

파일럿 프로젝트 결과 보고서

<Golf Score 관리자>

| 1조 | |
|----|----------|
| PM | 전북은행 김영목 |
| 팀원 | 광주은행 김은민 |
| | 광주은행 최대훈 |
| | 전북은행 박요온 |

I . Overview and Project Schedule

[Contents]

I . Project 목적 및 Schedule ... 2P

II. Project 진행 보고 ... 3P

III. Project 최종 결과 ... 8P

IV. Project 모니터링 ... 9P

□ 개요

Golf Score 관리자 구현

✓ 목적 : Golf Score 관리자 Package 구현

✓ 진행방식 : 참여율을 높이기 위한 미션별 토론 및 개별 코드 작성 후
최적화 선택 방식

✓ 사용 Tool : 파이참 (PyCharm)

□ Schedule

| Major Task | | 5월 | | | | | | | |
|---------------|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | 19(목) | 20(금) | 21(토) | 22(일) | 23(월) | 24(화) | 25(수) | 26(목) |
| Project 기획 | 주제 및 PM 선정 | | | | | | | | |
| | Project 진행방식 논의 | | | | | | | | |
| Project 설계 | 미션 세분화 | | | | | | | | |
| | 미션별 담당자 지정 | | | | | | | | |
| Project 구현 | 부문별 코드 작성 | | | | | | | | |
| | 함수 쪼개기 | | | | | | | | |
| | 패키지 작성 | | | | | | | | |
| 결과 보고 | 보고서 작성 | | | | | | | | |

II. Project 진행 보고

□ 단계별 Project

✓ Mission별 Project 구현 방법 및 함수 설명

Mission 1)
- 참가 선수들 등록



1) Excel file read 및 참가 선수들 등록하기

✓ Step1) Excel file 불러오기 실행

✓ Step2) 참가 선수를 리스트 형태로 변환

- 참가 선수 : '명단.txt' 파일 생성
- 선수 저장 : 선수 1명씩 가져온 후
리스트 형태로 저장
- 엑셀 등록 : 선수 list 개수만큼
엑셀 등록

```
def 선수등록(excel_file, filename):  
  
    # step1 : file 불러오기  
    file = open("{}".format(filename), "r", encoding='utf-8')  
    line = file.readlines()  
    file.close()  
  
    # step2 : 참가선수를 리스트형태로 변환  
    wb, ws, ws2 = loading_excel(excel_file)  
    peoples = []  
    for idx in range(len(line)):  
        people = line[idx].replace('\n', ' ') # 참가선수1명씩 가져옴.  
        peoples.append(people) # 전체참가선수를 리스트형태로 저장  
  
    for idx in range(len(peoples)): # 엑셀에 참가선수 뿌리기  
        b = ws.cell(row=idx + 3, column=1, value=peoples[idx])  
        print(b.value)  
  
    wb.save(excel_file)  
    print('참가선수 등록 완료')  
  
    return peoples
```

Mission 2)
- 선수 HOLE별 Score 랜덤 입력

Mission 3)
- Score 합산 입력

Mission 4)
- 랭킹 및 각종 기록 개수 등록

Mission 5)
- 결과 Sheet에 최종 성적 등록

Ⅱ. Project 진행 보고

□ 단계별 Project

Mission 1)

- 참가 선수들 등록

Mission 2)

- 선수 HOLE별 Score 랜덤 입력

Mission 3)

- Score 합산 입력

Mission 4)

- 랭킹 및 각종 기록 개수 등록

Mission 5)

- 결과 Sheet에 최종 성적 등록

✓ Mission별 Project 구현 방법 및 함수 설명

1) def 난수뿌리기(filename):

✓ Step1) 18홀까지의 홀별 타수 난수 추출

- 타수(par_ran) = randint()를 통한 난수 추출 (1~par*2)

✓ Step2) 선수별 18홀까지 SCORE값 입력

- Score = 타수(par_ran) - PAR

- ex) PAR 3, 타수 = 2

$$2 - 3 = -1$$

```
def 난수뿌리기(excel_file, player):  
    wb, ws, ws2 = loading_excel(excel_file)  
  
    for row_no in range(3, len(player)+3):  
        for col_no in range(3, 21):  
            inp_nm = '{}-{}'.format(get_column_letter(col_no), 2)  
            par_ran = randint(1, ws[inp_nm].value*2)  
            score = par_ran - ws[inp_nm].value  
            # ws.cell(column = col_no, row = row_no, value = par_ran)  
            ws.cell(column = col_no, row = row_no, value = score)  
        # return score  
  
    wb.save(excel_file)  
    print('난수뿌리기 등록 완료')  
    return
```

Ⅱ. Project 진행 보고

□ 단계별 Project

Mission 1)

- 참가 선수들 등록

Mission 2)

