



**МИНОБОРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**«Московский государственный технологический университет «СТАНКИН»  
(ФГБОУ ВО «МГТУ «СТАНКИН»)**

---

**Институт  
информационных  
технологий**

**Кафедра  
информационных систем**

**Основная образовательная программа 09.03.02**

**«Информационные системы и технологии»**

**Отчет по дисциплине «Интеллектуальные и экспертные системы»**

**по лабораторной работе № 2**

**по теме: «Работа со списками в среде SWI-Prolog»**

Студент  
группы ИДБ-21-07  
Преподаватель

Музафаров К. Р.  
Перепелкина Ю.В.

Москва 2023 г.

**ОГЛАВЛЕНИЕ**

ЗАДАНИЕ.....	3
ВВЕДЕНИЕ .....	4
ОПИСАНИЕ РАБОТЫ.....	5
ЛИСТИНГ КОДА .....	15
ВЫВОДЫ .....	15
СПИСОК ЛИТЕРАТУРЫ .....	27

## ЗАДАНИЕ

Предметная область – библиотека. Каждая книга может быть описана структурой: название, автор, издание. Автор может быть описан структурой: фамилия, имя, год рождения. Издание может быть описано структурой: издательство, номер издания, год издания, количество страниц, цена, гонорар автора.

Реализовать следующие типы запросов:

1. Найти автора, у которого книга переиздавалась максимальное число раз;
2. Найти все книги, изданные более одного раза;
3. Найти все книги, изданные в указанном издательстве в заданном году;
4. Найти все книги указанного автора;
5. Найти все книги, цена которых превышает заданную сумму.
6. Разработать графический интерфейс для доступа к базе знаний.

Требования к графическому интерфейсу:

- реализация всех запросов с форматированным выводом результатов;
- использование разнообразных элементов управления (вкладки, текстовые поля, радио-кнопки, выпадающие списки, кнопки);
- возможность добавления фактов в динамическую БД;
- возможность сохранения динамической БД в файл.

## **ВВЕДЕНИЕ**

**Цели работы:** рассмотреть основные конструкции рекурсивно-логического программирования: процедуры рекурсии и построение списков на языке Prolog и изучить приемы работы с интерпретатором SWI Prolog; приобрести практические навыки создания, отладки и выполнения программ в среде SWI Prolog.

## ОПИСАНИЕ РАБОТЫ

В ходе лабораторной работы была создана экспертная система по предметной области «Библиотека».

Факты, заданные системе представлены на рисунке 1.



```

1 :- use_module(library(pce)).
2 :- dynamic book/3.
3
4 % Факты
5 book('The Secrets of Prolog',
6     author('Johnson', 'Emily', 1974),
7     edition('TechPress', 1, 2015, 320, 4000, 5000)).
8
9 book('Advanced SWI-Prolog Techniques',
10    author('Johnson', 'Emily', 1974),
11    edition('TechPress', 2, 2018, 350, 4500, 5500)).
12
13 book('Mysteries of the Cosmos',
14    author('Hawking', 'Stephanie', 1982),
15    edition('Universe Publishing', 1, 2017, 280, 3000, 4000)).
16
17 book('Mysteries of the Cosmos',
18    author('Hawking', 'Stephanie', 1983),
19    edition('Universe Publishing', 2, 2017, 280, 3000, 4000)).
20
21 book('Journeys through Time',
22    author('Hawking', 'Stephanie', 1982),
23    edition('Universe Publishing', 1, 2020, 300, 3500, 4500)).
24
25 book('A Programmer\'s Guide to Logic',
26    author('Turing', 'Alan', 1912),
27    edition('CompSci Books', 3, 1995, 250, 2500, 3000)).
28
29 book('The Journey of Prolog',
30    author('Adams', 'Douglas', 1952),
31    edition('SciFi World', 1, 1990, 300, 1500, 1500)).
32
33 book('The Journey of Prolog',
34    author('Adams', 'Douglas', 1952),
35    edition('SciFi World', 2, 1995, 320, 1800, 1800)).
36
  
```

Рис. 1. Факты

```
book('The Journey of Prolog',
    author('Adams', 'Douglas', 1952),
    edition('SciFi World', 3, 2000, 350, 2000, 2000)).
```

```
book('Exploring AI',
    author('Curie', 'Marie', 1867),
    edition('Tech Advances', 1, 2010, 250, 2500, 2500)).
```

```
book('Exploring AI',
    author('Curie', 'Marie', 1867),
    edition('Tech Advances', 2, 2015, 300, 3000, 3000)).
```

```
book('Mysteries of Space',
    author('Hawking', 'Stephen', 1942),
    edition('Universe Publishing', 1, 2005, 400, 4000, 4000)).
```

Рис. 2. Факты

Далее была написана логика: алгоритмы, предикаты и функции для выполнения заданий и реализации графического интерфейса. (рис. 2-7).

