

# Логическое программирование и язык Пролог

# План лекции:

---

1. Понятие логического программирования.
2. Типы предложений в Прологе.
3. Объекты данных – термы.
4. Встроенные предикаты.
5. Структура программы Турбо Пролога.

# Литература:

---

1. Братко, И. Программирование на языке Пролог для искусственного интеллекта [Текст]/ И. Братко. – М.: Мир, 1990. – 560с.
2. Марселлус, Д. Программирование экспертных систем на Турбо Прологе [Текст]/ Д. Марселлус. – М.: Финансы и статистика, 1994. – 256с.
3. Прыкина, Е.Н. Основы логического программирования в среде Турбо Пролог [Текст]/ Е.Н. Прыкина. – Кемерово: КемГУКИ, 2006. – 68с.

# Краткие исторические сведения

---

Пролог – ПРОграммирование в терминах ЛОГики.

Разработан в Марселе в 1972 году.

Первая версия языка Турбо Пролог – в 1986 году.

Турбо Пролог 2.0. – в 1988 году.

# Особенности Пролога

---

1. Декларативный характер написанных на Прологе программ.
2. Вывод с поиском и возвратом.
3. Для хранения данных используются списки, а не массивы.
4. Одним из основных методов программирования является рекурсия.

# Программа на Прологе

---

Модель некоторого фрагмента предметной области, о котором идет речь в решаемой задаче.

Вместо алгоритма решения задачи составляется ее логическая спецификация.

# Логическое программирование

---

Метод программирования, предназначенный для решения задач искусственного интеллекта, в соответствии с которым программа описывает логическую структуру решения задачи, указывая преимущественно, что «нужно сделать», не вдаваясь в детали «как это делается». Практическим воплощением метода является язык Пролог.

# Основные области применения языка Пролог:

- экспертные системы и оболочки экспертных систем;
- создание естественно-языковых интерфейсов для существующих систем;
- быстрая разработка прототипов прикладных программ;
- символьные вычисления для решения уравнений, дифференцирования и интегрирования;
- системы автоматизированного проектирования;
- автоматизированное управление производственными процессами;
- проектирование динамических реляционных баз данных;
- автоматический перевод с одного языка на другой.

С точки зрения логики предикатов элементарные выражения имеют вид:

---

$$p(a_1, \dots, a_n),$$

где  $p$  – имя предиката;

$a_1, \dots, a_n$  - аргументы.

Аргументом предиката может быть константа, переменная или составной объект.

Число аргументов предиката называется его **арностью**.

# ПРИМЕР:

---

«Париж – столица Франции»

$$p(x, y)$$

где  $p$  – предикат, выражающий отношение «являться столицей»;

$x, y$  – объекты «Париж, Франция», связанные отношением  $p$ .



В Прологе имена всех отношений и объектов записываются со строчной буквы.

# Типы предложений в Прологе

---



! Все предложения заканчиваются точкой.

# Основные понятия Пролога

---

**Процедура** - множество предложений, имеющих в заголовке предикат с одним и тем же именем и одинаковым количеством аргументов.

# Факты

---

Факты используются для констатации того, что выполнено некоторое отношение между объектами.

Например,

likes (nick, helen).

likes (nick, helen).  $\neq$  likes (helen, nick).

# Примеры фактов

---

- |                                |                                 |
|--------------------------------|---------------------------------|
| 1. women (helen).              | 1. Елена – женщина.             |
| 2. father (victor, helen).     | 2. Виктор является отцом Елены. |
| 3. give (victor, book, helen). | 3. Виктор дает Елене книгу.     |

# Использование переменных в фактах

---

likes (nick, apple).

likes (helen, apple).       $\longrightarrow$       likes (X, apple).

likes (mary, apple).

...



Совокупность фактов в Прологе называется базой данных.

# Вопросы

---

Система рассматривает вопрос как цель.

? – have (mary, book).

Два факта сопоставимы, если их предикаты одинаковы и их соответствующие аргументы попарно совпадают.

# Вопросы

---

likes (alexander, fish).

likes (alexander, mary).

likes (mary, book).

likes (alexander, book).

? – likes (alexander, car).

нет

? – likes (mary, alexander).

нет

? – likes (mary, book).

да

# Использование переменных в вопросах

---

likes (alexander, fish).

likes (alexander, mary).

likes (mary, book).

likes (alexander, book).

? – likes (alexander, X).

X=fish

X=mary

X=book

? – likes (X, mary).

X=alexander



Переменные принято записывать с прописной буквы.

