**Final**

**REPORT**

*Library system*로고, 폰트, 상징, 등록 상표이(가) 표시된 사진

자동 생성된 설명

과목명: 객체지향프로그래밍

교수명: 하상호

학과: 컴퓨터공학과

이름/학번: 오지현 / 20204111

문규인 / 20224026

양서희 / 20224028

황지상 / 20224060

제출일자: 2023.11.20

목 차

1. **문제 명세2**

주제 및 기능2

1. **문제 분석3**

클래스 계층구조3

1. **설계4**

3-1. UML 다이어그램4

3-2. 클래스 설명5

[첨부1] 미팅일지116

[첨부2] 미팅일지217

**1. 문제 명세**

* 주제: Library System (도서 관리 시스템)
* 상세 기능

|  |  |
| --- | --- |
| 구분 | 기능 |
| 책 | * 도서 관리(조회, 삽입, 삭제 기능) * 도서 대여, 회원 등록, 로그인(회원 구분: 학생, 대학원생, 교수) * 도서 검색 (카테고리, 제목, 저자명, 고유번호) * 도서 검색 후 책 정보 출력(저자, 출판사, 장르, 리뷰 등) |
| 회원 | * 회원 등록, 로그인(회원 구분: 학생, 대학원생, 교수 등) * 연체 상태 보이기 * 읽었던 책의 별점과 리뷰 등록 * 대여했던 도서 기록 |
| 관리자 | * 회원 관리 (조회, 삭제 기능) * 도서 관리 (추가, 삭제 기능) |

* C++ 객체지향 기능 활용 계획

|  |  |
| --- | --- |
| 기능 | 활용 |
| 추상화 | * 클래스(도서 정보, 회원 정보 추상화) |
| 상속 | * 다양한 종류의 도서 모델링 * 다양한 사용자 -> 학부생, 대학원생, 교직원 |
| 동적 바인딩(가상함수) | * 사용자 유형에 따른 메시지 출력 |
| 연산자 중복 | * 여러 개의 도서 병합 (여러 권의 책 대여) |
| 예외처리 | * 대여하려는 책이 이미 대출중인 책이거나 대출 불가능인 상태 * 파일 읽기 실패 |
| 포괄적 함수 | * 도서 목록 처리 |

**2. 문제 분석**

* 클래스 계층구조

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

**3. 설계**

**3-1. 클래스 설명**

I. Member클래스

도서관 회원을 나타내며, 아이디, 비밀번호, 이름을 필드로 갖고 있다. 또한 회원이 대여한 책의 정보를 저장하기 위한 Book 클래스 배열과 최대 저장 가능한 책의 수, 책 대여 가능 기간을 포함한다. 생성자, 접근자, 설정자를 포함합니다. 정보를 보여주는 display 메서드가 있다.

II. Book 클래스

책의 정보를 담는 클래스로, 제목, 저자, 출판사, 장르, 고유번호 등의 필드를 포함한다. 생성자, 접근자, 설정자를 포함한다. 정보를 보여주는 display 메서드가 있다.

III. Member\_App 클래스

Member\_App 이라는 클래스는 로그인 회원가입과 같이 회원에 관련하여 화면에 보이고 정보를 입력 받기 위해 만든 클래스이다.

1. System: MemberSystem 객체
2. Run(): 사용자가 원하는 메뉴를 실행시킴
3. Showmenu():메서드는 어떤 메뉴가 있는지 보임
4. getChoice(): 메서드는 선택한 메뉴 번호 값을 반환
5. login (): 메서드는 로그인 과정을 보임
6. signup(): 클래스는 회원가입 과정을 보임

IV. MemberSystem 클래스

MemberSystem 클래스는 입력받은 정보를 처리하기 위해서 만들어진 클래스이다.

1. members: Member(회원)객체 배열
2. load\_member\_file: 멤버 정보를 가지고 있는 파일 읽어옴 (System 상속)
3. save\_member\_file: 파일에 멤버 정보를 저장함 (System 상속)
4. signup(회원 가입) 함수 :
5. 시퀀스 다이어그램

텍스트, 도표, 스크린샷, 평행이(가) 표시된 사진

자동 생성된 설명

1. 알고리즘 기술

|  |
| --- |
| signup(Member& member){  members.push\_back(member);  save\_Member\_File();  } |

1. login(도서 검색) 함수
2. 시퀀스 다이어그램

텍스트, 도표, 스크린샷, 평행이(가) 표시된 사진

자동 생성된 설명

1. 알고리즘 기술

|  |
| --- |
| bool login(const string& id, const string& password) {  for const Member& member : members {  if (member.getId() == id && member.getPassword() == password) then{  출력 << "로그인 성공!"  return true;  }  }  출력<< "로그인 실패. 아이디나 비밀번호가 일치하지 않습니다."  return false;  } |

V. BookApp 클래스

Book\_App 이라는 클래스는 로그인 회원가입과 같이 회원에 관련하여 화면에 보이고 정보를 입력 받기 위해 만든 클래스이다.

