

09.14 SoftwareProject2 – Group 3

20171656 유성현, 20171658 윤여환, 20171661 이다은, 20171662 이동윤

delete

```
#del 명령 수정
elif parse[0] == 'del':
    if len(parse) == 2:
        for i in scdb: #2중 for문으로 이름이 같은 모든 레코드 제거
            for p in scdb:
                if p['Name'] == parse[1]:
                    scdb.remove(p)

    else:
        print("이름을 정확히 입력해주세요.")
```

for 반복문을 1개 사용했더니 최대 두 개의 아이템만 지워져서 for 반복문을 2 개를 사용함.

```
#del 명령 추가
elif parse[0] == 'del':
    try:
        scdb_copy = scdb[:] #slice
        for p in scdb_copy:
            if p['Name'] == parse[1]: #p번째 딕셔너리의 Name의 value가 입력 값의 두번째 마디와 같으면
                scdb.remove(p) #p번째 딕셔너리를 삭제한다.
        #break
    except IndexError:
        continue
```

scdb list를 for 반복문을 돌려서 기존의 list의 내용을 수정하면 list에 변경이 있기 때문에 다음 for 반복문을 실행시킬 때 변화가 생긴다. 따라서 기존의 list를 복사해서 복사한 list를 수정하여 기존의 list에 타격이 없도록 한다.

전자의 방법이 더 좋은 것 같아서 전자의 방법을 채택했습니다.

Increase

```
# inc 명령 추가
elif parse[0] == 'inc':
    if len(parse) == 3:
        for p in scdb:
            if p['Name'] == parse[1]:
                try:
                    p['Score'] += int(parse[2]) # 입력한 값을 score 에 더해줌
                except ValueError: #정수가 아닌 값을 입력했을 때 예외처리
                    print("정수를 입력해주세요.")
    else:
        print("이름과 점수를 알맞게 입력해주세요.")
```

```

#inc 명령 추가
elif parse[0] == 'inc':
    try:
        amount = int(parse[2])
        for p in scdb:
            if p['Name'] == parse[1]:
                p['Score'] = int(p['Score'])
                p['Score'] = p['Score'] + amount
                # A += 1   ==   A = A + 1
                # 1 + 1 = 2 / '1' + '1' = 11
                # '1' + 1 = ? error
                # 스트링에서 int로 변환 후 연산하기.
            except IndexError:
                continue
    ---

```

이름 다음에 2번째로 입력 받는 인자를 amount로 설정
for 반복문을 사용하여 입력 받은 이름이 목록에 있는지 확인하고,
존재하면 string 타입의 score를 정수 타입으로 바꾸어 amount를 더해서 저장
try except를 사용하여 IndexError 일 때 예외 처리

Change to Integer

```

def readScoreDB():
    try:
        fh = open(dbfilename, 'rb')
    except FileNotFoundError as e:
        print("New DB: ", dbfilename)
        return []

    scdb = []
    try:
        scdb = pickle.load(fh)
        #print(scdb)
        for p in scdb:
            '''
            {'Age': '18', 'Name': 'Lee', 'Score': '91'}
            위는 예시이다.

            Age 와 Score는 Int값으로 처리해야함.

            현재 Age 의 값은 '18' -> 18
            '''
            p['Age'] = int(p['Age'])
            p['Score'] = int(p['Score'])
    except:
        print("Empty DB: ", dbfilename)
    else:
        print("Open DB: ", dbfilename)
    fh.close()
    return scdb

```

처음에 문서를 열 때 문서 안의 string 타입으로 저장되어있는 Age와 Score를 정수 타입으로 바꿔줌

ValueError, IndexError, KeyError

```
def doScoreDB(scdb):
    while (True):
        inputstr = (input("Score DB > "))
        if inputstr == "": continue
        parse = inputstr.split(" ")
        if parse[0] == 'add':
            if len(parse) == 4:
                try:
                    record = {'Name': parse[1], 'Age': int(parse[2]), 'Score': int(parse[3])}
                    scdb += [record]
                except ValueError: #정수가 아닌 다른 수를 입력했을 때 예외 처리
                    print("정수를 입력해주세요")
            else:
                print("이름, 나이, 점수를 알맞게 입력해주세요.")

    elif parse[0] == 'show':
        try:
            sortKey = 'Name' if len(parse) == 1 else parse[1]
            showScoreDB(scdb, sortKey)
        except KeyError: #다른 값을 넣었을 때 발생하는 에러 처리
            print("Name, Score, Age 중에 선택하여 입력해주세요.")

#find 명령 추가
elif parse[0] == 'find':
    try:
        for p in scdb:
            if p['Name'] == parse[1]:
                for i in sorted(p):
                    print(i + "=" + p[i], end=' ')
                print()
    except IndexError:
        continue
```

try expect 와 ValueError, IndexError, KeyError를 사용하여 다양한 입력 오류에 대한 예외 처리