REPORT

Library system

목 차

과목명: 객체지향 프로그래밍

학과: 컴퓨터공학과

학번:

이름: 오지현 양서희 문규인 황지상

1. 팀 프로젝트
   1. 팀 구성
   2. 역할 분담
2. 문제 명세

2-1. 문제 명세

2-2. 문제 분석

2-3. 설계

문제 명세

Library System

1. 책

1. 도서 관리(조회, 삽입, 삭제 기능)
2. 도서 대여. 회원 등록, 로그인(회원 구분: 학생, 대학원생, 교수 등)
3. 도서 검색 (카테고리, 제목, 저자명, 고유번호)

4-1. 도서 검색 후 책 정보 출력(저자, 출판사, 장르, 리뷰 등)

2. 회원

1. 회원 등록, 로그인(회원 구분: 학생, 대학원생, 교수 등)
2. 연체 상태 보이기
3. 읽었던 책의 별점과 리뷰 등록
4. 대여했던 도서 기록

3. 관리자

1. 회원 관리(조회, 삭제 기능)

2. 도서 관리(추가, 삭제 기능)

문제 분석

설계

텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명

Member 클래스:

도서관 회원을 나타내며, 아이디, 비밀번호, 이름을 필드로 갖고 있습니다. 또한 회원이 대여한 책의 정보를 저장하기 위한 Book 클래스 배열과 최대 저장 가능한 책의 수, 책 대여 가능 기간을 포함합니다. 대여한 책 정보를 조회하거나 책을 대여하고 반납하는 메서드를 제공합니다.

Student 클래스:

Member 클래스를 상속받아 학년과 전공을 추가로 가지고 있습니다. 또한 대여 기간과 대여 가능한 책의 수를 다르게 할 것 입니다.

Professor 클래스:

Member 클래스를 상속받아 소속 학과 정보를 추가로 가지고 있습니다. 또한 대여 기간과 대여 가능한 책의 수를 다르게 할 것 입니다.

Librarian 클래스:

Member 클래스를 상속받아 책 데이터를 관리하는 메서드를 제공합니다. 이를 통해 책을 추가하거나 삭제할 수 있습니다.

Book 클래스:

책의 정보를 담는 클래스로, 제목, 저자, 출판사, 장르, 고유번호 등의 필드를 포함합니다. 대여 여부와 대여 후 경과 기간을 확인할 수 있고, 파일에 책 정보를 저장하거나 불러오는 메서드를 제공합니다. 또한 연체 여부를 판별하는 기능을 갖추고 있습니다.

Member\_App 클래스:

MemberSystem 클래스:

첫번째로는 members라는 Member(회원)객체 배열을 가지고 있다. public에는 signup(회원가입),login(로그인) 기능의 메서드를 가지고 있다.

또한 멤버 정보를 가지고 있는 파일을 읽어들이거나 저장하는 load\_member\_file, save\_member\_file 함수를 private으로 가지고 있다.

signup(도서 검색) 함수 :

1. 시퀀스 다이어그램

회원가입 요청

Member\_system

Member\_app

USER

입력 받은 회원정보 멤버 객체화

입력 받은 회원의 정보

파일에 멤버 정보 저장

회원가입 결과 반환

결과 출력

1. 알고리즘 서술

members.push\_back(member);

save\_Member\_File();

입력 받은 정보를 member 객체화 시켜 memberSystem에 있는 members 객체 배열에 저장

그 이후 파일에 입력

login(도서 검색) 함수 :

1. 시퀀스 다이어그램

로그인 요청

Member\_system

Member\_app

USER

입력 받은 로그인정보 멤버 객체화

입력 받은 로그인 정보

로그인 정보랑 회원 정보랑 비교

로그인 결과 반환

결과 출력

Rent\_book(도서 대여) 함수 :

1)시퀀스 다이어그램

BookSystem

Book\_App

USER

도서 대여 요청

입력 받은 도서 정보

입력 받은 도서가 대여 중인지 확인 후 대여

대여 결과 반환

결과 출력

Return\_book(도서 반납) 함수 :

