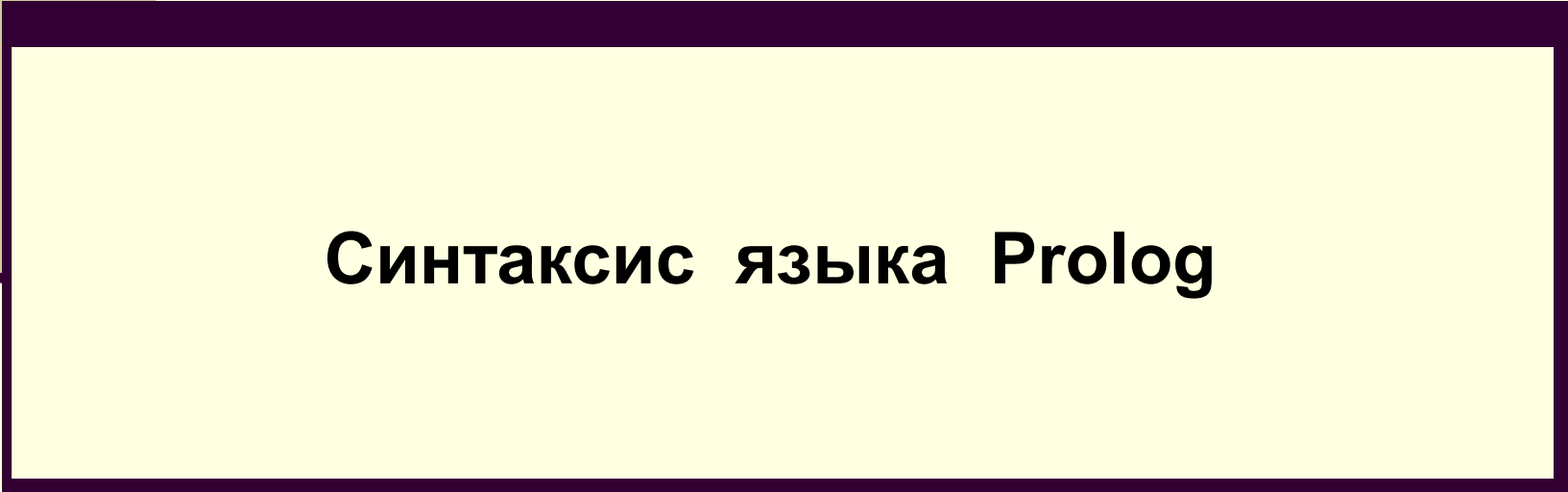




# Prolog



**Синтаксис языка Prolog**

# Основные элементы языка Пролог

**Алфавит языка Пролог** включает следующие символы:

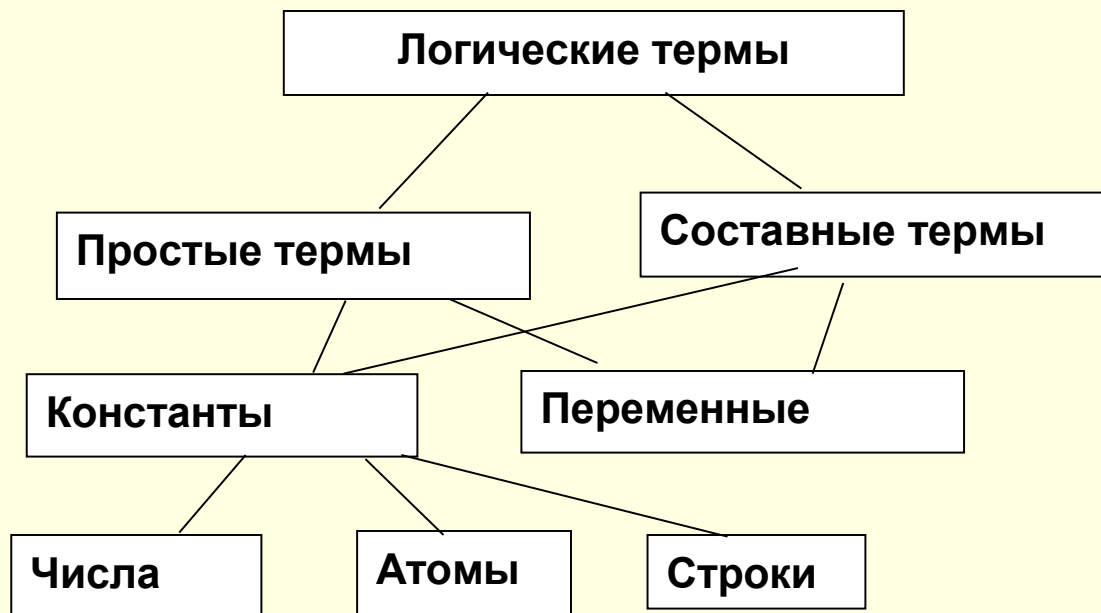
A, B, C, ..., Z, a, b, c, ..., z — прописные и строчные буквы латинского алфавита.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9 — цифры.

