C프로그래밍및실습

도서관 관리 시스템

진척 보고서 #1

제출일자: 2023-11-26
제출자명: 송재호
제출자학번: 185132

1. 프로젝트 목표

1) 배경 및 필요성

도서관은 도서의 정보를 확인, 대출, 등의 여러가지 기능을 함, 하지만 서적의 양이 많아짐에 따라, 사람이 직접 관리하는 것에 한계가 있음 이에 자동화 시스템이 필요하다고 생각한다. 고객이 원하는 분야의 서적들을 보여주고 각각의 서적을 선택한다면, 서적의 기본적인 정보들을 출력해준다면, 고객들은 직접 도서관에 방문하지 않아도 원하는 서적을 도서관에서 보유중인지 확인하거나, 원하는 분류만 선택을 한다면 서적의 목록을 보고 선택을 할 수 있을 것 이다.

2) 프로젝트 목표

많은 양의 서적들을 분야별로 분류하여 저장, 고객이 원하는 분야의 책 목록을 보여주고, 각 분야별에서 원하는 서적을 선택하면 서적의 정보를 출력, 고객이 서적을 대출한다면, 반납 예정일을 통보, 반납일의 일정 기간 전에 반납 요청 알림을 보낸다.

3) 차별점

기존 도서관 관리 시스템은 서적의 정보를 간략하게 보여주고 대출시 반납예정일만을 알려주었지만, 반납예정일의 일정 기간 전에 알림이 감.

2. 기능 계획

1) 분야 목록 출력 기능

- 설명: 원하는 분야를 선택하거나 입력한다면 같은 분야의 서적들의 목록을 출력하여 보여준다. 이러한 목록들은 아직 원하는 서적을 생각해 놓지 못한 고객들에게 서적 추천 기능의 역할을 할 수 있을 것 이다.

(1) 세부 기능

해당 기능을 사용하기 이전에 보유하고 있는 분류의 목록을 출력한다.

(2) 세부 기능

분야 선택 시 문자열 배열을 활용하여 분야별 도서 목록을 출력하고자 한다.

2) 대출, 반납 기능

- 설명: 원하는 도서가 있을 경우 대출 기능을 선택하여 대출을 하고 반납하는 기능이다.

(1) 세부 기능

대출 신청 시 해당 서적을 대출의 배열에 저장하는 기능

(2) 세부 기능

반납을 완료 할 경우 다시 서적을 원래 배열에 저장하는 기능

3) 관리자 암호 기능

- 설명: 도서 정보 입력에 있어서 접근 권한 제한이 없다면 잘못된 정보를 입력 받을 수 있다.

(1) 세부 기능

도서 정보 입력 전에 암호를 입력받아 암호가 일치 할 때 정보입력 기능을 하도록 하는 기능.

3. 진척사항

1) 기능 구현

(1) 도서 정보 입력 함수

- 입출력

입력:

Password: 관리자 암호

lib[*count].book_name: struct library 구조체 배열 lib[] 중 도서의 이름을 저장하는 배열

lib[*count].author: struct library 구조체 배열 lib[] 중 작가의 이름을 저장하는 배열

lib[*count].fields struct library 구조체 배열 lib[] 중 도서의 장르를 저장하는 배열

lib[*count].publisher struct library 구조체 배열 lib[] 중 도서의 출판사을 저장하는 배열

lib[*count].pages struct library 구조체 배열 lib[] 중 도서의 페이지을 저장하는 배열

lib[*count].price struct library 구조체 배열 lib[] 중 도서의 가격을 저장하는 배열

lib[*count].quantity : struct library 구조체 배열 lib[] 중 도서의 수량 저장하는 배열

*count : 도서의 종류를 저장하는 변수 count를 지시하는 포인터 배열의 인덱스 위치로도 사용가능

출력: Void 함수이기에 반환값 없으나 해당 도서의 정보를 입력하라는 안내문구를 출력한다.

- 설명

관리자 암호인 password 를 입력받고 password가 기존의 암호와 동일하다면 도서의 정보를 입력할 수 있고, 다르다면 다시 입력하라는 문구를 출력한다.

각각의 배열은 도서의 이름, 작가의 이름, 장르, 출판사, 전체 페이지, 가격, 수량을 저장하며 도서와 작가의 이름, 장르, 출판사는 char 로 전체 페이지와 수량은 int, 가격은 float로 저장한다.

