

# Prolog를 이용한 약인공지능

## 도서 대출 시스템 개발

수리논리학 01분반, 권기향 교수님

컴퓨터공학과

팀원 : 2343411 이재원, 2343388 김우진

### 목차

#### \* 서론

#### \* 본론

- 1) Prolog, 인공지능 소개
- 2) 도서관 대출 관리 시스템 구조, 기능
- 3) 구현 및 코드 설명

#### \* 결론

#### \* 출처 및 참고문헌

# 서론

## 1. 프로젝트 개요

본 프로젝트는 Prolog를 통해 도서관 대출 시스템을 구현하는 것을 목표로 하였다. 이 시스템은 도서 정보 관리, 도서 대출 및 반납, 대출 상태 확인, 유저 보드, 도서 검색 기능을 포함한다. Prolog의 논리 프로그래밍을 활용하여 복잡한 데이터베이스와 규칙 기반 로직을 효과적으로 처리할 수 있었다.

## 2. 프로젝트 배경

Prolog를 이용한 약인공지능 시스템 중, 도서관 대출 관리 시스템을 만들기로 정했다. 그 이유는 도서관 시스템은 도서 관리와 효율적인 대출/반납 절차가 필요하다. Prolog의 논리 프로그래밍 특성은 도서 검색, 대출 상태 관리 등에서 효율적으로 동작하여 사용자 편의성을 높이고, 도서관 운영의 개선을 할 수 있다.

## 3. 프로젝트 목표

- 데이터베이스 구축 (도서, 대출자 정보, 도서 대출 상태)
- 도서 검색 기능, 도서 대출 및 반납 기능 구현
- 유저 인터페이스 구현

# 본론

## 1. Prolog, 인공지능 소개

Prolog는 논리형 프로그래밍 언어로, 객체와 객체 간의 관계에 관한 문제를 해결한다. 프로그램 자체가 논리식의 모양을 띄고 있으며, 논리식을 간결하게 표현할 수 있다. 인공지능, 자연언어 처리 분야에 사용된다.

인공지능은 인간의 학습능력, 추론능력, 지각능력을 인공적으로 구현하려는 컴퓨터 과학의 세부 분야로, 약인공지능, 강인공지능이 있다. 약인공지능은 좁은 AI를 의미하며, 하나의 좁은 작업, 간단한 문제를 해결하는 데에 집중한다.

## 2. 도서관 대출 관리 시스템 구조, 기능

데이터베이스: 도서 정보, 대출 상태, 대출자 정보에 관한 데이터베이스 생성

- 도서 정보(book/4): 바코드, 제목, 저자, 장르 정보를 포함
- 대출 상태(loan\_status/2): 도서의 대출 여부를 관리
- 대출자 정보(borrower/3): 대출자의 이름, 전화번호, 대출 도서의 바코드를 저장

주요 기능: 도서의 대출 및 반납, 도서 목록의 출력, 도서 검색 기능, 유저보드

주요 기능에 대한 알고리즘 순서도



### 3. 구현 및 코드 설명

- 사용자 인터페이스 화면 (시연 영상은 ppt파일에 포함하였다)

## Welcome to the Library



book\_List(도서 목록), Check\_out(대출 및 반납), Usr\_Board(유저보드) 버튼

- 도서 목록 출력

| Barcode | Title                   | Author              | Genre     |
|---------|-------------------------|---------------------|-----------|
| 1001    | The Great Gatsby        | F. Scott Fitzgerald | Fiction   |
| 1002    | 1984                    | George Orwell       | Dystopian |
| 1003    | To Kill a Mockingbird   | Harper Lee          | Fiction   |
| 1004    | A Brief History of Time | Stephen Hawking     | Science   |
| 1005    | Pride and Prejudice     | Jane Austen         | Romance   |
| 1006    | The Catcher in the Rye  | J.D. Salinger       | Fiction   |
| 1007    | The Hobbit              | J.R.R. Tolkien      | Fantasy   |
| 1008    | The Da Vinci Code       | Dan Brown           | Thriller  |
| 1009    | Moby-Dick               | Herman Melville     | Adventure |
| 1010    | Brave New World         | Aldous Huxley       | Dystopian |

- 도서의 목록을 데이터프레임 형태를 출력한다.(바코드, 제목, 저자, 장르)

- 제목 검색, 장르 검색 기능

(도서의 데이터베이스는 예시로 10개만 저장하였다.)