*% Вспомогательные функции*

*% 1 Найми автора, у которого книга переиздавалась максимальное число раз:*

```
author_most_reprints(AuthorName) :-
    findall(Reprints-LastName-FirstName,
        (book(_, author(LastName, FirstName, _), edition(_, EditionNumber, _, _, _, _)),
         aggregate(count, book(_, author(LastName, FirstName, _), edition(_, _, _, _, _)), Reprints)),
        Counts),
    max_member(MaxReprints-LastName-FirstName, Counts),
    format(atom(AuthorName), '~w ~w', [FirstName, LastName]).

find_author_most_reprints(ResultText) :-
    author_most_reprints(AuthorName),
    send(ResultText, value, AuthorName).
```

*% 1.1 Кнопка для нахождения автора, у которого книга переиздавалась максимальное число раз:*

```
button_most_reprints :-
    new(Dialog, dialog('Find Author with Most Reprints')),

    new(ResultText, text_item('Result', '')),
    new(FindButton, button('Find Author',
        message(@prolog, find_author_most_reprints, ResultText))),

    send_list(Dialog, append, [FindButton, ResultText]),
    send(Dialog, open).
```

*% 2 Найми все книги, изданные более одного раза*

```
books_reprinted_more_than_once(Books) :-
    findall(Title,
        (book(Title, _, edition(_, EditionNumber, _, _, _, _)),
         EditionNumber > 1),
        Books).
```

Рис. 3. Алгоритм, предикаты и функции

```

% Функция для вызова books_reprinted_more_than_once и отображения результатов
find_books_reprinted_more_than_once(ResultText) :-
    books_reprinted_more_than_once(Books),
    atomic_list_concat(Books, ', ', BooksStr),
    send(ResultText, value, BooksStr).

% 2.1 Кнопка для нахождения всех книг, изданных более одного раза
button_find_books_reprinted_more_than_once :-
    new(Dialog, dialog('Find Books Reprinted More Than Once')),

    new(ResultText, text_item('Result', '')),
    new(FindButton, button('Find Books',
        message(@prolog, find_books_reprinted_more_than_once, ResultText))),

    send_list(Dialog, append, [FindButton, ResultText]),
    send(Dialog, open).

% 3 Найти все книги, изданные в указанном издательстве в заданном году
books_by_publisher_year(Publisher, Year, Books) :-
    findall(Title,
        (book(Title, _, edition(Publisher, _, Year, _, _))),
        Books).

% Функция для вызова books_by_publisher_year и отображения результатов
find_books_by_publisher_year(PublisherEntry, YearEntry, ResultText) :-
    get(PublisherEntry, selection, Publisher),
    get(YearEntry, selection, YearStr),
    atom_number(YearStr, Year), % Преобразование строки в число
    books_by_publisher_year(Publisher, Year, Books),
    (Books = [] -> BooksStr = 'No books found'; atomic_list_concat(Books, ', ', BooksStr)),
    send(ResultText, value, BooksStr).

```

Рис. 4. Алгоритм, предикаты и функции



*% 3.1 Кнопка для нахождения всех книг, изданных в указанном издательстве, в заданном году*

```
button_find_books_by_publisher_year :-
    new(Dialog, dialog('Find Books by Publisher and Year')),
    new(PublisherEntry, text_item('Enter publisher')),
    new(YearEntry, text_item('Enter year')),
    new(ResultText, text_item('Result', '')),
    new(FindButton, button('Find Books',
        message(@prolog, find_books_by_publisher_year, PublisherEntry, YearEntry, ResultText))),
    send_list(Dialog, append, [PublisherEntry, YearEntry, FindButton, ResultText]),
    send(Dialog, open).
```

*% 4 Найти все книги указанного автора*

```
books_by_author(LastName, FirstName, Books) :-
    findall(Title,
        book(Title, author(LastName, FirstName, _), _),
        Books).
```

*% Функция для вызова books\_by\_author и отображения результатов*

```
find_books_by_author(LastName, FirstName, ResultText) :-
    books_by_author(LastName, FirstName, Books),
    ( Books = []
    -> BooksStr = 'No books found'
    ; atomic_list_concat(Books, ' ', BooksStr)
    ),
    send(ResultText, value, BooksStr).
```

Рис. 6. Алгоритм, предикаты и функции

*% 4.1 Кнопка для нахождения всех книг указанного автора*

```
button_find_books_by_author :-
    new(Dialog, dialog('Find Books by Author')),

    new(LastNameEntry, text_item('Enter author\'s last name')),
    new(FirstNameEntry, text_item('Enter author\'s first name')),
    new(ResultText, text_item('Result', '')),
    new(FindButton, button('Find Books',
        message(@prolog, find_books_by_author, LastNameEntry?selection, FirstNameEntry?selection, ResultText))),

    send_list(Dialog, append, [LastNameEntry, FirstNameEntry, FindButton, ResultText]),
    send(Dialog, open).
```

*% 5 Найти все книги, цена которых превышает заданную сумму*

```
find_books_above_price(Pricelimit, ResultText) :-
    findall(Title,
        (book(Title, _, edition(_, _, _, Price, _)),
        Price > Pricelimit),
        Books),
    (Books = [] -> BooksStr = 'No books found'; atomic_list_concat(Books, ' ', BooksStr)),
    send(ResultText, value, BooksStr).
```

*% Функция для вызова books\_above\_price и отображения результатов*

```
find_books_above_price_handler(PricelimitEntry, ResultText) :-
    get(PricelimitEntry, selection, PricelimitStr),
    atom_number(PricelimitStr, Pricelimit), % Преобразование строки в число
    find_books_above_price(Pricelimit, ResultText).
```