1)시퀀스 다이어그램

BookSystem

Book\_App

USER

도서 반납 요청

입력 받은 도서 정보

입력 받은 도서가 대여 중인지 확인 후 반납

반납 결과 반환

결과 출력

Search\_book(도서 검색) 함수 :

1)시퀀스 다이어그램

BookSystem

Book\_App

USER

도서 검색 요청

입력 받은 도서 정보

입력 받은 정보와 책 정보들 비교

알맞은 책의 정보 가져옴

해당 도서 정보 출력

1. 알고리즘 기술

함수 searchBooks(library, searchKeyword) {

found = false // 검색 결과를 찾았는지 여부를 나타냄

출력: "searchKeyword"

모든 book을 library에서 가져옴 {

만약 book의 도서 이름 또는 저자에 searchKeyword가 포함되어 있다면 {

출력: book의 모든 정보를 출력

found = true // 검색어를 찾았음을 나타내는 변수를 참으로 설정

}

}

만약 found가 거짓이라면 {

출력: "해당 검색어의 결과를 찾을 수 없습니다."

}

}

checkOverdue(대여 연체 확인) 함수 : 다시

1. 시퀀스 다이어그램

Book

회원 확인

Member

USER

책의 정보 입력

책 정보와 대여 날짜 불러오기

대여 가능 기간과 대여날짜 이용하여 연체 유무 보이기

1. 기능 서술

대여한 날짜에서 대여가능기간을 더했을 때, 오늘 날짜를 지나쳤다면 연체 = TRUE, 아니라면 FALSE -> 책과 연체상태 출력

1. 알고리즘 서술

함수 checkOverdue() {

dueDate의 일자를 현재 일자에 대여 가능 기간을 더하여 설정

만약 현재 시간이 연체 날짜인 dueDate보다 이전이라면 {

연체되었음을 나타내는 참(true) 반환

} 그렇지 않으면 {

연체되지 않았음을 나타내는 거짓(false) 반환

}

}

addReview(리뷰 작성) 함수 : 다시

1. 시퀀스 다이어그램

Book

회원 확인

Member

USER

책의 정보 입력

책의 정보 불러오기

회원의 책 대여 여부 확인

리뷰와 별점 작성

1. 기능 서술

대여 후 반납했던 책 기록을 가져옴 -> 책 선택 -> 별점과 리뷰작성 -> 저장

1. 알고리즘 서술

함수 addReview() {

try {

만약 회원이 이 책을 빌린 적이 잇다면{

책 정보 출력

book의 정보를 출력하는 BookInfo() 호출

책에 대한 별점 및 리뷰 등록

"리뷰가 성공적으로 등록되었습니다. book의 제목." 출력

}

그렇지 않다면 {

대여되지 않은 책에 대한 오류를 처리하기 위해 예외 발생

}

}

예외 처리 {

만약 발생한 예외가 있으면 {

발생한 예외의 메시지 출력

}

}

}

add\_book(도서 추가) 함수 :

1)시퀀스 다이어그램

librarian\_System

librarian\_App

USER

도서 추가 요청

입력 받은 도서 정보

입력 받은 도서 파일에 추가

도서 추가 결과 반환

결과 출력

delete\_book(도서 추가) 함수 :

1)시퀀스 다이어그램

librarian\_System

librarian\_App

USER

도서 삭제 요청

입력 받은 도서 정보

입력 받은 도서 파일에 있는 도서 정보와 비교 후 삭제

도서 삭제 결과 반환

삭제 출력

deleteMember(회원 삭제) 함수 :

1. 시퀀스 다이어그램

회원 삭제 요청

librarian\_system

librarian\_app

USER

입력 받은 회원 아이디 정보

회원 아이디 정보와 회원들 정보 비교 후 삭제

삭제 결과 반환

결과 출력

displayMembers(회원 조회) 함수 :

1. 시퀀스 다이어그램

회원 조회 요청

librarian\_system

librarian\_app

USER

입력 받은 회원 아이디 정보

회원 아이디 정보와 회원들 정보 비교 후 조회

조회 결과 반환

결과 출력