



Въведение в логическото програмиране с Пролог

Добре дошли в света на Пролог! Тази презентация ще ви запознае с основите на този език. Ще разгледаме неговата история, синтаксис и приложения.

Защо изучаваме Пролог?

Предимства

Пролог е мощен инструмент за решаване на задачи. Той е особено полезен при задачи, свързани с изкуствен интелект.

Недостатъци

Пролог може да бъде труден за усвояване. Той има специфичен синтаксис и изисква логическо мислене.



Кратка история на Пролог

1972

Създаден от Ален Колмерое във Франция. Първоначално е предназначен за обработка на естествен език.

2

1980-те

Става популярен в областта на изкуствения интелект. Използван е за експертни системи и други приложения.

Днес

3

Продължава да се използва в различни области. Включително лингвистика, биоинформатика и роботика.

Приложения на Пролог



Изкуствен интелект

Експертни системи, планиране, разпознаване на образи.



Лингвистика

Обработка на естествен език, синтактичен анализ.

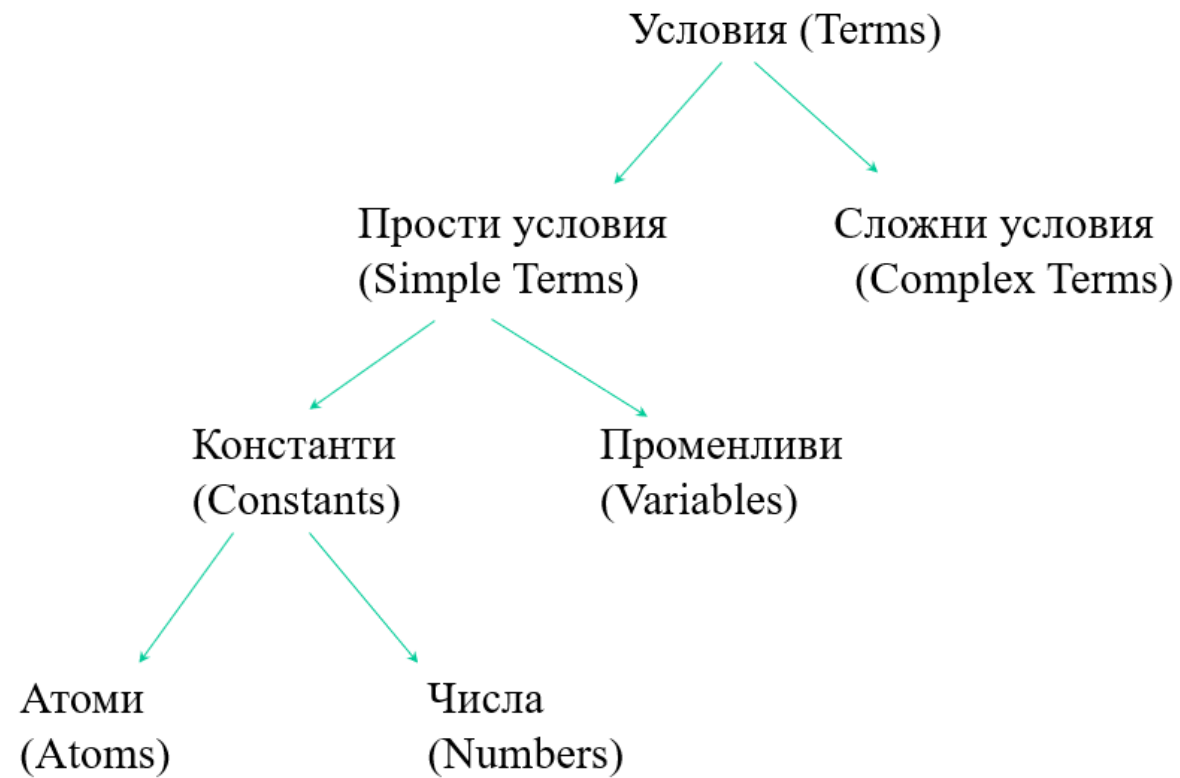


Биоинформатика

Анализ на ДНК последователности, моделиране на протеини.



Синтаксис на Пролог



1

Термове

Атоми, числа, променливи, структури.