Рис. 5. Алгоритм, предикаты и функции

```
% 5.1 Кнопка для нахождения всех книг, цена которых превышает заданную сумму:
button_find_books_above_price :-
    new(Dialog, dialog('Find Books Above a Price')),
    new(PriceLimitEntry, text_item('Enter price limit')),
    new(ResultText, text_item('Result', '')),
    new(FindButton, button('Find Books',
        message(@prolog, find_books_above_price_handler, PriceLimitEntry, ResultText))),
    send_list(Dialog, append, [PriceLimitEntry, FindButton, ResultText]),
    send(Dialog, open).
```

Рис. 6. Алгоритм, предикаты и функции

```
% 6 Добавление клиента
add_book(TitleStr, LastNameStr, FirstNameStr, BirthYearStr, PublisherStr, EditionNumberStr, YearStr, PagesStr, PriceStr) :
    atom_string(Title, TitleStr),
    atom_string(LastName, LastNameStr),
    atom_string(FirstName, FirstNameStr),
    atom_number(BirthYearStr, BirthYear),
    atom_string(Publisher, PublisherStr),
    atom_number(EditionNumberStr, EditionNumber),
    atom_number(YearStr, Year),
    atom_number(PagesStr, Pages),
    atom_number(PriceStr, Price),
    assertz(book(Title, author(LastName, FirstName, BirthYear), edition(Publisher, EditionNumber, Year, Pages, Price))),
    writeln('Book added to the database').

% 6.1 Кнопка для добавления клиента:
button_add_book :-
    new(Dialog, dialog('Add New Book')),

    new(TitleEntry, text_item('Enter book title')),
    new(LastNameEntry, text_item('Enter author\'s last name')),
    new(FirstNameEntry, text_item('Enter author\'s first name')),
    new(BirthYearEntry, text_item('Enter author\'s birth year')),
    new(PublisherEntry, text_item('Enter publisher')),
    new(EditionNumberEntry, text_item('Enter edition number')),
    new(YearEntry, text_item('Enter year of publication')),
    new(PagesEntry, text_item('Enter number of pages')),
    new(PriceEntry, text_item('Enter price')),

    new(AddButton, button('Add Book',
        message(@prolog, add_book, TitleEntry?selection, LastNameEntry?selection, FirstNameEntry?selection, Bi

    send_list(Dialog, append, [TitleEntry, LastNameEntry, FirstNameEntry, BirthYearEntry, PublisherEntry, EditionNumberEnt
    send(Dialog, open).
```

Рис. 7. Алгоритм, предикаты и функции

```

% 7. Интерфейс для сохранения базы данных
% Все определения save_db/1 группируются здесь

save_db(FileName) :-
    open(FileName, write, Stream),
    with_output_to(Stream, listing(book)),
    close(Stream),
    writeln('Database saved to file').

% 7.1 Кнопка для сохранения анкет в БД

button_save_db :-
    new(Dialog, dialog('Save Database')),
    new(FileNameEntry, text_item('Enter file name')),
    new(SaveButton, button('Save Database',
        message(@prolog, save_db, FileNameEntry?selection))),

    send_list(Dialog, append, [FileNameEntry, SaveButton]),
    send(Dialog, open).

% Основное окно
create_main_window :-
    new(MainDialog, dialog('Main Window')),
    new(OpenDialogBtn_most_reprints, button('Author most reprints', message(@prolog, button_most_reprints))),
    new(OpenDialogBtn_find_books_reprinted_more_than_once, button('Find Books', message(@prolog, button_find_books_reprinted_more_than_once))),
    new(OpenDialogBtn_find_books_by_publisher_year, button('Find books by publisher year', message(@prolog, button_find_books_by_publisher_year))),
    new(OpenDialogBtn_find_books_by_author, button('Find books by author', message(@prolog, button_find_books_by_author))),
    new(OpenDialogBtn_find_books_above_price, button('Find books above price', message(@prolog, button_find_books_above_price))),
    new(OpenDialogBtn_add_book, button('Add book', message(@prolog, button_add_book))),
    new(OpenDialogBtn_save_db, button('Save DataBase', message(@prolog, button_save_db))),
    send_list(MainDialog, append, [OpenDialogBtn_most_reprints, OpenDialogBtn_find_books_reprinted_more_than_once, OpenDialogBtn_find_books_by_publisher_year,
        OpenDialogBtn_find_books_by_author, OpenDialogBtn_find_books_above_price, OpenDialogBtn_add_book, OpenDialogBtn_save_db]),
    send(MainDialog, open).

:- create_main_window.

```

Рис. 8. Алгоритм, предикаты и функции

Реализация графического интерфейса представлена на рисунке 9.

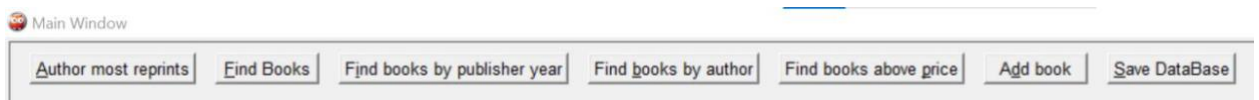


Рис. 9. Графический интерфейс

При нажатии на кнопку Find author, выводится окно, в нем при нажатии на кнопку «Find author» выводится автор книги.

Пример работы программы представлен на рисунке 10.