- 선수 HOLE별 Score 랜덤 입력

Mission 3)

- Score 합산 입력

Mission 4)

- 랭킹 및 각종 기록 개수 등록

Mission 5)

- 결과 Sheet에 최종 성적 등록

√ Mission별 Project 구현 방법 및 함수 설명

1) def 스트로크합산(excel_file, player):

√ Step1) 선수(행)의 18번홀 (열)까지 value 값을 arr_raw 리스트에 합산

√ Step2) 보정치는 0이므로 스트로크 = 최종성적이므로

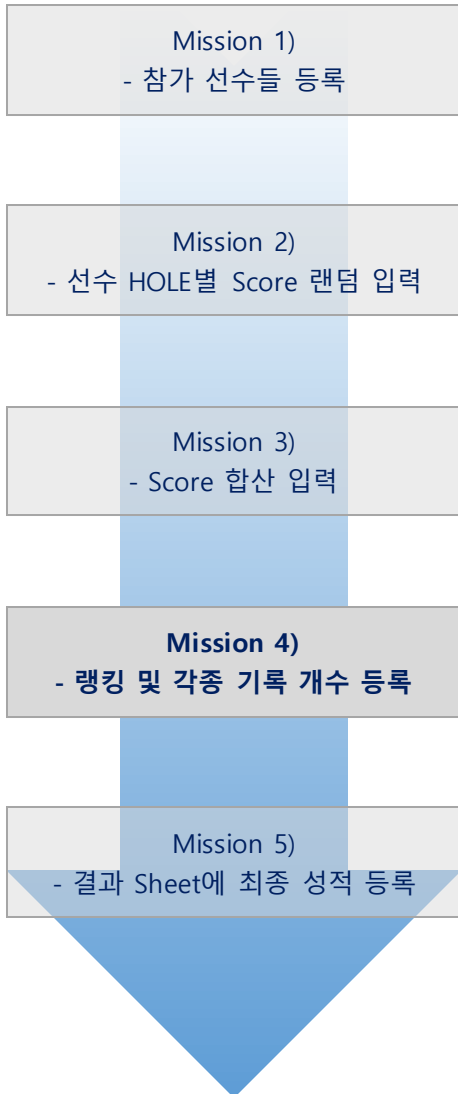
- 스트로크(21열) : Score 랜덤합 + 72(18번홀까지 PAR의 합)

- 최종성적(23열) : 스트로크

```
def 스트로크합산(excel_file, player):  
    wb, ws, ws2 = loading_excel(excel_file)  
  
    for col_no in range(3, len(player)+3):  
        arr_raw = list()  
        for a in range(3, 21):  
            ran = '{}{}'.format(get_column_letter(a), col_no)  
            arr_raw.append(ws[ran].value)  
        stroke = sum(arr_raw)+72  
        ws.cell(column=21, row=col_no, value=stroke)  
        ws.cell(column=23, row=col_no, value=stroke)  
  
    wb.save(excel_file)  
    return
```

II. Project 진행 보고

□ 단계별 Project



✓ Mission별 Project 구현 방법 및 함수 설명

1) def 선수점수정렬(filename):

2) def 등수만들기(filename):

1,2) Return 확인

```
선수점수정렬 : [68, 68, 68, 69, 70, 70, 71, 79, 80, 81]  
등수만들기    : {68: 1, 69: 4, 70: 5, 71: 7, 79: 8, 80: 9, 81: 10}
```

Step1) 최종성적만 가지고 와서 오름차순으로 정렬을 한다.

Step2) key 값은 최종성적, value 는 등수인 딕셔너리를 생성한다.

3) def 랭킹채우기(filename):

```
for row_idx in range(3, len(player) + 3):  
    score = ws.cell(column=23, row=row_idx).value  
    if score in rank_dict:  
        rank = rank_dict[score]  
        ws.cell(column=24, row=row_idx, value=rank)  
    else:  
        pass
```

엑셀에 적혀진 점수와 딕셔너리의 key 값이 같다면 그때의 딕셔너리 value 값을 삽입해준다.

4) def 기록 카운트(filename):

엑셀에 디폴트로 0값을 채워준 후, IF문을 통해 개수를 카운트해서 기록해준다.