2

Факти

Твърдения за света.

3

Правила

Отношения между факти.

4

Цели

Въпроси към системата.

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 кавички или апострофи
 - *Примери* : „**Мария**“ , „**Иван Петров**“ , „**@ \$%**“, ‘**Гераците**‘
- 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:
 - publication(book('Илинден',author('Димитър','Талев'),868),14.00).

Аргументи

- Броят на аргументите комплекс терминът се нарича негова същност
- Примери:
 - жена(maria)** е термин с 1 аргумент
 - обича(иван, maria)** има 2 аргумента



Факти в Пролог

Определение

Базови твърдения, които са винаги верни.

Пример

родител(иван, петър).
"Иван е родител на Петър".

Представяне

Записват се като име на предикат и аргументи.

Факти

Факти: Това, което е известно.

В Пролог, отношението между обектите се нарича предикат.

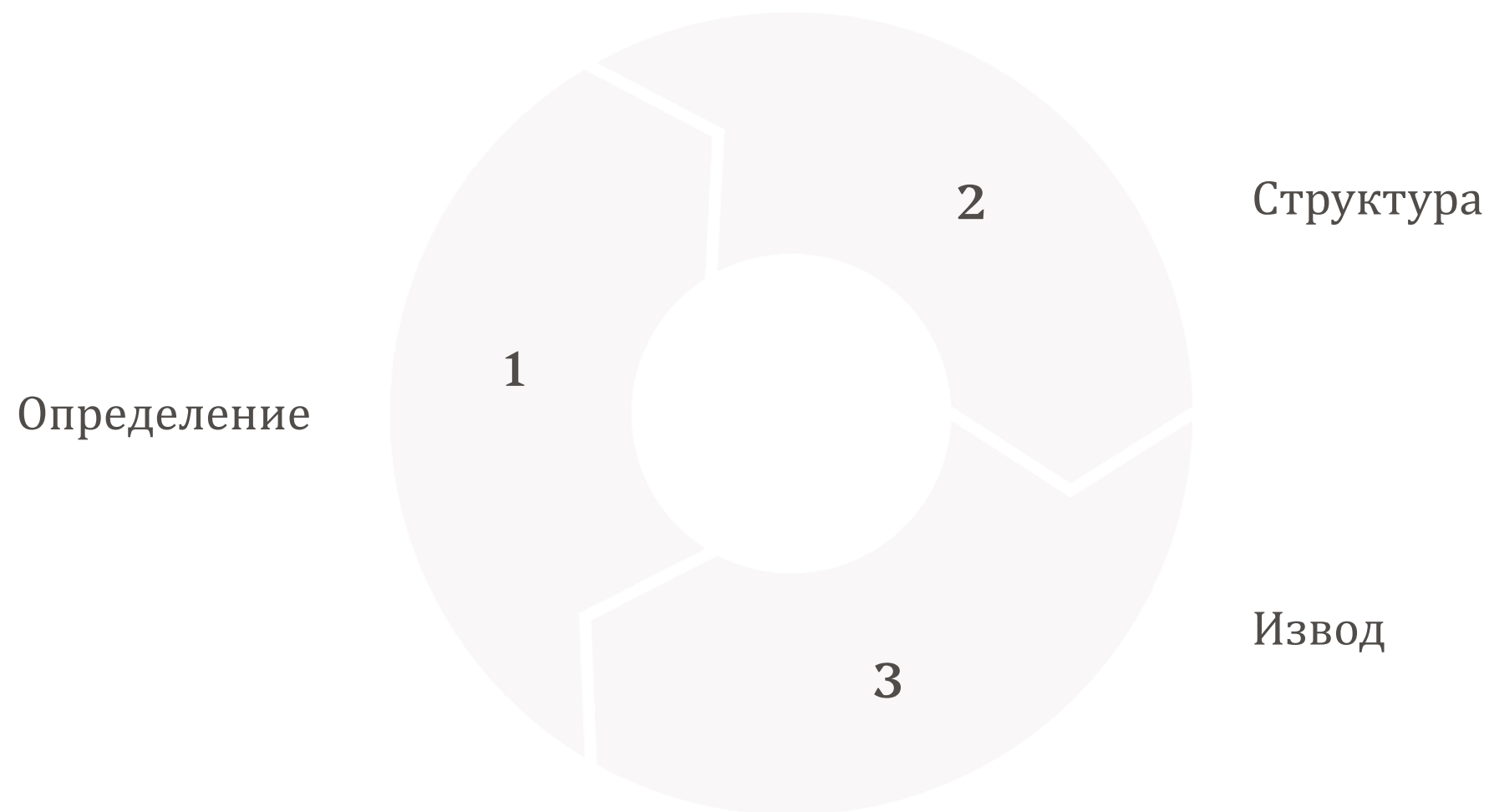
В естествения език, една връзка е символизирана от едно изречение.

