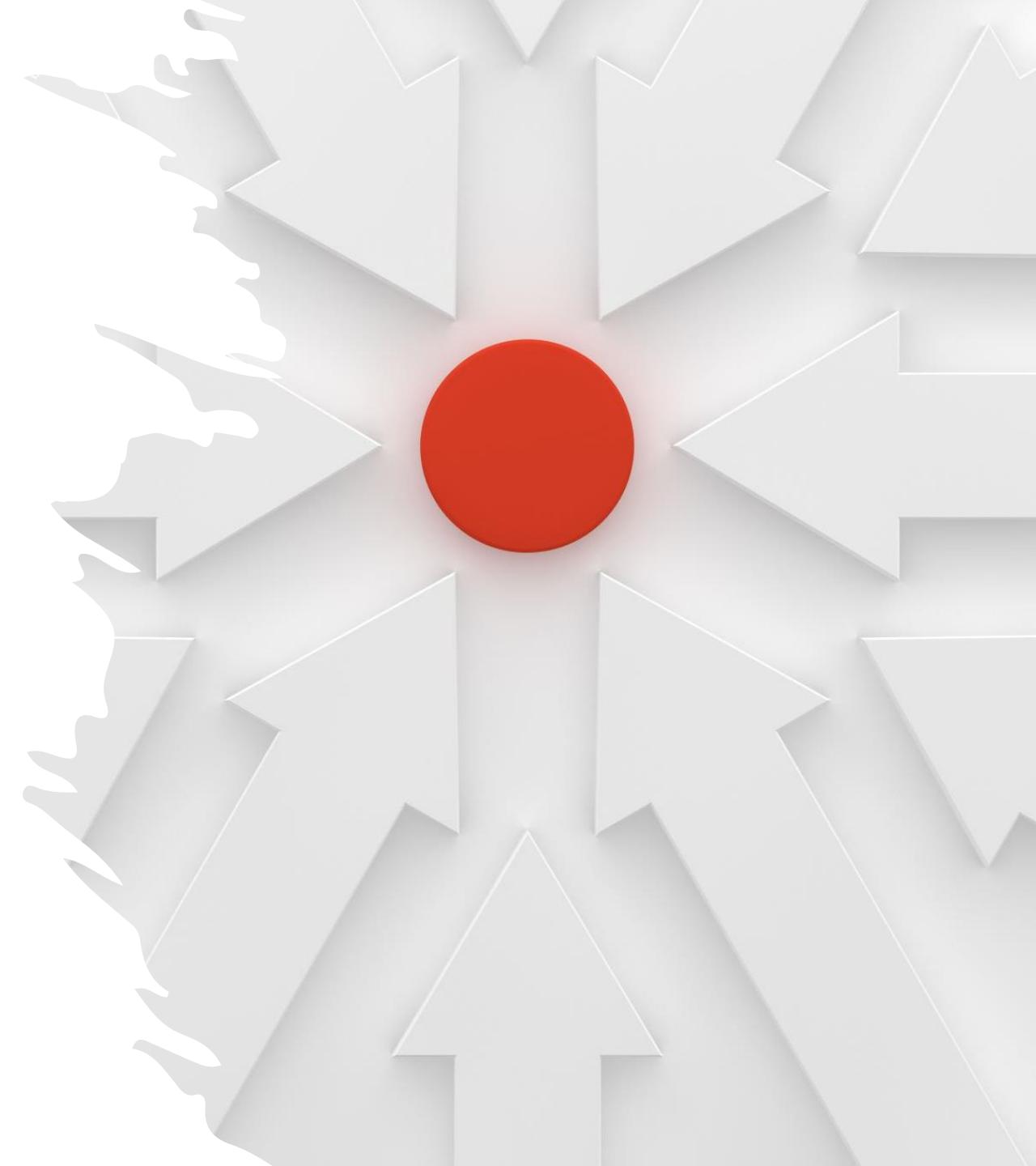
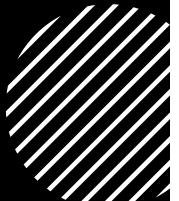


