

C프로그래밍및실습

대학생활 도우미

프로그램

진척 보고서 #1

제출일자: 2023-11-26

제출자명: 최지원

제출자학번: 224571

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

학교에 막 입학한 학생이나 복학한 학생들은 학교 생활에 적응하기 힘들 수도 있다. 수업 장소는 어딘지 정확한 수업 시간은 언제인지 쉬는 시간은 몇 분인지 정확하게 파악하기 힘든 경우가 많다. 또 학점을 관리하고 과제의 마감기한도 관리하기가 힘들다. 이 모든 걸 관리해주는 프로그램이 있으면 학교 생활에 더 도움이 될 것이라고 생각한다.

2) 프로젝트 목표

수업 장소와 시간, 교수님을 알려주고 목표학점을 입력하면 그에 맞게 학점을 분석해주고 과제의 마감기한을 관리할 수 있는 학교생활 도우미 프로그램

3) 차별점

기존에 있던 일반적인 대학생 앱과 같은 서비스와는 개인의 학교생활 관리에 더 중심적인 프로그램임. 더 직관적이게 원하는 항목을 선택 가능하게 하여 사용자가 더 쉽게 프로그램을 이용할 수 있음. 과제 마감기한과 학점 분석과 같은 기능을 추가하여 기존에 앱과는 차별화된 기능이 있음.

2. 기능 계획

1) 전체 틀 구성

프로그램의 전체 흐름과 틀을 구성

2) 시간표 정리

- 현재 시간표를 정리하는 기능

(1) 사용자가 수업 시간, 장소, 교수님을 입력함

(2) 사용자 입력 후 현재 시간표를 정리해서 보여준다

- 입력이 끝난 후 수업 시간, 수업 장소, 교수님 정보를 확인 가능함

(3) 시간표의 수업 수정, 삭제, 추가 기능

3) 성적 분석

- 목표 성적과 현재 각 과목의 성적을 입력하면 목표성적까지의 성적을 분석

(1) 목표 성적 입력

(2) 사용자의 과목별 성적 입력

(3) 사용자의 성적 평균 분석

(4) 목표 성적까지 얼마나 성적을 올려야 하는지 알려줌

4) 과제 관리&디데이 기능

- 과제의 마감 디데이와 과제 목록을 확인하는 기능

(1) 과제 추가, 수정, 삭제 기능

(2) 현재 컴퓨터 시간을 기반으로 디데이 계산

- 현재 컴퓨터 시간과 입력한 시간정보를 바탕으로 디데이를 계산함

(3) 현재 해야 될 과제 목록 나열

- 디데이가 끝난 과제들은 자동으로 삭제됨

3. 진척사항

1) 기능 구현

(1) 전체 틀 구성

- 입출력

- 입력: 원하는 메뉴의 번호
- 출력: 대학생활 도우미 프로그램 전체 메뉴 번호

- 설명: 대학생활 도우미 프로그램을 소개하고 원하는 번호를 사용자가 선택할 수 있게 함. 프로그램은 사용자가 종료하기 전까지 계속 돌아 갈 수 있도록 함.

- 적용된 배운 내용: if조건문, while문

- 코드 스크린샷

```
while (1) {
    printf(_Format: "₩₩₩₩대학생활 도우미 프로그램입니다.₩₩ 무엇을 도와드릴까요?₩₩");
    printf(_Format: "1. 시간표 관리₩₩2. 성적 관리₩₩3. 과제 관리₩₩4. 프로그램 종료₩₩");
    printf(_Format: "-----₩₩");
    printf(_Format: "원하는 번호를 입력하세요.₩₩");

    int num = 0;
    scanf_s(_Format: "%d", &num);

    if (num == 1) {
        if (!timetableCreated) {
            // 시간표를 입력받는 부분
        }
    }
}
```