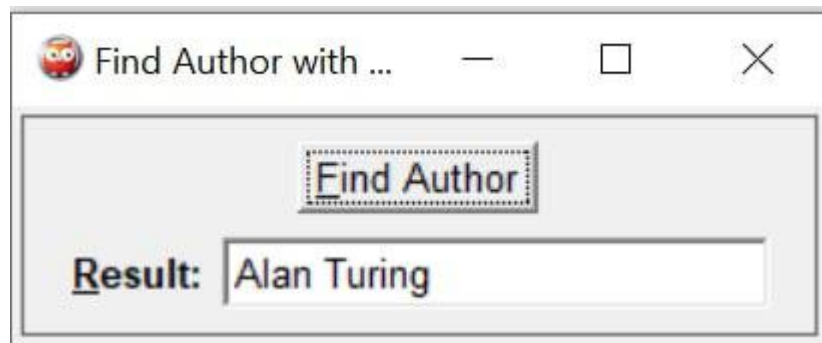


Рис. 10. Find author

При нажатии на кнопку Find books, выводится окно, в нем при нажатии на кнопку «Find books» выводится книга.

Пример работы программы представлен на рисунке 11.

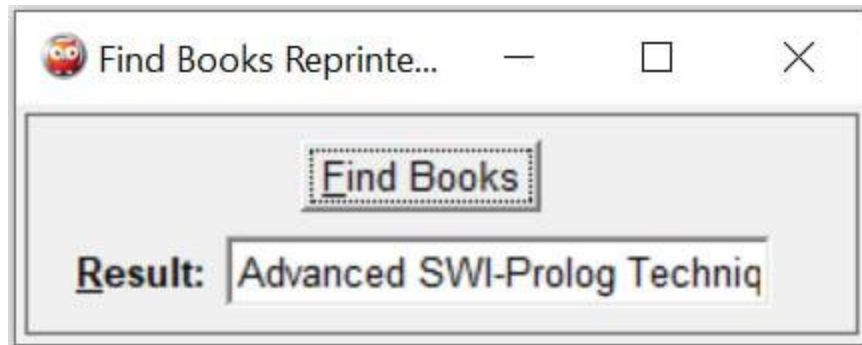


Рис. 11. Find books

При нажатии на кнопку Find books by publisher and year, выводится окно, в нем нужно издателя и год, программа выведет все подходящие книги с указанным издателем и выпущенные в указанный год. Пример работы программы представлен на рисунке 12.

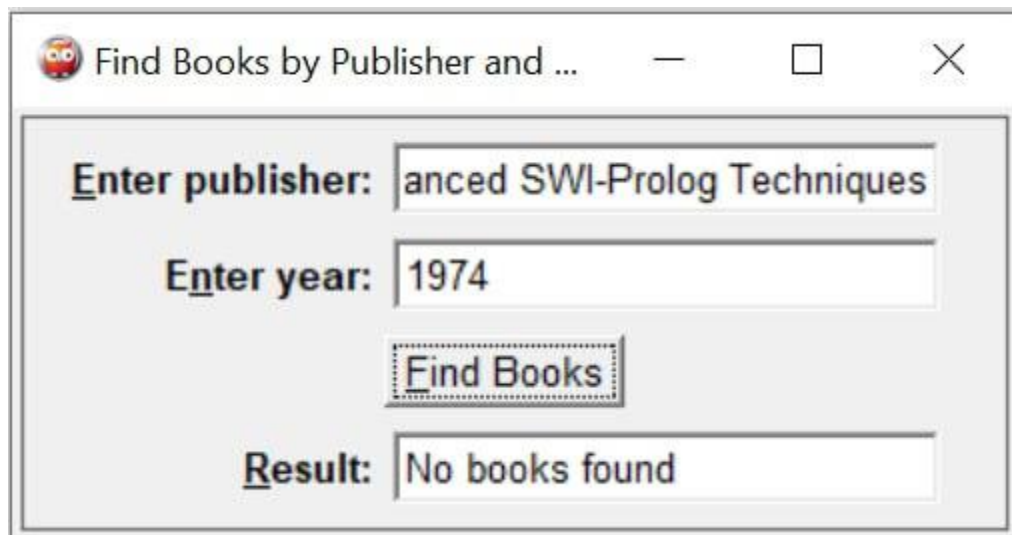


Рис. 12. Find books by publisher and year

При нажатии на кнопку Find books by author, выводится окно, в нем нужно ввести имя и фамилию автора, программа выведет все книги, с аднгим автором.

Пример работы программы представлен на рисунке 13.