# Основные этапы и направления исследований в области систем искусственного интеллекта





# Основные направления исследований в области искусственного интеллекта



---

Разработка интеллектуальных информационных систем или систем, основанных на знаниях

---

Разработка естественно-языковых интерфейсов и машинный перевод

---

Генерация и распознавание речи

---

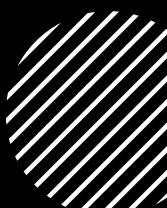
Обработка визуальной информации

---

Обучение и самообучение (системы добычи данных (Data-mining) и системы поиска закономерностей в компьютерных базах данных (Knowledge Discovery)



# Основные направления исследований в области искусственного интеллекта



---

Распознавание образов

---

Игры и машинное творчество

---

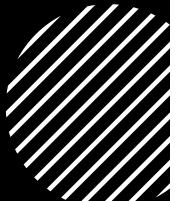
Программное обеспечение систем ИИ

---

Новые архитектуры компьютеров (компьютеры не фон-неймановской архитектуры, ориентированные на обработку символьной информации)

---

Интеллектуальные роботы



Инструментальные  
средства для  
разработки  
интеллектуальных  
систем включают:

специальные языки программирования,  
ориентированные на обработку символьной  
информации (LISP, SMALLTALK, РЕФАЛ),

языки логического программирования (PROLOG),  
языки представления знаний (OPS 5, KRL, FRL),

интегрированные программные среды,  
содержащие арсенал инструментальных средств  
для создания систем ИИ (KE, ARTS, GURU, G2),

оболочки экспертных систем (BUILD, EMYCIN, EXSYS Professional, ЭКСПЕРТ), которые позволяют  
создавать прикладные ЭС, не прибегая к  
программированию.

# Классификация интеллектуальных информационных систем

- Для ИИС характерны следующие признаки:
  - развитые коммуникативные способности;
  - умение решать сложные плохо формализуемые задачи;
  - способность к самообучению;
  - адаптивность.

**Системы с  
интеллектуальным  
интерфейсом**

- интеллектуальные базы данных
- естественные языковые интерфейсы
- гипертекстовые системы
- системы контекстной помощи
- когнитивная графика

**Экспертные  
системы**

- классифицирующие
- доопределяемые
- трансформирующие
- мультиагентные

**Самообучающиеся  
системы**

- индуктивные системы
- нейронные сети
- системы, основанные на precedентах
- информационные хранилища