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 в Пролог



Правилата определят как да се извличат нови факти. Използват се за дефиниране на отношения между факти. Те имат глава и тяло.

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(maria, 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.

/*Дефиниране на релации чрез правила */

мъж(димитър).

мъж(николай).

мъж(борис).

мъж(филип).

мъж(александър).

мъж(георги).

жена(виктория).

жена(катя).

жена(лили).

жена(ана).

жена(мариана).

родител(A,B):-син(B,A).

родител(A,B):- дъщеря(B,A).

глава

тяло

to/then

ako/if

баща(A,B):-родител(A,B), мъж(A).

глава

тяло

to/then

ako/if

Логическо „И“ / or

майка(A,B):-родител(A,B),жена(A).

Цели в Пролог

1	Въпроси
2	Решения
3	Отговори

Целите са въпроси, които задаваме на Пролог. Пролог се опитва да намери решения. Той връща отговори въз основа на фактите и правилата.

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(;).


```
/*Задаване на цели: */  
| ?- родител(георги, филип).  
true  
| ?- родител(филип, георги).  
false  
| ?- родител(филип, димитър).  
true  
| ?- родител(филип, X).  
X = димитър  
yes
```

преудовлетворяване на целта

```
| ?- родител(филип, X).  
X = димитър ;  
X = николай ;  
X = виктория ;  
false
```

**преудовлетворяване
на целта**

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.

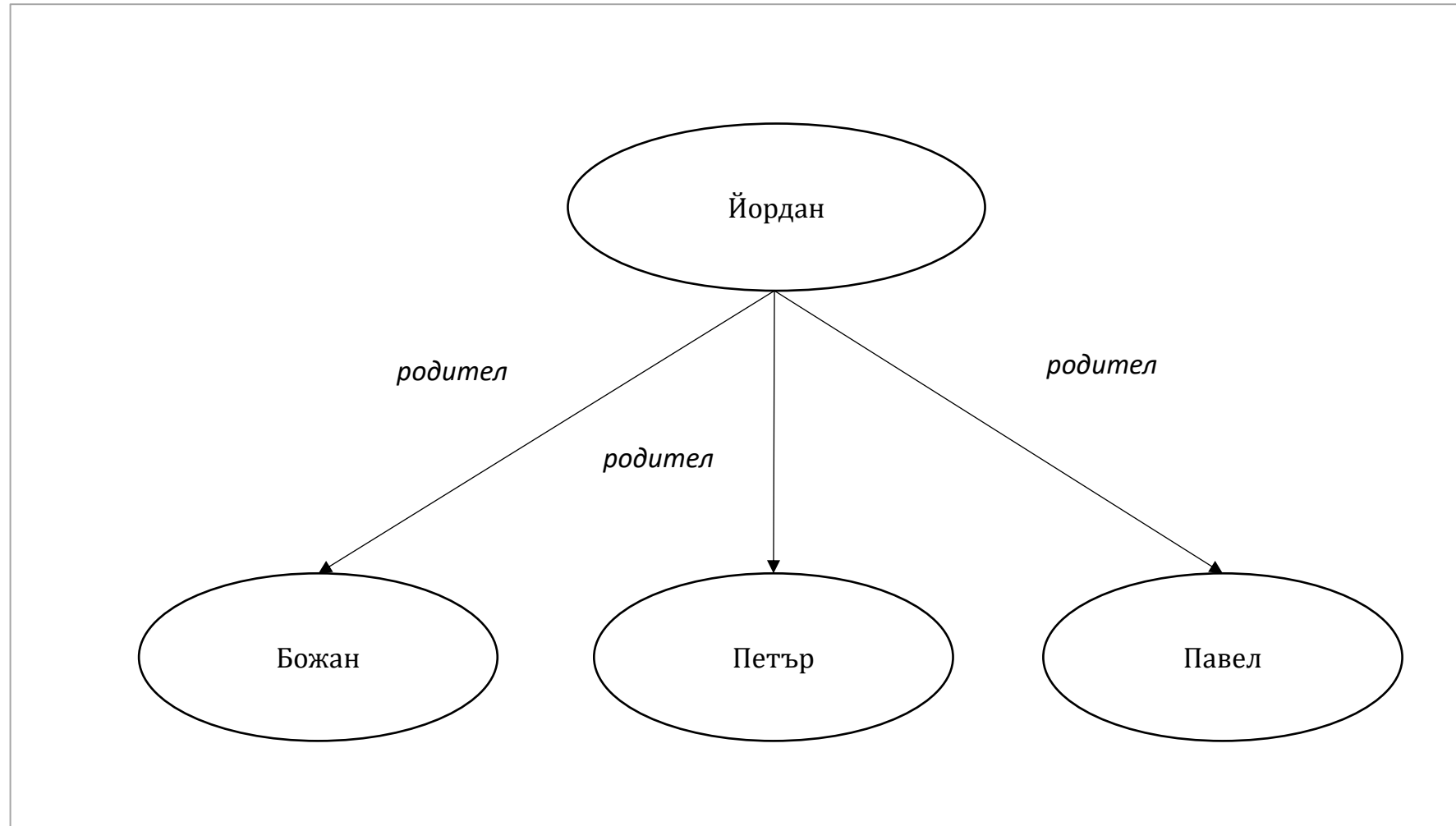
Използване на коментари.

