

C프로그래밍및실습

멘토링 자동 관리 프로그램

진척 보고서 #1

제출일자: 2023.11.26

제출자명: 김성빈

제출자학번: 205224

1. 프로젝트 목표

1) 배경 및 필요성

현재 비대면으로 중학교 3학년 학생들에게 영어를 가르쳐 주는 EBS 멘토링 활동을 하고 있다. 비대면으로 진행되는 프로그램이다 보니 학생들과 메시지를 이용해 소통을 하는 경우가 많다. 보통 메시지를 보내는 상황과 멘트가 비슷한데, 매번 메시지를 직접 보내다 보니 불편함을 느낀다. 그래서 멘티에게 보낼 메시지를 미리 저장해 출력해 주는 프로그램이 필요하다고 생각하게 되었다. 또 멘토링이 원활하게 진행되도록 돕고, 멘티에게 도움이 될 수 있는 프로그램을 만들어보고자 한다.

2) 프로젝트 목표

멘티에게 보낼 상황에 맞는 메시지를 멘티의 이름과 멘티의 수업 시간을 넣어서 바로 전송할 수 있도록 미리 저장된 메시지를 출력하는 프로그램을 만들고, 멘티의 수업 기록을 저장한 파일을 불러와서 멘티 별로 현재까지 진행된 수업의 횟수와 결석 횟수 출력과 급여 출력을 목표로 한다.

3) 차별점

기존 프로그램은 현재 진행하는 EBS 멘토링 프로그램을 기반으로 두고 구현한 프로그램이 아니다. 또 보통은 멘티만의 입장에서 멘티에게 도움이 되는 프로그램을 많이 구현하는데, 이 프로그램은 멘토의 편의도 고려하는 프로그램이라는 점에서 차별점이 있음.

2. 기능 계획

1) 멘티에게 보낼 메시지 출력

- 멘토링 1시간 전과 지각을 한 경우 그리고 수업이 끝난 경우에 매번 멘티에게 메시지를 보내야 하는데 매번 같은 문구를 보내는 불편을 줄이기 위해 구상

(1) 멘티의 이름과 수업 시간 정보를 미리 저장

(2) 조건문을 사용해 상황을 고르고 메시지 출력

- 멘티에게 수업 1시간 전 보낼 수업 안내 메시지를 출력한다.
- 멘티가 지각한 경우의 메시지를 출력해 준다.
- 수업이 끝난 경우의 마무리 멘트를 출력해 준다.

2) 수업 진행 현황 출력

- 멘티의 수업 기록 파일에서 총 수업 횟수와 남은 횟수, 멘티의 결석 횟수를 출력한다.

(1) 수업 기록 파일 불러오기

- 파일 입출력 기능을 사용해 멘티의 수업 기록이 저장된 파일을 불러온다.

(2) 수업 기록 횟수 계산 및 급여 계산

- 진행 시간을 기반으로 수업 시간을 계산해 멘티의 수업 횟수와 결석 횟수를 출력하고, 총 급여를 계산해 함께 출력한다.

3) 단어장 프로그램

- 멘티가 학습한 단어를 복습할 수 있는 단어장 프로그램 구현

(1) 학습할 단어 불러오기

- 미리 단어가 저장된 파일을 불러오거나 직접 입력하여 학습할 단어를 정한다.

