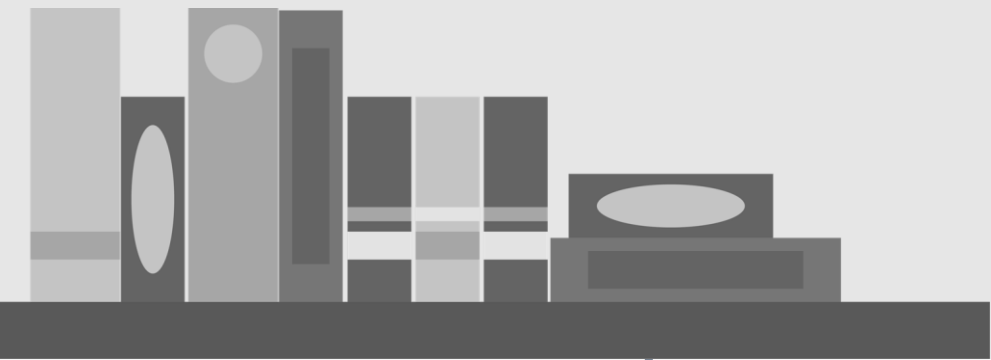


# 도서관 시스템

컴퓨터공학과  
18101208 김조은





## 1. 출력 화면 설계

-----\*서울과학기술대학교 도서관 시스템\*-----

1. 관리자
2. 이용자
3. 종료

-----

메뉴 번호 입력 :

//잘못 입력한 경우(뒤의 모든 경우에 적용)

메뉴 번호 입력 : 0

잘못된 입력입니다. 1~2 사이의 번호를 입력하세요.

메뉴 번호 입력 :

//1(관리자) 선택

-----\*서울과학기술대학교 도서관 관리\*-----

1. 학생 추가
2. 책 추가
3. 내일 반납 예정자 확인
4. 지난 연체날짜 숨기기
5. 종료

-----  
메뉴 번호 입력 :

//1(이용자 추가) 선택

-----\*1. 이용자 추가\*-----

1. 대학생
2. 대학원생
3. 졸업생
4. 교수
5. 강사
6. 직원

-----  
메뉴 번호 입력 :

## 1. 출력 화면 설계

//1.1(대학생) 선택

-----\*1. 대학생 추가\*-----

ID 입력(ex. 001) : 001

이름 입력(ex. 홍길동 / (한글)3자까지) : Joeun

전화번호 입력(ex. 01012345678) : 01012345678

학년 입력(ex. 1) : 1

휴학 여부 입력(ex. 0 / 재학 중이면 0, 휴학 중이면 1) : 1

입력하고자 하는 정보는 다음과 같습니다.

001/Joeun/01012345678/1(학년)/1(휴학중)

//뒤에는 동일

정보를 입력하시겠습니까? (y/n) : y

저장되었습니다.

메인 화면으로 돌아갑니다

정보를 입력하시겠습니까? (y/n) : n

메인 화면으로 돌아갑니다

## 1. 출력 화면 설계

//1.2(대학원생) 선택

-----\*2. 대학원생 추가\*-----

ID 입력(ex. 001) : 001

이름 입력(ex. 홍길동 / (한글)3자까지) : Joeun

전화번호 입력(ex. 01012345678) : 01012345678

학년 입력(ex. 1) : 1

과정 입력(ex. 0 / 석사면 0, 박사면 1) : 1

입력하고자 하는 정보는 다음과 같습니다.

001/Joeun/01012345678/1(학년)/1(휴학중)/1(박사)

//1.3(졸업생) 선택

-----\*3. 졸업생 추가\*-----

ID 입력(ex. 001) : 001

이름 입력(ex. 홍길동 / (한글)3자까지) : Joeun

전화번호 입력(ex. 01012345678) : 01012345678

입력하고자 하는 정보는 다음과 같습니다.

001/Joeun/01012345678

## 1. 출력 화면 설계

//1.4(교수) 선택

-----\*4. 교수 추가\*-----

ID 입력(ex. 001) : 001

이름 입력(ex. 홍길동 / (한글)3자까지) : Joeun

전화번호 입력(ex. 01012345678) : 01012345678

전공 입력(ex. 컴공 / (한글)6자까지) : 컴퓨터공학과

입력하고자 하는 정보는 다음과 같습니다.