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). жена(султана).
наследник(Y, X) :- родител(X, Y).

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

true.

Клаузи

Клаузите са: факти, правила и цели.

The screenshot shows the SWISH Prolog environment. On the left, a list of clauses is displayed, with a brown circle highlighting the first eight lines:

```
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
```

On the right, the query `grandfather(X,ivan)` is entered in the input field, highlighted by a green circle. Below it, the result `X = marin` is shown, also highlighted by a green circle. A red circle highlights the query `?- grandfather(X,ivan)` in the output area. The bottom of the interface includes buttons for `Examples`, `History`, `Solutions`, a checkbox for `table results`, and a `Run!` button.

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: */
```

```
мъж(димитър).
```

```
мъж(николай).
```

```
мъж(борис).
```

```
мъж(филип).
```

```
мъж(александър).
```

```
мъж(георги).
```

```
жена(виктория).
```

```
жена(катя).
```

```
жена(лили).
```

```
жена(ана).
```

```
жена(мариана).
```




SWI Prolog

Robust, mature, free. **Prolog for the real world.**

HOME

DOWNLOAD

DOCUMENTATION

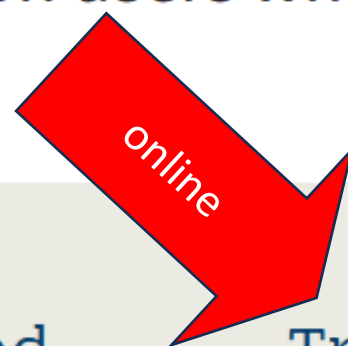
TUTORIALS

COMMUNITY

COMMERCIAL

WIKI

SWI-Prolog offers a comprehensive free Prolog environment. Since its start in 1987, SWI-Prolog development has been driven by the needs of real world applications. SWI-Prolog is widely used in research and education as well as commercial applications. Join over a million users who have downloaded SWI-Prolog. [more...](#)



Download SWI-Prolog



Get Started



Try SWI-Prolog online (SWISH)



Try SWI-Prolog in your browser (WASM)


← → ↻ 🔍 ☆ 📁 👤 ⋮ https://swish.swi-prolog.org


 **SWISH** 

New tab  

Create a **Program** **Notebook** here

based on **Empty** **Student** **CLP** **s(CASP)** profile


user:"me"  Filter ▾ Type ▾ 🔍

 No matching files