(2) 단어 맞추기 퀴즈를 통해 단어 학습 현황 확인

- 문자열 함수 strcmp를 사용해 멘티가 입력한 단어와 저장된 단어가 일치하는지 확인

4) 비밀번호 확인

- 멘티가 멘토가 사용하는 기능에 접근할 수 없도록 비밀번호 기능 구현

(1) 입력한 문자열이 비밀번호와 일치하는지 확인

- 사용자로부터 입력 받은 문자열이 미리 저장된 비밀번호와 일치하는지 확인해 일치하면 다음 기능으로 넘어가고, 일치하지 않으면 비밀번호 재입력

(2) 3회 이상 틀릴 시 프로그램 종료

- 입력한 문자열과 비밀번호가 3회 일치하지 않을 경우 프로그램을 종료하고, 3회가 되기 전에 맞추면 다음 기능으로 넘어가기

3. 진척사항

1) 기능 구현

(1) 멘티에게 보낼 메시지 출력

- 미리 저장된 2차원 배열에서 멘티의 이름과 수업 시간을 선택하기 위해 숫자를 입력 받고, 메시지를 보낼 상황을 선택하기 위해 마찬가지로 숫자를 입력 받아 미리 저장된 메시지를 출력
- 멘티의 이름과 메시지를 보낼 상황을 선택하고 각 상황에 맞는 메시지 출력
- 2차원 배열, 조건문, 함수
- 코드 스크린샷

```

int main() {
    int terminate = 0;
    int tryCount = 0;
    char studentNames[STUDENTS][CHARNUM] = {"가가", "나나", "다다", "라라"};
    char studentTimes[STUDENTS][CHARNUM] = {"6시", "7시 30분", "8시 30분", "9시"};

    printf("멘토링 자동 관리 프로그램\n");
    printf("-----\n");
    printf("메뉴를 선택해주세요.\n");
    printf("1. 수업 메시지 출력\n");
    printf("2. 수업 진행 현황\n");
    printf("3. 단어장\n");
    printf("4. 프로그램 종료\n");
    printf("-----\n");

    while (1) {
        int choiceA = -1;
        scanf_s("%d", &choiceA);

        switch (choiceA) {
            case 1:
                if (getPassword() == 1) {
                    return 0;
                }
                else {
                    int numName = -1;

                    printf("\n1: %s 2: %s 3: %s 4: %s\n",
                        studentNames[0], studentNames[1], studentNames[2], studentNames[3]);
                    printf("멘티의 이름을 선택해주세요 : ");
                    scanf_s("%d", &numName);

                    getMessage(numName);
                }
                break;

            void getMessage(int numName) {
                int situation;
                char studentNames[STUDENTS][CHARNUM] = { "가가", "나나", "다다", "라라" };
                char studentTimes[STUDENTS][CHARNUM] = { "6시", "7시 30분", "8시 30분", "9시" };

                printf("상황을 선택해주세요.\n");
                printf("1. 수업 시작 전\n");
                printf("2. 멘티의 지각\n");
                printf("3. 수업 끝\n");
                scanf_s("%d", &situation);

                if (situation == 1) {
                    printf("%s 멘티님, 잠시 후 오후 %s에 멘토링이 진행될 예정입니다~! 교재와 함께 시간 맞춰 접속해주세요 :) \n",
                        studentNames[numName - 1], studentTimes[numName - 1]);
                }
                else if (situation == 2) {
                    printf("%s 멘티님, 수업 시작 후 10분이 경과되었습니다. 수업 참여에 어려움이 있을까요? \n",
                        studentNames[numName - 1]);
                }
                else if (situation == 3) {
                    printf("%s 멘티님, 오늘 수업도 수고하셨습니다! 다음 수업 시간에 다시 봐요 :) \n",
                        studentNames[numName - 1]);
                }
            }
        }
    }
}

```

(2) 비밀번호 확인

- 문자열을 입력 받고 비밀번호의 일치 여부를 판단해 주는 문자열 출력
- 3회 이상 틀리면 프로그램이 종료되고, 3회 이내에 비밀번호를 맞추면 다음 기능에 접근 가능

- 반복문, 조건문, 함수, 문자열 함수

- 코드 스크린샷

```
        case 1:
            if (getPassword() == 1) {
                return 0;
            }
            else {
                int numName = -1;

                printf("#n1: %s 2: %s 3: %s 4: %s\n",
                    studentNames[0], studentNames[1], studentNames[2], studentNames[3]);
                printf("멘티의 이름을 선택해주세요 : ");
                scanf_s("%d", &numName);

                getMessage(numName);
            }
            break;

int getPassword() {
    int tryCount = 0;

    while (1) {
        printf("비밀번호를 입력하세요: ");
        scanf_s("%s", password, (int)sizeof(password));

        if (strcmp(password, "mentoring") == 0) {
            printf("맞았습니다!\n");
            return 0;
        }
        else {
            tryCount++;
            printf("틀렸습니다. 다시 입력해주세요. 기회는 %d번 남았습니다.\n", 3 - tryCount);

            if (tryCount >= 3) {
                printf("비밀번호 3회 이상 실패. 프로그램을 종료합니다.\n");
                return 1;
            }

            // 입력 버퍼 비우기
            int c;
            while ((c = getchar()) != '\n' && c != EOF);
        }
    }
}
```

