

Изкуствен интелект / интелигентни системи

*гл. асистент д-р Венета Табакова-Комсалова
ФМИ, ПУ „П. Хилендарски“, Пловдив*

Упражнение 1-3

Въведение. Пролог.

Факти, правила, цел.

- ❖ Въведение

- ❖ Пролог

- Защо Пролог?

- Малко история

- Приложения на Пролог

Теоретични основи на езика

Пролог

Клаузна форма

Пролог се стреми да извлече логически извод от данни, с които разполага. Обикновено, програмата на Пролог не е последователност от действия, както при алгоритмичните езици, а е съвкупност от факти и правила за извеждането на заключения от тези факти. За разлика от императивните езици като С или Java това е език за декларативно програмиране. Това означава, че вместо да уточняваме как да се постигне определена цел в дадена ситуация, се описва самата ситуация (**правилата и фактите**) и се посочва **целта**. След това интерпретаторът на Пролог извежда решението на проблема.

Пролог е особено полезен за решаване на проблеми в ИИ, както и в области като търсене, планиране и представяне на знания.

Важна характеристика на Пролог е, че освен намиране на отговори на поставени въпроси, той може да се справи с търсене на алтернативи като намери всички възможни решения, а не само едно.

Пролог е особено полезен за решаване на проблеми в ИИ, както и в области като търсене, планиране и представяне на знания.

Важна характеристика на Пролог е, че освен намиране на отговори на поставени въпроси, той може да се справи с търсене на алтернативи като намери всички възможни решения, а не само едно.

Факти

Факти: Това, което е известно.

В Пролог, отношението между обектите се нарича предикат.

В естествения език, една връзка е символизирана от едно изречение.

В предикатната логика, която Пролог използва, връзката е обобщена в една простичка фраза, която се състои от отношението име, последвано от обект или обекти (затворени в скоби). Както и при изреченията фактът, завършва с точка (.).

Тъй като Пролог е декларативен език, използването на програми на Пролог означава задаване на въпроси върху описания преди това свят. Най-простият начин за описване на света е чрез излагане на факти, като този:

<u>На естествен език</u>	<u>В Пролог</u>
Иван харесва Мария.	харесва(иван, maria).
Мария харесва Иван.	харесва(maria, иван).
Иван харесва кучета.	харесва(иван, кучета).

Правила

Правила определят зависими отношения между факти и други правила. Правилата позволяват да се изведат, факти, от други факти. Друг начин да се каже това е, че по правило, заключенията се възприемат за истина, ако едно или повече други заключения или факти се установят, че са истина. В Пролог всички правила, имат две части: глава и тяло, разделени от специален маркер „:-“ . Главата е факт, който би бил верен, ако описаните условия в тялото са верни.

Тялото е съвкупност от условия, които трябва да бъдат проверени за истинност, за да може да се твърди, че главата е истина.

Следователно правилата извличат информация от фактите.

Пример: Нека приемем, че съществуват следните факти и правила:

Иван харесва Мария.

Небето е синьо.

Мария харесва всичко, което Иван харесва.

Надежда харесва всичко, което е синьо.

От тях могат да се изведат следните изводи:

Мария харесва Мария.

Надежда харесва небето.

За да се напишат на Пролог тези правила е необходимо следното:

<u>На естествен език</u>	<u>В Пролог</u>
Иван харесва Мария. Небето е синьо. Мария харесва всичко, което Иван харесва.	харесва(иван, maria). синьо(небе). харесва(мария, Нещо):- харесва (иван, Нещо).
Надежда харесва всичко, което е синьо.	харесва(надежда, Нещо):-синьо (Нещо).

Символното означение „:-“ се чете като „ако”(if). В конкретния случай *Нещо* е променлива и поради тази причина е с главна буква. Пролог ще търси всички възможни обвързвания на променливата „Нещо“ спрямо фактите така, че да се намерят всички възможни решения.