+ - = \* / < > [ ] : ; , | . —специальные знаки.

**Основные конструкции** логического программирования - термы и утверждения.

# Определение и классификация термов



# Числовые константы

---

Числа в языке Пролог используются **целые и вещественные**.

**Целые числа** записываются так же, как в любом другом языке программирования; целые отрицательные числа записываются со знаком, в записи положительных чисел знак можно опустить, например, 135, 0, -89.

**Вещественные** константы могут быть представлены в форме с фиксированной точкой и с плавающей точкой, например, 135.712 и 0.135712E+3, соответственно.

# АТОМЫ

## Атомы

Последовательность латинских букв, цифр и знака подчеркивания, начинающая со строчной буквы.

Например, `may`, `all`, `factorial` — символьные атомы.

Последовательность специальных знаков есть атом

Например, `:-`, `>=`, `*`.

Любое слово из букв латинского и русского алфавита и цифр, заключенное в апострофы

Например, `'Mary'`, `'Маша'`, `'1_мая'`, `'Arity_Prolog'`.

Символьные атомы не должны содержать пробелы.

# Переменные

---

**Имя переменной в Прологе** — это последовательность латинских букв, цифр и знака подчеркивания, начинающаяся с прописной буквы или знака подчеркивания.

Например, X, All, S1 — переменные.

Переменные используются для представления объектов, значения которых определяются в ходе решения задачи. Переменные записываются в качестве аргументов предикатов в Пролог-программе и в запросах.

# Анонимные переменные

---

Если значение аргумента предиката не принимается во внимание, то этот аргумент обозначается **анонимной переменной**, то есть вместо имени переменной указывается знак подчеркивания «\_».

# Строки

---

Строки — это последовательности символов, заключенная в апострофы. Строки используются в задачах обработки текстов на естественных языках. " — пустая строка. Строки могут включать пробелы, например, '1 января 2003 года ' есть строка, и 'Turbo-Prolog' тоже строка.



# Составные термы

---

Составной терм — это конструкция вида  $f(t_1, t_2, \dots, t_k)$ , где  $f$  — символьный атом, определяющий функциональную константу или главный функтор, а  $t_1, t_2, \dots, t_k$  — термы, каждый из которых может быть составным термом. Составной терм по-другому называется структурой.

# Примеры составных термов

---

book(Author, Title, Year) — пример составного терма.

.(H, T) – список, пример составного терма.

# Представление фактов

---

Самая простая программа на Прологе является множеством фактов.

**Факт** — это предикатная структура, заканчивающаяся точкой, которая записывается следующим образом:

<имя предиката>(<терм<sub>1</sub>>, <терм<sub>2</sub>>, ..., <терм<sub>n</sub>>).

## Представление фактов (продолжение)

---

Факт представляет собой утверждение о том, что соблюдается некоторое отношение. С их помощью можно выражать произвольные отношения между объектами.

Например, `student('Иванов', 'МГГУ')`.

Этот факт определяет отношение между термами, первым из которых является фамилия студента, а вторым — место его учебы.

# Синтаксические правила записи фактов:

---

- Имя предиката в утверждении-факте есть символьный атом.
- После имени предиката записывается список аргументов в круглых скобках. Между именем предиката и открывающей скобкой '(' не должно быть пробела.

# Синтаксические правила записи фактов:

---

3. Возможны нуль—местные предикаты в фактах, т.е. предикаты, не имеющие аргументов.
4. В качестве аргументов в списке могут быть как переменные, так и константы.
5. В конце факта обязательна точка.

# Переменные в фактах

Переменные в фактах неявно связываются квантором всеобщности. Это означает, что факт

$p(T_1, T_2, \dots, T_n)$ .

