**chapter 1**

도서 관리 프로그램을 만들기 위해 책의 권수를 파악하는 프로그램을 만들어 보자.

1층에 들어갈 수 있는 전체 원형 책장의 개수와 이를 이용해 가능한 도서의 총 수를 구하자.

<조건>

도서관은 원형으로 되어 있으며, 내벽의 형태에 맞춰 둥근 형태의 원형 책장이 벽전체에 들어서 있다.

1층의 공간 형태와 원형 책장의 배치 형태는 다음과 같다.

계단

입구

1층 공간의 지름은 총 22m 이다.

책장의 길이는 2m 이다.

책장 의 선반당 들어갈 수 있는 총 책의 권수는 70권이다.

책장의 선반은 총 14개 이다.

소수점 이하는 반 내림(강제형변환)한다.

**chapter 3**

도서관리 프로그램을 만들어 보자.

<조건>

- 추가 할 메뉴와 각 기능은 다음과 같다.

1. 전체 도서 입력
   1. 전체 도서 입력은 순서대로 책의 제목, 내용을 입력 받아 저장한다.
   2. 전체 도서의 수는 총 3권이다.
   3. 입력이 완료되면 프로그램의 처음 UI로 돌아간다.
2. 전체 도서 출력
   1. 입력한 전체 도서의 제목, 내용을 순서대로 출력한다.
   2. 출력이 완료되면 프로그램의 처음 UI로 돌아간다
3. 프로그램 종료
   1. 프로그램을 종료한다.

<실행화면>

|  |
| --- |
| 1. 프로그램 시작 시 |
| ===== 도서관리 프로그램 =====  1. 전체 도서 입력  2. 전체 도서 출력  0. 프로그램 종료  : |

|  |
| --- |
| 1. 전체 도서 입력 |
| ===== 도서관리 프로그램 =====  1. 전체 도서 입력  2. 전체 도서 출력  0. 프로그램 종료  : 1  ===== 1. 전체 도서 입력 =====  [1/3]  도서 제목:  [PB] JAVA SE  도서 내용:  Hello JAVA SE!!  [2/3]  도서 제목:  [PB] JAVA EE  도서 내용:  Hi JAVA EE~~  [3/3]  도서 제목:  [PB] JAVA SE/EE  도서 내용:  JAVA SE/EE ~~~~!!!!  ===== 도서관리 프로그램 =====  1. 전체 도서 입력  2. 전체 도서 출력  0. 프로그램 종료  : |

|  |
| --- |
| 1. 전체 도서 출력 |
| ===== 도서관리 프로그램 =====  1. 전체 도서 입력  2. 전체 도서 출력  0. 프로그램 종료  : 2  ===== 2. 전체 도서 출력 =====  [1/3]  - 도서제목:  [PB] JAVA SE  - 도서내용:  Hello JAVA SE!!  [2/3]  - 도서제목:  [PB] JAVA EE  - 도서내용:  Hi JAVA EE~~  [3/3]  - 도서제목:  [PB] JAVA SE/EE  - 도서내용:  JAVA SE/EE ~~~~!!!!  ===== 도서관리 프로그램 =====  1. 전체 도서 입력  2. 전체 도서 출력  0. 프로그램 종료  : |

|  |
| --- |
| 1. 프로그램 종료 |
| ===== 도서관리 프로그램 =====  1. 전체 도서 입력  2. 전체 도서 출력  0. 프로그램 종료  : 0  [Exit] 프로그램을 종료합니다... |

**chapter 4**

chapter 3에서 만들었던 프로그램에 객체지향 개념을 적용시켜 보자.

<조건>

- 세 개의 클래스로 각각의 역할을 분담한다.

