


**Концепция логического  
программирования.  
Программа на языке Пролог.**

Лектор:  
доцент каф. АОИ  
Салмина Нина  
Юрьевна



# Концепция логического программирования (ЛП)

**ЛП** – программирование в терминах целей.

**Процедурные языки** - «КАК» что-либо делать.

**ЛП** – «ЧТО» делать.

**Программист** сообщает, что ему известно (факты и правила) и задает вопросы: больше интересуют знания, меньше – алгоритмы.

**Система** **сама** строит дедуктивный вывод.

# Пролог: набор механизмов

- Сопоставление образцов
- Древовидное представление структур данных
- Автоматический возврат

Программирование в терминах логики:

Декларативная точка зрения на программирование

Символьная обработка данных

## Пример области знаний «Логика»

- Если Дэвид интересуется логикой, то он либо запишется в следующем семестре на занятия по курсу Логика, либо он ленив.
- Если Дэвид самостоятельно изучал литературу по логике, то он интересуется логикой.
- Дэвид самостоятельно изучал литературу по логике.
- Дэвид не ленив.

# Пример «Логика»

## Оформление знаний в логике высказываний

<u>Высказывания</u>	<u>Обозначение</u>	<u>Формализация знаний:</u>
Дэвид интересуется логикой	D	
Дэвид запишется в следующем семестре на занятия по курсу Логика	A	$D \supset A \vee \sim B$
Дэвид не ленив	B	$C \supset D$
Дэвид самостоятельно изучал литературу по логике	C	C
		B
		<hr/> A - ?

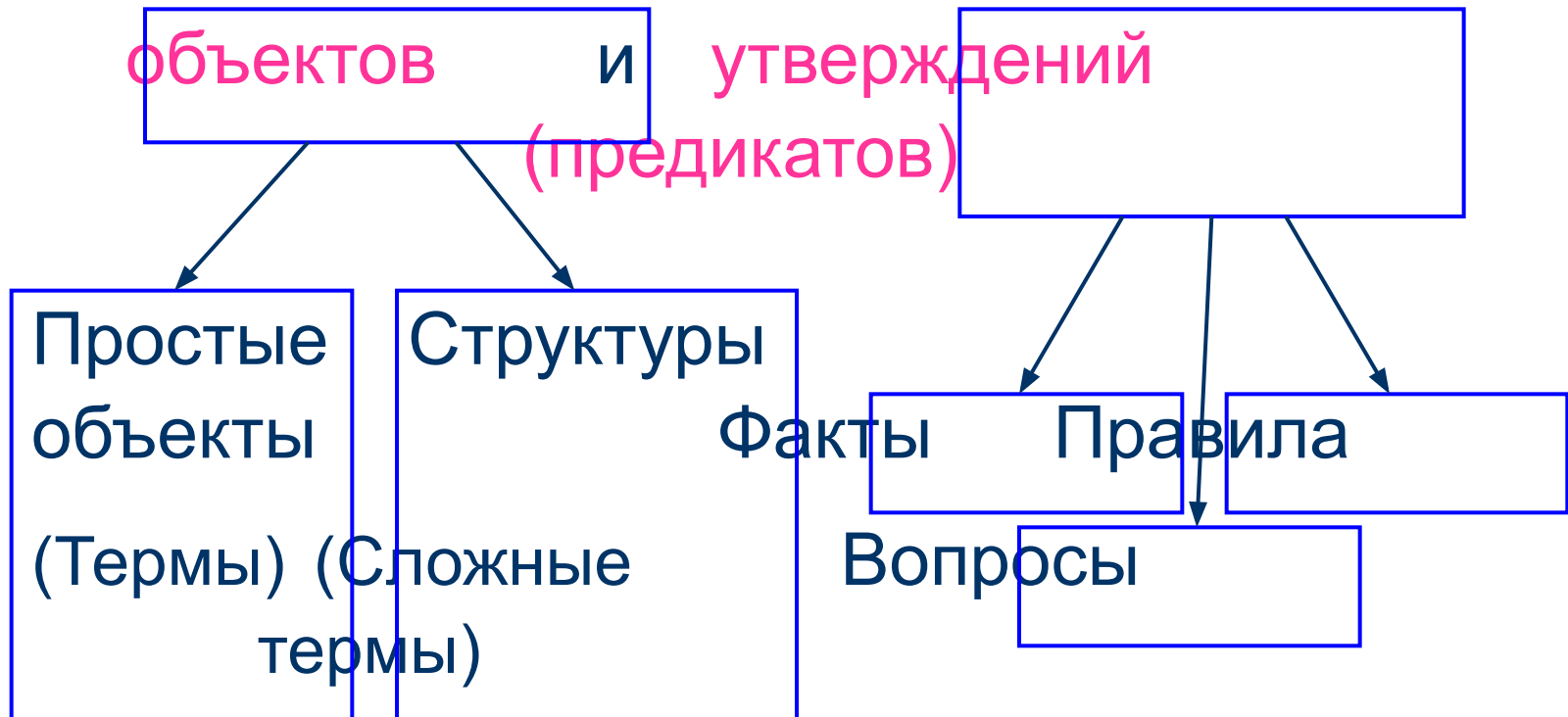
# Пример «Логика»