# КОНЪЮНКЦИЯ

likes (alexander, fish).

likes (alexander, mary).

likes (mary, book).

likes (alexander, book).

Цель: «Существует ли что-нибудь, что нравится Мэри и Александру?»

Подцели:  
«Существует ли что-нибудь, что нравится Мэри?»

**И**

«Нравится ли Александру найденное значение X?»

? – likes (mary, X)

, likes (alexander, X).

# Конъюнкция

---

Процесс нахождения соответствия между целью и фактом или правилом называется *унификацией*.

! Обработывая конъюнкцию целей, Пролог, согласовывая с базой данных каждую цель, просматривает вопрос слева направо.

# Правила

---

Правила описывают (содержит)  
утверждения, зависящее от условий.

Правила используются для выражения  
определений.

Пример:

$X$  является сестрой  $Y$ , если

$X$  является женщиной и

$X$  и  $Y$  имеют одних и тех же родителей.

# Правила

---

child (X, Y) :-

parent (Y, X).

grandfather (X, Y) :-

parent (X, Z), parent (Z, Y), man (X).

parent (X, Y) :-

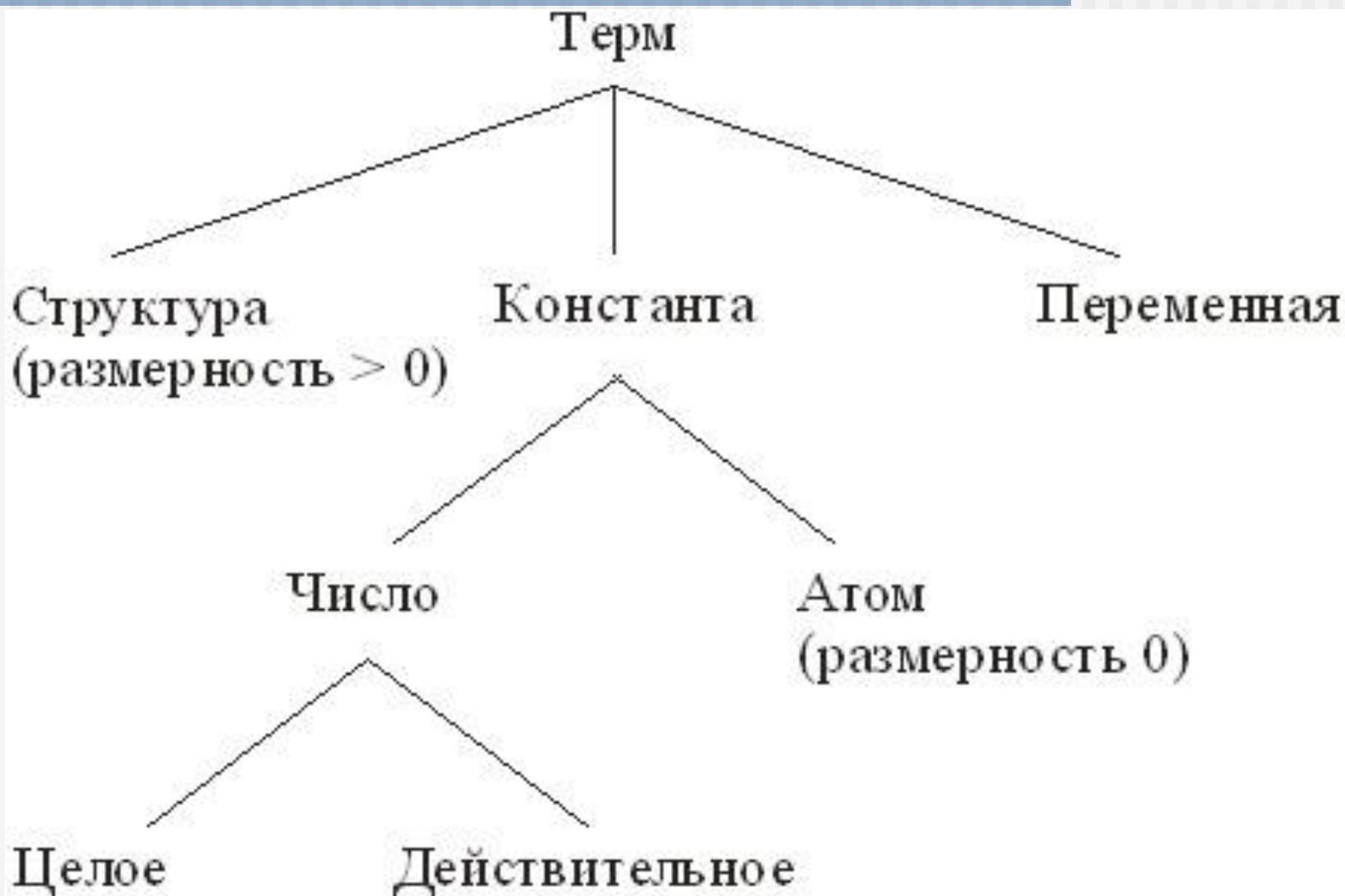
mother (X, Y); father (X, Y).