(2)-1. 사용자의 시간표 입력

- 입출력

- 입력: 수업 과목명, 교수명, 수업 장소와 총 듣는 수업의 개수
- 출력: 수업 시간표

- 설명: 사용자의 수업 시간표를 입력하고 입력한 시간표를 보여주는 기능

- 적용된 배운 내용: if문, 포인터, 동적 배열, 함수

- 코드 스크린샷

```

if (num == 1) {
    if (!timetableCreated) {
        // 시간표를 입력받는 부분
        int grade = 0;
        int semester = 0;

        printf(_Format: "시간표 관리 메뉴입니다.\n");
        printf(_Format: "몇학년 몇학기 수업인가요? (띄어쓰기로 구분해서 학년, 학기 작성)\n");
        scanf_s(_Format: "%d %d", &grade, &semester);
        printf(_Format: "총 몇개의 수업을 들으시나요?");
        scanf_s(_Format: "%d", &hakjum);

        timetable = (struct Course*)malloc(_Size:hakjum * sizeof(struct Course));

        if (timetable == NULL) {
            printf(_Format: "메모리 할당 오류\n");
            return -1;
        }

        printf(_Format: "듣는 수업의 수업명, 교수명, 장소명을 차례대로 입력하세요. (띄어쓰기로 구분)\n");

        for (int i = 0; i < hakjum; i++) {
            scanf_s(_Format: "%49s %49s %49s", timetable[i].className, MAX_LENGTH, timetable[i].professorName, MAX_LENGTH, timetable[i].classPlace, MAX_LENGTH);
        }

        timetableCreated = 1;
    }

    // 시간표 출력
    displayTimetable(timetable, hakjum);
}

```

● 시간표를 보여주는 함수

```

void displayTimetable(struct Course* timetable, int hakjum) {
    printf(_Format: "\n시간표:\n-----\n");
    for (int i = 0; i < hakjum; i++) {
        printf(_Format: "%s | %s | %s\n", timetable[i].className, timetable[i].professorName, timetable[i].classPlace);
    }
}

```

(2)-2. 시간표 입력 후 수업 추가 기능

- 입출력

- 입력: 수업 추가
- 출력: 수업이 추가 됐다는 메시지

- 설명: 처음에 시간표를 입력하고 그 이후에 수업을 추가할 수 있는 기능

- 적용된 배운 내용: 배열, if문

- 코드 스크린샷

```

if (option == 1) {
    // 수업 추가
    printf("추가할 수업의 수업명, 교수명, 장소명을 차례대로 입력하세요. (띄어쓰기로 구분)\n");
    scanf_s("%s %s %s", timetable[hakjum].className, MAX_LENGTH, timetable[hakjum].professorName, MAX_LENGTH, timetable[hakjum].classPlace, MAX_LENGTH);
    hakjum++;
    printf("수업이 추가되었습니다.\n");
}

```

(2)-3. 시간표 입력 후 수업 삭제 기능

- 입출력

- 입력: 수업 삭제
 - 출력: 수업이 삭제됐다는 메시지
- 설명: 처음에 시간표를 입력하고 그 이후에 수업을 삭제할 수 있는 기능
- 적용된 배운 내용: 반복문, 배열, if문
- 코드 스크린샷

```

else if (option == 2) {
    // 수업 삭제
    if (hakjum > 0) {
        printf("삭제할 수업의 번호를 입력하세요 (1부터 시작)\n");
        int courseNumber = 0;
        scanf_s("%d", &courseNumber);

        if (courseNumber >= 1 && courseNumber <= hakjum) {
            for (int i = courseNumber - 1; i < hakjum - 1; i++) {
                timetable[i] = timetable[i + 1];
            }
            hakjum--;
            printf("수업이 삭제되었습니다.\n");
        }
        else {
            printf("잘못된 번호입니다.\n");
        }
    }
    else {
        printf("현재 수업이 없어 삭제할 수 없습니다.\n");
    }
}

```

(3)-1. 성적관리 프로그램

- 입출력
- 입력: 성적 관리 프로그램에서 원하는 항목 번호
 - 출력: 사용자가 입력한 항목 번호에 따른 기능
- 설명: 성적관리 프로그램의 전체적인 틀, 목표 성적을 기준에 입력 했다면 목표 성적이 처음에 뜨도록 함. 사용자는 성적관리 프로그램에서 원하는 항목을 선택할 수 있고 볼 수 있음.