1. System: BookSystem 객체
2. Run(): 사용자가 원하는 메뉴를 실행시킴
3. Showmenu(): 어떤 메뉴가 있는지 보임
4. getChoice(): 선택한 메뉴 번호 값을 반환
5. rentBook() : 책을 빌리는 과정을 보임
6. returnBook(): 책을 반납하는 과정을 보임

VI. BookSystem 클래스

1. Rent\_book(도서대여) 함수:
2. 시퀀스 다이어그램

텍스트, 도표, 스크린샷, 평행이(가) 표시된 사진

자동 생성된 설명

1. 알고리즘 기술

|  |
| --- |
| bool rent\_Book(const string& memberId, const string& bookName) {  auto it = find\_if(members.begin(), members.end(), [&](const Member& member) {  return member.getId() == memberId;  });    if (it != members.end()) then {  it->addRentedBook(bookName);  save\_Member\_File();  출력: bookName "이(가) 대여되었습니다."  return true;  }  else then {  출력: "해당 아이디의 회원을 찾을 수 없습니다."  return false;  }  } |

1. return\_book(도서반납) 함수:
2. 시퀀스 다이어그램

텍스트, 스크린샷, 도표, 평행이(가) 표시된 사진

자동 생성된 설명

1. 알고리즘 기술

|  |
| --- |
| bool returnBook(const string& memberId, const string& bookName) {  auto it = find\_if(members.begin(), members.end(), [&](const Member& member) {  return member.getId() == memberId;  });    if (it != members.end()) then {  it->removeRentedBook(bookName);  save\_Member\_File();  출력: bookName "이(가) 반납되었습니다."  return true;  }  else then {  출력: "해당 아이디의 회원을 찾을 수 없습니다."  return false;  }  } |

1. search\_book(도서 검색) 함수
2. 시퀀스 다이어그램

텍스트, 도표, 평면도, 평행이(가) 표시된 사진

자동 생성된 설명

1. 알고리즘 기술

|  |
| --- |
| void searchBooks(const string& searchKeyword) {  bool found = false;  cout << "\"" << searchKeyword << "\"에 대한 검색결과" << endl;  for (const Book& book : books) {  if (book.getBookName().find(searchKeyword) != string::npos ||  book.getWriter().find(searchKeyword) != string::npos) then {  출력: 제목, 저자, 출판사, 장르.고유번호, 페이지수, 대여여부;  found = true;  }  }  if (!found) then  cout << "해당 검색어의 결과를 찾을 수 없습니다." << endl;  }  } |

VII. LibrarianApp 클래스

LibrarianApp 이라는 클래스는 시스템 이용 항목들을 사용자에게 보여주고, 사용자의 선택에 따라 해당 서비스를 제공하는 클래스이다.

1. System: BookSystem 객체
2. Run(): 사용자가 원하는 메뉴를 실행시킴
3. Showmenu(): 어떤 메뉴가 있는지 보임
4. getChoice(): 선택한 메뉴 번호 값을 반환
5. removeBook() : 책을 파일에서 제거하는 과정
6. AddBook(): 책을 파일에 추가하는 과정
7. removeMember(): 멤버를 파일에서 제거하는 과정
8. displayMember(): 멤버를 보이고 조회하는 과정

VIII. LibrarianSystem 클래스

1. add\_book(도서 추가) 함수:
2. 시퀀스 다이어그램

텍스트, 도표, 스크린샷, 평행이(가) 표시된 사진

자동 생성된 설명

1. 알고리즘 기술

|  |
| --- |
| void AddBook (const Book& book) {  books.push\_back(book);  save\_Book\_File();  } |

1. removeBook(도서 삭제) 함수
2. 시퀀스 다이어그램

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

1. 알고리즘 기술

|  |
| --- |
| void removeBook(const string& book\_Name) {  auto it = std::find\_if(books.begin(), books.end(), [&](const Book& book) {  return book.getBookName() == book\_Name;  });  if (it != books.end()) then{  books.erase(it);  save\_Book\_File();  출력: book\_Name "이 삭제 되었습니다.";  }  else then  출력: book\_Name "이라는 도서가 없습니다.";  } |