도서의 정보를 성공적으로 추가 하였을 때 안내문구를 출력하고 포인터를 통해 변수 count 를 1 증가 시킨다.

- 적용된 배운 내용

조건문, 배열, 포인터, 함수

-코드스크린샷

```c
void addBook(struct library lib[], int* count) {
    int password;
    printf("관리자 암호를 입력하시오 : ");
    scanf_s("%d", &password);

    if (password == 123) {
        printf("도서의 이름을 입력하시오 : ");
        scanf_s("%s", lib[*count].book_name, (int)sizeof(lib[*count].book_name));

        printf("작가의 이름을 입력하시오 : ");
        scanf_s("%s", lib[*count].author, (int)sizeof(lib[*count].author));

        printf("도서의 장르를 입력하시오 : ");
        scanf_s("%s", lib[*count].fields, (int)sizeof(lib[*count].fields));

        printf("도서의 출판사를 입력하시오 : ");
        scanf_s("%s", lib[*count].publisher, (int)sizeof(lib[*count].publisher));

        printf("전체 페이지를 입력하시오 : ");
        scanf_s("%d", &lib[*count].pages, (int)sizeof(lib[*count].pages));

        printf("도서의 가격을 입력하시오 :  ");
        scanf_s("%f", &lib[*count].price, (int)sizeof(lib[*count].price));

        printf("도서의 수량을 입력하시오 :  ");
        scanf_s("%d", &lib[*count].quantity, (int)sizeof(lib[*count].quantity));

        printf("₩n도서 추가 완료!₩n₩n");
        (*count)++;
    } else {
        printf("올바르지 못한 암호입니다.₩n₩n");
    }
```

(2) 보유 장르 출력 함수

- 입출력

입력: char printedGenres[20][20] : 출력한 장르를 저장하는 2차원 문자열

int printedCount : 출력한 횟수를 저장하는 int printedCount 변수

int j : 반복문에서 사용할 변수

출력 : void 함수이기 때문에 반환값 없으나 해당 장르와 일치하는 도서의 정보를 장르를 제외하고 전부 출력한다

- 설명

장르를 출력하기에 앞서 장르의 중복출력을 방지하기위해 출력한 장르를 따로 저장하기 위한 2차원 문자열char printedGenres[20][20] 선언.

출력한 횟수를 확인하기 위한 변수 int printedCount 선언.

도서의 종류 수만큼 for문을 실행 하면서 장르를 출력. 2중 for문을 통해서 이전에 동일한 장르를 출력한적이 있는지 확인 (외부 for문은 책의 종류 수만큼 반복, 내부 for 문은 출력 횟수+1 만큼 반복)

char printedGenres[] 와 lib[].fields 를 strcpy_s 통해 비교하여 이전에 출력한적이 있는지 확인.

이전에 출력한적이 없다면 printf 을 통해 해당 장르를 출력하고 printedCount 1증가

- 적용된 배운 내용

반복문, 조건문, 함수, 배열

- 코드 스크린샷

```c
void displayGenres(struct library lib[], int count) {
    printf("보유하고 있는 장르 목록₩n");

    char printedGenres[20][20];
    int printedCount = 0;

    for (int j = 0; j < count; j++) {
        int genrePrinted = 0;
        for (int k = 0; k < printedCount; k++) {
            if (strcmp(lib[j].fields, printedGenres[k]) == 0) {
                genrePrinted = 1;
                break;
            }
        }

        if (!genrePrinted) {
            printf("%s₩n", lib[j].fields);
            strcpy_s(printedGenres[printedCount], 20, lib[j].fields);
            printedCount++;
        }
    }
}
```

(3) 장르별 도서 확인

- 입출력

입력: struct library lib[]: struct library 를 원소로 갖는 배열

char field_nm[30]: 입력하는 장르를 저장하기위한 2차원배열

int found: 입력한 장르가 저장되어있는 장르와 일치하는지 여부를 확인하기위한 변수

출력: void 함수라 반환값은 없지만 입력 받은 장르와 일치하는 도서의 정보를 장르를 제외하고 출력한다.

- 설명

입력받은 장르를 저장하기위해 2차원배열 char field_nm[30]을 선언.