001/Joeun/01012345678/컴퓨터공학과

//1.5(강사) 선택

-----\*5. 강사 추가\*-----

ID 입력(ex. 001) : 001

이름 입력(ex. 홍길동 / (한글)3자까지) : Joeun

전화번호 입력(ex. 01012345678) : 01012345678

입력하고자 하는 정보는 다음과 같습니다.

001/Joeun/01012345678

## 1. 출력 화면 설계

//1.6(직원) 선택

-----\*6. 직원 추가\*-----

ID 입력(ex. 001) : 001

이름 입력(ex. 홍길동 / (한글)3자까지) : Joeun

전화번호 입력(ex. 01012345678) : 01012345678

부서 입력(ex. 행정 / (한글)6자까지) : 장학

입력하고자 하는 정보는 다음과 같습니다.

001/Joeun/01012345678/장학

//2(책 추가) 선택

-----\*2. 책 추가\*-----

인덱스 입력(ex. 001) : 001

책 이름 입력(ex. 셜록홈즈 / (한글)12자까지) : 셜록홈즈

책 저자 입력(ex. 홍길동 / (한글)4자까지) : 코난도일

입력하고자 하는 정보는 다음과 같습니다.

001/셜록홈즈/코난도일

정보를 입력하시겠습니까? (y/n) :

//이후는 1(학생 추가)와 동일

//3(종료)입력

도서관 관리 프로그램을 종료합니다.

## 1. 출력 화면 설계



//3(내일 반납 예정자 확인)입력

//있으면

내일 반납해야하는 이용자 ID는 001번 003번 입니다.

//없으면

내일 반납해야하는 이용자 ID는 없습니다.

//4(지난 연체날짜 숨기기)입력

정상적으로 지난 연체 날짜를 숨겼습니다.

//5(종료)입력

도서관 관리 프로그램을 종료합니다.

//2(이용자)선택

-----\*이용자 로그인\*-----

ID 입력: 001

-----\*서울과학기술대학교 도서관\*-----

~님, 환영합니다.

1. 도서 목록 보기
2. 도서 검색
3. 예약
4. 대출
5. 반납
6. 예약 및 대출 현황
7. 종료

-----  
메뉴 번호 입력 :



## 1. 출력 화면 설계

// 1(도서 목록 보기) 입력

-----\*1. 도서 목록 보기\*-----

No.	인덱스	책제목	저자	대출여부	예약정보
001	001	일이삼사오육칠팔구십십일	ㅌㅇㅇ	X	X
002	002	설록홈즈사랑해요	코난도일	X	X

// 2(도서 검색하기) 입력

-----\*2. 도서 검색하기\*-----

1. 인덱스로 검색
2. 이름으로 검색
3. 작가로 검색
4. 종료

-----  
메뉴 번호 입력 :

## 1. 출력 화면 설계

//1(인덱스로 검색)선택

책 인덱스 입력 : 002

검색할 책 인덱스 : 002

//2(이름으로 검색)선택

책 이름 입력 : 셜록

검색할 책 이름 : 셜록

//3(작가로 검색)선택

책 작가 입력 : 코난도일

검색할 책 작가 : 코난도일



## 1. 출력 화면 설계

//책이 없는 경우(뒤의 모든 경우에 적용)  
존재하지 않는 책 제목입니다.

1. 메인으로 돌아가기
2. 다시 검색하기

메뉴 번호 입력 :

//책이 있는 경우

-----\*검색결과\*-----

No.	인덱스	책제목	저자	대출여부	예약정보
1	001	셜록홈즈	코난도일	X	2명

1. 메인으로 돌아가기
2. 다시 검색하기
3. 예약하기

메뉴 번호 입력 :

## 1. 출력 화면 설계

//3(예약하기) 입력

-----\*3. 예약하기\*-----

//사용자가 예약이 불가능할 경우  
~님은 더 이상 예약 할 수 없습니다.