**Адаптивные  
информационные  
системы**

- CASE-технологии
- компонентные технологии

# Этапы разработки интеллектуальной системы

## Этап 1: Выбор подходящей проблемы+

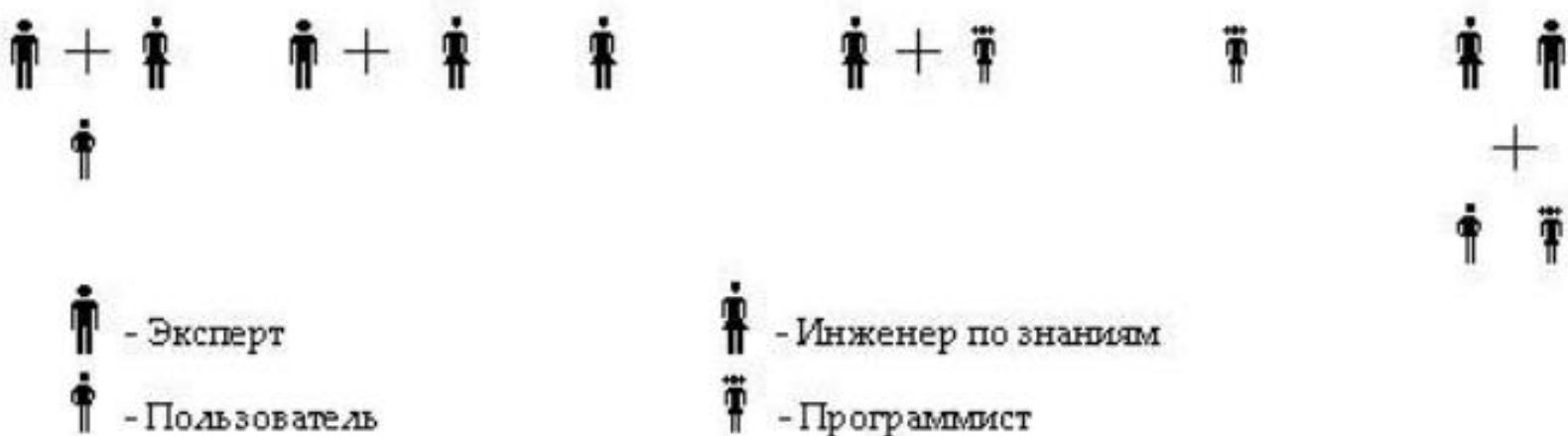
- определение проблемной области и задачи;
- нахождение эксперта, желающего сотрудничать при решении проблемы, и назначение коллектива разработчиков;
- определение предварительного подхода к решению проблемы;
- анализ расходов и прибыли от разработки;
- подготовку подробного плана разработки.

## Этап 2: Разработка прототипной системы

- **Прототипная система** является усеченной версией экспертной системы, спроектированной для проверки правильности кодирования фактов, связей и стратегий рассуждения эксперта.
- Объем прототипа - несколько десятков правил, фреймов или примеров.

## Стадии разработки прототипа ЭС

### Домашний этап разработки



## Этап 3: Развитие прототипа до промышленной ЭС

Демонстрационный прототип ЭС	Система решает часть задач, демонстрируя жизнеспособность подхода (несколько десятков правил или понятий)
Исследовательский прототип ЭС	Система решает большинство задач, но не устойчива в работе и не полностью проверена [несколько сотен правил или понятий)
Действующий прототип ЭС	Система надежно решает все задачи на реальных примерах, но для сложной задачи требует много времени и памяти
Промышленная система	Система обеспечивает высокое качество решений при минимизации требуемого времени и памяти: переписывается с использованием более эффективных средств представления знаний
Коммерческая система	Промышленная система, пригодная к продаже, т.е. хорошо документирована и снабжена сервисом

## Этап 4: Оценка системы

- критерии пользователей (понятность и "прозрачность" работы системы, удобство интерфейсов и др.);
- критерии приглашенных экспертов (оценка советов-решений, предлагаемых системой, сравнение ее с собственными решениями, оценка подсистемы объяснений и др.);
- критерии коллектива разработчиков (эффективность реализации, производительность, время отклика, дизайн, широта охвата предметной области, непротиворечивость БЗ, количество тупиковых ситуаций, когда система не может принять решение, анализ чувствительности программы к незначительным изменениям в представлении знаний, весовых коэффициентах, применяемых в механизмах логического вывода, данных и т.п.).

## Этап 5: Стыковка системы

- На этом этапе осуществляетсястыковка экспертной системы с другими программными средствами в среде, в которой она будет работать, и обучение людей, которых она будет обслуживать.

## Этап 6: Поддержка системы

# Модели представления знаний



# Построение концептуальной модели

- создается целостное и системное описание используемых знаний, отражающее сущность функционирования проблемной области

Результат концептуализации проблемной области обычно фиксируется в виде наглядных графических схем на объектном, функциональном и поведенческом уровнях моделирования:

- объектная модель описывает структуру предметной области как совокупности взаимосвязанных объектов;
- функциональная модель отражает действия и преобразования над объектами;
- поведенческая модель рассматривает взаимодействия объектов во временном аспекте.

Первые две модели описывают статические аспекты функционирования проблемной области, а третья модель - динамику изменения ее состояний.

