

# 소프트웨어 프로젝트

1조

정진혁, 조가성, 조경상, 조상연

```
1  ●import pickle
2
3  ●dbfilename = 'assignment3.dat'
4  ●def readScoreDB():
5      try:
6          ●fh = open(dbfilename, 'rb')
7          ●except FileNotFoundError as e:
8              print('New DB: ', dbfilename)
9              return []
10
11     scdb = []
12     try:
13         scdb = pickle.load(fh)
14     ●except:
15         print('Empty DB: ', dbfilename)
16     else:
17         print('Open DB: ', dbfilename)
18     for record in scdb:
19         ●for t in record:
20             if t == 'Name':
21                 continue
22             record[t] = int(record[t])
23     fh.close()
24     return scdb
25
26 ●def writeScoreDB(scdb):
27     ●fh = open(dbfilename, 'wb')
28     pickle.dump(scdb, fh)
29     fh.close()
30
```

#Age와 Score을 정수형으로 받는 부분

# Cord Review

(완성된 코드)

```

31 def doScoreDB(scdb):
32     while(True):
33         inputstr = input('Score DB >>> ')
34         if inputstr == '':
35             continue
36         parse = inputstr.split(" ")
37         if parse[0] == 'add':
38             if len(parse) != 4:
39                 print('There are not enough arguments to "add command".') # 예외(에러) 처리 부분
40                 continue
41             record = {'Name':parse[1], 'Age':int(parse[2]), 'Score':int(parse[3])}
42             scdb += [record]
43         elif parse[0] == 'command':
44             command_manual()
45         elif parse[0] == 'find':
46             for p in scdb:
47                 if p['Name'] == parse[1]:
48                     person_name = [p]
49                     findScoreDB(person_name)
50             if not parse[1] in p['Name']:
51                 print(''+parse[1]+'', 'is not name in list.') # 예외(에러) 처리 부분
52                 print('Please add to the list')
53         elif parse[0] == 'inc':
54             for p in scdb:
55                 if p['Name'] == parse[1]:
56                     p['Score'] += int(parse[2])
57                     break
58             if not parse[1] in p['Name']:
59                 print(''+parse[1]+'', 'is not name in list.') # 예외(에러) 처리 부분
60                 print('Please add to the list')
61         elif parse[0] == 'del':
62             for p in scdb:
63                 if p['Name'] == parse[1]:
64                     scdb.remove(p)
65             print('delete', ''+parse[1]+'')
66             if not parse[1] in p['Name']:
67                 print(''+parse[1]+'', 'is not name in list.') # 예외(에러) 처리 부분
68         elif parse[0] == 'show':
69             if len(parse) >= 2:
70                 print("Invalid command input")
71                 continue
72             else:
73                 sortKey = 'Name'
74                 showScoreDB(scdb, sortKey)
75         elif (parse[0] == 'quit') or (parse[0] == 'exit'):
76             break
77         else:
78             print('Invalid command: ' + parse[0])
79

```

# Cord Review

(완성된 코드)

```
79
80 •def showScoreDB(scdb, keyname):
81 •    for p in sorted(scdb, key=lambda person: person[keyname]):
82         for attr in sorted(p):
83             print(attr + '=' + str(p[attr]), end=' ')
84         print()
85
86 •def findScoreDB(pname):
87 •    for p in pname:
88         for attr in sorted(p):
89             print(attr + '=' + str(p[attr]), end=' ')
90         print()
91
92 •def command_manual():
93     print('add: "add + [Name] + [Age] + [Score]"')
94     print('find: "find + [Name]"')
95     print('increase: "inc + [Name] + [Amount]"')
96     print('delete: "del + [Name]"')
97     print('show: "show"')
98     print('exit: "quit OR exit"')
99
100
101 #start!
102 print('-----Score Manager-----')
103 print('If you want a manual, type the following command: "command"')
104 •scoredb = readScoreDB()
105 print()
106 print('-----File contents-----')
107 showScoreDB(scoredb, 'Name') #처음 데이터 파일의 상태를 파악하기위해...
108 doScoreDB(scoredb)
109 writeScoreDB(scoredb)
110
```

# Cord Review

(추가 or 수정 or 에러(예외 처리))

```
1  •import pickle
2
3  •dbfilename = 'assignment3.dat'
4  •def readScoreDB():
5      try:
6          fh = open(dbfilename, 'rb')
7          •except FileNotFoundError as e:
8              print('New DB: ', dbfilename)
9              return []
10
11     scdb = []
12     try:
13         scdb = pickle.load(fh)
14     •except:
15         print('Empty DB: ', dbfilename)
16     else:
17         print('Open DB: ', dbfilename)
18     for record in scdb:                #Age와 Score을 정수형으로 받는 부분
19     •    for t in record:                #
20         if t == 'Name':                #
21             continue                    #
22         record[t] = int(record[t])     #
23     fh.close()
24     return scdb
25
26  •def writeScoreDB(scdb):
27  •    fh = open(dbfilename, 'wb')
28        pickle.dump(scdb, fh)
29        fh.close()
```

