

프로그래밍 입문 – 소프로젝트 2

1616557 IT공학과 박아정

1. 코드 설명

A. ScoreCalculator.py

- 클래스를 정의한 파일로 클래스의 멤버변수, 멤버함수가 정의되어 있다.

멤버변수	멤버함수
<ul style="list-style-type: none">· 1차 점수: fir· 2차 점수: sec· 3차 점수: thi· 4차 점수: fort· 5차 점수: fif· 6차 점수: six· 7차 점수: sev· 점수 리스트: score_list	<ul style="list-style-type: none">· 생성자: __init__· 최고점 제거 함수: del_max· 최저점 제거 함수: del_min· 평균 점수 함수: avg_score

- del_max 함수

```
def del_max(self):  
    maxS = max(self.score_list) #int 값으로 max 값 나눔.  
                                # list 의 최대값 구해주는 내장함수 max  
    i = self.score_list.index(maxS)  
    del self.score_list[i]      #최대값을 찾아 삭제한다.  
    return self.score_list
```

- del_min 함수

```
def del_min(self):  
    minS = min(self.score_list) #내장함수 min  
    i = self.score_list.index(minS)  
    del self.score_list[i]      #min 값을 찾아 삭제  
    return self.score_list
```

- avg_score 함수

```
def avg_score(self):  
    sum = 0
```

```

num = len(self.score_list) #list의 길이를 구해서
for i in range(num):
    sum += self.score_list[i] #총 합계 구하기
return sum/num

```

B. Score_Calculator.py

- 변수 설명

fir, sec, thi, fort, fif, six, sev	선수의 1~7차 점수를 받아오는 변수
Gymnast_cal	클래스 생성
delMax_list	최대값을 삭제한 list를 받아오는 변수
delMin_list	최소값을 삭제한 list를 받아오는 변수
agv_score	평균이 저장되는 변수

- 코드 설명

```

from ScoreCalculator import ScoreCalculator
import time #시간을 지연시키기 위해

print("="*26)
print("Gymnast Score Calculator")
print("="*26)

#이름과 점수를 받아온다.
name = input("Enter the Gymnast's name: ")
scores = input("Enter the scores: ")

#받아온 점수를 띄어쓰기로 구분하여 각 변수에 점수 대입한다.
fir, sec, thi, fort, fif, six, sev = scores.split(" ")

# 클래스 생성
Gymnast_cal = ScoreCalculator(int(fir), int(sec), int(thi),
                               int(fort), int(fif), int(six), int(sev))

#ScoreCalculator 클래스 멤버함수 사용
delMax_list = Gymnast_cal.del_max() #최대값 삭제
delMin_list = Gymnast_cal.del_min() #최소값 삭제

avg_score = Gymnast_cal.avg_score() #평균 구하기

print("Calculator processing", end = "")

```

*#time.sleep 함수를 사용하여 마치 컴퓨터가 계산한 것 처럼 보여준다.
#time.sleep()에서 괄호 안에 초를 넣어준다.*

```
for i in range(9):
    time.sleep(1)
    print(".", end = "")

print()

print(name+"'s final score is: %0.2f" %avg_score)
```

C. GymnastAwards.py

- 변수 설명

filename	엑셀 파일을 받아올 변수
Gymnast_cal	ScoreCalculator 클래스를 사용하기 위해 생성
Score_list	7명의 선수들의 점수 list
max_row, max_col	엑셀의 행과 열의 개수
name_score	(선수, 평균점수)쌍 딕셔너리 변수
delMax_list	최대값을 삭제한 list 변수
delMin_list	최소값을 삭제한 list 변수
avg_score	평균점수를 저장하는 변수
sortedScore	정렬된 (이름:평균점수)쌍을 저장하는 변수
sortedScoreList	딕셔너리 -> 리스트

- 코드 설명

: for문을 사용해서 엑셀파일의 한 행을 읽고 Score_list에 값을 저장한다. 그 값을 함수를 사용해 계산하고 새로운 파일의 셀에 저장한 뒤 다음 행을 읽기 위해 다시 초기화 시킨다.