# Объектная модель

- отражает знание о составе объектов, их свойств и связей
- Элементарной единицей структурного знания является **факт**, описывающий одно свойство или одну связь объекта, который представляется в виде триплета:
  - *предикат* (*Объект*, *Значение*).
- Если предикат определяет название свойства объекта, то в качестве значения выступает конкретное значение этого свойства, например:
  - *профессия* ("Иванов", "Инженер").
- Если предикат определяет название связи объекта, то значению соответствует объект, с которым связан первый объект, например:
  - *Работает* ("Иванов", "Механический цех").



В качестве важнейших типизированных видов отношений рассматриваются следующие отношения:

"род" - "вид" (обобщение);

"целое" - "часть" (агрегация);

"причина" - "следствие";

"цель" - "средство";

"функция" - "аргумент";

"ассоциация";

"хронология";

"пространственное положение" и др.

# Функциональная модель

описывает преобразования фактов, зависимости между ними, показывающие, как одни факты образуются из других. В качестве единицы функционального знания определим функциональную зависимость фактов в виде импликации:

$$A_1 \wedge A_2 \wedge A_3 \wedge \dots \wedge A_n \rightarrow B,$$

означающей, что факт В имеет место только в том случае, если имеет место конъюнкция фактов или их отрицаний A<sub>1</sub>, A<sub>2</sub>, ..., A<sub>n</sub>, например:

сбыт (Товар, "Слабый") и

прибыль (Товар, "Ничтожная") и

потребители (Товар, "Любители нового") и

число\_конкурентов (Товар,"Небольшое") ->

жизненный\_цикл (Товар, "Выведение на рынок").

# Поведенческая модель

- отражает изменение состояний объектов в результате возникновения некоторых событий, влекущих за собой выполнение определенных действий (процедур).
- **Состояние объекта** – это изменяющиеся во времени значения некоторого свойства.

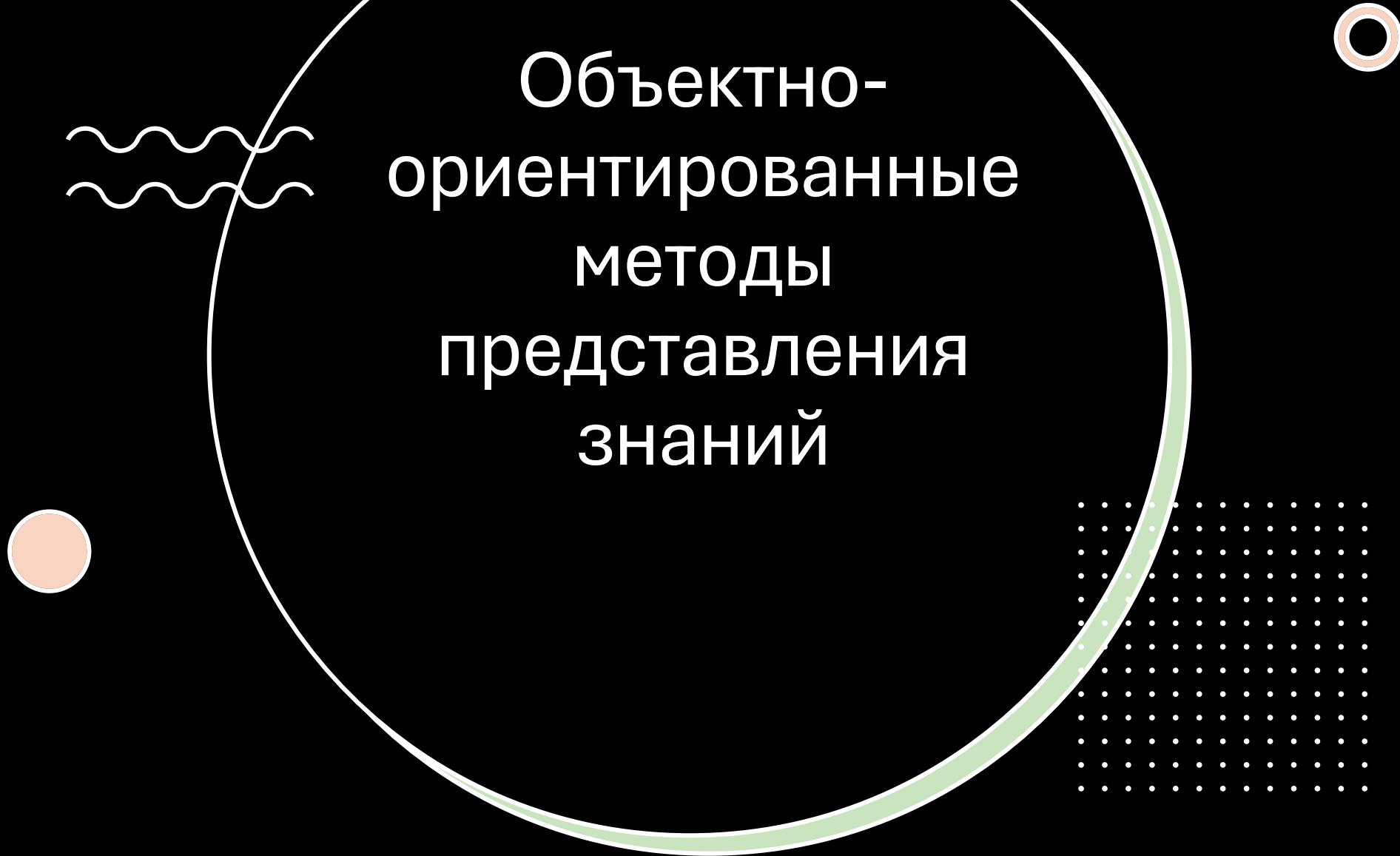
# Логическая модель представления знаний