입력한 장르가 저장되어있는 장르와 일치하는지 여부를 확인하기위한 변수int found선언하고 0으로 초기화시킨다.

scanf_s통해 장르를 입력받고 char field_nm[30] 저장하고 이전에 저장한 장르 lib[].fields 와 strcmp를통해 비교한다.

for 문을 사용하여 도서의 종류 수만큼 비교 과정을 반복한다.

일치하는 장르가 나온다면 해당 도서의 정보를 출력한다.

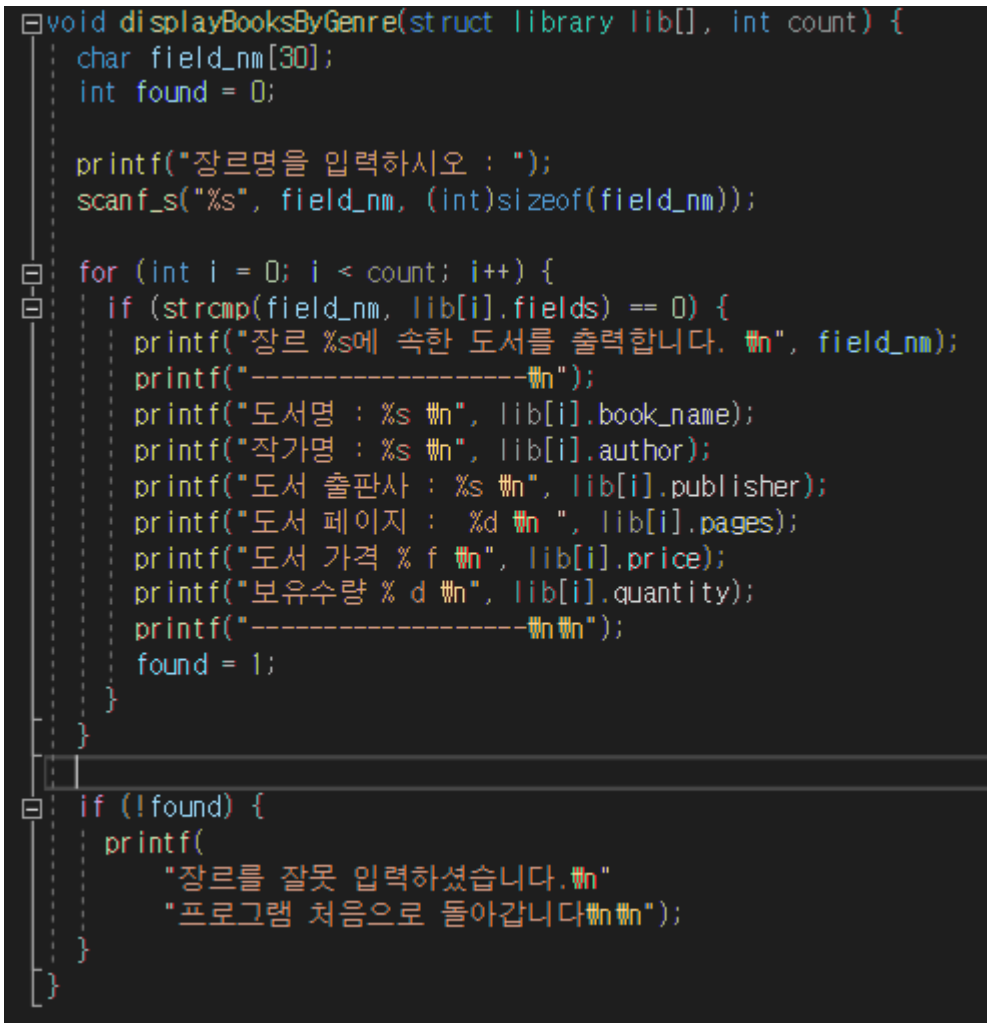
일치하는 장르가 없다면 found 를 1로 저장한다.

반복문을 빠져나와 조건문 if 을 통해 잘못 입력했다는 안내문구를 출력한다.

