

C++ 프로그래밍 및 실습

도서 관리 시스템

진척 보고서 #2

제출일자: 2024.12.01

제출자명: 고희림

제출자학번: 214766

1. 프로젝트 목표

1) 배경 및 필요성

도서관의 이용을 편하게 만들고, 도서를 관리하는 것도 편하게 만들기 위함이다.

2) 프로젝트 목표

이용자들에게 도서의 대출/반납 및 관리자의 관리가 쉽도록 한다.

3) 차별점

도서관의 대출 및 반납을 편하게 할 수 있고, 동시에 책을 검색하거나 추천도 받을 수 있다.

2. 기능 계획

1) 도서의 대출과 반납

- 도서를 대출하고 반납하는 기능

(1) 대출

- 책 목록에서 사용자가 입력한 책을 대출

(2) 반납

- 대출된 책 목록에서 사용자가 입력한 책을 반납

2) 도서 추가와 삭제

- 책 목록에 책을 추가하거나, 원래 있던 책을 삭제

(1) 추가

- 책 목록에 존재하지 않는 책을 추가

(2) 삭제

- 책 목록에 존재하는 책을 삭제

3) 도서 목록 보기

- 현재 도서관에 있는 책들의 목록을 보여줌

(1) 장르별 목록

4) 책 추천

- 원하는 장르의 책을 추천해 줌

3. 진척 사항

1) 기능 구현

(1) 도서 대출

```
11 // 책을 대출하는 함수
12 void Library::borrowBook(const string& title) {
13     for (auto& book : books) {
14         if (book.title == title) {
15             // 책이 이미 대출되어있는 경우
16             if (book.isBorrowed == true) {
17                 cout << title << " 은/는 이미 대출 중입니다." << endl;
18             } else {
19                 book.isBorrowed = true;
20                 cout << title << " 이/가 대출되었습니다." << endl;
21             }
22             return;
23         }
24     }
25     // 책이 존재하지 않는 경우
26     cout << title << " 을/를 대출할 수 없습니다." << endl;
27 }
28
```

library.cpp

- 입출력

books: Library 클래스에서 선언된 벡터 (vector<Book> books;), Book 클래스에서 생성된 책의 정보를 모아 놔

isBorrowed: Book클래스에서 객체를 생성할 때(title, author, isBorrowed) 같이 생성되는 항목 중 하나, false로 자동 생성되고 대출 시 true 로 바뀌 줌

- 설명

13. for문을 이용해 books에 있는 책 중에서 입력한 제목에 맞는 책을 찾음

14 ~ 23: 찾은 책의 isBorrowed를 확인하여 대출이 되어있지 않은 경우 대출해 줌

26: for문에서 책을 찾지 못한 경우 대출할 수 없다고 출력

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)

벡터를 이용해 book객체를 벡터 books에 저장

const 키워드와 &을 이용해 객체의 변경을 방지함

(1) 도서 반납

```
29 // 책을 반납하는함수
30 void Library::returnBook(const string& title) {
31     for (auto& book : books) {
32         if (book.title == title) {
33             // 책이 대출되어있지 않는 경우
34             if (book.isBorrowed == false) {
35                 cout << title << " 은/는 대출 되어있지 않습니다." << endl;
36             } else {
37                 book.isBorrowed = false;
38                 cout << title << " 이/가 반납되었습니다." << endl;
39             }
40             return;
41         }
42     }
43     // 책이 존재하지 않는 경우
44     cout << title << " 을/를 반납할 수 없습니다." << endl;
45 }
```

library.cpp

- 입출력

books: Library 클래스에서 선언된 벡터 (vector<Book> books;), Book 클래스에서 생성된 책의 정보를 모아 놔

isBorrowed: Book클래스에서 객체를 생성할 때(title, author, isBorrowed) 같이 생성되는 항목 중 하나, false로 자동 생성되고 대출 시 true 로 바뀌 줌

- 설명

31. for문을 이용해 books에 있는 책 중에서 입력한 제목에 맞는 책을 찾음

32 ~ 41: 찾은 책의 isBorrowed를 확인하여 반납이 되어있지 않은 경우 반납해 줌

44: for문에서 책을 찾지 못한 경우 반납할 수 없다고 출력

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)