- 적용된 배운 내용: if문, 함수, while문, 포인터
- 코드 스크린샷

```
else if (num == 2) {
    //성적 관리 프로그램 작성
    displayTimetable(timetable, hakjum);

    if (*goalGrade != 0.0) {
        printf("당신의 목표 성적은 %.2lf 입니다.\n\n", *goalGrade);
    }
    else {
        printf("아직 목표 성적이 입력되지 않았습니다.\n1번을 눌러 목표 성적을 입력하세요.\n");
    }

    int setGrade = 0;
    int option = 0;
    while (1) {
        printf("\n\n성적관리 프로그램\n-----\n");
        printf("1. 목표성적 입력 및 수정\n2. 각 수업 별 성적 입력 및 수정\n3. 성적분석\n4. 이전으로\n");
        scanf_s("%d", &option);
    }
}
```

(3)-2. 목표 성적 입력 및 수정

- 입출력
 - 입력: 사용자의 목표 성적
 - 출력: 사용자가 입력한 목표 성적
- 사용자가 목표 성적을 입력하고 수정할 수 있는 기능
- 적용된 배운 내용: if문, 포인터
- 코드 스크린샷

```
//목표 성적 입력 및 수정
if (option == 1) {
    if (!setGrade) { //목표 성적이 입력되어 있지 않은 경우
        printf("목표 성적을 입력하세요: ");
        scanf_s("%lf", goalGrade);
        printf("당신의 목표 성적은 %.2lf 입니다.", *goalGrade);
        setGrade = 1;
    }
    else { //목표 성적 수정
        printf("수정할 목표 성적을 입력하세요.\n");
        scanf_s("%lf", goalGrade);
        printf("수정한 당신의 목표 성적은 %.2lf입니다.", *goalGrade);
    }
}
```

(3)-3. 과목 별 성적 입력

- 입출력

- 입력: 사용자의 목표 성적
- 출력: 사용자가 입력한 목표 성적

- 사용자가 수강하고 있는 과목 별 성적을 입력할 수 있는 기능. 현재는 실행 시 모든 과목의 성적이 수정되므로 수정할 과목을 선택할 수 있도록 수정 할 필요가 있음.

- 적용된 배운 내용: if문, 포인터, 함수, 반복문, 배열

- 코드 스크린샷

```
//각 과목 별 목표 성적 입력 및 수정
else if (option == 2) {
    for (int i = 0; i < hakjum; i++) {
        printf("%s 과목의 성적을 입력하세요: \n", timetable[i].className);
        scanf_s("%lf", &timetable[i].grade);
    }
    displayGradeTimetable(timetable, hakjum);
}
```

```
void displayGradeTimetable(struct Course* timetable, int hakjum) {
    printf("\n수업 별 성적:\n-----\n");
    for (int i = 0; i < hakjum; i++) {
        printf("%s | 성적: %.2lf\n", timetable[i].className, timetable[i].grade);
    }
}
```

(3)-4. 목표 성적과 현재 성적 비교

- 입출력

- 입력: 사용자가 기존에 입력했던 목표 성적과 과목 별 성적
- 출력: 목표 성적과 사용자의 성적 비교

- 목표 성적과 현재 사용자의 성적을 비교하여 알려준다. 지금은 현재 성적이 입력되지 않은 경우에 대해서만 조건문을 걸어놓았는데 목표 성적이 입력되지 않은 경우 목표 성적이 입력되지 않았다는 문구를 표시할 예정이다.