Цел

Целта е получаване на отговор от системата. Намирането на удовлетворение на една цел изисква използването на един от всичките предикати, описани в програмата. Много често обаче е възможно да се търсят няколко свързани цели, всяка от която се нарича подцел. Обвързването на няколко подцели оформя сложна цел, в която могат да се използват логическите операции конюнкция и дизюнкция (логическо „и” и „или”), обозначаването на двете операции е съответно за конюнкция – запетая(,), а за дизюнкция – точка и запетая(;).

Променливи

В Пролог, променливите позволяват да се пишат общи факти и правила и да се зададат обобщаващи въпроси.

Променливите в Пролог получават своите стойности след обвързването им с фактите или правилата. След като получат стойност, те я „съхраняват“ докато машината за извод намери едно решение и веднага след това променливата се освобождава и става необвързана, за да може да се намери друго решение.

В Пролог е възможно използването и на анонимни променливи.

Когато се изисква точно определена информация от конкретна заявка, могат да се използват анонимни променливи, за да се игнорират стойностите, които не са необходими.

Разликата е, че анонимната променлива никога няма да има установена стойност.

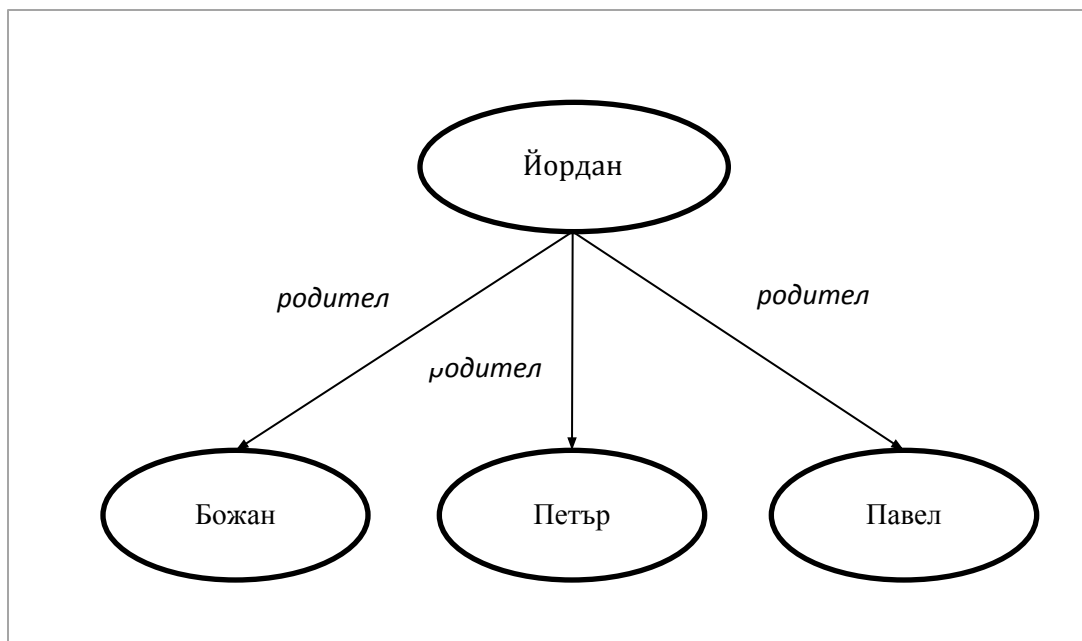
Използване на коментари.

В Пролог е удачно използването на коментари. Съществуват два варианта за коментиране:

% - всичко, което е след знака за процент до края на реда, се смята за коментар

/* */ - служи за поставяне на коментар на няколко реда

Да разгледаме повестта на Елин Пелин „Гераците“



База данни (БД) от факти

Нека представим компютърните моделите на двата сценария използвайки средствата на езика за програмиране Пролог.

/* БД: сценарий „Гераците“ (първа версия) */

родител(йордан, божан).

родител(йордан, петър).

родител(йордан, павел).

Примери за БД „Гераците“:

?- родител(йордан, божан).

true.

?- родител(йордан, кочо).

false.

?- родител(йордан, X).