где  $T_i$  – переменные следует понимать так

$\forall(T_1) \forall(T_2) \dots \forall(T_n) p(T_1, T_2, \dots, T_n)$  —

истина.

Факты, содержащие переменные, называются **универсальными**.

# Примеры универсальных фактах

---

Примерами универсальных фактов являются следующие утверждения:

$\text{plus}(X, 0, X)$ .

что означает, сумма любого числа  $X$  с нулем равна  $X$ .

$\text{proiz}(X, 0, 0)$ .

что означает, произведение любого числа  $X$  с нулем равна  $0$ .



# Представление правил

---

**Правила** — это средство определения новых утверждений с помощью уже существующих в Пролог-программе утверждений (фактов и правил).

С точки зрения синтаксиса языка Пролог правило — это утверждение вида

$A:— B_1, B_2, \dots, B_n. \quad (n \geq 0)$

где  $A$  — заголовок правила, а конъюнкция предикатов  $B_1, B_2, \dots, B_n$  называется телом правила.

# Переменные в правилах

В правилах, так же как и в фактах, переменные неявно связаны **квантором всеобщности**.

Например, утверждение «Точка с координатами  $(X, Y)$  принадлежит окружности с радиусом, равным 2, и с центром в точке начала координат, если  $X^2 + Y^2 = 4$ .»

На Прологе будет записано следующим образом:

`circle(X, Y):— X*X+Y*Y=4.`

что означает для **любых**  $X$  и  $Y$ , таких что  $X^2 + Y^2 = 4$ , точка  $(X, Y)$  принадлежит окружности с радиусом, равным 2, и с центром в точке начала координат.

# Процедуры

---

Набор правил, заголовки которых имеют одно и то же имя и арность (число аргументов), описывает одно и то же отношение и называется **процедурой**.

Правила, составляющие одну и ту же процедуру, должны следовать в тексте программы подряд. В процедуру нельзя включать правила с другим заголовком.

# Область действия переменных

---

Логические переменные служат для обозначения неопределенных объектов.

**Областью действия переменной** является одно утверждение (факт или правило).

# Подстановки

---

**Подстановкой** называется конечное (возможно, пустое) множество пар вида  $X_i = t_i$ , где  $X_i$ —переменная, а  $t_i$ —терм, не содержащий переменную  $X_i$ .

# Подстановки и конкретизация переменных

---

**Результат применения подстановки  $\theta$  к утверждению  $A$  обозначается  $A\theta$  и получается путем замены каждого вхождения в  $A$  каждой переменной  $X_i$  на соответствующий терм  $t_i$ .**

С помощью подстановок производится конкретизация переменных (аналог присвоения значений) .

# Примеры утверждений

Утверждение В является **примером** утверждения А, если найдется такая подстановка  $\theta$ , что  $V=A\theta$ .

Например, факт  $\text{summa}(1,2,3)$ , которое означает, что  $1+2=3$ , является примером утверждения  $\text{summa}(X,Y,3)$  при применении подстановки  $\theta=\{X=1, Y=2\}$ .

Утверждение С называется **общим примером** утверждений А В, если найдутся такие подстановки  $\theta_1$  и  $\theta_2$ , что  $C=A\theta_1$  и  $C=B\theta_2$ , т.е. С является примером А и В одновременно. Например, факт  $\text{summa}(1,2,3)$  является общим примером утверждений  $\text{summa}(X,2,Z)$  и  $\text{summa}(1,Y,Z)$  при применении подстановок  $\theta_1=\{X=1, Z=3\}$  и  $\theta_2=\{Y=2, Z=3\}$ .

# Примеры утверждений

---

Утверждение В является **примером** утверждения А, если найдется такая подстановка  $\theta$ , что  $B=A\theta$ .

Например, факт  $\text{summa}(1,2,3)$ , которое означает, что  $1+2=3$ , является примером утверждения  $\text{summa}(X,Y,3)$  при применении подстановки  $\theta=\{X=1, Y=2\}$ .



# Вопросы (запросы).

---

Вопрос (целевое утверждение) — это средство извлечения информации из логической программы.

С помощью вопроса выясняется, истинно ли некоторое утверждение или нет. С точки зрения логики поиск ответа на вопрос состоит в определении того, является ли утверждение (вопрос) логическим следствием программы или нет.