II. Project 진행 보고

□ 단계별 Project

Mission 1)
- 참가 선수들 등록

Mission 2)
- 선수 HOLE별 Score 랜덤 입력

Mission 3)
- Score 합산 입력

Mission 4)
- 랭킹 및 각종 기록 개수 등록

Mission 5)
- 결과 Sheet에 최종 성적 등록

✓ Mission별 Project 구현 방법 및 함수 설명

5) def 대회결과등록(excel_file, player) : 결과 Sheet에 최종 성적 등록하기

✓ Step1) 선수별 기록 List화

- 선수별 스트로크 및 각종 기록 List화 작업 (파일명 : infos)

```
def 선수정보담기():  
    #sheet2번에서 활용할 용도 (선수, 이름 등 가져오기)  
    infos = list()  
    for row_idx in range(3, len(player) + 3):  
        person_dict = {}  
        person_dict['선수명'] = ws.cell(column=1, row=row_idx).value  
        person_dict['스트로크'] = ws.cell(column=21, row=row_idx).value  
        person_dict['행킹'] = ws.cell(column=24, row=row_idx).value  
        person_dict['버디'] = ws.cell(column=26, row=row_idx).value  
        person_dict['파'] = ws.cell(column=27, row=row_idx).value  
        person_dict['보기'] = ws.cell(column=28, row=row_idx).value  
        person_dict['양파'] = ws.cell(column=29, row=row_idx).value  
        #채우기  
        infos.append(person_dict)  
    return infos  
선수정보담기()
```

✓ Step2) 대회 결과 (결과표 ①)

- Step1에서 생성한 list 자료(infos)를 스트로크순(순위)으로 정렬 후 상위 3명 / 하위 1명 read

```
def 대회결과():  
    #sheet2번 첫번째 표 채우기 (대회결과)  
    infos = 선수정보담기()  
    sorted_infos = sorted(infos, key=_lambda x: (x['스트로크']))  
  
    row_idx = 2 #2행부터 채울거니까  
    for i in [0,1,2,-1]: #앞에 상위 3명, 뒤에 하위 1명 가져온다.  
        info = sorted_infos[i]  
        ws2.cell(column=1, row=row_idx, value='골프' if i == -1 else str(info['행킹']) + '등')  
        ws2.cell(column=2, row=row_idx, value=info['선수명'])  
        ws2.cell(column=3, row=row_idx, value=info['스트로크'])  
  
        row_idx += 1 #2,3,4,5행에 넣을 거니까  
    대회결과()
```

✓ Step3) 대회 결과 (결과표 ②)

- Step2와 동일한 방식으로 각 기록별 내림차순 후 해당 선수 및 기록 입력

```
def 버디최고기록(infos):  
    many_birdie = sorted(infos, key=lambda x: (x['버디']), reverse=True)[0]  
    ws2.cell(column=2, row=8, value=many_birdie['선수명'])  
    ws2.cell(column=3, row=8, value=many_birdie['버디'])
```

Ⅲ. Project 최종 결과

□ 단계별 Project

1. [시트1. 스코어]

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | AA | AB |
|----|------|------|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|-------|-------|---------|-------|-------|-----|-------|
| 1 | NAME | HOLE | H1 | H2 | H3 | H4 | H5 | H6 | H7 | H8 | H9 | H10 | H11 | H12 | H13 | H14 | H15 | H16 | H17 | H18 | STROKE | HANDY | SCORE | RANKING | EAGLE | BIRDY | PAR | BOGGY |
| 2 | 선수명 | PAR | 4 | 4 | 3 | 5 | 4 | 5 | 4 | 3 | 4 | 5 | 3 | 4 | 4 | 3 | 4 | 4 | 5 | 4 | 스트로크 | 보정치 | 최종성적 | 랭킹 | 이글 | 버디 | 파 | 보기 |
| 3 | 김은민 | | -1 | -2 | 0 | 0 | -1 | -1 | 0 | 0 | -3 | -3 | 2 | 1 | 0 | 2 | 0 | -1 | 1 | 4 | 70 | | 70 | 2 | 1 | 4 | 6 | 2 |
| 4 | 박승현 | | 4 | 2 | -2 | -4 | 1 | -2 | -1 | 2 | -3 | 2 | 1 | 4 | -3 | 3 | -3 | 1 | -2 | 4 | 76 | | 76 | 4 | 3 | 1 | 0 | 3 |
| 5 | 이건호 | | -3 | -3 | -2 | -4 | -1 | -4 | -1 | 3 | 1 | -1 | 2 | 3 | 0 | -2 | 2 | -1 | -4 | 2 | 59 | | 59 | 1 | 2 | 4 | 1 | 1 |
| 6 | 정은지 | | 3 | 2 | -1 | -3 | 4 | -1 | -2 | -2 | -3 | 3 | 0 | 0 | 0 | 0 | 3 | -1 | -4 | 3 | 72 | | 72 | 3 | 2 | 4 | 3 | 0 |
| 7 | 최다훈 | | -1 | 0 | 3 | 5 | -1 | 4 | -3 | -1 | 2 | -3 | -2 | -3 | -1 | 3 | 0 | 4 | -4 | 2 | 76 | | 76 | 4 | 1 | 4 | 2 | 0 |
| 8 | 한혜형 | | 4 | 3 | 1 | -3 | 3 | 5 | 4 | 1 | -2 | 4 | -2 | -3 | -1 | -1 | 4 | 4 | 2 | -1 | 94 | | 94 | 11 | 2 | 3 | 0 | 2 |
| 9 | 김영목 | | 3 | 2 | 1 | -4 | -3 | -3 | 0 | 3 | 1 | -3 | 3 | 1 | 3 | 1 | 3 | 4 | 5 | -2 | 87 | | 87 | 7 | 1 | 0 | 1 | 4 |
| 10 | 박성실 | | 1 | 4 | 1 | 1 | 2 | -1 | 1 | 3 | -2 | 5 | 3 | 4 | -2 | 0 | -1 | -1 | -3 | 2 | 89 | | 89 | 10 | 2 | 3 | 1 | 4 |
| 11 | 박오은 | | 2 | 3 | 3 | 2 | -1 | 2 | -3 | 2 | 2 | 1 | 1 | -3 | 1 | 0 | -3 | 3 | 4 | -1 | 87 | | 87 | 7 | 0 | 2 | 1 | 3 |
| 12 | 오승현 | | -3 | -2 | 0 | -2 | -1 | 0 | 3 | 1 | 4 | 3 | 0 | 4 | 0 | 0 | 0 | 0 | 3 | 1 | 83 | | 83 | 6 | 2 | 1 | 7 | 2 |
| 13 | 김진수 | | 1 | 0 | 2 | -2 | 2 | 3 | 1 | 3 | 1 | 2 | -1 | 4 | -1 | -1 | -2 | 4 | 1 | -1 | 88 | | 88 | 9 | 2 | 4 | 1 | 4 |