If you are a new user you may

- Use the Examples menu from the navigation bar
- Use the Program or Notebook button above

[help on search](#)





?- Your query goes here ...



Examples▴ History▴ Solutions▴ ☐ table results **Run!**




SWI-Prolog

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


SWISH -- SWI-Prolog for SHarir  

← → ↻ 🔍 ☆ 📁 👤 ⋮ https://swish.swi-prolog.org

 **SWISH** 

 Program  


1 Your Prolog rules and facts go here ...



?- Your query goes here ...

Examples▴ History▴ Solutions▴ ☐ table results **Run!**

Download SWI-Prolog

 SWI-Prolog (AMD64, Multi-threaded, version 9.1.2)

File Edit Settings Run Debug Help

Welcome to SWI-Prolog (threaded, 64 bits, version 9.1.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run `?- license.` for legal details.

For online help and background, visit <https://www.swi-prolog.org>
For built-in help, use `?- help(Topic).` or `?- apropos(Word).`

?-

Create new Prolog source

← → ↕

> Този компютър > Документи > Prolog

Търсене в Prolog

Организиране ▾ Нова папка

Този компютър

3D обекти

Видеоклипове

Документи

Audacity

IBM

IBM SPSS Statistics 27.0.1 IF026 (x64) Multilingual

Prolog

ViberDownloads

Име

birds

birds1

birds2

native

Състояние

Дата на промяна

21.9.2024 г. 13:05

4.12.2024 г. 10:32

4.12.2024 г. 10:31

3.6.2024 г. 16:54

Тип

PL файл

PL файл

PL файл

PL файл

Размер

5 КБ

6 КБ

8 КБ

5 КБ

Име на файла: proba1

Запиши като тип: Prolog Source

Скрий папките

Запиши

Отказ

Consult ...

Edit ...

New ...

Reload modified files

Navigator ...

Exit

threaded, 64 bits, version 9.1.2)

ABSOLUTELY NO WARRANTY. This is free software.

for legal details.

For online help and background, visit <https://www.swi-prolog.org>

For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-

Welcome to SWI-Prolog (threaded, 64 bits, version 9.1.2)

SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.

Please run ?- license. for legal details.

For online help and background, visit <https://www.swi-prolog.org>