- 적용된 배운 내용

반복문, 조건문, 함수, 배열

- 코드 스크린샷



```
void displayBooksByGenre(struct library lib[], int count) {
    char field_nm[30];
    int found = 0;

    printf("장르명을 입력하시오 : ");
    scanf_s("%s", field_nm, (int)sizeof(field_nm));

    for (int i = 0; i < count; i++) {
        if (strcmp(field_nm, lib[i].fields) == 0) {
            printf("장르 %s에 속한 도서를 출력합니다. ₩n", field_nm);
            printf("-------------------₩n");
            printf("도서명 : %s ₩n", lib[i].book_name);
            printf("작가명 : %s ₩n", lib[i].author);
            printf("도서 출판사 : %s ₩n", lib[i].publisher);
            printf("도서 페이지 :  %d ₩n ", lib[i].pages);
            printf("도서 가격 % f ₩n", lib[i].price);
            printf("보유수량 % d ₩n", lib[i].quantity);
            printf("-------------------₩n₩n");
            found = 1;
        }
    }

    if (!found) {
        printf(
            "장르를 잘못 입력하셨습니다.₩n"
            "프로그램 처음으로 돌아갑니다₩n₩n");
    }
}
```

(4) 보유 도서의 종류 수 출력

- 입출력

입력: `struct library lib[]`: `struct library` 를 원소로 갖는 배열

출력: void 함수라 반환값은 없지만 도서의 종류수 `count` 출력한다.

- 설명

도서의 종류수`count`를 출력한다.

- 적용된 배운 내용

함수

- 코드 스크린샷

```
void displayBookCount(struct library lib[], int count) {
    printf("₩n총 도서의 종류:  %d₩n", count);
}
```

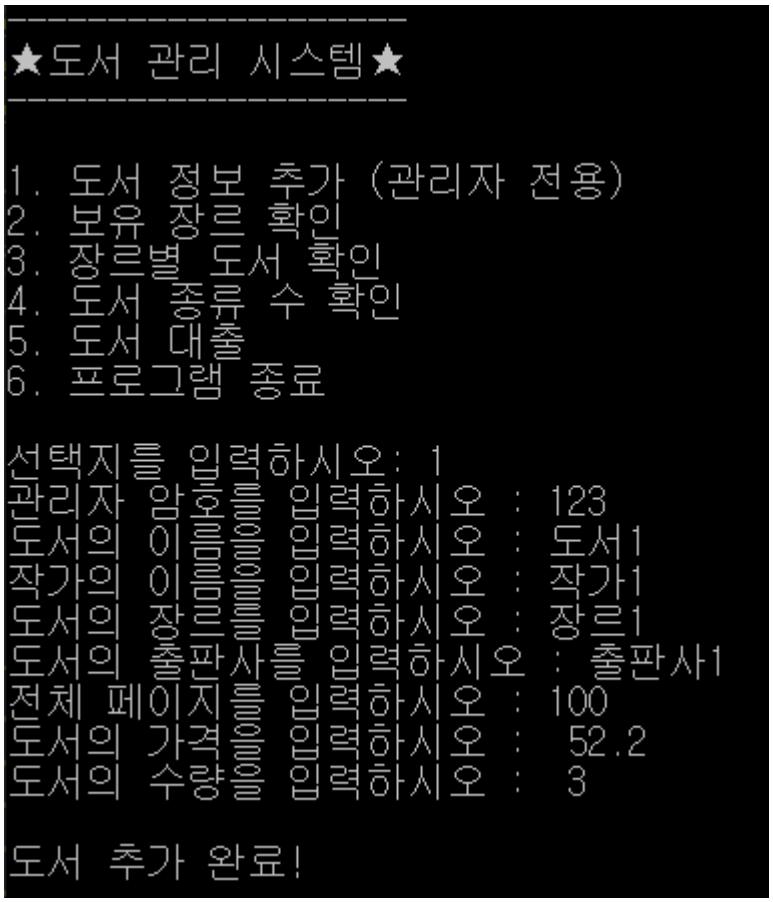
2) 테스트 결과

(1) 도서의 정보 입력 함수

- 설명

암호를 입력하고, 도서의 정보를 입력 받아 구조체 배열에 저장한다.