벡터를 이용해 book객체를 벡터 books에 저장

const 키워드와 &을 이용해 객체의 변경을 방지함

(3) 도서 추가

```
6 // 책을 추가하는 함수
7 void Library::AddBook(const string& title, const string& author) {
8     books.emplace_back(title, author);
9     // push_back은 객체를 복사해서 벡터에 추가, 객체가 미리 있을 때 유용
10    // emplace_back은 객체를 즉시 생성
11    cout << title << " 이/가 도서관에 추가되었습니다." << endl;
12 }
```

library.cpp

```
61 while (true) {
62     cout << "관리자 메뉴" << endl;
63     cout << "0. 시스템 종료" << endl;
64     cout << "1. 도서 추가" << endl;
65     cout << "2. 도서 삭제" << endl;
66     cout << "3. 이전으로 돌아가기" << endl;
67
68     cin >> choice;
69
70     if (choice == 0) {
71         exit(0); // 프로그램의 즉시 종료
72     } else if (choice == 1) {
73         cout << "추가할 도서 제목: ";
74         cin >> title;
75         cout << "도서의 저자: ";
76         cin >> author;
77         library.AddBook(title, author);
78     }
79 }
```

main.cpp

- 입출력

books: Library 클래스에서 선언된 벡터 (vector<Book> books;), Book 클래스에서 생성된 책의 정보를 모아 놔

choice: 사용자의 선택을 입력 받아 저장

- 설명

사용자에게 제목과 저자를 입력 받고 AddBook()을 호출

책이 추가되었다는 메시지 출력

- 적용된 배운 내용

벡터에서 객체를 추가하는 것

(4) 도서 삭제

```
14 // 책 삭제
15 void Library::RemoveBook(const string& title) {
16     for (auto i = books.begin(); i != books.end(); ++i) {
17         if (i->title == title) {
18             if (i->isBorrowed == true) {
19                 cout << title << "이/가 대출 중이라 삭제가 불가능합니다." << endl;
20                 return;
21             }
22
23             books.erase(i);
24             cout << title << " 이/가 삭제되었습니다." << endl;
25             return;
26         }
27     }
28     cout << title << " 은/는 도서관에 없습니다." << endl;
29 }
30
```

library.cpp

- 입출력

books: Library 클래스에서 선언된 벡터 (vector<Book> books;), Book 클래스에서 생성된 책의 정보를 모아 놔

isBorrowed: Book클래스에서 객체를 생성할 때(title, author, isBorrowed) 같이 생성되는 항목 중 하나, false로 자동 생성되고 대출 시 true 로 바뀌 줌

- 설명

for문으로 books 벡터의 시작부터 끝까지 입력한 제목의 도서를 찾음

isBorrowed가 true일경우 도서가 대출 중이므로 삭제 불가

도서를 삭제한 후에는 삭제했다는 메시지 출력

도서가 없을 경우 도서가 존재하지 않다는 메시지 출력

- 적용된 배운 내용

벡터에서 객체를 삭제하는 방법

벡터를 순환하는 방법

(5) 도서 목록

```
67 // 도서관의 책 목록을 볼 수 있는 함수
68 void Library::BookList() {
69     if (books.empty()) {
70         cout << "도서관에 책이 없습니다. " << endl;
71         return;
72     }
73
74     cout << "- 현재 도서 목록 -" << endl;
75     cout << left << setw(20) << "Title" << setw(20) << "Author" << setw(10) << "Status" << endl;
76     cout << "-----" << endl;
77     for (const auto& book : books) {
78         cout << left << setw(20) << book.title;
79         cout << setw(20) << book.author;
80         cout << setw(10);
81         if (book.isBorrowed == true) {
82             cout << "대출 중" << endl;
83         } else {
84             cout << "대출 가능" << endl;
85         }
86     }
87
88     cout << endl;
89 }
```

library.cpp

- 입출력

books: Library 클래스에서 선언된 벡터 (vector<Book> books;), Book 클래스에서 생성된 책의 정보를 모아 놔

- 설명

벡터 books가 비어 있으면 도서관에 책이 없다는 메시지 출력

그렇지 않으면 현재 도서목록 출력

도서의 정렬의 위해 #include <iomanip>의 setw()함수를 사용

left로 왼쪽 정렬을 함

- 적용된 배운 내용

벡터가 비어있는지 확인하는 함수 사용

for문을 벡터 전체를 출력하는데 이용함

2) 테스트 결과

(1) 도서 대출

```
도서관리 시스템
1. 도서 대출
2. 도서 반납
3. 시스템 종료
1
대출할 도서 제목: Justice
Justice 이/가 대출되었습니다.
도서관리 시스템
1. 도서 대출
2. 도서 반납
3. 시스템 종료
█
```