For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-

ERROR: c:/документи/обучение/ии-ис/изкуствен интелект/2023 си редовно/zadachi/5 деца_таланти (1).pl:18:

ERROR: catch/3: Unknown procedure: use_rendering/2

Warning: c:/документи/обучение/ии-ис/изкуствен интелект/2023 си редовно/zadachi/5 деца_таланти (1).pl:18:

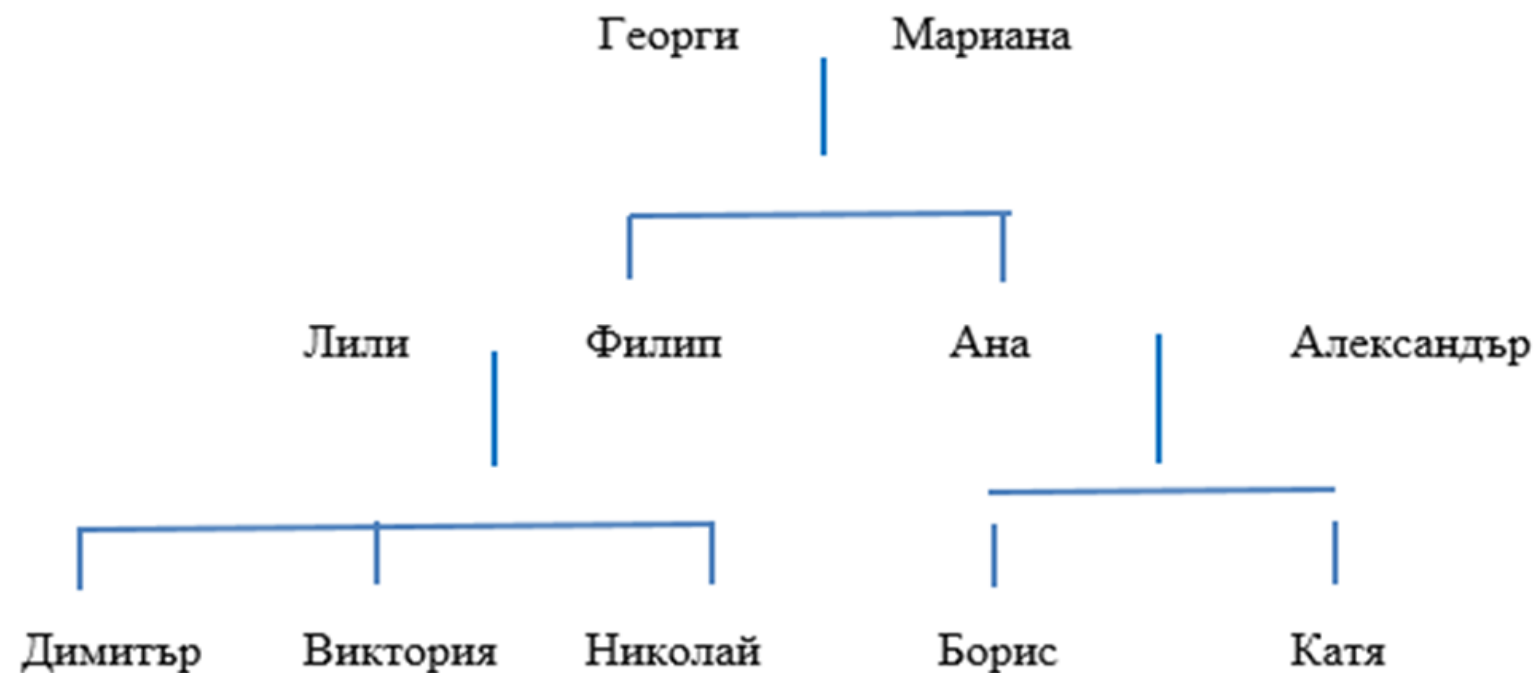
Warning: Goal (directive) failed: user:use_rendering(table,[header(d('P\u0098PjPμ','PŷP°P»P°PSC','P°PsPrPëPSPë'))])

% c:/ДОКУМЕНТИ/ОБУЧЕНИЕ/ИИ-ИС/ИЗКУСТВЕН ИНТЕЛЕКТ/2023 СИ редовно/Zadachi/5 деца_таланти (1).pl compiled 0.00 sec, 2 clauses

?- deca(Ds).

Ds = [d(ivanka, piano, 4), d(stanko, matematika, 5), d(yoana, literatura, 6), d(ivan, kitara, 7), d(nevena, programist, 8)]

Създайте базата знания за родословното дърво от фигурата



Въведете фактите, които виждате, например
мъж/1, жена/1, син/2, дъщеря/2

Дефинирайте общовалидни дефиниции
(правила) за:
родител/2, майка/2, баща/2, баба/2, дядо/2,
брат/2, сестра/2

леля/2, чичо/2, братовчед/2, братовчедка/2,
братовчеди/2,
племенник/1, племенница/2

и други

Примерен Пролог код

```
мъж(димитър). мъж(николай). мъж(борис).  
мъж(филип). мъж(александър). мъж(георги).
```

```
жена(виктория). жена(катя). жена(лили).  
жена(ана). жена(мариана).
```

```
син(филип, георги). син(филип, мариана).  
син(димитър, филип). син(димитър, лили).  
син(николай, филип). син(николай, лили).  
син(борис, ана). син(борис, александър).
```

```
дъщеря(виктория, филип). дъщеря(виктория, лили).  
дъщеря(ана, георги). дъщеря(ана, мариана).  
дъщеря(катя, ана). дъщеря(катя, александър).
```

```
родител(иван, петър).
```

```
родител(мария, петър).
```

```
майка(X, Y) :- родител(X, Y), жена(X).
```

```
жена(мария).
```

```
?- майка(X, петър).
```

```
X = мария.
```

Този код дефинира отношения родител и жена.

Правилото майка извлича майките на даден човек. Целта пита коя е майката на Петър.

Регистрация

`https://tinyurl.com/
29c7bnc1`



Благодаря за вниманието!