X = божан ;

X = петър ;

X = павел.

?- родител(X, Y).

X = йордан,

Y = божан ;

X = йордан,

Y = петър ;

X = йордан,

Y = павел.

сценарий „Железният светилник“

родител(стоян,кочо).

родител(стоян,лазар).

родител(стоян,манда).

родител(стоян,нона).

родител(стоян,катерина).

БД от правила. Примерните програми могат да бъдат разширени с нови предикати. Нека първо разширим БД на „Железният светилник“ с два нови предиката, деклариращи пола на героите от романа, а именно мъж/1 и жена/1.

/* БД: сценарий „Железният светилник“ (втора версия) */

родител(стоян, кочо).

родител(стоян, лазар).

родител(стоян, манда).

родител(стоян, нона).

родител(стоян, катерина).

мъж(стоян). мъж(кочо). мъж(лазар).

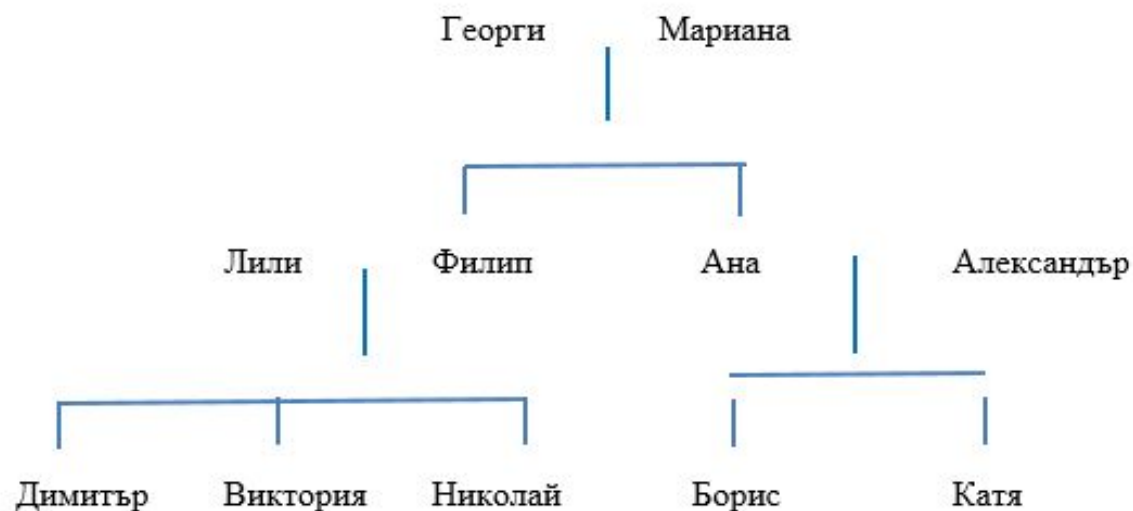
жена(манда). жена(нона). жена
(катерина).

наследник(Y, X) :- родител(X, Y).

Можем да зададем следния въпрос: |- наследник(лазар, стоян).

true.

Пример: Да разгледаме фамилните отношения от следното родословно дърво:



- Опишете фактите в даденото родословно дърво.
- Определете как се дефинира родител, майка, баща ...

Допълнения към задачата:

1. Да се добави предикат и клауза за намиране на „баба”.
2. Да се добави предикат и клауза за намиране на „чичо”.
3. Да се добави предикат и клауза за намиране на „леля”.

Указания за точка 1:

- 1) Трябва да се добави предикат **daughter** (дъщеря);
- 2) Трябва да се добави предикат **mother** (майка);
- 3) Трябва да се добави предикат **grandmother** (баба);
- 4) Трябва да се опише действието на предиката **grandmother**

SWI-Prolog

<https://swish.swi-prolog.org/>

The screenshot shows the SWISH web interface at <https://swish.swi-prolog.org/example/lists.pl>. The interface includes a menu bar with 'File', 'Edit', 'Examples', and 'Help'. A search bar is located in the top right. The main area is divided into two panes. The left pane contains a Prolog program with the following code:

```
1 son(ivan,jordan).
2 son(jordan,marin).
3 sister(maria,silvia).
4 brother(petyr,jordan).
5 married(jordan,maria).
6 married(petyr,stela).
7 father(A,B):-son(B,A).
8 grandfather(A,B):-father(A,C),father(C,B).
9
```

The right pane shows the execution of the query `grandfather(X,ivan)`. The result is `X = marin`. Below the result, the query `?- grandfather(X,ivan)` is shown. The interface also features a large owl illustration in the background and a 'Run!' button at the bottom right.