1. removeMember(회원 삭제) 함수
2. 시퀀스 다이어그램

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

1. 알고리즘 기술

|  |
| --- |
| bool removeMember(const string& id) {  auto it = find\_if(members.begin(), members.end(), [&](const Member& member) {  return member.getId() == id; });  if (it != members.end()) then{  members.erase(it);  Load\_Member\_File();  출력: id " 회원이 삭제되었습니다.";  return true;  }  else then{  출력: "해당 아이디의 회원을 찾을 수 없습니다.";  return false;  }  } |

1. displayMember(회원 조회) 함수
2. 시퀀스 다이어그램

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

1. 알고리즘 기술

|  |
| --- |
| void displayMember() {  if (members.empty ) then {  출력:"회원 목록이 비어 있습니다.";  }  else {  출력:"회원 목록: ";  for (size\_t i = 0; i < members.size(); ++i) {  출력: "회원 번호: " << i + 1 ;  members[i].display();  }  }  } |

[첨부1]

**4. 결과 및 구현**

기술

객체지향프로그래밍 \_ 텀 프로젝트 **미 팅 일 지 1**

|  |  |  |  |
| --- | --- | --- | --- |
| 참여자 | 오지현, 문규인, 양서희, 황지상 | | |
| 미팅날짜 | 2023.11.02 목 | 시작 시간 | 14:40 |
| 종료 시간 | 15:40 |
| 장소 | 순천향대학교 멀티미디어관 619호 | | |
| 미팅 내용 | * 역할 분담 – 조장:오지현 / 기록:양서희 / 발표: 보류/ 자료 수집: 분배 * 주제 선택 – 도서관 관리 시스템 * 사용할 개발 환경 : vs 또는 깃허브 -> 더 의논해보기 * 시스템에 들어갈 기능  1. 도서 관리 (조회, 삽입, 삭제 기능 등)   2. 도서 대여, 회원 등록, 로그인 (회원 구분: 학생, 대학원생, 교수 등)  3. 도서 검색 (카테고리, 제목, 저자명)   1. 검색하면 책 정보 간략하게 뜨기 (저자, 출판사, 장르, 페이지 수 등), 책 검색하고 선택하면 리뷰 보이게 2. 연체상태 (회원별로 대여 가능기간 다르게) 3. 읽었던 책 별점, 리뷰 등록 4. 대여했던 도서 기록 5. 책에 고유번호를 붙여서 검색 쉽도록  * 필요 클래스에 대한 대략적인 설계   Book클래스{  변수: 책이름, 저자, 출판사, 분야(상속), 고유번호, 별점, 리뷰, 페이지 수, 대여 여부  함수: 생성자, 호출자, 접근자 }  Member클래스{  변수: 직급(상속), 아이디, 비번, 이름, 대여가능 권 수, 대여 가능 기간  함수: 생성자, 호출자, 접근자 } | | |
| 미팅 결과 | 시스템에 들어갈 기능을 다음과 같이 분배하여 설계해오기 | | |
| 비고 | 다음 회의: 11/6 월 4시 30분 | | |

[첨부2]

객체지향프로그래밍 \_ 텀 프로젝트 **미 팅 일 지 2**

|  |  |  |  |
| --- | --- | --- | --- |
| 참여자 | 오지현, 문규인, 양서희, 황지상 | | |
| 미팅날짜 | 2023.11.06 월 | 시작 시간 | 16:30 |
| 종료 시간 | 17:10 |
| 장소 | 온라인(디스코드) | | |
| 미팅 내용 | * 각자 준비해 온 내용 발표 * 들으면서 궁금한 부분 질문 및 보완점 공유 * MVC패턴, 시퀀스 다이어그램, UML 다이어그램으로 표현해보자 * 코드 실행 시 데이터가 저장된 파일을 읽고 시작할지 시작하면서 읽을지 의논 -> 보류 * 보고서 준비 * 각자 맡은 부분에 대해 분석 및 설계를 더 자세하고 명확하게 준비해서 취합하자 * 추상화, 상속, 동적 바인딩, 연산자 중복, 예외처리, 포괄적 함수 등과 같은 기능을 포함시키고 명시해주자 * 공유 코드에는 주석을 달아주자   PPT를 만들어야 할 수도 있으니 기간을 고려 | | |
| 미팅 결과 | * 추상화, 상속, 동적 바인딩, 연산자 중복, 예외처리, 포괄적 함수 등의 기능을 본인이 맡은 부분에서 활용 가능하면 수정하기 * 결과를 보고서 활용에 편하게 최대한 잘 정리해 조장에게 보내기 * 마감 기한은 정하지 않았지만 최대한 빠르게 끝내기 | | |
| 비고 | 각자 수정 및 추가하는 대로 조장에게 보내고, 취합 | | |