# Cord Review

(추가 or 수정 or 에러(예외 처리))

```
def doScoreDB(scdb):
```

```
    while(True):
```

```
        inputstr = input('Score DB >>> ')
```

```
        if inputstr == '':
```

```
            continue
```

```
        parse = inputstr.split(" ")
```

```
        if parse[0] == 'add':
```

```
            if len(parse) != 4:
```

#예외(에러) 처리 부분

```
                print('There are not enough arguments to "add command".') #
```

```
                continue
```

```
            record = {'Name':parse[1], 'Age':int(parse[2]), 'Score':int(parse[3])}
```

```
            scdb += [record]
```

↳ 저장이 될 때 'Age'와 'Score'은 정수형 으로 저장

```
        elif parse[0] == 'command':
```

>>> 사용자가 매뉴얼을 보길 원할 때 호출 되는 함수

```
            command_manual()
```

```
        elif parse[0] == 'find': >>> 'find' 명령어 구현
```

```
            for p in scdb:
```

```
                if p['Name'] == parse[1]:
```

```
                    person_name = [p]
```

```
                    findScoreDB(person_name) >>> 'findScoreDB' 함수 구현
```

```
            if not parse[1] in p['Name']: >>> 찾는 이름이 없을 때 실행됨(에러) 처리 부분
```

```
                print('"' + parse[1] + '"', 'is not name in list.')
```

```
                print('Please add to the list')
```

```
                #
```

# Cord Review

(추가 or 수정 or 에러(예외 처리))

```
53 elif parse[0] == 'inc':
54     for p in scdb:
55         if p['Name'] == parse[1]:
56             p['Score'] += int(parse[2])
57             break
```

```
58     if not parse[1] in p['Name']: #예외(에러) 처리 부분
59         print('"' + parse[1] + '"', 'is not name in list.')#
60         print('Please add to the list') #
```

```
61 elif parse[0] == 'del':
62     for p in scdb:
63         if p['Name'] == parse[1]:
64             scdb.remove(p)
65     print('delete', '"' + parse[1] + '"')
```

```
66     if not parse[1] in p['Name']: >>> 찾는 이름이 없을 때 실행됨 #예외(에러) 처리 부분
67         print('"' + parse[1] + '"', 'is not name in list.')#
```

```
68 elif parse[0] == 'show':
69     if len(parse) >= 2:
70         print("Invalid command input")
71         continue
72     else:
73         sortKey = 'Name'
74         showScoreDB(scdb, sortKey)
```

```
75 elif (parse[0] == 'quit') or (parse[0] == 'exit'):
76     break >>> 'exit' 명령어 추가
```

```
77 else:
78     print('Invalid command: ' + parse[0])
79
```

# Cord Review

(추가 or 수정 or 에러(예외 처리))

```
80 def showScoreDB(scdb, keyname):
81     for p in sorted(scdb, key=lambda person: person[keyname]):
82         for attr in sorted(p):
83             print(attr + '=' + str(p[attr]), end=' ')
84             print()
```

↑ 'Age'와 'Score'는 정수형이므로 str으로 묶어줘야함

```
86 def findScoreDB(pname): >>>
87     for p in pname:
88         for attr in sorted(p):
89             print(attr + '=' + str(p[attr]), end=' ')
90             print()
```

↑ 'Age'와 'Score'는 정수형이므로 str으로 묶어줘야함

```
92 def command_manual():
93     print('add: "add + [Name] + [Age] + [Score]"')
94     print('find: "find + [Name]"')
95     print('increase: "inc + [Name] + [Amount]"')
96     print('delete: "del + [Name]"')
97     print('show: "show"')
98     print('exit: "quit OR exit"')
```

```
100
101 #start!
102 print('-----Score Manager-----')
103 print('If you want a manual, type the following command: "command"')
104 scoredb = readScoreDB()
105 print()
106 print('-----File contents-----')
107 showScoreDB(scoredb, 'Name') #처음 데이터 파일의 상태를 파악하기위해...
108 doScoreDB(scoredb)
109 writeScoreDB(scoredb)
```



# Before

> > > > >

# After

수정 방향: 사용자가 좀 더 편하게(?) 이용할 수 있도록 수정함  
(진짜 사용자가 이 프로그램을 사용한다면 어떨까 라는 생각해봄)

# Before

> > > > >

# After

## Cord Review

(Before > After)