Примерно решение:

мъж(димитър).

мъж(николай).

мъж(борис).

мъж(филип).

мъж(александър).

мъж(георги).

жена(виктория).

жена(катя).

жена(лили).

жена(ана).

жена(мариана).

семејство(георги, мариана).

семејство(мариана, георги).

семејство(филип, лили).

семејство(лили, филип).

семејство(ана, александър).

семејство(александър, ана).

син(филип, георги).

син(филип, мариана).

син(димитър, филип).

син(димитър, лили).

дъщеря¹⁹(виктория, филип).

дъщеря(виктория, лили).

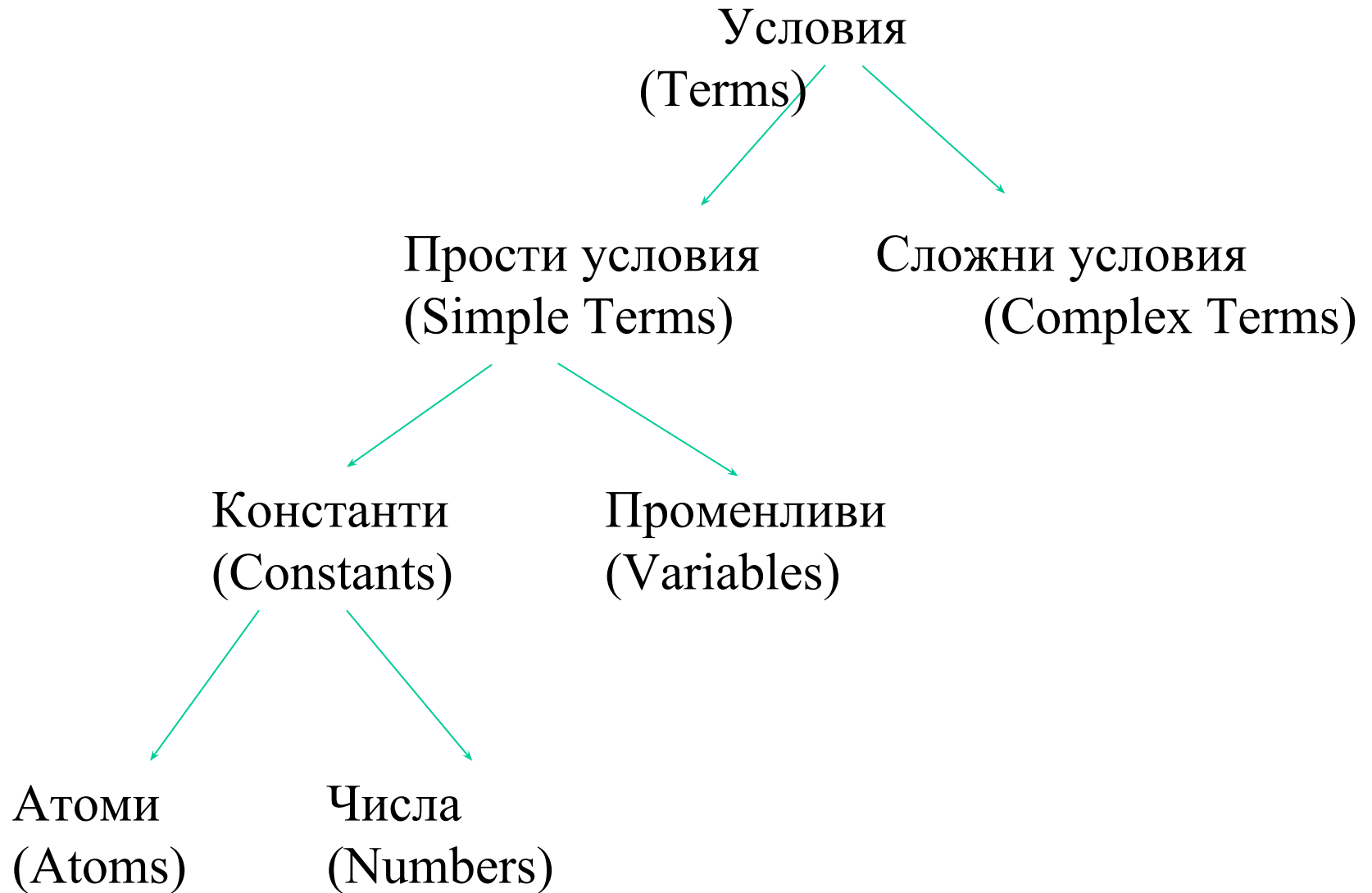
син(николай, филип).

