

# Интеллектуальные информационные системы. Семинар 1.

Основные понятия и определения

# Определение интеллектуальной информационной системы

## Интеллектуальная информационная система

(ИИС) - комплекс программных, лингвистических и логико-математических средств для реализации основной задачи – осуществления поддержки деятельности человека и поиска информации в режиме продвинутого диалога на естественном языке. ИИС являются разновидностью интеллектуальной системы, а также одним из видов информационных систем.

## Информационная система - это

совокупность данных (или баз данных), систем управления базами данных и прикладных программ, функционирующих на вычислительных средствах как единое целое для решения определенных задач.

Информационная система является системой обработки данных, хранящихся в базах данных, для манипулирования данными используется язык запросов.

## **Интеллектуальная система - это**

**Интеллектуальная система** (ИнтС, англ. *intelligent system*) — это техническая или программная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы.

## **Структура интеллектуальной системы**

включает три основных блока — базу знаний, механизм вывода решений и интерфейс на естественном языке.

Интеллектуальная система является системой обработки знаний. Для манипулирования знаниями используются языки представления и извлечения знаний.

## Языки программирования ИИС

Для разработки интеллектуальных систем могут применяться различные языковые средства. Учитывая многообразие форм построения и областей применения современных интеллектуальных систем, в качестве языковых средств разработки могут применяться как традиционные языки программирования достаточно низкого уровня, такие как C++, Object Pascal (Delphi), Java или Basic, так и специализированные языки разработки интеллектуальных систем.

# Специализированные языки программирования ИИС

Специализированные языки программирования ИИС – это языки программирования, ориентированные на разработку интеллектуальных информационных систем, как правило имеющие парадигму, отличную от традиционных для языков программирования общего назначения процедурных парадигм.

# Процедурная и декларативная парадигмы программирования

Процедурные (императивные) парадигмы (процедурный подход) – ***как сделать?***

Процедурные языки служат для описания алгоритмов.

Декларативные парадигмы (декларативный подход) – ***что мы знаем? что хотим узнать?***

Декларативные языки служат для описания объектов предметной области и связей между объектами.

# Объектно-ориентированная парадигма программирования

Промежуточное положение между процедурным и декларативным подходом занимает объектно-ориентированный подход (ООП): при рассмотрении объектов в программе, как моделей объектов реального мира – ООП можно отнести к декларативному подходу, при рассмотрении реализации поведения объектов – ООП можно отнести к императивному подходу.

# Декларативные языки программирования

К языкам программирования интеллектуальных систем можно отнести:

LISP - функциональный язык программирования;

PROLOG — логический язык программирования;

CLIPS -специализированный язык разработки экспертных систем

и многие другие.

# Язык Пролог

Язык программирования Пролог (PROgramming LOGic – программирование в терминах логики) предполагает получение решения задачи при помощи логического вывода из ранее известных фактов и правил. Пролог был создан в 1971 г. на факультете естественных наук в Марселе.

# Язык Пролог — декларативный язык

Программа на языке Пролог не является последовательностью действий – она состоит из описания задачи (набор фактов и правил), а Пролог-система сама строит логический вывод на основе введённых данных. В связи с этим Пролог считается **декларативным** языком программирования.

# Предложения программы на языке Пролог

Программа на прологе состоит из ***предложений***.  
Предложения могут быть трех видов: ***факты,***  
***правила, вопросы***. Все предложения строятся из термов.

# Термы

**Терм** – синтаксический объект одной из следующих категорий:

**константы;**

**переменные;**

**предикаты, структуры** (составные термы или функции), состоящие из имени предиката и списка аргументов-термов; имена предикатов начинаются со строчной буквы.

# Константы

***Константы*** - это поименованные конкретные объекты или отношения. ***Константа*** начинается со *строчной буквы*, либо заключаются в одинарные кавычки. Также константа может быть числом.

# Переменные

**Переменные** служат для обозначения объектов, значения которых меняются в ходе выполнения программы. Имена переменных начинаются с заглавных букв или знака «\_» Область действия переменной – предложение. Одноименные переменные в разных предложениях могут иметь разные значения.

# Предикат

Предикат вида  $p(t_1, t_2, \dots, t_n)$ ,  
где  $p$  — имя предиката, а  $t_1, t_2, \dots, t_n$  —  
термы, является **элементарной  
формулой**.

Примеры предикатов:

`student('Дмитриев', m, 'МЭИ', 1)`

`student('Петрова', f, 'МЭИ', 1)`

Предикату `student` соответствует схема  
отношения:

`student(фамилия, пол, вуз, курс)`

## Понятие предиката

Предикат вида  $p(t_1, t_2, \dots, t_n)$ ,

где  $p$  — имя предиката, а  $t_1, t_2, \dots, t_n$  —  
термы, является **элементарной  
формулой**.

Примеры предикатов:

`student('Дмитриев', m, 'МЭИ', 1)`

`student('Петрова', f, 'МЭИ', 1)`

Предикату `student` соответствует схема  
отношения:

`student(фамилия, пол, вуз, курс)`

# Факты в языке Пролог

Простейшим типом предложения является факт.

***Факт*** – это безусловно истинное утверждение Пролог-программы. Факт записывается в виде предиката с точкой на конце.

## Примеры фактов

student('Дмитриев',m,'МЭИ',1).

student('Петрова',f,'МЭИ',1).

part('двигатель', 'автомобиль').

parent(tom, bob).

# 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

$$A: \text{---}V_1, V_2, \dots, V_n. \quad (n \geq 0)$$

где  $A$  — заголовок правила, а конъюнкция предикатов  $V_1, V_2, \dots, V_n$  называется телом правила.

## Представление правил

При такой записи «, $\rangle$  соответствует знаку конъюнкции  $\&$  (логическое И), а знак “:—” соответствует знаку импликации  $\leftarrow$  (ЕСЛИ).

Правило — это условное предложение языка Пролог.

$A: \text{—} V_1, V_2, \dots, V_n. \quad (n \geq 0)$

Читается как  $A \text{—}$  истинно, если истинны предикаты  $V_1, V_2, \dots, V_n$  одновременно.

# 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).
```

# Простейшая программа на языке Пролог

Простейшая программа на языке Пролог состоит из фактов.

Любой логической программе можно задавать вопросы.

# 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.

# 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.

# План практических занятий для бакалавров по дисциплине «Интеллектуальные информационные системы»

ТЕМЫ ЛАБОРАТОРНЫХ РАБОТ	
	Семестр 4, весна 2017 г.
ЛР1	Язык Пролог. Программа «Родственники» в среде SWI Prolog.
ЛР2	Встроенные предикаты языка SWI Prolog. Управление логическим выводом.
ЛР3	Рекурсия в Прологе. Вычислительные программы на Прологе
	Семестр 5, осень 2017 г.
ЛР4	Язык Пролог. Разработка экспертной системы в среде SWI Prolog.
ЛР5	Язык CLIPS. Функциональный стиль программирования. Определение функции на языке CLIPS для вычисления значения функции $Y = f(X)$ с выбором формулы в соответствии с условием.
ЛР6	Язык CLIPS. Процедурные стандартные функции. Рекурсия в языке CLIPS.
ЛР7	Язык CLIPS. Построение баз знаний на языке CLIPS. Факты, правила и шаблоны.