

HW 1

1. 제목

자료구조를 사용한 학생 관리 시스템

2. 이름

김민성

3. 제출일

24.10.14

4. 과제 목표

자료구조를 잘 이용할 줄 알기

5. 코드 작성 과정

```
def main():
    student_scores = {}

    while True:
        print("\n학생 성적 관리 프로그램")
        print("1. 학생 추가")
        print("2. 성적 조회")
        print("3. 성적 수정")
        print("4. 전체 학생 목록 출력")
        print("5. 종료")

        choice = input("원하는 작업을 선택하세요 (1-5): ")

        if choice == '1':
            name = input("학생 이름을 입력하세요: ")
            score = input("학생 성적을 입력하세요: ")
            student_scores[name] = score
            print(f"{name}의 성적이 추가되었습니다.")
```

```

elif choice == '2':
    name = input("조회할 학생의 이름을 입력하세요: ")
    if name in student_scores:
        print(f"{name}의 성적: {student_scores[name]}")
    else:
        print("해당 학생이 존재하지 않습니다.")

elif choice == '3':
    name = input("수정할 학생의 이름을 입력하세요: ")
    if name in student_scores:
        new_score = input("새로운 성적을 입력하세요: ")
        student_scores[name] = new_score
        print(f"{name}의 성적이 수정되었습니다.")
    else:
        print("해당 학생이 존재하지 않습니다.")

elif choice == '4':
    print("\n전체 학생 목록:")
    for name, score in student_scores.items():
        print(f"{name}: {score}")

elif choice == '5':
    print("프로그램을 종료합니다.")
    break

else:
    print("잘못된 선택입니다. 다시 시도하세요.")

if __name__ == "__main__":
    main()

```

6. 코드 실행 결과

학생 성적을 입력하세요 : A +
김민성의 성적이 추가되었습니다 .

학생 성적 관리 프로그램

1. 학생 추가
2. 성적 조회
3. 성적 수정
4. 전체 학생 목록 출력
5. 종료

원하는 작업을 선택하세요 (1-5): 2

조회할 학생의 이름을 입력하세요 : 김민성

김민성의 성적 : A +

7. 문제 해결 과정 및 배운점

수업도중에 배운 자료구조를 어려운 프로그램에 적용시키는건 아무래도 조금은 어려웠다..그렇기에 나는 좋은 도구이자 친구인 GPT를 사용하여 진짜 좋은 프로그램을 만들었다.

Assignment 4-2

1. 제목

ChatGPT를 활용한 자료구조 선택과 최적화

2. 이름

김민성

3. 제출일

24.10.14 (월)

4. 과제 목표

Chat Gpt로 더 complex한 문제들을 푸는것

5. ChatGPT 활용

질문 내용

ChatGPT에게 했던 질문을 기록합니다. → 자료구조, 기능(어떤식으로 써야 제일 간편할까?)(이전 코드와 비슷한 형식으로 해줘.--> HW 1)

ChatGPT 답변 요약

저에게 코드를 알려주었습니다

6. 자료구조 선택 이유

딕셔너리: 선생님의 이름을 키로 사용하고, 해당 선생님이 담당하는 과목 목록(리스트)을 값으로 저장하여 빠르게 검색할 수 있습니다. 이 구조는 선생님 이름을 통해 직접 과목에 접근할 수 있는 장점이 있습니다.

리스트: 각 선생님이 여러 과목을 담당할 수 있도록 과목을 리스트로 저장했습니다. 리스트를 사용함으로써 과목의 추가 및 삭제가 용이합니다.

7. 코드 작성 과정

```
def main():
    # 선생님 이름과 담당 과목을 저장할 딕셔너리
    teacher_subjects = {}

    while True:
        print("\n선생님 관리 프로그램")
        print("1. 선생님 추가")
        print("2. 특정 과목 담당 선생님 조회")
        print("3. 선생님 과목 수정")
        print("4. 전체 선생님 목록 출력")
        print("5. 종료")

        choice = input("원하는 작업을 선택하세요 (1-5): ")

        if choice == '1':
            name = input("선생님 이름을 입력하세요: ")
            subjects = input("담당 과목을 입력하세요 (쉼표로 구분): ").split(',')
            subjects = [subject.strip() for subject in subjects] # 과목 공백 제거
            teacher_subjects[name] = subjects
            print(f"{name} 선생님의 과목이 추가되었습니다.")

        elif choice == '2':
            subject = input("조회할 과목을 입력하세요: ")
            teachers = [name for name, subjects in teacher_subjects.items() if subject
in subjects]
            if teachers:
                print(f"{subject} 담당 선생님: {' '.join(teachers)}")
            else:
                print("해당 과목을 담당하는 선생님이 없습니다.")

        elif choice == '3':
            name = input("과목을 수정할 선생님의 이름을 입력하세요: ")
            if name in teacher_subjects:
```

```

        new_subjects = input("새로운 담당 과목을 입력하세요 (쉼표로 구분): ")
        new_subjects = new_subjects.split(',')
        new_subjects = [subject.strip() for subject in new_subjects]
        teacher_subjects[name] = new_subjects
        print(f"{name} 선생님의 과목이 수정되었습니다.")
    else:
        print("해당 선생님이 존재하지 않습니다.")

    elif choice == '4':
        print("\n전체 선생님 목록:")
        for name, subjects in teacher_subjects.items():
            print(f"{name}: {' '.join(subjects)}")

    elif choice == '5':
        print("프로그램을 종료합니다.")
        break

    else:
        print("잘못된 선택입니다. 다시 시도하세요.")

if __name__ == "__main__":
    main()

```

8. 코드 실행 결과

선생님 관리 프로그램

1. 선생님 추가
2. 특정 과목 담당 선생님 조회
3. 선생님 과목 수정
4. 전체 선생님 목록 출력
5. 종료

원하는 작업을 선택하세요 (1-5): 4

전체 선생님 목록 :

김민성 : 바이오

물짱 : 포켓몬

9. 문제 해결 과정 및 배운점

ChatGPT와의 상호작용을 통해 얻은것은 참 많다..일단 코드에 대해서 이해도를 높여주었고 정말 모든걸 알려주어서 고맙다.