2) 테스트 결과

(1) 멘티에게 보낼 메시지 출력

- 멘티의 이름과 메시지를 보낼 상황을 선택하고 각 상황에 맞는 메시지 출력

- 테스트 결과 스크린샷

```

멘토링 자동 관리 프로그램
-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어장
4. 프로그램 종료
-----
1
비밀번호를 입력하세요: mentoring
맞았습니다!

1: 가가 2: 나나 3: 다다 4: 라라
멘티의 이름을 선택해주세요 : 3
상황을 선택해주세요.
1. 수업 시작 전
2. 멘티의 지각
3. 수업 끝
1
다다 멘티님, 잠시 후 오후 8시 30분에 멘토링이 진행될 예정입니다~! 교재와 함께 시간 맞춰 접속해주세요 :)

```

(2) 비밀번호 확인

- 입력한 문자열과 미리 저장된 비밀번호를 비교해 3회 이상 틀리면 프로그램이 종료되고, 3회 이내에 비밀번호를 맞추면 다음 기능에 접근 가능

- 테스트 결과 스크린샷

① 비밀번호를 바로 맞춘 경우

```

멘토링 자동 관리 프로그램
-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어장
4. 프로그램 종료
-----
2
비밀번호를 입력하세요: mentoring
맞았습니다!
수업 진행 현황 출력 기능 구현 예정

```

② 비밀번호를 3회 이내에 맞춘 경우

멘토링 자동 관리 프로그램

메뉴를 선택해주세요.

1. 수업 메시지 출력
 2. 수업 진행 현황
 3. 단어장
 4. 프로그램 종료
-

2

비밀번호를 입력하세요: aa
틀렸습니다. 다시 입력해주세요. 기회는 2번 남았습니다.
비밀번호를 입력하세요: bb
틀렸습니다. 다시 입력해주세요. 기회는 1번 남았습니다.
비밀번호를 입력하세요: mentoring
맞았습니다!
수업 진행 현황 출력 기능 구현 예정

③ 비밀번호를 3회 이상 틀릴 경우

멘토링 자동 관리 프로그램

메뉴를 선택해주세요.

1. 수업 메시지 출력
 2. 수업 진행 현황
 3. 단어장
 4. 프로그램 종료
-

2

비밀번호를 입력하세요: a
틀렸습니다. 다시 입력해주세요. 기회는 2번 남았습니다.
비밀번호를 입력하세요: b
틀렸습니다. 다시 입력해주세요. 기회는 1번 남았습니다.
비밀번호를 입력하세요: c
틀렸습니다. 다시 입력해주세요. 기회는 0번 남았습니다.
비밀번호 3회 이상 실패. 프로그램을 종료합니다.

C:\Users\USER\Desktop\Test\x64\Debug\Test.exe(프로세스 31660개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

4. 계획 대비 변경 사항

1) 비밀번호 확인

- 이전 : 비밀번호를 확인하는 과정 없이 바로 프로그램의 기능 실행
- 이후 : 멘토가 사용하는 기능은 비밀번호를 확인하는 과정 추가
- 사유 :

수업 메시지 출력이나 수업 현황은 멘토가 사용하는 기능이고, 단어장 출력은 멘티가 사용하는 기능으로 멘티와 멘토가 함께 사용할 수 있는 프로그램이다.

따라서 멘토가 사용하는 기능에는 멘티가 접근할 수 없도록 비밀번호 기능 구현이 필요하다고 판단하였다.

2) 프로젝트 일정 변경

- 이전 : 기능 2와 기능 3을 11/24일까지 완료
- 이후 : 기능 2와 기능 3을 다음 중간 구현물 과제 제출일인 12/10일까지 완료
- 사유 : 파일 입출력이 사용되는 기능이므로 해당 범위까지 수업을 듣고 난 후 기능을 구현할 수 있을 것이라고 판단하였다.

5. 프로젝트 일정

| 업무 | 11/3 | 11/10 | 11/17 | 11/26 | 12/10 | 12/15 | 12/23 |
|-----------|------|-------|-------|--------|--------|--------|-------|
| 제안서 작성 | 완료 | | | | | | |
| 메시지 출력 기능 | | 완료 | | | | | |
| 비밀번호 기능 | | | 완료 | | | | |
| 진척 보고서 #1 | | | 완료 | | | | |
| 수업 현황 기능 | | | | -----> | | | |
| 단어장 기능 | | | | -----> | | | |
| 진척 보고서 #2 | | | | -----> | | | |
| 추가 기능 고안 | | | | | -----> | | |
| 최종 결과물 제출 | | | | | | -----> | |