## Оформление знаний в логике предикатов

<u>Предикат</u>	<u>Обозначение</u>	<u>Формализация</u> <u>знаний:</u>
X интересуется Y	D(X,Y)	D(X,Y) □
X запишется в следующем семестре на занятия по курсу Y	A(X,Y)	A(X,Y) ∨ ~B(X)
X не ленив	B(X)	C(X,Y) □ D(X,Y)
X самостоятельно изучал литературу по Y	C(X,Y)	C(дэвид,логика)
		B(дэвид)
X – некоторое лицо		<hr/>
Y – некоторый предмет (курс)		A(дэвид,логика) - ?

# Модель на Прологе

строится в терминах:



# Простые объекты (термы)

- Константы
  - Атомы (со строчной буквы или в кавычках)
    - Char (отдельный символ в апострофах)
    - Symbol
    - String
  - Числа (целые и вещественные)
    - Integer
    - Real
- Переменные (с прописной буквы или подчеркивания)



# Утверждения (предикаты)

**Факт** – безусловно истинное утверждение.

отсутствие факта означает его неудачное выполнение!

**Правило** – условно истинное утверждение.

**Вопрос** – обращение к Пролог-программе, определяющее достижимость цели.

Записываются в виде **предложений** и заканчиваются точкой.

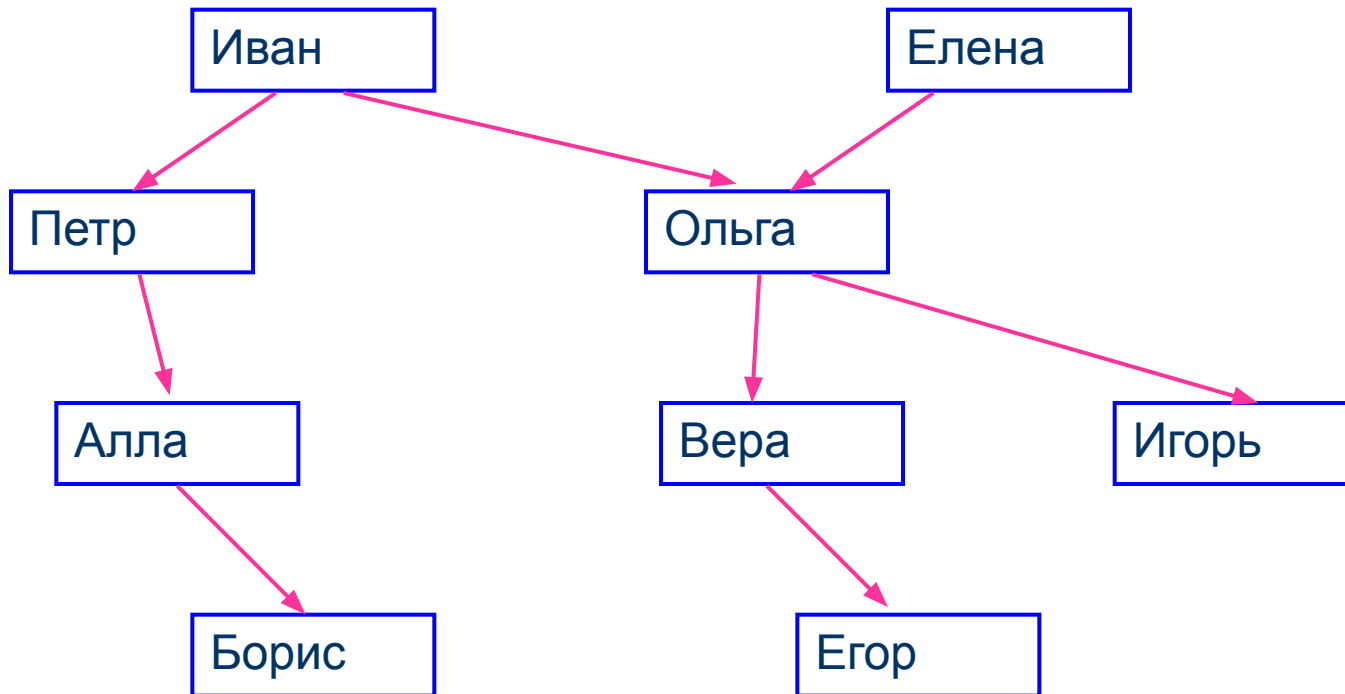
# ФАКТЫ

**<предикат>(объект1,...,объектN).**

КОНСТАНТЫ

*Совокупность фактов – база данных.*

# Пример. Дерево родственных отношений



## Пример. Предикаты.

родитель(иван, ольга).