- 테스트 결과 스크린샷

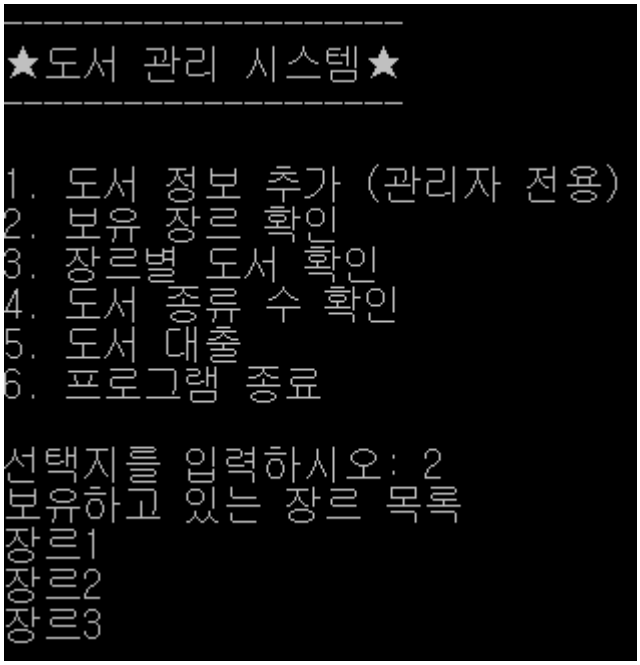


(2) 보유 장르 출력 함수

- 설명

도서의 정보를 입력받고, 보유중인 함수들을 중복없이 출력한다.