- 적용된 배운 내용: if문, 포인터, 배열

- 코드 스크린샷

```
//목표 성적과 현재 성적 비교
else if (option == 3) {
    printf("목표 성적: %.2lf\n", *goalGrade);
    double currentGrade = 0.0;
    double sum = 0.0;
    for (int i = 0; i < hakjum; i++) {
        sum += timetable[i].grade;
    }
    currentGrade = sum / hakjum;
    if (sum != 0.0) {
        printf("현재 당신의 성적: %.2lf\n", currentGrade);
        if (*goalGrade > currentGrade) {
            printf("목표 성적까지 %.2lf만큼 남았습니다.\n", *goalGrade - currentGrade);
            printf("조금 더 열심히 공부하세요!\n");
        }
        else if (*goalGrade < currentGrade) {
            printf("목표 성적보다 %.2lf만큼 높습니다.\n", currentGrade - *goalGrade);
            printf("멋집니다!\n");
        }
        else {
            printf("목표성적과 현재 성적이 같습니다.\n");
        }
    }
    else
    {
        printf("현재 성적을 입력해주세요.\n");
    }
}
```

(4). 과목 별 과제 입력

- 입출력

- 입력: 과목 별 과제 입력
- 출력: 없음

- 과목 별 목표 성적을 입력할 수 있는 기능. 아직 전체적인 틀만 짜놓았습니다.

- 적용된 배운 내용: if문, 배열

- 코드 스크린샷

```
//과제 관리 프로그램 작성
else if (num == 3) {
    int index = 0;
    if (hakjum==0) {
        printf("아직 시간표가 입력되지 않았습니다.\n시간표를 먼저 입력해주세요.\n");
    }
    printf("\n과제 관리 프로그램입니다.\n");
    printf("-----\n");
    printf("과제를 입력 할 과목을 선택하세요.\n");
    for (int i = 0; i < hakjum; i++) {
        printf("%d. %s\n", i + 1, timetable[i].className);
    }
    scanf_s("%d",&index);
}
```

2) 테스트 결과

(1)-1. 시간표 입력 및 출력

- 사용자가 시간표를 입력하고 입력한 시간표를 보여준다
- 테스트 결과 스크린샷

```
대학생활 도우미 프로그램입니다.
무엇을 도와드릴까요?
1. 시간표 관리
2. 성적 관리
3. 과제 관리
4. 프로그램 종료
-----
원하는 번호를 입력하세요.
1
시간표 관리 메뉴입니다.
몇학년 몇학기 수업인가요? (띄어쓰기로 구분해서 학년,학기 작성)
2 2
총 몇개의 수업을 들으시나요?3
듣는 수업의 수업명, 교수명, 장소명을 차례대로 입력하세요. (띄어쓰기로 구분)
c언어 김미수 용대
c++ 박태준 공6
알고리즘 임형석 공7

시간표:
-----
c언어 | 김미수 | 용대
c++ | 박태준 | 공6
알고리즘 | 임형석 | 공7

1. 수업 추가
2. 수업 삭제
3. 이전으로
원하는 번호를 입력하세요.
```

(1)-2. 수업 추가

- 수업을 추가할 수 있음. 추가한 수업은 시간표에 반영되어서 보여줌.
- 테스트 결과 스크린샷

```
원하는 번호를 입력하세요.
1
추가할 수업의 수업명, 교수명, 장소명을 차례대로 입력하세요. (띄어쓰기로 구분)
역문자 조상현 용대
수업이 추가되었습니다.

대학생활 도우미 프로그램입니다.
무엇을 도와드릴까요?
1. 시간표 관리
2. 성적 관리
3. 과제 관리
4. 프로그램 종료
-----
원하는 번호를 입력하세요.
1

시간표:
-----
c언어 | 김미수 | 용대
c++ | 박태준 | 공6
알고리즘 | 임형석 | 공7
역문자 | 조상현 | 용대

1. 수업 추가
2. 수업 삭제
3. 이전으로
원하는 번호를 입력하세요.
```

(1)-3. 수업 삭제

- 수업을 삭제할 수 있음. 삭제한 수업은 시간표에 반영되어서 보여줌.
- 테스트 결과 스크린샷

```
1. 수업 추가
2. 수업 삭제
3. 이전으로
원하는 번호를 입력하세요.
2
삭제할 수업의 번호를 입력하세요 (1부터 시작)
4
수업이 삭제되었습니다.
```

```
대학생활 도우미 프로그램입니다.
무엇을 도와드릴까요?
```

```
1. 시간표 관리
2. 성적 관리
3. 과제 관리
4. 프로그램 종료
```

```
-----
원하는 번호를 입력하세요.
```

```
1
```

```
시간표:
```

```
-----
c언어 | 김미수 | 용대
c++ | 박태준 | 공6
알고리즘 | 임형석 | 공7
```

```
1. 수업 추가
2. 수업 삭제
```