родитель(иван, петр).

родитель(елена, ольга).

родитель (ольга, вера).

родитель (ольга, игорь).

родитель (петр, алла).

родитель (алла, борис).

родитель (вера, егор).

# Пример. Вопросы.

? – родитель(иван, ольга).  $\implies$  Yes

? – родитель (X, вера).  $\implies$  X=ольга

? – родитель (борис, \_).  $\implies$  No

? – родитель (иван, \_).  $\implies$  Yes

? – родитель (X, Y).  $\implies$

X=иван, Y=петр

Анонимная переменная

(значение роли не играет)

8 решений

# Правила

**[A] : – [B].**

A – заголовок / голова предложения

B – тело правила (может быть пусто):  
состоит из фактов и правил.

A: – B1, B2, B3, ... BN.

«A выполнимо **ЕСЛИ** выполнимы B1 **И** B2 **И**  
... BN»

# Пример «Логика»

## Оформление знаний на Прологе

<u>Предикат</u>	<u>Обозначение</u>	<u>Формализация</u> <u>знаний:</u>
X интересуется Y	D(X,Y)	A(X,Y)
X запишется в следующем семестре на занятия по курсу Y	A(X,Y)	:-D(X,Y),B(X).
X не ленив	B(X)	D(X,Y) :- C(X,Y).
X самостоятельно изучал литературу по Y	C(X,Y)	C(дэвид,логика).
		B(дэвид).
X – некоторое лицо		
Y – некоторый предмет (курс)		
		<hr/> A(дэвид,логика) - ?

## Пример. Правила родственных отношений

отпрыск(X, Y):-родитель(Y, X).

женщина(елена).

женщина(ольга).

мужчина(петр).

мужчина(борис).

мать(X, Y):-родитель(X, Y), женщина(X).

отец(X, Y):-родитель(X, Y), мужчина(X).

сестра(X, Y):- родитель(Z, X), родитель(Z, Y),  
женщина(X).

имеет\_ребенка(X):-родитель(X, \_).



# Переменные в предложениях.

Диапазон имени переменной – одно предложение!

Разные предложения – разные переменные!

# Процедура

Набор правил с одним именем предиката, сгруппированных в одном месте.

A:- B,C.        ИЛИ

A:- D.        ИЛИ

...        ИЛИ

A:- Z,S.

---

**Вызов процедур** – по очереди, в порядке записи.

Если **вызов** процедуры **успешен** – происходит **переход к следующей** процедуре,

если **нет** – **возврат** к ближайшему из предыдущих **вызовов**.

## Пример. Процедура родственных отношений

кузина( $X, Y$ ):-родитель( $X_1, X$ ),родитель( $X_2, Y$ ),  
сестра( $X_1, X_2$ ),женщина( $X$ ).

кузина( $X, Y$ ):-родитель( $X_1, X$ ),родитель( $X_2, Y$ ),  
брат( $X_1, X_2$ ),женщина( $X$ ).

*МОЖНО*

кузина( $S, Y$ ):- женщина( $S$ ),родитель( $X_1, S$ ),  
родитель( $X_2, Y$ ),сестра( $X_1, X_2$ ),

# Унификация термов

Два терма унифицируемы (сопоставимы), если:

- термы абсолютно одинаковы (константы);
- 1-й терм – переменная, 2-й – любой атом;
- у них одни и те же функтор и арность и все их аргументы сопоставимы (структуры).

---

женщина(елена)  $\cong$  женщина(X)

мать(алла, борис)  $\not\cong$  мать(X, егор)

родитель(иван, Y)  $\cong$  родитель(иван, ольга)

# Поиск решения

*Резольвента* – множество целей, которые необходимо выполнить.

## **Поиск (начало – с вопроса):**

формируется резольвента;

выбирается очередная цель из резольвенты;

ищется предложение в программе, чей заголовок унифицируется с целью;

цель заменяется на тело выбранного предложения;

поиск продолжается с новой резольвентой, полученной из старой.

## **Завершение поиска:**

резольвента пуста – решение найдено;

все предложения просмотрены – решение не найдено.