Променливи

Променливите са последователност от символи, която започва с главна буква “A..Z” и може да съдържа малки букви “a..z”, главни букви “A..Z”, цифри “0..9” и символа долна черта “_”. Това са именуваните променливи.

Съществуват и анонимни променливи, които се означават само с долна черта “_”.

Пример: X, Y, A12, My_variable, XY_8, _.



Атоми

- Поредица от символи на главни букви, малки букви, цифри или долна черта, започвайки с малка буква
 - *Примери* : **синьо** , **чете_книга** , **свириКитара**

- Произволна последователност от символи, затворени в единични кавички

- *Примери* : „**Мария**“ , „**Иван Петров**“ , „@ \$%“

- Поредица от специални символи

- *Примери* : **:** , **;** , **.** : **-**

Числа

- Цели числа: 12, -34, 22342
- С плаваща запетая: 34573.3234, 0.3435

Променливи

- Поредица от символи: големи букви, малки букви, цифри, или долна черта, като се започне или с главна буква, или долна черта.

- Примери:

X, Y, Променлива, Мария, час

- Свободна променлива – записва се с долна черта „“

Сложни условия

- Атомите, числата и променливите са градивни елементи за **сложни термини**

- Сложните условия са изградени от а **функтор**, последван от последователност на **аргументи**

- Аргументите са поставени в кръгли скоби, разделени със запетаи

- Функторът трябва да е атом

Примери за сложни термини

- Примери, които сме виждали преди:
 - свириКитара(иван)
 - обича(иван, maria)
 - ревнив(петър, X)
- Сложни термини в сложни термини:
 - скрий(X, баща (баща (баща (иван))))

Аргументи

- Броят на аргументите комплекс терминът се нарича негова същност

- Примери:

жена(maria) е термин с 1 аргумент

обича(иван, maria) има 2 аргумента

баща(баща (иван)) 1 аргумент

Аргументът е важен

- Можете да дефинирате два предиката с един и същ функтор, но с различна същност
- Пролог би третираше това като два различни предиката!
- В документацията на Prolog, аргумент на предикатът обикновено се обозначава с суфикс "/", последван от число за посочване на аргументите

Пример за аргумент

- Тази база знания определя
 - щастлив / 1
 - listens2music / 1
 - playAirGuitar / 1
- щастлив(ваня).
listens2music (миа).
listens2music (ема): -
щастлив(ема).
playsGuitar (миа): -
listens2music(миа).
playsGuitar (ема): -
listens2music(ема).

Клаузи

Клаузите са: факти, правила и цели.