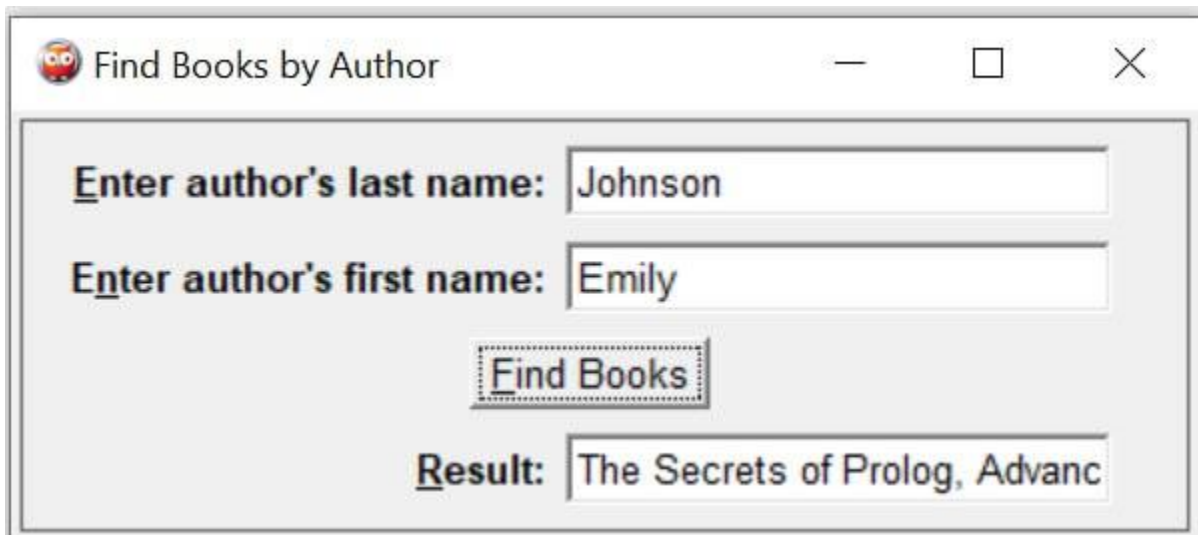


Рис. 13. Find books by author

При нажатии на кнопку Find books above price, выводится окно, в нем нужно ввести цену в рублях, программа выведет все книги, с подходящей ценой и выше.

Пример работы программы представлен на рисунке 14.

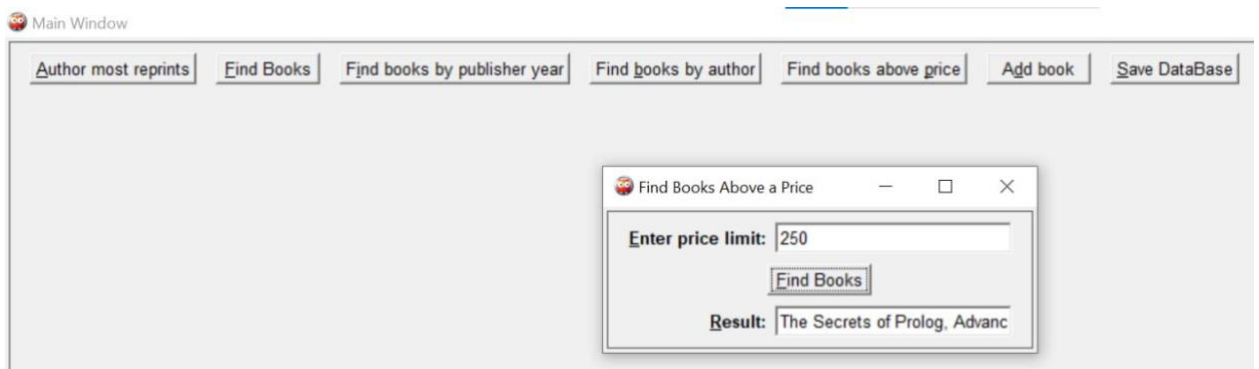


Рис. 14. Find books above price

При нажатии на кнопку Add book, выводится окно, в нем нужно ввести название, ФИО автора, год рождения автора, издателя, год написания книги, количество страниц и цену, программа добавит всю о соревнованиях в базу данных. Затем, при нажатии на Add book сохранится.

Пример работы программы представлен на рисунке 15.



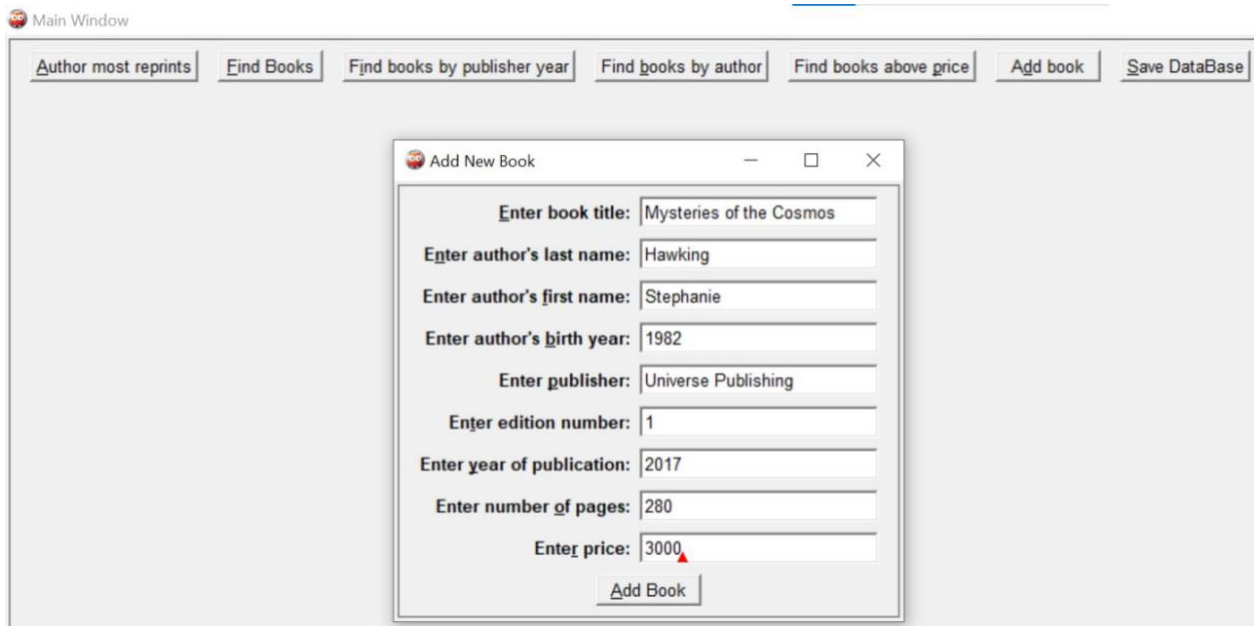


Рис. 15. Add book

Save database, выводится окно, куда нужно ввести название базы данных, программа сохранит базу в отдельный файл на компьютере.