# Простые вопросы

---

Вопросы, состоящие из одной цели, называются простыми вопросами.

# Конъюнктивные вопросы

---

Вопрос, включающий в себя конъюнкцию предикатов  $p_1, p_2, \dots, p_n$ , называется **конъюнктивным вопросом**.

Каждый предикат  $p_i$  называется целью.  
Конъюнктивный вопрос — это конъюнкция целей.

# Переменные в вопросах

---

Вопрос, не содержащий переменных, называется **основным вопросом**, а вопрос, включающий переменные, называется **неосновным**.

**Переменные в вопросах** неявно связаны квантором существования.

# Переменные в вопросах

Вопрос

Goal:  $p(X_1, X_2, \dots, X_n)$ .

где  $X_1, X_2, \dots, X_n$ —переменные, предполагает утвердительный ответ, если существует такой набор термов  $t_1, t_2, \dots, t_n$ , что подстановка  $\{X_1=t_1, X_2=t_2, \dots, X_n=t_n\}$  в предикат  $p$  дает результат “истина”.

Если существует, хотя бы одна такая подстановка, то вопрос

Goal:  $p(X_1, X_2, \dots, X_n)$ .

выводим из логической программы, т.е. является логическим следствием программы.

# Общие переменные в конъюнктивных вопросах

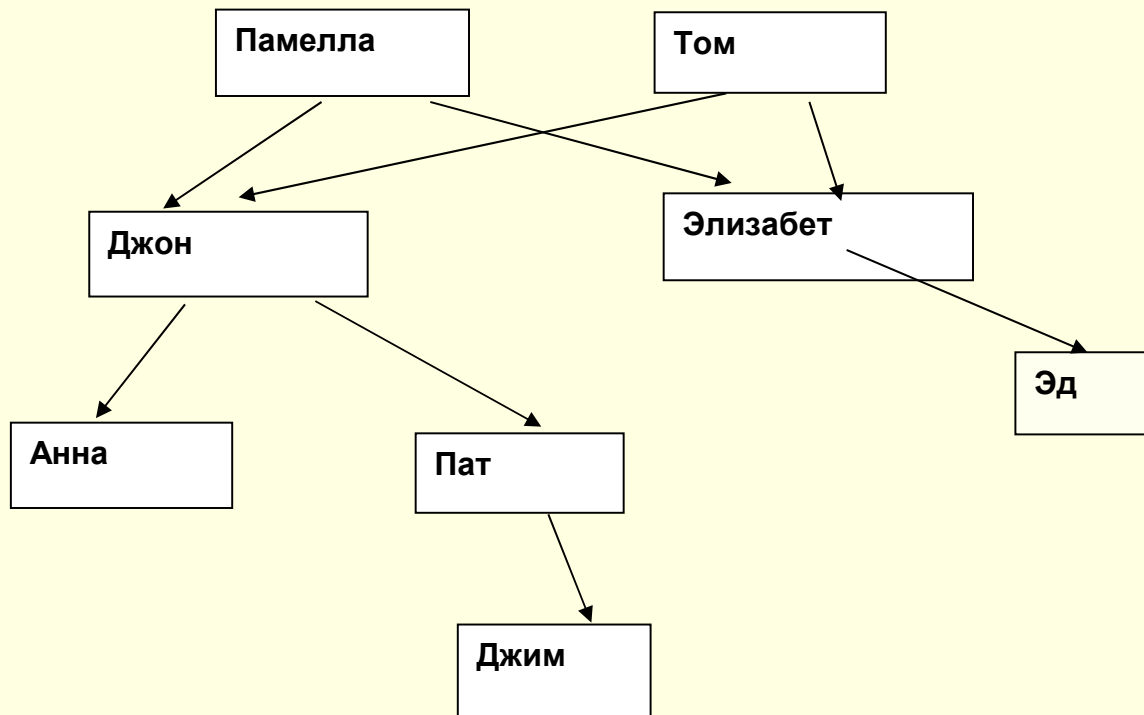
---

Конъюнктивные вопросы обычно содержат **общие переменные**. Переменные называются **общими**, если они входят в две или более цели конъюнктивного запроса.

# Пример простой Пролог—программы

Программа «Родственники» является примером простой Пролог—программы.

Пусть имеется генеалогическое дерево, определяющее степень родства между людьми.