예약할 책의 인덱스 입력 :

// 예약할 책이 없는 경우  
책이 존재하지 않습니다.

1. 메인으로 돌아가기
2. 다시 예약하기

메뉴 번호 입력 :

## 1. 출력 화면 설계

// 예약 완료된 경우  
000번 책 예약이 완료되었습니다.  
현재 해당 책의 대기자는 1명이고,  
~님의 예약 권수는 1권입니다.

// 더 이상 예약 할 수 없는 경우  
000번 책은 더 이상 예약 할 수 없습니다.

// 4(대출하기) 입력

-----\*4. 대출하기\*-----

//이용자가 연체자일 경우

~님은 연체되었습니다.

20190623까지 대출이 불가합니다.

//이용자의 대출 한도가 넘어갈 경우

~님은 더 이상 대출이 불가능합니다.

대출할 책의 인덱스 입력 :

// 대출할 책이 없는 경우

책이 존재하지 않습니다.

//이미 대출 중일 경우

대출 중인 책이므로 대출이 불가합니다.

## 1. 출력 화면 설계



//대출 중은 아니지만 예약 중인 경우  
현재 1명이 예약 중이므로 대출이 불가능합니다.

//대출이 완료된 경우  
000번 책 대출이 완료되었습니다.

// 5(반납하기) 입력  
-----\*5. 반납하기\*-----  
//사용자가 대출 한 책이 없을 경우  
반납할 책이 존재하지 않습니다.

//대출한 책이 있는 경우  
현재 대출 중인 책은 001번, 002번 입니다.  
반납할 책의 인덱스 입력 :

//반납이 완료된 경우  
000번 책 반납이 완료되었습니다.  
현재 대출 중인 책은 001번 입니다.

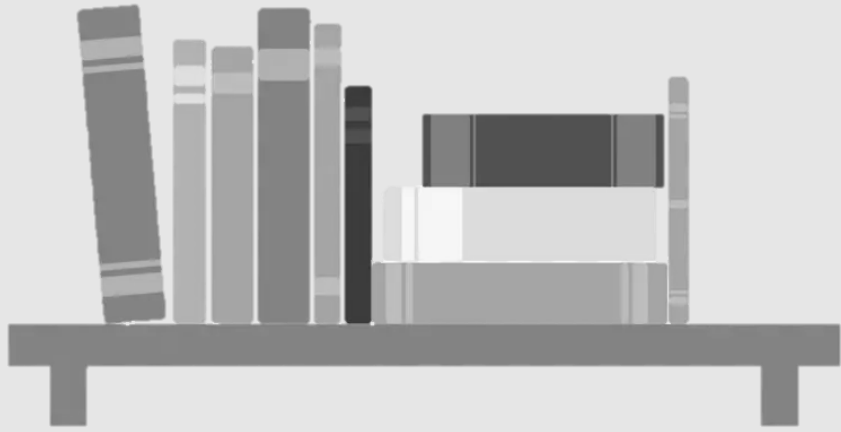
## 1. 출력 화면 설계

//연체된 경우  
해당 책은 연체되었습니다.  
~님은 20190626까지 대출이 불가능합니다.

//잘못된 입력  
대출되지 않은 책의 인덱스입니다.  
// 6(종료) 입력  
도서관 시스템을 종료합니다.



## 1. 출력 화면 설계



## 2. 레코드 설계

1. user.bin : 이용자들의 정보를 저장한 파일 총 40바이트
  - 1) 공통사항 : ID(3), name(6), phone\_num(11), rent\_rsv\_check(2), segmentNum(1), Bcount(1) 총 24바이트
  - 2) Undergrad : grade(1), off\_check(1) 총 26바이트 + 공백 14바이트
  - 3) Postgrad : grade(1), off\_check(1), course(1) 총 27바이트 + 공백 13바이트
  - 4) Grad : X 총 24바이트 + 공백 16바이트
  - 5) Professor : major(16) 총 40바이트
  - 6) Lecture : X 총 24바이트 + 공백 16바이트
  - 7) Staff : department(16) 총 40바이트