1. BooK  
   : 책의 정보를 담는 클래스
   1. 필드
      1. int bookNum: 책의 번호
      2. String title: 책의 제목
      3. String content: 책의 내용
   2. 메서드
      1. setter: 각 필드의 정보를 설정할 수 있는 기능
      2. getter: 각 필드의 정보를 가져올 수 있는 기능
      3. toString(): 책의 정보를 모두 출력하는 기능(오버라이드 메서드)
   3. 생성자
      1. Book(): 기본 생성자
      2. Book(int bookNum, String title, String content): 생성시 책의 정보를 모두 저장
2. BookManager  
   : 도서관리 프로그램의 기능을 담당하는 클래스
   1. 필드
      1. Book[] bookArray: 생성된 Book를 저장할 수 있는 배열, 크기는 10
      2. int baIndex: bookArray 에 인덱스로 사용되는 변수
   2. 메서드
      1. insertBook(): isFull() 메서드로 배열이 가득 찾는지 체크 한 후, 가득 차지 않았다면 전달받은 Book 객체를 만들고 배열에 집어넣는다.
      2. getBookList(): bookArray 에 저장되어 있는 책의 개수만큼만 들어 있는 배열을 새로 만들어 반환한다.
   3. 생성자
      1. BookManager(): 기본 생성자. 객체 생성시 bookArray의 크기는 10, baindex는 0으로 초기화 한다.
3. BookMain  
   : 프로그램의 시작인 main() 메서드가 있는 클래스
   1. 메서드
      1. main(): 프로그램의 시작점인 메인 메서드  
         메뉴 UI를 여기에 구현한다.  
         프로그램 종료 이외에 각 기능의 실행이 완료되면 처음 화면으로 돌아가서 프로그램실행을 반복한다

<클래스 다이어그램>

|  |
| --- |
| C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ch4.classdiagram-2.png |

<실행화면>

|  |
| --- |
| 프로그램 시작 시 |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력(전체)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 도서 입력 |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력(전체)  0. 프로그램 종료  : 1  ===== 1. 도서 입력 =====  책 번호:  001  제목:  [PB] JavaSE  내용:  Hello JAVA SE!  ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력(전체)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 도서 출력(2권만 입력한 경우) |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력(전체)  0. 프로그램 종료  : 2  ===== 2. 도서 전체 출력 =====  책 번호:  1  책 제목:  [PB] JavaSE  책 내용:  Hello JAVA SE!  책 번호:  2  책 제목:  [PB] JAVA EE  책 내용:  Hello JAVA EE~~~  ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력(전체)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 프로그램 종료 시 |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력(전체)  0. 프로그램 종료  : 0  [Exit] 프로그램을 종료합니다... |

**chapter 5**

chapter 4에서 만들었던 프로그램의 Book 클래스에 상속을 적용시켜 보자.

<조건>

- 여섯 개의 클래스로 각각의 역할을 분담한다.

1. Book  
   : 이전과 동일  
   : PaperBook 과 AudioBook 의 부모 클래스
2. PaperBook  
   : 일 층의 책들을 나타내는 클래스  
   : Book 클래스를 상속받는다.
   1. 필드
      1. pbData: PaperBook의 전용 데이터를 담는 변수(임의로 문자열로 지정함)
   2. 메서드
      1. setPbData: 필드 pbData의 정보를 설정할 수 있는 기능
      2. getpbData: 필드 pbData의 정보를 가져올 수 있는 기능
      3. toString(): 책의 정보를 모두 출력하는 기능(오버라이드 메서드)
   3. 생성자
      1. PaperBook(): 기본 생성자
      2. PaperBook(int bookNum, String title, String content, String pbData): 생성 시 책의 정보를 모두 저장
3. AudioBook  
   : 이 층의 책들을 나타내는 클래스  
   : Book 클래스를 상속받는다.
   1. 필드
      1. abData: AudioBook의 전용 데이터를 담는 변수(임의로 문자열로 지정함)
   2. 메서드
      1. setAbData: 필드 abData의 정보를 설정할 수 있는 기능
      2. getAbData: 필드 abData의 정보를 가져올 수 있는 기능
      3. toString(): 책의 정보를 모두 출력하는 기능(오버라이드 메서드)
   3. 생성자
      1. AudioBook(): 기본 생성자
      2. AudioBook(int bookNum, String title, String content, String abData): 생성 시 책의 정보를 모두 저장
4. BookManager  
   : 도서관리 프로그램의 기능을 담당하는 클래스  
   : 기본적으로 chapter 4의 클래스와 동일  
   : 추가적인 사항은 아래와 같다.
   1. 필드  
      : 추가사항 없음
   2. 메서드
      1. searchBook(int bookNum): 전체 배열의 Book 객체에서 전달받은 bookNum 과 같은 bookNum을 가지고 있는지를 체크 한 후 일치하는 Book 객체를 반환한다.
      2. getBookListByType(int typeNum): 전달받은 typeNum을 체크하여 typeNum에 해당하는 PaperBook/AudioBook 객체의 배열을 만들어 반환한다.
      3. checkBookType(String type): 책 입력시 제목의 앞부분 네 글자를 가져와서 가져온 문자열을 반환한다.
   3. 생성자
      1. BookManager(int bookArraySize): 전달받은 bookArraySize 크기의 bookArray 를 생성한다.
5. BookCUI  
   : 프로그램의 CUI(Command User Interface) 즉, 메뉴 출력 기능을 담당하는 클래스
   1. 필드
      1. input: 사용자의 입력값을 처리하는 Scanner
      2. menuNum: 사용자가 입력한 메뉴값을 저장하는 변수
      3. bm: BookManager 객체를 생성 후 보관하기 위한 변수
   2. 메서드
      1. cuiStart(): 전체 메뉴 출력 기능, 사용자에게 menuNum을 입력받아 각 값에 해당하는 기능을 호출하고 호출이 완료되면 다시 돌아와 전체메뉴를 다시 출력한다.
      2. insertBook(): 입력 메뉴 출력 기능
      3. printBook(): 출력 메뉴 출력 기능
      4. searchBook(); 검색 메뉴 출력 기능
   3. 생성자
      1. BookCUI(int bookArraySize): 생성자 호출시 input 객체 생성, munuNum 초기화, bm 을 생성한다. bm 생성시 전달받은 bookArraySize 를 bm의 생성자에게 전달해준다.
6. BookMain  
   : 프로그램의 시작인 main() 메서드가 있는 클래스
   1. 메서드
      1. main(): 실행 시 BookCUI 클래스를 만들고 cuiStart() 메서드를 호출한다.