## - 도서 대출 및 반납

### Borrow Book

Name:

Phone:

Barcode:

도서의 대여할 때는

개인정보(이름, 전화번호)와 도서의 바코드 번호를 입력해야 한다.

## - 유저보드

### Book Bulletin Board

Title:  Author:

Content-type: text/html; charset=UTF-8

### Book Posts

| Title   | Author           |
|---|------------------|
| Prolog  | Alain Colmerauer |
| <input type="button" value="Go to User Recommend"/> |                  |
| <input type="button" value="Home"/>                 |                  |

사용자가 임의로 추가하고 싶은 책과 저자를 입력하면 게시판처럼 테이블이 작성된다.

위의 사진에서는 Title: Prolog, Author: Alain Colmerauer 를 추가하였다.

## - 코드 설명

library\_db.pl

%1 book db

책(바코드, 제목, 저자, 장르). 도서의 데이터베이스, 도서에 관한 정보를 저장하고 있다.

```
%1 book database (Barcode, Title, Author, Genre)
book(1001, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Fiction').
book(1002, '1984', 'George Orwell', 'Dystopian').
book(1003, 'To Kill a Mockingbird', 'Harper Lee', 'Fiction').
book(1004, 'A Brief History of Time', 'Stephen Hawking', 'Science').
book(1005, 'Pride and Prejudice', 'Jane Austen', 'Romance').
book(1006, 'The Catcher in the Rye', 'J.D. Salinger', 'Fiction').
book(1007, 'The Hobbit', 'J.R.R. Tolkien', 'Fantasy').
book(1008, 'The Da Vinci Code', 'Dan Brown', 'Thriller').
book(1009, 'Moby-Dick', 'Herman Melville', 'Adventure').
book(1010, 'Brave New World', 'Aldous Huxley', 'Dystopian').
```

%2 loan\_status\_db

대출\_상태(바코드, 상태). 책의 대출 상태는 available: 대출 가능, checked\_out : 대출 불가로 나타난다.

사용자가 책을 대출, 반납할 때마다 추가되거나 삭제되어야 하기 때문에 ':- dynamic loan\_status/2.' 를 추가

```
%2 loan_status_db (Barcode, Status)
:- dynamic loan_status/2.
loan_status(1001, checked_out).
loan_status(1002, available).
loan_status(1003, available).
```

%3 borrower db

대여자(이름, 전화번호, 바코드)

사용자가 책을 대출, 반납할 때마다 추가되거나 삭제되어야 하기 때문에 ':- dynamic borrower/3.' 를 추가

```
:- dynamic borrower/3.
borrower('Tom', '010-1234-5678', 1001).
```

%5 title search

제목 검색 기능, Part에 제목 일부분 저장 후, Title과 비교 후 참인 것들을 book\_status(%7) 함수를 통해, 책 정보를 출력한다.(Barcode, Title, Author, Genre, Status)

```
%5 title search
title_search(Part, Title) :-
    book(Barcode, Title, _, _),
    sub_string(Title, _, _, Part),
    book_status(Barcode).
```

%6 classification\_genre

장르 검색 기능, 장르를 입력하면, 해당 장르인 책들을 book\_status(%7) 함수를 통해, book\_list와 동일한 과정으로 책 정보를 출력한다.(Barcode, Title, Author, Genre, Status)

```
classification_genre(Genre) :-
    book(Barcode, _, _, Genre),
    book_status(Barcode).
```