# Пример поиска решения

## Программа

$p(X):-q(X),s(X),r(X).$

$p(X):-u(X).$

$q(a).$

$q(b).$

$s(c).$

$s(b).$

$r(a).$

$u(d).$

$u(a).$

?-  $p(a).$

## Поиск решения

Начальная резольвента:  $p(a).$

1-е предложение:  $p(X):-q(X),s(X),r(X).$

Унификация:  $X=a.$

Новая резольвента:  $q(X),s(X),r(X)$

После унификации:  $q(a),s(a),r(a)$

...  $s(a)$  выполнить не удастся

Выбирается 2-е предложение:

$p(X):-u(X).$

Новое предложение после унификации:

$u(a).$

- нет тела, резольвента пуста –  
вычисление заканчивается успешно.

# Общая структура программы

[include «файл, добавляемый к модулю»]

[global domains]

[Domains]

<описание сложных термов>

[global predicates]

Predicates

<описание предикатов>

Clauses

<набор правил и фактов>

Goal

<цель – задаваемый вопрос>

# Predicates

**<предикат>(тип\_арг1, ..., тип\_аргN).**

*Можно:*

**<предикат>(тип\_арг1 комментарий1, ... ,  
тип\_аргN комментарийN)**

***Допустимо многократное объявление предиката с одним именем, но с разным количеством аргументов или типами данных***



# Пример программы «Семейные отношения»

## predicates

parent(Symbol parent, string child)

man(string)

woman(symbol)

mother(SYMBOL, SYMBOL)

father(SYMBOL, SYMBOL)

sister(string, string)

brother(string, string)

kuzina(string, string)

## clauses

parent(ivan, peter).

parent(ivan, olga).

parent(elena, olga).

parent(peter, alla).

parent(olga, vera). ...

man(«ivan»).

man(peter). ...

woman(olga).

woman(alla). ...

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

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

sister(X, Y):-parent(Z, Y), parent(Z, X), woman(X)

.

brother(X, Y):-parent(Z, Y), parent(Z, X), man(X).

kuzina(X, Y):-woman(X),

parent(X1, X), parent(X2, Y), sister(X1, X2).

kuzina(S, Y):-woman(X),

parent(X1, S), parent(X2, Y), brother(X1, X2).

## goal

kuzina(vera, alla).

# Сложные термы

- Структуры
- Перечислимые термы
- Списки

# Структуры

## $S(X_1, X_2, \dots, X_N)$

**S** – имя структуры (функтор).

**X<sub>i</sub>** – компоненты (аргументы) структуры: константы; переменные; структуры.

**N** – число аргументов (арность).

---

дата(1, май, 1983)

дата(День, Месяц, Год)

точка(4,2)

треугольник(точка(4,2),точка(6,4),точка(7,1))

---

Описание структуры:

## $T = S(X_1, X_2, \dots, X_N)$

**T** – имя типа (может совпадать с именем структуры)

# Сложные термы (описание)

## Перечислимые термы

**$T = k_1; k_2; \dots k_N$**

$T$  – имя типа

$k_i$  – одно из возможных значений

## Списки

**$T = \langle \text{тип элементов списка} \rangle^*$**

## Примеры:

$\text{direction} = \text{north}; \text{south}; \text{west}; \text{east}$

$\text{sp} = \text{integer}^*$

$\text{list} = \text{direction}^*$

$\text{list2} = \text{sp}^*$

# «Семейные отношения», структуры данных

## domains

pol = man; woman

date = day(integer **day**, integer **month**, real **year**)

ass = string\*

## predicates

Person (string **fam**, symbol **name**, pol, date)

Child (string, ass)

Family (string **fam**, string **name\_husband**, string  
**name\_wife**, ass **children**)

age\_husband (string **fam**, date **data\_today**, integer **age**)

Age (date, date, integer)

# «Семейные отношения», программа

## clauses

Family (ivanov, ivan, lena, [olga,peter]).

Family (sidorov, david, olga, [vera, igor]).

child(X,Y) :- family(X,\_,\_,Y).

Person (ivanov, "ivan", man, day(5,1,1960)).

Person (ivanov, lena, woman, day(21,4,1962)).

Person (ivanov, olga, woman, day(13,12,1990)).

Age (day(D,M,G), day(A,B,C), Y) :- B>M, Y=G-C-1.

Age (day(D,M,G), day(A,B,C), Y) :- B<M, Y=G-C.

Age (day(D,M,G), day(A,B,C), Y) :- B=M, D>A, Y=G-C-1.

Age (day(D,M,G), day(A,B,C), Y) :- B=M, D<=A, Y=G-C.

age\_husband (X,C,Y) :- family(X,Z,\_,\_), person(X,Z,\_,A), age(C,A,Y).

## goal

age\_husband (ivanov, day(1,9,2017), Y).