Пример работы программы представлен на рисунке 16.

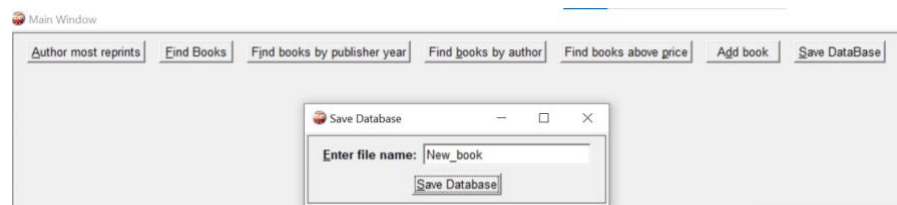


Рис. 16. Save database

**ЛИСТИНГ КОДА**

```
:- use_module(library(pce)).
```

```
:- dynamic book/3.
```

```
% Факты
```

```
book('The Secrets of Prolog',  
    author('Johnson', 'Emily', 1974),  
    edition('TechPress', 1, 2015, 320, 4000, 5000)).
```

```
book('Advanced SWI-Prolog Techniques',  
    author('Johnson', 'Emily', 1974),  
    edition('TechPress', 2, 2018, 350, 4500, 5500)).
```

```
book('Mysteries of the Cosmos',  
    author('Hawking', 'Stephanie', 1982),  
    edition('Universe Publishing', 1, 2017, 280, 3000, 4000)).
```

```
book('Mysteries of the Cosmos',  
    author('Hawking', 'Stephanie', 1983),  
    edition('Universe Publishing', 2, 2017, 280, 3000, 4000)).
```

```
book('Journeys through Time',  
    author('Hawking', 'Stephanie', 1982),  
    edition('Universe Publishing', 1, 2020, 300, 3500, 4500)).
```

```
book('A Programmer\'s Guide to Logic',  
    author('Turing', 'Alan', 1912),  
    edition('CompSci Books', 3, 1995, 250, 2500, 3000)).
```

```
book('The Journey of Prolog',
```

```
author('Adams', 'Douglas', 1952),
edition('SciFi World', 1, 1990, 300, 1500, 1500)).
```

```
book('The Journey of Prolog',
author('Adams', 'Douglas', 1952),
edition('SciFi World', 2, 1995, 320, 1800, 1800)).
```

```
book('The Journey of Prolog',
author('Adams', 'Douglas', 1952),
edition('SciFi World', 3, 2000, 350, 2000, 2000)).
```

```
book('Exploring AI',
author('Curie', 'Marie', 1867),
edition('Tech Advances', 1, 2010, 250, 2500, 2500)).
```

```
book('Exploring AI',
author('Curie', 'Marie', 1867),
edition('Tech Advances', 2, 2015, 300, 3000, 3000)).
```

```
book('Mysteries of Space',
author('Hawking', 'Stephen', 1942),
edition('Universe Publishing', 1, 2005, 400, 4000, 4000)).
```

% Вспомогательные функции

% 1 Найти автора, у которого книга переиздавалась максимальное число раз:

```
author_most_reprints(AuthorName) :-
    findall(Reprints-LastName-FirstName,
```



```

        (book(_, author(LastName, FirstName, _), edition(_, EditionNumber, _, _, _,
        _))),
        aggregate(count, book(_, author(LastName, FirstName, _), edition(_, _, _, _,
        _, _)), Reprints)),
        Counts),
        max_member(MaxReprints-LastName-FirstName, Counts),
        format(atom(AuthorName), '~w ~w', [FirstName, LastName])).

```

```

find_author_most_reprints(ResultText) :-
    author_most_reprints(AuthorName),
    send(ResultText, value, AuthorName).

```

% 1.1 Кнопка для нахождения автора, у которого книга переиздавалась  
максимальное число раз:

```

button_most_reprints :-
    new(Dialog, dialog('Find Author with Most Reprints')),

    new(ResultText, text_item('Result', "")),
    new(FindButton, button('Find Author',
        message(@prolog, find_author_most_reprints, ResultText))),

    send_list(Dialog, append, [FindButton, ResultText]),
    send(Dialog, open).

```

% 2 Найти все книги, изданные более одного раза

```

books_reprinted_more_than_once(Books) :-
    findall(Title,
        (book(Title, _, edition(_, EditionNumber, _, _, _, _)),
        EditionNumber > 1),

```

Books).

% Функция для вызова books\_reprinted\_more\_than\_once и отображения результатов

find\_books\_reprinted\_more\_than\_once(ResultText) :-

books\_reprinted\_more\_than\_once(Books),

atomic\_list\_concat(Books, ' ', BooksStr),

send(ResultText, value, BooksStr).