%7 book\_status

Barcode를 입력하면, loan\_status(%2)와 book(%1)을 통해 각 정보(Barcode, Title, Author, Genre, Status)를 가져온 후, 데이터프레임을 구성할 'Barcode: ~w, Title: ~w, Author: ~w, Genre: ~w, Status: ~w' 한 줄을 생성한다.

```
%7 book_status (Barcode, Title, Author, Genre, Status)
book_status(Barcode) :-
    loan_status(Barcode, Status),
    book(Barcode, Title, Author, Genre),
    format('Barcode: ~w, Title: ~w, Author: ~w, Genre: ~w, Status: ~w', [Barcode, Title, Author, Genre, Status]), nl.
```

%8 book\_loan

Name, Phone\_number, Barcode 책 대여에 필요한 정보를 기입 후 checkout\_book(%9)에 해당 매개변수들을 가져간다.

```
book_loan :-
    write('Enter your name : '),
    read(Name),
    write('Enter your phone number : '),
    read(Phone_number),
    write('Enter the book barcode'),
    read(Barcode),
    checkout_book(Name, Phone_number, Barcode).
```

%9 checkout\_book -1

loan\_status(%2)로 대여가 가능한지 확인(available) 후, loan\_status(%2)와 borrower(%3)의 db 수정  
수정과정

retract : loan\_status(%2)에 기존의 (Barcode, available(대여 가능)) 삭제

assertz : loan\_status(%2)에 새로운 (Barcode, checked\_out(대여 불가)) 생성

assertz : borrower(%3)에 새로운 대여자 정보 생성(Name, Phone\_number, Barcode)

만약에 해당 %9의 첫 명령문 loan\_status를 수행할 때, 두번째인자가 available이 아니라 checked\_out이라면

거짓이되므로 (-1)은 종료되고 (-2)를 실행

%9-2

loan\_status(%2)로 대여가 불가능한지 확인, 불가능이 확인되면 불가능하다는 메시지 write 출력

```
%9 checkout_book-1
checkout_book(Name, Phone_number, Barcode) :-
    loan_status(Barcode, available),
    retract(loan_status(Barcode, available)),
    assertz(loan_status(Barcode, checked_out)),
    assertz(borrower(Name, Phone_number, Barcode)),
    write('Book checked out successfully. '), nl.
% -2
checkout_book(Name, Phone_number, Barcode) :-
    loan_status(Barcode, checked_out),
    write('Book is already checked out. '), nl.
```

%10 return\_book

Barcode를 넣어서 반납, loan\_status(%2)를 통해, 미반납 상태인지 확인 후 참이라면, 반납 수행,  
loan\_status(%2)와 borrower(%3) db 수정

수정과정

retract : loan\_status(%2)에 기존의 (Barcode, checked\_out(대여 불가)) 삭제

assertz : loan\_status(%2)에 새로운 (Barcode, available(대여 가능)) 생성

retract : borrower(%3)에 기존의 대여자 정보 삭제(Name, Phone\_number, Barcode)



```
return_book(Barcode) :-  
    loan_status(Barcode, checked_out),  
    retract(loan_status(Barcode, checked_out)),  
    assertz(loan_status(Barcode, available)),  
    retract(borrower(Name, Phone_number, Barcode)).
```

library\_server.pl

```
:- use_module(library(http/thread_httpd)).  
:- use_module(library(http/http_dispatch)).  
:- use_module(library(http/html_write)).  
:- use_module(library(http/http_parameters)).
```