Символ «:-» имеет смысл логического «Если».

(«;») в последнем правиле соответствует связке «или».

# Объекты данных – термы



# Константы

---

*Константами* являются целые и действительные числа. Диапазон изменения целых чисел от **-32768** до **+ 32767**.

Диапазон изменения действительных чисел от **1E-307** до **1E+308**.

Как и в других языках программирования, константы обозначают конкретные элементарные объекты, а все другие типы данных в Прологе составлены из сочетаний констант и переменных.

# Переменные

---

Переменные служат для обозначения объекта, на который нельзя сослаться по имени. Областью действия переменной в Прологе является одно *предложение*.

Исключением из правила определения области действия является *анонимная переменная*, которая обозначается символом подчеркивания "\_".

# Переменные

---

Переменные, отличные от *анонимных*, называются *именованными*, а неконкретизированные (переменные, которым не было присвоено значение) называются *свободными*.

Переменная, которая получила какое-то значение и оказалась связанной с определенным объектом, называется *связанной*.

# Переменные

---

Имена переменных должны начинаться с заглавной буквы или символа подчеркивания и содержать только символы букв, цифр и подчеркивания.

Примеры переменных:

X, Переменная, \_3, \_переменная.

# Структуры

---

Структура считается более сложной единицей. Она состоит из предиката и аргументов. Аргументы разделяются запятыми и заключаются в круглые скобки.

Например, структура

`likes(john, bird).`

определяет, что Джон любит птиц. Предикатом здесь служит `likes`, а двумя аргументами – `john` и `bird`. Аргументами предиката могут быть константы, переменные и другие структуры.

# Встроенные предикаты

---

*Встроенным* предикатом называется предикат, определение которого уже имеется в Пролог системе.

Существуют следующие встроенные предикаты:

- ввода-вывода;
- для управления файлами;
- выполнения арифметических операций;
- операций сравнения и других полезных операций.

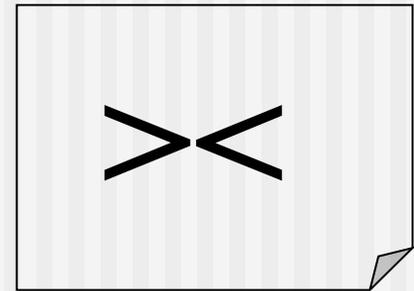
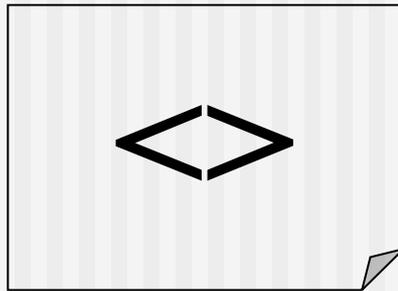
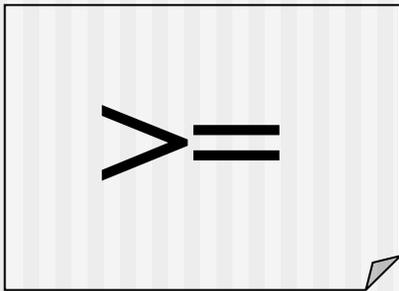
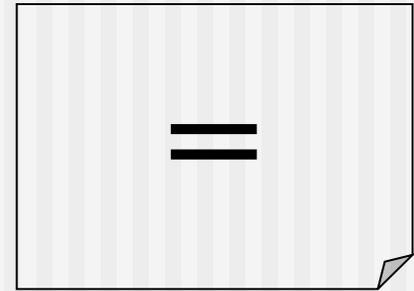
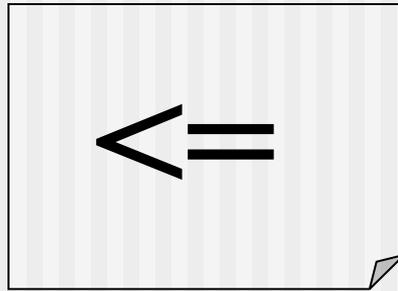
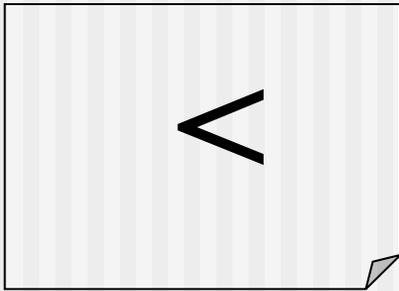
# Арифметические операции