: 받아온 데이터에서 (이름 : 평균점수)딕셔너리를 만들고 sorted함수를 사용해 내림차순으로 정렬한다. 리스트의 상위 3번째까지 값을 받아오기 위해서 딕셔너리를 리스트로 바꾸고 결과를 출력한다.

```
from ScoreCalculator import ScoreCalculator
import openpyxl
import operator
```

```

# player_score.xlsx 파일 불러와서 데이터를 받아온다.
filename = "player_score.xlsx"

wb_obj = openpyxl.load_workbook(filename)
sheet_obj = wb_obj.active

max_row = 7    #행의 개수
max_col = 8    #열의 개수

currentCell_obj = sheet_obj.cell(row=1, column=1)
Score_list = []    #7 명의 선수들의 점수 list
name_score = {}    #(선수, 평균점수) 쌍 딕셔너리

for i in range(2, max_row+1):    #player_score.xlsx 파일에서
    '점수'와 '이름'을 가져온다.
        for j in range(1, max_col+1):
            currentCell_obj = sheet_obj.cell(row = i, column=j)
            Score_list.append(currentCell_obj.value)    # 셀의
            value 를 Score_list 에 추가한다.

    #클래스 생성해서 값으로 받아온 list 값들 넣어준다.
    Gymnast_cal = ScoreCalculator(Score_list[1], Score_list[2],
    Score_list[3], Score_list[4], Score_list[5], Score_list[6],
    Score_list[7])
    delMax_list = Gymnast_cal.del_max()    # 최대값 삭제
    delMin_list = Gymnast_cal.del_min()    # 최소값 삭제
    avg_score = Gymnast_cal.avg_score()    # 평균 구하기

    #이름-점수 쌍을 name_score 에 삽입한다.
    name_score[Score_list[0]] = avg_score
    Score_list = []    # 한 행을 처리한 뒤 Score_list 를 초기화 한다.

#받아온 name_score 를 내림차순으로 정렬한다
# operator.itemgetter(1)로 하여 value 값을 기준으로 한다.
#reverse = True 로 하여 내림차순으로 출력되게 한다.
sortedScore = sorted(name_score.items(), key =
operator.itemgetter(1), reverse=True)
sortedScoreList = list(sortedScore)    #list 로 만들고

print("="*50)
print("Gymnast Awards Ceremony")
print("="*50)

#sortedScoreList 에서 앞에 1,2,3 요소가 1,2,3 등이므로 출력(앞에서
정렬했기 때문에)

```

```

print("Gold medal : ", sortedScoreList[0][0], "(final score = ",
sortedScoreList[0][1], ")")
print("Silver medal : ", sortedScoreList[1][0], "(final score = ",
sortedScoreList[1][1], ")")
print("Bronze medal : ", sortedScoreList[2][0], "(final score = ",
sortedScoreList[2][1], ")")

print("="*50)
print("Congratulations!")

```

2. 실행 결과

```

C:\Users\qkrdk\PycharmProjects\IT_1616557_Project2\venv\Scripts\py
C:/Users/qkrdk/PycharmProjects/IT_1616557_Project2/Score_Calculat
=====
Gymnast Score Calculator
=====
Enter the Gymnast's name: Susan
Enter the scores: 7 8 9 10 9 8 0
Calculator processing.....
Susan's final score is: 8.20

Process finished with exit code 0
|

```

➔ 실행했을 때 “형광펜” 부분이 마치 컴퓨터가 계산하는 것처럼 보여 진다.

```

C:\Users\qkrdk\PycharmProjects\IT_1616557_Project2\venv\Scripts\pyth
C:/Users/qkrdk/PycharmProjects/IT_1616557_Project2/GymnastAwards.py
=====
Gymnast Awards Ceremony
=====
Gold medal : Nathan (final score = 7.8 )
Silver medal : Kevin (final score = 6.0 )
Bronze medal : Mike (final score = 5.4 )
=====
Congratulations!

Process finished with exit code 0

```