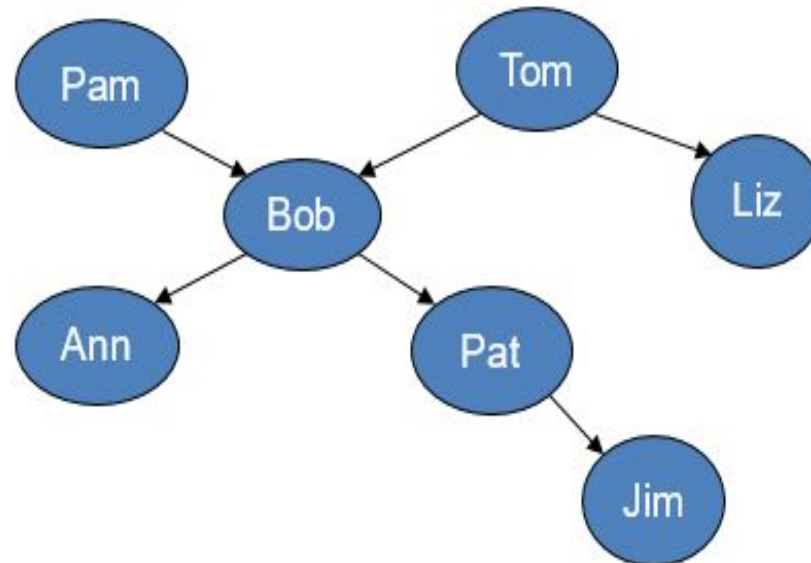
Символни константи

Символните константи са последователност от символи, която започва с малка буква “a..z” и може да съдържа малки букви “a..z”, главни букви “A..Z”, цифри “0..9” и символа долна черта “_”.
Например: my_function, a12, b, x_y, f_15, delete и т.н.

Предикати

Предикатите позволяват дефиниране на различни релации (отношения) чрез факти. Записват се с име и брой параметри разделен с „/“. Например: родител/2

Пример: Да разгледаме фамилните отношения от следното родословно дърво:



Примерна програма в SWI

Prolog:

/* База данни: */

parent(tom,bob).

parent(tom,liz).

parent(pam,bob).

parent(bob,ann).

parent(bob,pat).

parent(pat,jim).

/*Задаване на цели: */

| ?- parent(tom,bob).

true

| ?- parent(bob,tom).

false

| ?- parent(tom,sam).

true

| ?- parent(tom,X).

преудовлетворяване на целта

X = bob ? yes

| ?- parent(tom,X).

X = bob ? ;

X = liz ? ;

false

| ?- parent(X,bob).

X = tom ? ;

X = pam ? ;

false

**преудовлетворяване
на целта**

/* дефиниране на релации чрез правила */

male(tom).

male(bob).

male(jim).

female(pam).

female(liz).

female(ann).

female(pat).

father(X,Y):- male(X), parent(X,Y).

и / and конюнкция

глава

тяло

to/then

ако/if

mother(X,Y):-female(X),parent(X,Y).

brother(X,Y):- male(X), parent(Z,X),parent(Z,Y),X \neq Y.

sister(X,Y):- female(X),parent(Z,X), parent(Z,Y),X \neq Y.

/* програма publication*/

/* Автор - структура author(Собствено име,Фамилия)

Книга - структура book(Заглавие,Автор,Брой страници)

Публикация - предикат publication(Книга,Цена)

Националност - предикат nationality(Автор,Град в който е роден) */

publication(book('Железният светилник',author('Димитър','Талев'),498),10.90).

publication(book('Осъдени души',author('Димитър','Димов'),456),25.00).

publication(book('Илинден',author('Димитър','Талев'),868),14.00).

publication(book('Ветрена мелница',author('Елин','Пелин'),526),25.00).

publication(book('Песента на колелетата',author('Йордан','Йовков'),560),9.00).

publication(book('Да бъде ден',author('Христо','Смирненски'),450),6.00).

publication(book('Поглед зад очилата',author('Атанас','Далчев'),496),9.00).

publication(book('Хъшове',author('Иван','Вазов'),546),25.00).

publication(book('В полите на Витоша',author('Пейо','Яворов'),362),20.00).

publication(book('Немили-Недраги',author('Иван','Вазов'),366),9.00).

publication(book('През води и гори',author('Емилиян','Станев'),144),4.20).