thread\_httpd, http\_dispatch는 prolog에서 html을 사용하기 위해서 필수로 가지는 모듈

```

server(Port) :-
    http_server(http_dispatch, [port(Port)]).

%
:- http_handler(root(.), home_page, []).
:- http_handler(root(books), books_page, []).
:- http_handler(root(checkout), checkout_page, []).
:- http_handler(root(search), search, []).
:- use_module(library(http/http_client)).

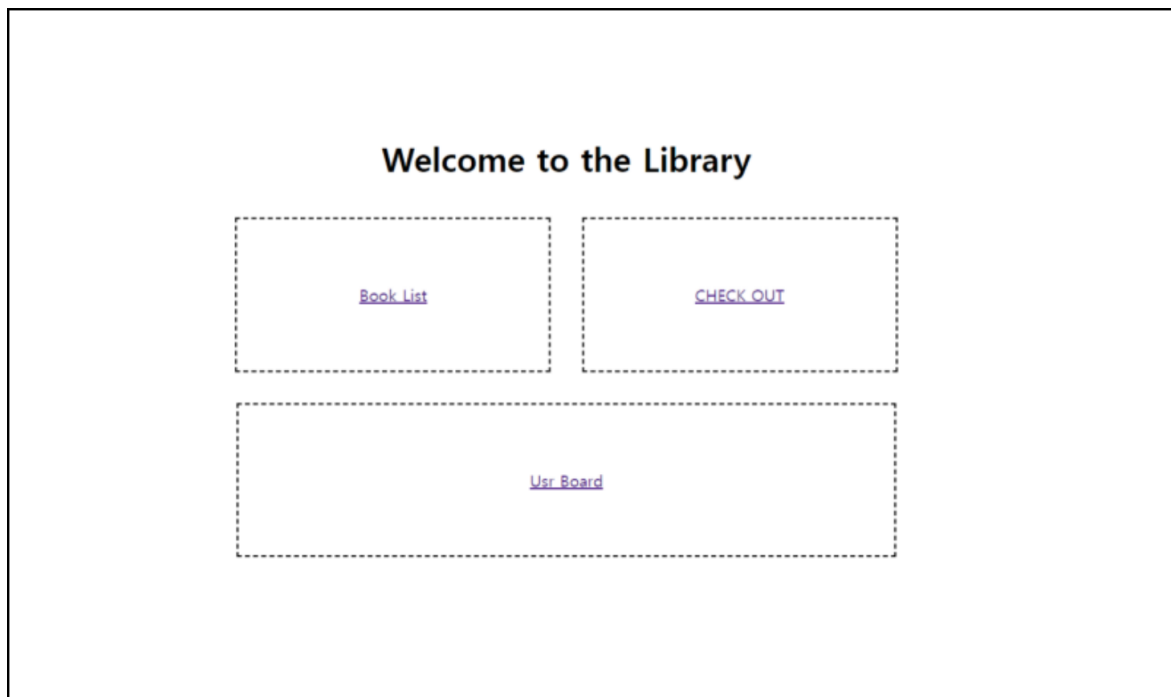
```

http\_server를 port를 지정하면 http 서버에 들어갈 수 있음

예) port = 8080, address = localhost:8080

http\_handler를 통해서 html 페이지를 여러 개 설정할 수 있음

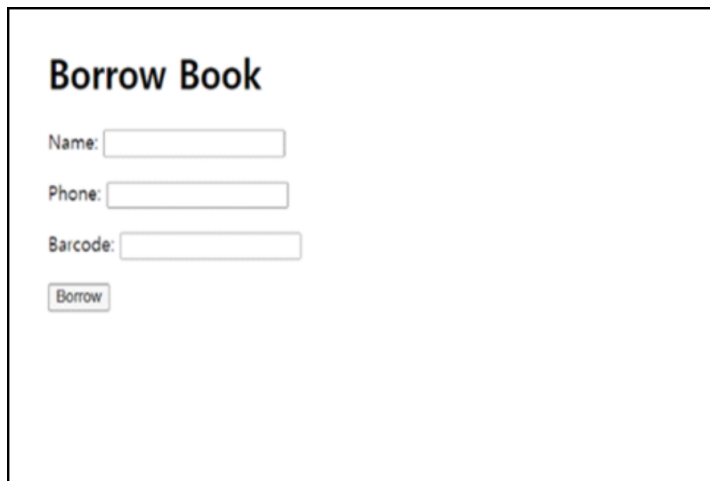
root는 현재 위치를 나타내며, 내가 이미 존재하고 있는 페이지를 root로 설정



h1 태그로 "Welcome to the Library" 제목을 표시

div 태그를 이용해서 Book List, CHECK OUT, USR Board, 박스로 묶음

anchor인 a와 주소 입력 href 태그를 사용해서 각각 다음 페이지로 넘어가는 하이퍼링크 제작



**Borrow Book**

Name:

Phone:

Barcode:

input태그를 이용해서 입력받을 박스를 제작

Button 태그를 이용해서 Borrow 버튼을 만들

Borrow 버튼에도 a, href 태그를 이용해서 특정 페이지로 이동

(library\_server.pl은 화면을 구현하는 코드가 너무 길어서 추가적인 내용은 깃허브 주소에서 확인  
요함)

## 결론

### 배운 점

#### 1. 논리 프로그래밍의 기능

Prolog를 배우면서 논리적 사고와 규칙 기반 프로그래밍에 대해 이해할 수 있었다. 이를 통해 복잡한 데이터베이스 질의와 규칙을 간단하게 구현할 수 있었다.

#### 2. 협업과 커뮤니케이션

팀원과 프로젝트를 진행하면서 효과적인 협업과 커뮤니케이션의 중요성을 느꼈다. 서로의 아이디어를 공유하고, 문제를 함께 해결하는 과정에서 전달력을 길렀다.

#### 3. 프로젝트 관리

프로젝트를 계획하고, 각 단계별로 진행 상황을 점검하며 일정 관리의 중요성을 깨달았다. 이러한 경험은 향후 다른 프로젝트를 진행할 때 좋은 도움이 될 것이다.

#### 4. 기술적 성장

Prolog뿐만 아니라 html과 데이터베이스 관리에 대한 학습을 할 수 있었다. 이러한 기술적 성장

은 앞으로 더 복잡한 시스템을 설계하고 구현하는 데 도움이 될 것이다.

## 결과

도서관 대출 시스템은 도서 대출 및 반납을 원활하게 처리하기 위한 도구로서 이용자들에게 서비스를 제공하도록 만들었다. 이 시스템을 통해 도서관은 도서의 대출 상태를 파악하고, 도서의 정보를 쉽게 관리할 수 있다. 또한 이용자는 간단한 절차를 통해 도서를 대출하고 반납할 수 있으며, 대출한 도서의 상태를 확인할 수 있다.

이러한 시스템은 도서관의 운영 효율성을 향상시키고 이용자들에게 편의성을 제공하여, 도서관 서비스를 발전시킨다.

## 출처, 참고문헌

<https://ko.wikipedia.org/wiki/%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5>

[https://ko.wikipedia.org/wiki/%ED%94%84%EB%A1%A4%EB%A1%9C%EA%B7%B8\\_\(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D\\_%EC%96%B8%EC%96%B4\)](https://ko.wikipedia.org/wiki/%ED%94%84%EB%A1%A4%EB%A1%9C%EA%B7%B8_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D_%EC%96%B8%EC%96%B4))

<https://youtu.be/SyxxWpFwMGs?si=RNq6hOfCf1CbcbTh>

<https://youtu.be/0fyE8Fuj3sg?si=NvrnVRfi71-nG5ae>

chatgpt

## 깃허브 주소

[https://github.com/Plametal/prolog\\_pro/tree/063e2a30337e41bbc90621bed201c0ce69c0cc81/%EC%B5%9C%EC%A2%85%EA%B2%B0%EA%B3%BC%EB%AC%BC](https://github.com/Plametal/prolog_pro/tree/063e2a30337e41bbc90621bed201c0ce69c0cc81/%EC%B5%9C%EC%A2%85%EA%B2%B0%EA%B3%BC%EB%AC%BC)

## 내용 스크린샷

|                                      |  |
|--------------------------------------|--|
| Prolog를 이용한 약인공지능도서 대출 시스템 개발_완.docx | 최종 발표본   |
| library_db.pl                        | Add files via upload                           |
| library_server.pl                    | Rename library_server to library_server.pl     |
| prolog를 이용한 약인공지능 도서 대출 시스템 개발.pptx  | Rename prolog를 이용한 약인공지능 도서 대출 시스템 개발.pptx to  |
| 시연.zip                               | Add files via upload                           |
| 주석 설명.txt                            | Update and rename 주석 설명.txt to 최종결과물/주석 설명.txt |