사용자 편의(커맨드 매뉴얼 변경, 프로그램 시작 시 깔끔한(?) 화면 )

```
78 def showScoreDB(scdb, keyname):
79     for p in sorted(scdb, key=lambda person: person[key
80         for attr in sorted(p):
81             print(attr + '=' + str(p[attr]), end=' ')
82         print()
83
84 def findScoreDB(pname):
85     for p in pname:
86         for attr in sorted(p):
87             print(attr + '=' + str(p[attr]), end=' ')
88         print()
89
90 def command_manual():
91     print('add: "add + Name + Age + Score"')
92     print('find: "find + Name"')
93     print('increase: "inc + Name + Amount"')
94     print('delete: "del + Name"')
95     print('show: "show + SoretingKey(Can be omitted)"')
96     print('exit: "quit OR exit"')
97
98
99     print('If you want a manual, type the following command')
100 scoredb = readScoreDB()
101 doScoreDB(scoredb)
102 writeScoreDB(scoredb)
```

```
79 def showScoreDB(scdb, keyname):
80     for p in sorted(scdb, key=lambda person: person[keyname]):
81         for attr in sorted(p):
82             print(attr + '=' + str(p[attr]), end=' ')
83         print()
84
85
86 def findScoreDB(pname):
87     for p in pname:
88         for attr in sorted(p):
89             print(attr + '=' + str(p[attr]), end=' ')
90         print()
91
92 def command_manual():
93     print('add: "add + [Name] + [Age] + [Score]"')
94     print('find: "find + [Name]"')
95     print('increase: "inc + [Name] + [Amount]"')
96     print('delete: "del + [Name]"')
97     print('show: "show"')
98     print('exit: "quit OR exit"')
99
100
101 #start!
102 print('-----Score Manager-----')
103 print('If you want a manual, type the following command: "command"')
104 scoredb = readScoreDB()
105 print()
106 print('-----File contents-----')
107 showScoreDB(scoredb, 'Name') #처음 데이터 파일의 상태를 파악하기위해...
108 doScoreDB(scoredb)
109 writeScoreDB(scoredb)
```

# Before

> > > > >

# After

## Cord Review

(Before > After)

예외처리 코드 수정('show' 커맨드, del 안내문 중복 출력 수정 등)

```
55 elif parse[0] == 'inc':
56     for p in scdb:
57         if p['Name'] == parse[1]:
58             p['Score'] += int(parse[2])
59             break
60     if not parse[1] in p['Name']: #예외
61         print(">"+parse[1]+"", 'is not name in list.')#
62         print('Please add to the list') #
63 elif parse[0] == 'del':
64     for p in scdb:
65         if p['Name'] == parse[1]:
66             scdb.remove(p)
67             print('delete', ""+parse[1]+"")
68     if not parse[1] in p['Name']: #예외
69         print(">"+parse[1]+"", 'is not name in list.')#
70 elif parse[0] == 'show':
71     sortKey = 'Name' if len(parse) == 1 else parse[1]
72     showScoreDB(scdb, sortKey)
73 elif (parse[0] == 'quit') or (parse[0] == 'exit'):
74     break
75 else:
76     print('Invalid command: ' + parse[0])
77
```

```
53 elif parse[0] == 'inc':
54     for p in scdb:
55         if p['Name'] == parse[1]:
56             p['Score'] += int(parse[2])
57             break
58     if not parse[1] in p['Name']: #예외(예러) 처리 부분
59         print(">"+parse[1]+"", 'is not name in list.')#
60         print('Please add to the list') #
61 elif parse[0] == 'del':
62     for p in scdb:
63         if p['Name'] == parse[1]:
64             scdb.remove(p)
65             print('delete', ""+parse[1]+"")
66     if not parse[1] in p['Name']: #예외(예러) 처리 부분
67         print(">"+parse[1]+"", 'is not name in list.')#
68 elif parse[0] == 'show':
69     if len(parse) >= 2:
70         print("Invalid command input")
71         continue
72     else:
73         sortKey = 'Name'
74         showScoreDB(scdb, sortKey)
75 elif (parse[0] == 'quit') or (parse[0] == 'exit'):
76     break
77 else:
78     print('Invalid command: ' + parse[0])
79
```

# 실행결과 화면

## Cord Review (Result)

```
-----Score-----  
If you want a manual, type the  
Open DB: assignment3.dat
```

```
-----File contents-----  
Age=18 Name=Choi Score=84  
Age=19 Name=Kim Score=98  
Age=18 Name=Lee Score=91  
Age=23 Name=Park Score=79  
Score DB >>>
```

```
-----Score Manager-----  
If you want a manual, type the following command: "command"  
Open DB: assignment3.dat
```

```
-----File contents-----  
Age=18 Name=Choi Score=84  
Age=19 Name=Kim Score=98  
Age=18 Name=Lee Score=91  
Age=23 Name=Park Score=79  
Score DB >>> command  
add: "add + [Name] + [Age] + [Score]"  
find: "find + [Name]"  
increase: "inc + [Name] + [Amount]"  
delete: "del + [Name]"  
show: "show"  
exit: "quit OR exit"  
Score DB >>> show  
Age=18 Name=Choi Score=84  
Age=19 Name=Kim Score=98  
Age=18 Name=Lee Score=91  
Age=23 Name=Park Score=79  
Score DB >>> find Kim  
Age=19 Name=Kim Score=98  
Score DB >>> inc Choi 2000  
Score DB >>> add Jo 20 0  
Score DB >>> show  
Age=18 Name=Choi Score=2084  
Age=20 Name=Jo Score=0  
Age=19 Name=Kim Score=98  
Age=18 Name=Lee Score=91  
Age=23 Name=Park Score=79  
Score DB >>> exit  
>>>
```

**감사합니다!**