 각 명령어 별로 함수를 정의해서 사용하며 필요 시 추가로 정의하여 사용한다.

이 때 사용되는 구상 가능한 의사코드(Pseudo Code)는 다음과 같다.

|  |
| --- |
| **Pseudo-algorithm for grade of students processing** |
| def calculate(score):  CALCULATE average, grade  def show(sID):  PRINT students AFTER sort  def search(sID):  IF student\_ID IN students THEN  **return** student  ELSE  PRINT NO SUCH PERSON  ENDIF  def searchgrade(grade):  IF grade IN studetns THEN  **return** student[grade]  ELSE  PRINT NO RESULTS  ENDIF  def changescore(sID):  IF search(sID) THEN  INPUT exam, score  UPDATE students(exam, score)  PRINT changed students  ENDIF  def add(sID):  IF search(sID) THEN  PRINT ALREADY EXISTS  ENDIF  INPUT name, mid, **final**  ADD student(name, mid, **final**)  def remove():  INPUT sID  IF sID IN students THEN  REMOVE student(sID)  def quit()  IF save to yes THEN  SAVE file AND Quit  ELSE  **do** not SAVE file AND Quit  def main()  INPUT file name  IF EXISTS file name THEN  WHILE  INPUT command  CASE  show: show()  search: search()  searchgrade: searchgrade()  changescore: changescore()  add: add()  remove: remove()  quit(): quit()  ENDCASE  ENDWHILE  ENDIF |

대략적인 프로그램의 진행 방법을 기술하였다.

* + calculate(): 파라미터로 넘겨받은 딕셔너리에 대해 Mid, Final 성적으로 Average와 grade를 계산한다.
  + show(): Students 딕셔너리를 출력하는 함수이다.
  + search(): Students\_ID 를 사용해 해당 ID가 딕셔너리에 존재하는지를 검사하는 함수이다.
  + searchgrade(): 학생의 성적 등급을 사용해 존재하는 학생을 출력하는 함수이다.
  + changescore(): ID와 변경할 시험 타입, 시험 성적 등을 입력받아 해당하는 ID의 시험 점수를 변경하는 함수이다.
  + add(): 새로운 학생 정보를 추가하는 함수이다. 이미 존재하는 학생일 경우 에러처리를 해준다.
  + remove(): ID를 사용해 해당 ID를 딕녀서리에서 제거하는 함수이다.
  + quit(): 프로그램 전체를 종료하고 파일 저장 여부를 결정한다.

## Flowchart

위의 의사 알고리즘을 Flowchart를 통해 표현하면 아래와 같다.

|  |
| --- |
|  |

## 프로그램 구조 및 설명

1. 커널을 통해 실행할 Text File 명을 입력

- 커널 실행 시 불러올 Text File 명을 함께 입력하여 kernel 상에서 입력된 File명을 변수로 선언해 file open을 진행한다.

1. 불러온 Text File (Students.txt)에 대해 데이터 처리를 진행
   * 최초 실행 시 Students.txt에 있는 데이터를 읽어와 Students 딕셔너리에 저장 후 show()함수 호출을 통해 데이터를 출력한다.
   * while문을 통해 파일이 종료되기 전까지 필요한 함수 호출을 입력받은 command로 진행한다.
   * 호출 가능한 함수는 show, search, searchgrade, changescore, add, remove, quit 으로 총 7개이며 해당 함수들은 적절한 반환 값을 가지고 필요 시 내부에서 다른 함수를 호출하여 사용한다.
2. 필수 함수 이외의 사용자 정의 함수
   * 필수 포함 함수인 7개의 함수를 제외한 세부 기능을 하는 함수를 몇가지 추가하였다.
   * calculate() 함수는 Students에 입력된 mid, final 성적을 통해 average와 grade를 판별하여 Students 딕셔너리에 추가한다.
   * initStudents() 함수는 전달 받은 파라미터 (Students, sID, name, score)를 바탕으로 계산된 avg, grade 값을 포함한 키,값 쌍을 Students 딕셔너리에 초기화한다.
3. 추가 고려 사항
   * 함수 호출 시 매개변수의 자료형을 리스트가 아닌 딕셔너리를 사용함으로써 search와 같은 탐색 기능 수행 시 키 값을 통해 빠르게 찾을 수 있도록 구현하였다.
   * 전역 변수가 아닌 지역 변수를 사용할 수 있도록 구현하여 전역 변수 사용 시 발생 가능한 충돌 문제 등에 대한 고려를 하였다.

## 프로그램 실행방법 및 예제

1. 토론
   * Students 딕셔너리를 출력 시 오른쪽 정렬하여 출력 구문의 가독성을 신경썼다.
   * 성적 계산 함수인 calculate() 함수와 계산된 성적을 추가하여 초기화해주는 initStudents() 함수를 추가로 작성하여 코드를 간소화하고 코드를 재사용하여 사용할 수 있게 구현하였다.
   * 처음 구현 시 Students를 이중 리스트로 구현하여 함수 호출 시 전달할 수 있도록 구현했으나 가독성 및 코드 간소화를 위해 딕셔너리로 수정하여 key 값을 통해 Students를 제어할 수 있도록 수정하였다.
   * 처음은 Students를 전역변수로 선언하여 데이터를 업데이트 했으나 이후 코드의 유지 보수와 프로그램 규모가 커졌을 때를 고려해 전역 변수 사용을 자제하고 지역 변수를 선언해 자료를 Update 할 수 있게 수정하였다.
   * changescore 함수 내부에 change 라는 성적 수정 함수를 추가로 구현해 mid 또는 final 성적을 수정 시 반복되는 코드를 간소화 하였다.
   * add() 함수에서 잘못된 성적 범위를 입력시 오류를 판별할 수 있도록 추가하였다.
2. 결론
   * 본 과제에서는 성적 관리 프로그램이라는 주제로 필요한 함수를 작성하고 사용하는 방법을 터득하였다. 특히 전역 변수와 지역 변수, immutable 객체와 mutable 객체에 대한 차이점을 명확히 이해하였고, 함수를 재사용할 수 있게 구현하는 것이 중요함을 알게 되었다.
   * 어떠한 프로그램을 구현할 때에 무작정 코드부터 짜는 것이 아닌 개념을 이해하고 설계 단계에 충분한 고민을 한 이후 구현을 하는 것이 오히려 시간을 절약하고 코드의 효율성을 높이는 길임을 깨닫게 되었다.
3. 개선방향
   * Students를 클래스로 구현하여 필요한 함수를 클래스 내부 메소드로 구현하였다면 훨씬 더 코드가 간소화되고 가벼워질 것이라 생각한다. 그리고 성적 입력 시 잘못된 범위 입력, 저장 시 yes/no 이외의 입력에 대한 에러 처리 구문 등을 추가하여 좀 더 완성도 높은 프로그램을 구현한다면 쓰임새가 상승할 것이라 기대된다.