## 2. 레코드 설계

2. book.bin : 책 목록 정보를 저장한 파일  
인덱스(3) 책제목(27) 저자(8) 대출&예약여부(2) 총 40바이트

3. Ruser.bin : 대출자 목록 파일  
학번(3) 예약정보(3\*3권) 대출정보(3\*5권) 총 27바이트

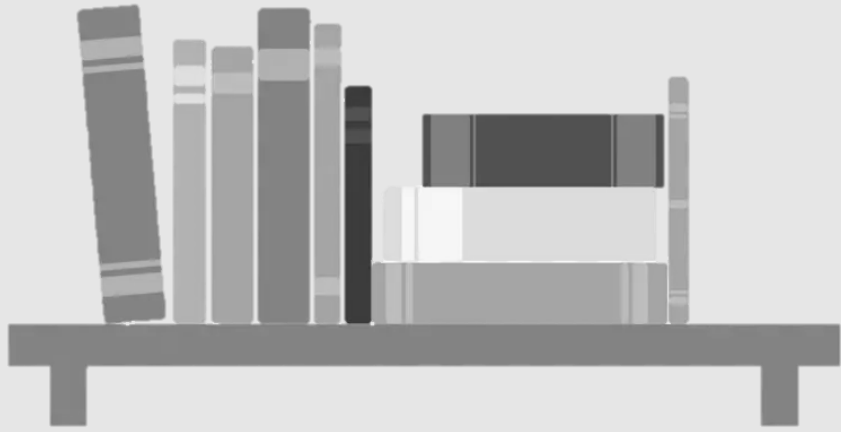
4. Rbook.bin : 대출&예약 중인 책 목록  
인덱스(3) 예약자목록(3\*3명) 대출자정보(3) 총 15바이트

5. rentdate.bin : 반납날짜 관리하는 파일  
학번(3) 대출한 책의 인덱스(3) 반납예정날짜(8) 총 14바이트

6. overdue.bin : 연체자 관리하는 파일  
학번(3) 연체일수(8) 총 11바이트



## 2. 레코드 설계



### 3. Class 설계

User	
멤버 변수	String ID, name, phone_num, off_check, rent_rsv_check, segmentNum, Bcount final int record_size=60
멤버 함수	<ul style="list-style-type: none"> <li>• User(String ID, name, phone_num){}               <ul style="list-style-type: none"> <li>: rent_date 초기화도 해주기</li> <li>• String getUserInfo(){}</li> <li>: 하위 클래스에서 사용하는 함수</li> </ul> </li> <li>• void addRuser(String fn_Ruser){}</li> <li>: 대출&amp;예약한 이용자 파일 생성</li> </ul>

Undergrad	
멤버 변수	String grade, off_check : 0이면 휴학 1이면 재학
멤버 함수	<ul style="list-style-type: none"> <li>• Undergrad(String grade, off_check){}</li> <li>: segmentNum=1, Bcount=3</li> <li>• void addUndergrad(String fn_user){}</li> </ul>

Postgrad	
멤버 변수	String course : 0이면 석사 1이면 박사
멤버 함수	<ul style="list-style-type: none"> <li>• Postgrad(String course){}</li> <li>: segmentNum=2, Bcount=3</li> <li>• void addPostgrad(String fn_user){}</li> </ul>

Grad	
멤버 변수	X
멤버 함수	<ul style="list-style-type: none"> <li>• Grad(){}</li> <li>: segmentNum=3, Bcount=2</li> <li>• void addGrad(String fn_user){}</li> </ul>

### 3. Class 설계

User

Professor

멤버 변수

String major

멤버 함수

- Professor(String major){}  
: segmentNum=4, Bcount=5
- void addProfessor(String fn\_user){}

Lecturer

멤버 변수

X

멤버 함수

- Lecturer(){}
- : segmentNum=5, Bcount=3
- void addLecturer(String fn\_user){}

Staff

멤버 변수

String department

멤버 함수

- Staff(String department){}
- : segmentNum=6, Bcount=3
- void addStaff(String fn\_user){}

### 3. Class 설계



## Book

멤버 변수     String index, Bname, author, Brent\_rsv\_check  
                 final int record\_size=50