# Отношение parent

---

Родственные отношения можно записать с помощью фактов, соответствующие отношению `parent(<имя родителя>, <имя ребёнка>)`

`parent('Памелла', 'Джон').`

`parent('Памелла', 'Элизабет').`

`parent('Том', 'Джон').`

`parent('Том', 'Элизабет').`

`parent('Джон', 'Анна').`

`parent('Джон', 'Пат').`

`parent('Элизабет', 'Эд').`

`parent('Пат', 'Джим').`



## Отношение person

---

Расширим эту программу фактами, определяемыми схемой отношения person(<имя>, <пол>, <возраст>):

person('Памелла', 'ж', 72).

person('Том', 'м', 78).

person('Джон', 'м', 42).

person('Элизабет', 'ж', 35).

person('Эд', 'м', 14).

person('Анна', 'ж', 20).

person('Пат', 'ж', 25).

person('Джим', 'м', 3).

Отношение person определяет характеристики человека

# Текст программы

```
parent('Памелла', 'Джон').  
parent('Памелла', 'Элизабет').  
parent('Том', 'Джон').  
parent('Том', 'Элизабет').  
parent('Джон', 'Анна').  
parent('Джон', 'Пат').  
parent('Элизабет', 'Эд').  
parent('Пат', 'Джим').  
person('Памелла', 'ж', 72).  
person('Том', 'м', 78).
```

# Текст программы

---

```
person('Джон', 'м', 42).  
person('Элизабет', 'ж', 35).  
person('Эд', 'м', 14).  
person('Анна', 'ж', 20).  
person('Пат', 'ж', 25).  
person('Джим', 'м', 3).
```

# Примеры вопросов к программе «Родственники»

---

## Вопрос 1.

Вопрос "Является ли Пат родителем Джима?" на Прологе можно задать следующим образом:  
? - parent('Пат', 'Джим').

Пролог-система будет искать в программе факт, совпадающий с вопросом, и, обнаружив такой факт, система выдаст ответ 'YES'.

В случае, когда соответствующий факт не обнаружен, система выдаст ответ 'NO'.

# Примеры вопросов к программе «Родственники»

---

## Вопрос 2.

Вопрос "Кто отец Элизабет и сколько ему лет?" на Прологе можно задать следующим образом:

? - parent(X,'Элизабет'),person(X,'м', Y).

Пролог-система выдаст ответ:

X=Том

Y=78

YES

Если возраст не интересует пользователя, то в вопросе используется анонимные переменные, обозначаемые знаками подчеркивания '\_'.  
\_

# Примеры вопросов к программе «Родственники»

---

## Вопрос 3.

Вопрос "Кто отец Элизабет?" на Прологе можно задать следующим образом:

? - parent(X, 'Элизабет'), person(X, 'м', \_).

Пролог-система выдаст ответ:

X=Том

YES

# Примеры вопросов к программе «Родственники»

---

## Вопрос 3.

Приведенные примеры вопросов относятся к программе, состоящей из одних фактов. Для того чтобы сократить и упростить вопросы в Пролог программах задаются правила.

Вопрос 3 можно упростить, если задать следующее правило:

“X является отцом Y, если X является родителем Y, и X – мужчина.”

На языке Пролог это правило записывается так:

```
father(X,Y):-parent(X,Y),person(X,'м',_).
```

# Примеры вопросов к программе «Родственники»

---

## Вопрос 3.

А вопрос 3 записывается следующим образом:

? - father(X, 'Jim').

Пролог-система выдаст тот же ответ:

X=Том

YES



# Лабораторная работа № 1. Простейшая программа на языке Пролог.

Необходимо выполнить следующие действия:

1. Описать с помощью фактов 4-уровневое генеалогическое дерево в Пролог—программе “Родственники”, включающей предикаты `parent` и `person`.

2. Написать правила, определяющие следующие отношения:

“X является отцом Y” .

“X является бабушкой Y” .

“X является сестрой Y” .

“X является племянником Y” .

“X является племянницей Y” .

# Лабораторная работа № 1. Простейшая программа на языке Пролог.

---

“X является родителем родителя Y” .

“X является прадедушкой Y” .

“X является двоюродным братом Y” .

3. Отладить программу с помощью интерпретатора SWI Prolog.

4. Продемонстрировать работу программы с помощью вопросов.

5. Составить отчет по лабораторной работе.