(2)-1. 목표 성적 입력

- 목표 성적 입력

- 테스트 결과 스크린샷

```
아직 목표 성적이 입력되지 않았습니다.
1번을 눌러 목표 성적을 입력하세요.
```

```
성적관리 프로그램
```

```
-----
1. 목표성적 입력 및 수정
2. 각 수업 별 성적 입력 및 수정
3. 성적분석
4. 이전으로
```

```
1
```

```
목표 성적을 입력하세요: 4.0
당신의 목표 성적은 4.00 입니다.
```

```
성적관리 프로그램
```

```
-----
1. 목표성적 입력 및 수정
2. 각 수업 별 성적 입력 및 수정
3. 성적분석
4. 이전으로
```

(2)-2. 목표 성적 수정

- 목표 성적 수정
- 테스트 결과 스크린샷

성적관리 프로그램

```
-----
1. 목표성적 입력 및 수정
2. 각 수업 별 성적 입력 및 수정
3. 성적분석
4. 이전으로
1
수정할 목표 성적을 입력하세요.
4.5
수정한 당신의 목표 성적은 4.50입니다.
```

(2)-3. 각 수업 별 성적 입력

- 각 수업 별 성적을 사용자가 입력하고 보여줌
- 테스트 결과 스크린샷

성적관리 프로그램

```
-----
1. 목표성적 입력 및 수정
2. 각 수업 별 성적 입력 및 수정
3. 성적분석
4. 이전으로
2
c언어 과목의 성적을 입력하세요:
4.0
c++ 과목의 성적을 입력하세요:
3.5
알고리즘 과목의 성적을 입력하세요:
3.0

수업 별 성적:
-----
c언어 | 성적: 4.00
c++ | 성적: 3.50
알고리즘 | 성적: 3.00
```

(2)-4. 목표 성적과 현재 성적 분석

- 사용자가 입력한 과목 별 성적을 평균 내어 목표 성적과 비교한 결과값을 보여줌
- 테스트 결과 스크린샷

```
성적관리 프로그램
-----
1. 목표성적 입력 및 수정
2. 각 수업 별 성적 입력 및 수정
3. 성적분석
4. 이전으로
3
목표 성적: 4.50
현재 당신의 성적: 3.50
목표 성적까지 1.00만큼 남았습니다.
조금 더 열심히 공부하세요!
```

(3) 프로그램 종료

- 프로그램 종료
- 테스트 결과 스크린샷

```
대학생활 도우미 프로그램입니다.
무엇을 도와드릴까요?
1. 시간표 관리
2. 성적 관리
3. 과제 관리
4. 프로그램 종료
-----
원하는 번호를 입력하세요.
4
프로그램을 종료합니다.
```

4. 계획 대비 변경 사항

1) 성적 분석

- 이전: 기존에는 사용자의 성적을 분석하여 어떤 과목에서 점수를 더 올리면 목표 성적을 받을 수 있을지 알려주려고 하였음
- 이후: 목표 성적까지의 차이만 알려줌

- 사유: 구현 할 실력 부족.. 하지만 공부를 더 해서 구현할 수 있다면 최종 프로젝트 제출 전 까지 구현하도록 하겠음.

5. 프로젝트 일정

업무		11/3	11/10	11/17	11/26	12/1	12/8	12/15
제안서 작성		진행완료						
기능1			진행완료					
시간표 정리	시간표 입력			진행 완료				
	수업 추가			진행완료				
	수업 삭제			진행완료				
성적 관리	목표 성적				진행 완료			
	과목별 성적				진행중	----->		
	성적 분석				진행중	----->		
과제 정리	과목별 과제					----->		
	과제 디데이						----->	
	과제 목록						----->	