- предполагает унифицированное описание объектов и действий в виде предикатов первого порядка.
- Под **предикатом** понимается логическая функция на N - аргументах (признаках), которая принимает истинное или ложное значение в зависимости от значений аргументов

# Продукционные модели представления знаний

- используются для решения более сложных задач, которые основаны на применении эвристических методов представления знаний, позволяющих настраивать механизм вывода на особенности проблемной области и учитывать неопределенность знаний.
- В продукционной модели основной единицей знаний служит правило в виде: - "**если <посылка>, то <заключение>**", с его помощью выражены:
  - пространственно-временные,
  - причинно-следственные,
  - функционально-поведенческие (ситуация - действие) отношения объектов.

Объектно-  
ориентированные  
методы  
представления  
знаний



**Данные** - это отдельные факты, характеризующие объекты, процессы и явления в предметной области, а также их свойства.

При обработке на ЭВМ данные трансформируются, условно проходя следующие этапы:

- данные как результат измерений и наблюдений;
- Данные на материальных носителях информации (таблицы, протоколы, справочники);
- модели (структуры) данных в виде диаграмм, графиков, функций;
- данные в компьютере на языке описания данных;
- базы данных на машинных носителях.



Знания связаны с данными, основываются на них, но представляют результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе выполнения какой-либо практической деятельности. Они получаются эмпирическим путем.

**Знания** - это выявленные закономерности предметной области (принципы, связи, законы), позволяющие решать задачи в этой области.

При обработке на ЭВМ знания трансформируются аналогично данным:

- знания в памяти человека как результат мышления;
- материальные носители знаний (учебники, методические пособия);
- **поле знаний** - условное описание основных объектов предметной области, их атрибутов и закономерностей, их связывающих;
- знания, описанные на языках представления знаний (продукционные языки, семантические сети, фреймы);
- **базы знаний.**

- **Представление знаний** - структурирование знаний с целью формализации процессов решения задач в определенной проблемной области.
- **Модель представления знаний** - формализм, предназначенный для отображения статических и динамических свойств предметной области.
- В искусственном интеллекте основными моделями представления знаний являются:
  - **продукционные системы,**
  - **семантические сети,**
  - **фреймы,**
  - **формальные логические модели.**

## Продукционные правила

В этой модели знания представляются в виде предложений типа:

***Если (условие), то (действие).***

Под ***условием*** понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под ***действием*** - действия, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее как условия, и терминальными или целевыми, завершающими работу системы).