- ① 랭킹 : 동점시 동일랭킹 출력 (X4, X7 참고)
- ② 보정치 Null → 스트로크 = 최종성적
- ③ 이글, 버디 등 카운트 → 없을시 0으로 처리

2. [시트2. 스코어]

| | A | B | C |
|----|------|-----|------|
| 1 | 대회결과 | 선수명 | 최종성적 |
| 2 | 1등 | 이건호 | 59 |
| 3 | 2등 | 김은민 | 70 |
| 4 | 3등 | 정은지 | 72 |
| 5 | 꼴등 | 한혜형 | 94 |
| 6 | | | |
| 7 | 다기록 | 선수명 | 기록수 |
| 8 | 다버디 | 김은민 | 4 |
| 9 | 다파 | 오승현 | 7 |
| 10 | 다보기 | 김영목 | 4 |
| 11 | 다양파 | 한혜형 | 5 |

- ① 대회결과 : 동점시 동일랭킹 출력 (A9, A11 참고), 양식 준수를 위하여 꼴등동일시 1명만 출력
- ② 다기록 : 양식 준수를 위하여 동점시 1명만 출력(다파 : AB9, AB10 참고)

IV. Project 모니터링

□ Project 후기

| 구분 | | Lessons Learned |
|----|-----|---|
| PM | 김영목 | <ul style="list-style-type: none"> □ 시작전 요건 정의의 중요성 □ 일정관리 및 적절한 업무배정 □ 모듈화와 패키지 기반의 코딩 및 확인법 |
| 팀원 | 김은민 | 무작정 코드부터 작성하는 것이 습관이었는데 이번 프로젝트를 통해서 시작 전 기획을 하는 것이 효율적이라는 것을 알게 되었습니다. 특히, 미니 프로젝트를 통해서 함수를 패키지로 구현하는 것을 직접 적용해 볼 수 있었던 뜻 깊은 기회라고 생각합니다. |
| | 최대훈 | 골프 5개월 차에게 매우 흥미로운 주제로 다가왔으나 영원히 골프와 이별할 뻔 했습니다. 그러나 다행히 훌륭한 실력들을 가진 조원들 덕분에 혼자서 고민하던 문제가 해결되고, 함수를 구현하는 과정에서 '아, 여기서 이렇게 하는구나!' 하며 무릎을 여러 번 탁 쳤던 것 같습니다. 이번 파일럿 프로젝트를 통해 지금까지 배운 내용을 곱씹어가며 실력을 쌓을 수 있었던 유익한 시간이었습니다. |
| | 박요운 | <ul style="list-style-type: none"> - 좋았던 점 : 자유로운 분위기에서 열정적인 PM을 필두로 모든 팀원이 적극적으로 참여하여 수월하게 진행되었음. 또한, 개별적 Coding 작성 후 공유를 통한 방식으로 미처 놓쳤던 부분에 대해서도 학습할 수 있었음. - 아쉬운 점 : 팀원간 Coding 공유 시, 개인적인 Coding 실력 부족으로 이해하는데 많은 시간이 소요되었으며 이에 따른 프로젝트 시간이 다소 지연됨. |