<클래스 다이어그램>

|  |
| --- |
| C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ch5.classdiagram-3.png |

<실행화면>

|  |
| --- |
| 1. 프로그램 시작 시 |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 도서 입력(AudioBook) |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : 1  ===== 1. 도서 입력 =====  책 번호:  1  제목:  [AB] Java Audio SE  내용:  Hello Java Audio SE~  AudioBook Data:  Audio Data One  ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 도서 입력(PaperBook) |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : 1  ===== 1. 도서 입력 =====  책 번호:  2  제목:  [PB] Java Paper SE  내용:  Hello Java Paper SE!!  PaperBook Data:  paper Data one  ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 도서 출력(PaperBook) |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : 2  ===== 2. 도서 출력 =====  1. 전체 출력  2. PaperBook 출력  3. AudioBook 출력  :  2  책 번호:  2  책 제목:  [PB] Java Paper SE  책 내용:  Hello Java Paper SE!!  PaperBook Data:  paper Data one  ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 도서 출력(AudioBook) |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : 2  ===== 2. 도서 출력 =====  1. 전체 출력  2. PaperBook 출력  3. AudioBook 출력  :  3  책 번호:  1  책 제목:  [AB] Java Audio SE  책 내용:  Hello Java Audio SE~~  AudioBook Data:  Audio Data One  ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 도서 출력(전체) |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : 2  ===== 2. 도서 출력 =====  1. 전체 출력  2. PaperBook 출력  3. AudioBook 출력  :  1  책 번호:  1  책 제목:  [AB] Java Audio SE  책 내용:  Hello Java Audio SE~~  AudioBook Data:  Audio Data One  책 번호:  2  책 제목:  [PB] Java Paper SE  책 내용:  Hello Java Paper SE!!  PaperBook Data:  paper Data one  ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 도서 검색 |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : 3  ===== 3. 도서 검색(책 번호) =====  검색할 책 번호:  1  책 번호:  1  책 제목:  [AB] Java Audio SE  책 내용:  Hello Java Audio SE~~  AudioBook Data:  Audio Data One  ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : |

|  |
| --- |
| 1. 프로그램 종료 |
| ===== 도서관리 프로그램 =====  1. 도서 입력  2. 도서 출력  3. 도서 검색(책번호)  0. 프로그램 종료  : 0  [Exit] 프로그램을 종료합니다... |

**chapter 6**

chapter 5 에서 만들었던 프로그램에 예외처리를 추가해 보자.

<조건>

- 일곱 개의 클래스로 각각의 역할을 분담한다.

1. Book: chapter 5 와 동일
2. PaperBook: chapter 5 와 동일
3. AudioBook: chapter 5 와 동일
4. BookManager  
   : 추가하는 예외처리 부분은 다음과 같다.
   1. 메서드
      1. isFull(): 추가되는 메서드. 배열의 길이를 체크하여 가득 찼을 때 false. 가득 차지 않았을 때 true 를 반환한다.
      2. insertBook(Book):
      3. searchBook(int):
      4. getBookList():
      5. getBookListByType(int):
      6. checkBookType(String):
5. BookCUI
6. BookMain
7. BookException

<클래스 다이어그램>

<실행화면>