Любое производственное правило, содержащееся в БЗ, состоит из двух частей: антецедента и консеквента.

**Антецедент** – это посылка правила (условная часть), состоит из элементарных предложений, соединенных логическими связками И, ИЛИ.

**Консеквент** (заключение) включает одно или несколько предложений, которые выражают либо некоторый факт, либо указание на определенное действие, подлежащее исполнению.

Производственные правила записываются в виде  
АНТЕЦЕНДЕНТ → КОНСЕКВЕНТ

# Пример производственных правил

## Правило 1:

**Если** топливо поступает в двигатель и двигатель вращается,  
**то** проблема в свечах зажигания.

## Правило 2:

**Если** двигатель не вращается и фары не горят,  
**то** проблема в аккумуляторе или проводке.

## Правило 3:

**Если** двигатель не вращается и фары горят,  
**то** проблема в стартере.

## Правило 4:

**Если** в баке есть топливо и топливо поступает в карбюратор,  
**то** топливо поступает в двигатель.

В производственных системах знания представляются в виде совокупности специальных информационных единиц, имеющих следующую структуру:

**Имя продукции:** «сфера применения»

**Предусловие Условие ядра**

**Если A, то B Постусловие.**

Имя сферы указывает ту предметную область, к которой относятся знания, зафиксированные в данной продукции.

Предусловия устанавливают на множестве правил из интересующей сферы некоторый порядок, приоритет их использования.

Условия определяют возможность применения того или иного правила.

Ядро продукции «Если A, то B» описывает преобразование, которое составляет суть производственного правила, где A и B могут иметь разные значения. Постусловие говорит о том, что надо делать, когда данное производственное правило сработало.

Например:

Интерпретация результатов тестирования

Использовать в первую очередь

Шкала «лжи» L<70 Т-баллов

Если (шкала ошибок F – шкала коррекции K) <-11, то  
вывести сообщение: «Результаты тестирования  
недостоверны»

Закончить интерпретацию результатов

## Примеры простых продукций

- «Если сверкнет молния, то гремит гром».
- «Если в доме вспыхнул пожар, то вызывайте по телефону 01 пожарную команду».
- «Если в путеводителе указано, что в городе есть театр, то надо пойти туда».

В общем случае производственная система включает следующие компоненты:

- *базу данных, содержащую множество фактов;*
- *базу правил, содержащую набор продукции;*
- *интерпретатор (механизм логического вывода) или правила работы с продукциями.*

Продукционные системы делят на два типа — с **прямыми и обратными выводами**.

- При прямом выводе рассуждение ведется от данных к гипотезам.
- При обратном производится поиск доказательства или опровержения некоторой гипотезы (от цели к данным).

Часто используются комбинации прямой и обратной цепи рассуждений.

Например, имеется фрагмент базы знаний из двух правил:

71: Если "отдых - летом" и "человек - активный", то  
"ехать в горы",

72: Если "любит солнце", то "отдых летом",

Пусть в систему поступили данные :

"человек активный" и "любит "солнце"

## Прямой вывод

Необходимо, исходя из данных, получить ответ.

### 1-й проход.

- Шаг 1. Правило П1, не работает (не хватает данных "**отдых - летом**").
- Шаг 2. Правило П2, работает, в базу поступает факт "**отдых - летом**".

### 2-й проход.

- Шаг 3. Правило 71, срабатывает, активируется цель "**ехать в горы**", которая и выступает как совет, который дает ЭС.