---

A black asterisk symbol representing multiplication, centered within a white square box with a folded bottom-right corner.A black forward slash symbol representing division, centered within a white square box with a folded bottom-right corner.A black plus sign symbol representing addition, centered within a white square box with a folded bottom-right corner.A black minus sign symbol representing subtraction, centered within a white square box with a folded bottom-right corner.The text "div" representing integer division, centered within a white square box with a folded bottom-right corner.The text "mod" representing modulo operation, centered within a white square box with a folded bottom-right corner.

# Операции отношения

---



# ФУНКЦИИ

---

$\text{abs}(X)$

$\text{sin}(X)$

$\text{cos}(X)$

$\text{tan}(X)$

$\text{arctan}(X)$

$\text{exp}(X)$

$\text{ln}(X)$

$\text{log}(X)$

$\text{sqrt}(X)$

$\text{round}(X)$

$\text{trunk}(X)$

# Встроенные предикаты - Отсечение

---

*Отсечение* – обозначается «!».

1. блокируется возврат к предшествующим ему подцелям в данном утверждении
2. предотвращается использование всех утверждений, следующих за данным утверждением в данной процедуре и которые могли бы быть использованы для согласования цели, соответствующей данной процедуре.

# Процедура отсеечения

---

Пусть в программе задана база данных:  
man (ivan).  
man (petr).  
woman (svetlana).  
woman (varvara).

? - man(X).

выведет список всех мужчин;

? – man(X), !.

выведет первого мужчину, найденного в базе данных;

? - man(X), woman(Y).

выведет список возможных пар мужчина – женщина;

? man(X), woman(Y), !.

выведет первую найденную пару.

# Структура программы Турбо Пролога

---

Программа на Турбо Прологе состоит из следующих разделов:

**CONSTANTS** – раздел описания констант;

**DOMAINS** – раздел описания доменов;

**DATABASE** – раздел описания предикатов внутренней базы данных;

**PREDICATES** – раздел описания предикатов;

**CLAUSES** – раздел описания предложений;

**GOAL** – раздел описания внутренней цели.

# Раздел описания констант

---

Имя константы может состоять из английских букв, цифр и знака подчеркивания, причем не может начинаться с цифры.

Каждое определение константы должно размещаться в отдельной строке.

## **CONSTANTS**

const1 = значение 1

const2 = значение 2

# Раздел описания доменов

---

Существуют следующие типы доменов:

- целые – `integer` из промежутка (- 32768...32767);
- вещественные – `real` (лежащее между  $\pm 1e-307$ ... $\pm 1e308$ );
- символьные – `char` (один символ, заключенный в одиночные апострофы);
- строковые – `string` (последовательность символов, заключенная в двойные кавычки);
- символические – `symbol` (последовательность букв латинского алфавита, цифр и знаков подчеркивания начинающихся со строчной буквы, или последовательность любых символов, заключенная в кавычки);
- файловые – `file`.

# Раздел описания доменов

---

Объявление домена имеет следующий вид:

`<имя домена>=<определение домена>`

Списковый домен задается следующим образом:

`<имя спискового домена>=<имя домена  
элементов списка>*`

Например, список целых чисел описывается так:

`list_of_integer=integer*`

# Раздел описания предикатов внутренней базы данных

---

описываются те предикаты, которые можно в процессе выполнения программы добавлять во внутреннюю базу данных или удалять оттуда.

## **DATABASE**

pred1(...).

pred2(.....).

# Раздел описания предикатов

---

Описание предиката имеет следующий вид:

<имя предиката> (<имя домена первого аргумента>, ... <имя домена n-го аргумента>).

PREDICATES

mother(string,string).

# Раздел описания предложений

---

## **CLAUSES**

$p(\dots):-p1(\dots), p2(\dots), \dots .$

$p(\dots):-p1(\dots), p2(\dots), \dots .$

# Раздел описания внутренней цели

---

## **GOAL**

Если этот раздел отсутствует, то после запуска программы Пролог-система выдает приглашение вводить вопросы в диалоговом режиме (внешняя цель).

# Структура программы на языке Пролог

---

domains

... операторы доменов...

predicates

... операторы предикатов...

clauses

... факты и правила...

goal

... цель...