멤버 함수     • void addBook(String fn\_book){}

## Library

멤버 변수     너무 많아 생략(각 파일의 이름, 한 레코드 사이즈, 한 레코드안에 변수들 사이즈, 이용자의 중요한 정보 등)

멤버 함수     • void displayBookinfo(int n){}  
                 : 해당 인덱스(or 순서)의 책 목록 출력  
                 • boolean userSetting()  
                 : 사용자가 로그인해서 사용자의 중요한 정보만 따로 저장  
                 • boolean displayBookinfo(RandomAccessFile in, int num){}  
                 : No.가 num이고 해당하는 위치의 한 레코드를 출력  
                 • void listBook(){}  
                 : 전체 책 목록을 보여주는 메뉴얼  
                 • void searchBook(){}  
                 : 책을 검색하는 메뉴얼

• void searchBook\_index, void searchBook\_name, void searchBook\_author  
: 각각 인덱스, 제목, 작가로 검색하는 함수  
• void reserveBook(){  
: 책을 예약하는 메뉴얼  
• int illegalAccess(int n){}  
: 예약 & 대출 & 반납이 불가할 때 띄우는 메뉴, n이 1이면 예약, 2이면 대출, 3이면 반납  
• void rentBook(){  
: 책을 대출하는 메뉴얼  
• void returnBook(){  
: 책을 반납하는 메뉴얼  
• void displayRsvRent(){  
: 이용자의 예약 및 대출 현황 보기

## 3. Class 설계

## ManageMenu

멤버 변수

```
final String fn_student="student.bin"  
final String fn_book="book.bin";
```

멤버 함수

- void displayManageMenu(){}  
: 관리자 메뉴 표시
- void addUserMenu(){}  
: 이용자 추가 메뉴 표시
- String[] getUserInfo(){}  
: 이용자 공통되는 정보 얻기
- int addUndergradMenu(){}- int addPostrgradMenu(){}
  - int addGradMenu(){}  - int addProfessorMenu(){}  - int addLectureMenu(){}  - int addStaffMenu(){}  - : 각 이용자 구분에 따른 정보 추가 메뉴
    - int addBookMenu(){}  
: 책 정보 추가 메뉴
- void displayTomorrowReturn(){}  
: 내일 반납 예정자 표시
- void hidePastOverdue(){}  
: 연체지난 정보는 빈칸으로 채우기

## LibraryMain

멤버 변수

X

멤버 함수

- void main(String[] args)

### 3. Class 설계

## Function

멤버 변수

X

멤버 함수

- int menuNumber(int n){}  
: 올바른 메인 메뉴를 고르는지
- void fillWithBlank(String str, int size){}  
: 나머지를 빈칸으로 채우기
- String inputCorrectSize(int size){}  
: 정해진 크기로 입력받게 하기
- String inputNotOverSize(int size){}  
: 초과되지 않는 크기로 입력받게 하기
- String plusminusone(int n, char ch){}  
: 숫자인 문자를 +1이나 -1해서 문자열로 반환  
: n이 0이면 -1, 1이면 +1
- boolean isLeapYear(int year){}  
: year이 윤년이면 true
- int getDaysInMonth(int year, in month){}  
: year의 month에 전체 날짜 수 반환
- int[] toIntDate(String date\_s){}  
: date\_s를 인트형 배열로 변환
- String toStringDate(int[] date){}  
: date를 문자열로 변환
  - String getDate(){  
: 오늘 날짜 반환

## UserMenu

멤버 변수

final String fn\_user="user.bin",  
fn\_Ruser="Ruser.bin", fn\_book="book.bin",  
fn\_Rbook="Rbook.bin";

멤버 함수

- void displayUserMenu(){  
: 이용자 메뉴 보여주는 함수

- String plusDate(int n, String org\_date){}  
: 날짜 더해서 반환(n<29)
- int getTotalDays(int[] date, int year) {}  
: 기준 년도부터 전체 날 구하기
- int getExceedDay(String date\_s, String org\_date\_s) {}  
: 날짜가 비교 날짜보다 몇일 더 지났는지, 안지났거나 같으면 0

## 3. Class 설계