% 2.1 Кнопка для нахождения всех книг, изданных более одного раза

button\_find\_books\_reprinted\_more\_than\_once :-

new(Dialog, dialog('Find Books Reprinted More Than Once')),

new(ResultText, text\_item('Result', "")),

new(FindButton, button('Find Books',

message(@prolog, find\_books\_reprinted\_more\_than\_once, ResultText))),

send\_list(Dialog, append, [FindButton, ResultText]),

send(Dialog, open).

% 3 Найти все книги, изданные в указанном издательстве в заданном году

books\_by\_publisher\_year(Publisher, Year, Books) :-

findall(Title,

(book(Title, \_, edition(Publisher, \_, Year, \_, \_))),

Books).

% Функция для вызова books\_by\_publisher\_year и отображения результатов

find\_books\_by\_publisher\_year(PublisherEntry, YearEntry, ResultText) :-

get(PublisherEntry, selection, Publisher),

get(YearEntry, selection, YearStr),

atom\_number(YearStr, Year), % Преобразование строки в число

books\_by\_publisher\_year(Publisher, Year, Books),

(Books = [] -> BooksStr = 'No books found'; atomic\_list\_concat(Books, ' ', BooksStr)),

send(ResultText, value, BooksStr).

% 3.1 Кнопка для нахождения всех книг, изданных в указанном издательстве, в заданном году

button\_find\_books\_by\_publisher\_year :-

new(Dialog, dialog('Find Books by Publisher and Year')),

new(PublisherEntry, text\_item('Enter publisher')),

new(YearEntry, text\_item('Enter year')),

new(ResultText, text\_item('Result', "")),

new(FindButton, button('Find Books',

message(@prolog, find\_books\_by\_publisher\_year, PublisherEntry, YearEntry, ResultText))),

send\_list(Dialog, append, [PublisherEntry, YearEntry, FindButton, ResultText]),

send(Dialog, open).

% 4 Найти все книги указанного автора

books\_by\_author(LastName, FirstName, Books) :-

findall(Title,

book(Title, author(LastName, FirstName, \_), \_),

Books).

% Функция для вызова books\_by\_author и отображения результатов

find\_books\_by\_author(LastName, FirstName, ResultText) :-

books\_by\_author(LastName, FirstName, Books),

( Books = []

-> BooksStr = 'No books found'

; atomic\_list\_concat(Books, ' ', BooksStr)

),

send(ResultText, value, BooksStr).

% 4.1 Кнопка для нахождения всех книг указанного автора

button\_find\_books\_by\_author :-

new(Dialog, dialog('Find Books by Author')),

new(LastNameEntry, text\_item('Enter author\'s last name')),

new(FirstNameEntry, text\_item('Enter author\'s first name')),

new(ResultText, text\_item('Result', "")),

new(FindButton, button('Find Books',

message(@prolog, find\_books\_by\_author, LastNameEntry?selection,

FirstNameEntry?selection, ResultText))),

send\_list(Dialog, append, [LastNameEntry, FirstNameEntry, FindButton,  
ResultText]),

send(Dialog, open).

% 5 Найти все книги, цена которых превышает заданную сумму

find\_books\_above\_price(PriceLimit, ResultText) :-

findall(Title,

```

(book(Title, _, edition(_, _, _, Price, _)),
 Price > PriceLimit),
Books),
(Books = [] -> BooksStr = 'No books found'; atomic_list_concat(Books, ' ',
BooksStr)),
send(ResultText, value, BooksStr).

```

% Функция для вызова books\_above\_price и отображения результатов

find\_books\_above\_price\_handler(PriceLimitEntry, ResultText) :-

```

get(PriceLimitEntry, selection, PriceLimitStr),
atom_number(PriceLimitStr, PriceLimit), % Преобразование строки в число
find_books_above_price(PriceLimit, ResultText).

```

% 5.1 Кнопка для нахождения всех книг, цена которых превышает заданную сумму:

button\_find\_books\_above\_price :-

```

new(Dialog, dialog('Find Books Above a Price')),
new(PriceLimitEntry, text_item('Enter price limit')),
new(ResultText, text_item('Result', "")),
new(FindButton, button('Find Books',
    message(@prolog, find_books_above_price_handler, PriceLimitEntry,
ResultText))),
send_list(Dialog, append, [PriceLimitEntry, FindButton, ResultText]),
send(Dialog, open).

```

% 6. Add new client

% Enter surname: "Ivanov" - Фамилия клиента.

% Enter name: "Ivan" - Имя клиента.

% Enter patronymic: "Ivanovich" - Отчество клиента.

% Enter age: "30" - Возраст клиента. Введите число.

% Enter nationality: "Russian" - Национальность клиента.

% Education level: Выберите "higher" - Уровень образования клиента. Выберите из списка (например, "higher", "secondary\_vocational" или "secondary").

% Enter monthly income: "5000" - Ежемесячный доход клиента. Введите число.

% Owns living space: Выберите "yes" - Имеет ли клиент собственную жилую площадь. Выберите "yes" или "no"

% Has children: Выберите "no" - Имеет ли клиент детей. Выберите "yes" или "no".

% No bad habits: Выберите "yes" - Отсутствие вредных привычек у клиента. Выберите "yes" или "no".

% Partner education: "Higher" - Требуемый уровень образования партнера.

% Partner age: "35" - Максимальный возраст партнера. Введите число.

% Partner nationality: "Russian" - Национальность партнера.

% Partner income: "4000" - Минимальный ежемесячный доход партнера. Введите число.