- 테스트 결과 스크린샷

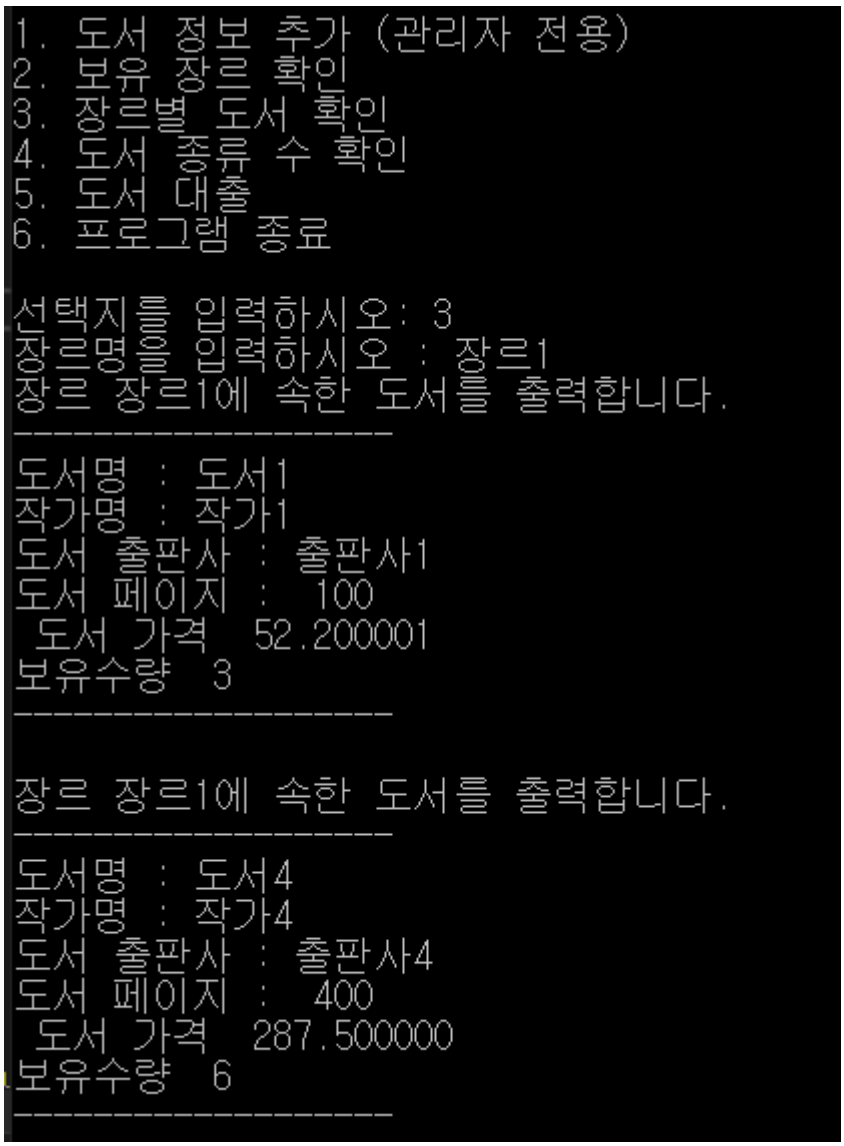


(3) 장르별 보유 도서 출력함수

- 설명

원하는 장르를 입력하면 입력받은 장르와 일치하는 장르에 속하는 도서들의 정보를 장르를 제외하고 출력한다

- 테스트 결과 스크린샷

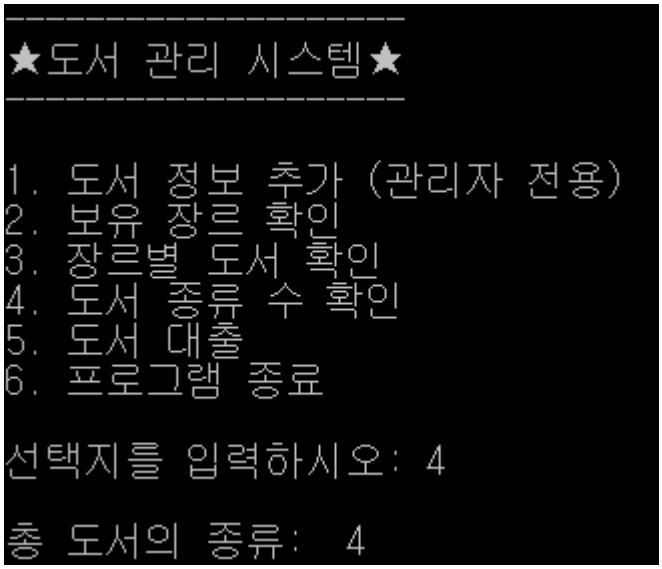

1. 도서 정보 추가 (관리자 전용)
2. 보유 장르 확인
3. 장르별 도서 확인
4. 도서 종류 수 확인
5. 도서 대출
6. 프로그램 종료
선택지를 입력하시오: 3
장르명을 입력하시오 : 장르1
장르 장르1에 속한 도서를 출력합니다.
도서명 : 도서1
작가명 : 작가1
도서 출판사 : 출판사1
도서 페이지 : 100
도서 가격 52.200001
보유수량 3
장르 장르1에 속한 도서를 출력합니다.
도서명 : 도서4
작가명 : 작가4
도서 출판사 : 출판사4
도서 페이지 : 400
도서 가격 287.500000
보유수량 6

(4) 도서의 종류수 출력

- 설명

입력받은 도서의 종류수를 출력한다.

- 테스트 결과 스크린샷



4. 계획 대비 변경 사항

1) 관리자 암호 추가

- 이전: 도서의 정보 입력에 제한이 없었다.

- 이후: 관리자 암호를 추가하여 관리자만 도서 정보를 입력할 수 있게 하였다.

- 사유: 고객들이 도서의 정보를 잘못입력하는 것을 방지하고자함

2) 반납 시간 알람 기능

- 이전: 도서의 대출 이후 반납일시 이전에 알람으로 알려주는 기능

- 이후: 잠정 보류

- 사유: 한 학기 배우는 기능으로 구현하기 어려울 것 같다는 의견으로 잠정 보류

5. 프로젝트 일정

<table>
<tr><th colspan="2">업무</th><th>11/3</th><th>11/13</th><th>11/30</th><th>12/15</th><th>12/20</th></tr>
<tr><td colspan="2">제안서 작성</td><td>완료</td><td></td><td></td><td></td><td></td></tr>
<tr><td>기능1</td><td>세부기능1</td><td colspan="2">완료</td><td></td><td></td><td></td></tr>
<tr><td></td><td>세부기능2</td><td colspan="2">완료</td><td></td><td></td><td></td></tr>
<tr><td>기능2</td><td>세부기능1</td><td></td><td colspan="2">----------></td><td></td><td></td></tr>
<tr><td></td><td>세부기능2</td><td></td><td colspan="2">----------></td><td></td><td></td></tr>
<tr><td>기능3</td><td>세부기능1</td><td></td><td></td><td colspan="2">완료</td><td></td></tr>
</table>

완료 기능

기능 1 : 장르 출력, 장르별 도서 출력

기능 3 : 관리자 암호