관리시스템 메뉴를 선택 후 제목을 입력하면 도서를 대출해 줌

```
대출할 도서 제목: Justice
Justice 은/는 이미 대출 중입니다.
도서관리 시스템
```

이미 대출 중인 도서일 경우 대출 중이라는 메시지 출력

```
대출할 도서 제목: qqww
qqww 을/를 대출할 수 없습니다.
도서관리 시스템
```

도서관에 없는 책 입력 시 대출할 수 없다는 메시지 출력

(2) 도서 반납

```
도서관리 시스템
1. 도서 대출
2. 도서 반납
3. 시스템 종료
2
반납할 도서 제목: Justice
Justice 이/가 반납되었습니다.
도서관리 시스템
1. 도서 대출
2. 도서 반납
3. 시스템 종료
```

관리시스템 메뉴를 선택 후 제목을 입력하면 도서를 반납해 줌

```
반납할 도서 제목: Justice
Justice 은/는 대출 되어있지 않습니다.
도서관리 시스템
```

대출되지 않은 도서일 경우 대출 되어있지 않다는 메시지 출력

```
반납할 도서 제목: qqwwwee
qqwwwee 을/를 반납할 수 없습니다.
도서관리 시스템
```

도서관에 없는 책 입력 시 반납할 수 없다는 메시지 출력

(3) 도서 추가

```
도서관리 시스템
0. 시스템 종료
1. 도서 대출
2. 도서 반납
3. 도서 목록
4. 관리자 메뉴
4
관리자 메뉴
0. 시스템 종료
1. 도서 추가
2. 도서 삭제
3. 이전으로 돌아가기
1
추가할 도서 제목: apple
도서의 저자: banana
apple 이/가 도서관에 추가되었습니다.
```

관리자 메뉴로 들어가 도서 추가를 선택

제목과 저자를 입력하면 도서관에 추가되었다는 메시지 출력

(4) 도서 삭제

```
도서관리 시스템
0. 시스템 종료
1. 도서 대출
2. 도서 반납
3. 도서 목록
4. 관리자 메뉴
4
관리자 메뉴
0. 시스템 종료
1. 도서 추가
2. 도서 삭제
3. 이전으로 돌아가기
2
삭제할 도서 제목: qwerty
qwerty 은/는 도서관에 없습니다.
```

관리자 메뉴로 들어가 도서 삭제를 선택

도서관에 없는 도서를 입력하면 없는 도서라는 메시지 출력

```
관리자 메뉴
0. 시스템 종료
1. 도서 추가
2. 도서 삭제
3. 이전으로 돌아가기
2
삭제할 도서 제목: Justice
Justice 이/가 삭제되었습니다.
```

도서를 입력하면 도서 삭제한 후 메시지 출력

```
관리자 메뉴
0. 시스템 종료
1. 도서 추가
2. 도서 삭제
3. 이전으로 돌아가기
2
삭제할 도서 제목: Justice
Justice이/가 대출 중이라 삭제가 불가능합니다.
```

대출중인 도서일 경우 대출 중이라 삭제 불가능하다는 메시지 출력

(5) 도서 목록

```
도서관리 시스템
0. 시스템 종료
1. 도서 대출
2. 도서 반납
3. 도서 목록
4. 관리자 메뉴
3
- 현재 도서 목록 -
```

Title	Author	Status
Human Acts	Han Kang	대출 가능
Chunhyangjeon	unknown	대출 가능
apple	banana	대출 가능
happy	sad	대출 가능
hungry	chicken	대출 가능

도서 목록을 선택하면 도서 목록과 도서의 상태(대출 가능한지)가 출력됨

4. 프로그램 실행 방법

대출과 반납 기능을 테스트 하기 위해 3권의 책을 미리 생성해 놓음

(제목, 작가) = (Human Acts, Han Kang), (Justice, Michael J. Sandel), (Chunhyangjeon)

대출과 반납할 때 제목을 입력하면 해당 기능을 이용할 수 있음

5. 계획 대비 변경 사항

1) 변경 내역 아직 없음

6. 프로젝트 일정

업무		11/3	11/10	11/17	11/24	12/1	12/8	12/15
제안서 작성		완료						
기능1	대출/반납		완료					
기능2	추가/삭제			완료				
기능3	도서 목록				완료			
기능4	추천					진행 중		
확인 및 수정								----->