% 6 Добавление клиента

add\_book(TitleStr, LastNameStr, FirstNameStr, BirthYearStr, PublisherStr, EditionNumberStr, YearStr, PagesStr, PriceStr) :-

```
    atom_string(Title, TitleStr),
    atom_string(LastName, LastNameStr),
    atom_string(FirstName, FirstNameStr),
    atom_number(BirthYearStr, BirthYear),
    atom_string(Publisher, PublisherStr),
    atom_number(EditionNumberStr, EditionNumber),
```

```

atom_number(YearStr, Year),
atom_number(PagesStr, Pages),
atom_number(PriceStr, Price),
assertz(book(Title, author(LastName, FirstName, BirthYear), edition(Publisher,
EditionNumber, Year, Pages, Price))),
writeln('Book added to the database').

```

% 6.1 Кнопка для добавления клиента:

button\_add\_book :-

```

    new(Dialog, dialog('Add New Book')),

    new(TitleEntry, text_item('Enter book title')),
    new(LastNameEntry, text_item('Enter author\'s last name')),
    new(FirstNameEntry, text_item('Enter author\'s first name')),
    new(BirthYearEntry, text_item('Enter author\'s birth year')),
    new(PublisherEntry, text_item('Enter publisher')),
    new(EditionNumberEntry, text_item('Enter edition number')),
    new(YearEntry, text_item('Enter year of publication')),
    new(PagesEntry, text_item('Enter number of pages')),
    new(PriceEntry, text_item('Enter price')),

    new(AddButton, button('Add Book',
        message(@prolog,          add_book,          TitleEntry?selection,
        LastNameEntry?selection,  FirstNameEntry?selection,  BirthYearEntry?selection,
        PublisherEntry?selection, EditionNumberEntry?selection,  YearEntry?selection,
        PagesEntry?selection, PriceEntry?selection))),

```

```

send_list(Dialog, append, [TitleEntry, LastNameEntry, FirstNameEntry,
BirthYearEntry, PublisherEntry, EditionNumberEntry, YearEntry, PagesEntry,
PriceEntry, AddButton]),
send(Dialog, open).

```

% 7. Интерфейс для сохранения базы данных

% Все определения save\_db/1 группируются здесь\

save\_db(FileName) :-

```

    open(FileName, write, Stream),
    with_output_to(Stream, listing(book)),
    close(Stream),
    writeln('Database saved to file').

```

% 7.1 Кнопка для сохранения анкет в БД

button\_save\_db :-

```

    new(Dialog, dialog('Save Database')),

    new(FileNameEntry, text_item('Enter file name')),
    new(SaveButton, button('Save Database',
        message(@prolog, save_db, FileNameEntry?selection))),

    send_list(Dialog, append, [FileNameEntry, SaveButton]),
    send(Dialog, open).

```

% Основное окно



create\_main\_window :-

```

    new(MainDialog, dialog('Main Window')),
    new(OpenDialogBtn_most_reprints, button('Author most reprints',
message(@prolog, button_most_reprints))),
    new(OpenDialogBtn_find_books_reprinted_more_than_once, button('Find
Books', message(@prolog, button_find_books_reprinted_more_than_once))),
    new(OpenDialogBtn_find_books_by_publisher_year, button('Find books by
publisher year', message(@prolog, button_find_books_by_publisher_year))),
    new(OpenDialogBtn_find_books_by_author, button('Find books by author',
message(@prolog, button_find_books_by_author))),
    new(OpenDialogBtn_find_books_above_price, button('Find books above price',
message(@prolog, button_find_books_above_price))),
    new(OpenDialogBtn_add_book, button('Add book', message(@prolog,
button_add_book))),
    new(OpenDialogBtn_save_db, button('Save DataBase', message(@prolog,
button_save_db))),
    send_list(MainDialog, append, [OpenDialogBtn_most_reprints,
OpenDialogBtn_find_books_reprinted_more_than_once,
OpenDialogBtn_find_books_by_publisher_year,
OpenDialogBtn_find_books_by_author,
OpenDialogBtn_find_books_above_price, OpenDialogBtn_add_book,
OpenDialogBtn_save_db]),
    send(MainDialog, open).

```

:- create\_main\_window.

## **ВЫВОДЫ**

В ходе этой лабораторной работы мы научились использовать SWI-Prolog и библиотеку XPCЕ для создания графического пользовательского интерфейса (GUI) и работы с динамической базой данных (БД).

В рамках работы были изучены основные конструкции рекурсивно-логического программирования, включая термы и утверждения. Мы также освоили унификацию термов на языке Prolog и изучили методы работы с интерпретатором SWI Prolog. Были рассмотрены элементы интерфейса SWI Prolog, такие как окно запросов и встроенный редактор текстов, и получены практические навыки создания, отладки и выполнения простых программ в среде SWI Prolog.

## СПИСОК ЛИТЕРАТУРЫ

1. "Логическое программирование на языке Prolog"  
([http://www.habarov.spb.ru/book\\_prolog\\_2013/SerpBook\\_Prolog.pdf](http://www.habarov.spb.ru/book_prolog_2013/SerpBook_Prolog.pdf))
2. "Введение в логическое программирование на языке Prolog"  
(<https://habr.com/ru/articles/124636/>)
3. "Логическое программирование на языке Visual Prolog. Учебное пособие для вузов" - Цуканова Н.И., Дмитриева Т.А.
4. "Искусство программирования на языке Пролог" - Л. Стерлинг, Э. Шапиро