publication(book('Патиланци',author('Ран','Босилек'),272),12.90).

publication(book('Под Игото',author('Иван','Вазов'),600),10.00).
publication(book('В лунната стая',author('Валери','Петров'),80),17.00).
publication(book('Аз съм българче',author('Иван','Вазов'),80),12.90).
publication(book('Палечко',author('Валери','Петров'),10),10.00).
publication(book('Мамино детенце',author('Любен','Каравелов'),96),4.00).
publication(book('Зайченцето бяло',author('Леда','Милева'),80),12.90).
publication(book('Тютюн',author('Димитър','Димов'),864),40.00).
publication(book('Крадецът на праскови',author('Емилиян','Станев'),112),13.00).
publication(book('Бай Ганьо',author('Алеко','Константинов'),256),5.00).
publication(book('Ян Бибиян',author('Елин','Пелин'),200),4.00).
publication(book('До Чикаго и назад',author('Алеко','Константинов'),120),4.90).
publication(book('Лавина',author('Блага','Димитрова'),344),12.00).
publication(book('Нощем с белите коне',author('Павел','Вежинов'),464),18.00).

% Националност - предикат nationality(Автор,Град в който е роден)
nationality(author('Димитър','Талев'),'гр.Прилеп').
nationality(author('Димитър','Димов'),'гр.Ловеч').
nationality(author('Елин','Пелин'),'с.Байлово').
nationality(author('Йордан','Йовков'),'гр.Жеравна').
nationality(author('Христо','Смирненски'),'гр.Кукуш').
nationality(author('Атанас','Далчев'),'гр.Солун').
nationality(author('Иван','Вазов'),'гр.Сопот').
nationality(author('Пейо','Яворов'),'гр.Чирпан').
nationality(author('Емилиян','Станев'),'гр.Велико Търново').
nationality(author('Ран','Босилек'),'гр.Габрово').
nationality(author('Валери','Петров'),'гр.София').
nationality(author('Любен','Каравелов'),'гр.Копривщица').
nationality(author('Леда','Милева'),'гр.София').
nationality(author('Алеко','Константинов'),'гр.Свищов').
nationality(author('Блага','Димитрова'),'гр.Бяла Слатина').
nationality(author('Павел','Вежинов'),'гр.София').

/*

1. Да се изведът всички книги с цена над 4.90
2. Да се изведът всички книги с повече от 200 страници:
 - а) книгите да се изведът с отделни променливи за име, автор и страници; б) книгите да се изведът като цели структури.
3. Да се намери автор с две различни книги.
4. Да се намери автор с точно една книга.
5. Всички двойки различни автори с едно и също първо име.
6. Заглавията на книги автори родени в гр.София.

*/

1. Да се изведът всички книги с цена над 4.90

?-publication(X,Y),Y>4.90.

2. Да се изведът всички книги с повече от 200 страници:

а) книгите да се изведът с отделни променливи за име, автор и страници;

?-publication(book(X,Y,Z),_,Z>200.

б) книгите да се изведът като цели структури.

?-publication(X,_,X=book(_,_,Z),Z>200,write(X),nl,fail.

3. Да се намери автор с две различни книги.

two_books(X):-publication(book(Y,X,_,_),publication(book(Z,X,_,_),Z\=Y.

?-two_books(X).

4. Да се намери автор с точно една книга.

one_book(X):-publication(book(Y,X,_,_), \+ (publication(book(Z,X,_,_),Z\=Y).

?-one_book(X).

5. Всички двойки различни автори с едно и също първо име.

same_fname(X,Y):-publication(book(_,author(Z,X),_,_),

publication(book(_,author(Z,Y),_,_),X\=Y.

%или

?-same_fname(X,Y).

6. Заглавията на книги автори родени в гр.София.

?-publication(book(X,Y,_,_),nationality(Y,'гр.София'),write(X),nl,fail.

7. Да се намери най-евтината книга.

?-publication(X,Y),\+ (publication(_,Z),Z<Y).