## Обратный вывод

Необходимо подтвердить выбранную цель при помощи имеющихся правил и данных.

### 1 й проход.

- Шаг 1. Цель - "**ехать в горы**":

пробуем П1 - данных, "**отдых - летом**" нет, они становятся новой целью, и ищется правило, где она в правой части.

- Шаг 2. Цель "**отдых - летом**":

правило П2 подтверждает цель и активирует ее.

### 2-й проход.

- Шаг 3. Пробуем П1, подтверждается искомая цель.

Продукции выгодны для выражения знаний, которые могут принимать форму перехода между состояниями:

- ситуация → действие;
- посылка → заключение;
- причина → следствие.

# Преимущества продукции

- модульность;
- наглядность;
- единообразие структуры (основные компоненты производственной системы могут применяться для построения интеллектуальных систем с различной проблемной ориентацией);
- естественность (вывод заключения в производственной системе во многом аналогичен процессу рассуждений эксперта);
- легкость внесения дополнений и простота механизма логического вывода;
- гибкость родовидовой иерархии понятий, которая поддерживается только как связи между правилами (изменение правила влечет за собой изменение в иерархии).

## Недостатки продукции

- отличие от структур знаний, свойственных человеку;
- этот процесс трудно поддается управлению;
- сложно представить родовидовую иерархию понятий.
- неясность взаимных отношений правил;
- сложность оценки целостного образа знаний;
- низкая эффективность обработки знаний.

Продукционная модель чаще всего применяется в промышленных экспертных системах. Она привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода.

Имеется большое число программных средств, реализующих продукционный подход (язык OPS 5; "оболочки" или "пустые" ЭС - EXSYS, ЭКСПЕРТ; инструментальные системы СПЭИС и др.), также промышленных ЭС на его основе (ФИАКР) и др.

# Семантическая сеть

Термин *семантическая* означает смысловая, а сама семантика - это наука, устанавливающая отношения между символами и объектами, которые они обозначают, т.е. наука, определяющая смысл знаков.

Более наглядными являются языки, опирающиеся на сетевую модель представления знаний.

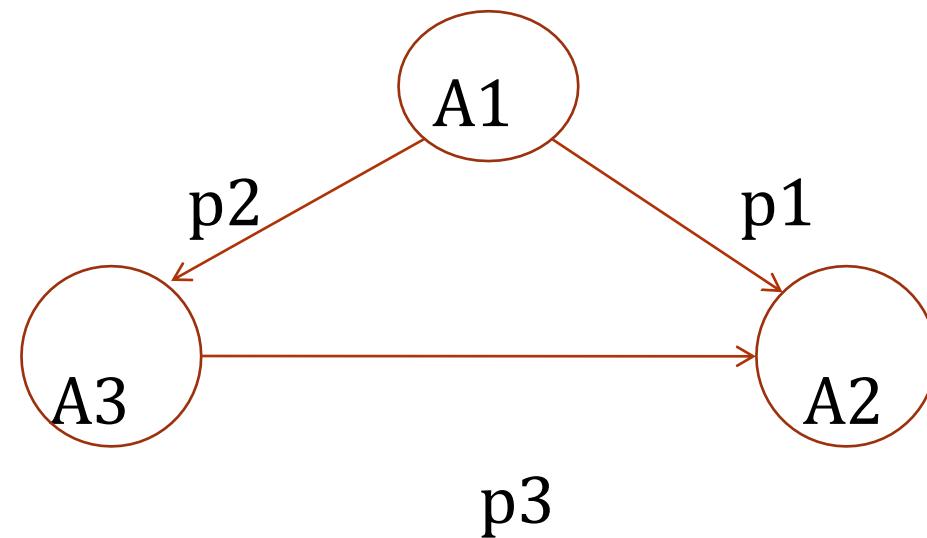
В основе такой модели лежит идея о том, что любые знания можно представить в виде совокупности объектов (понятий) и связей (отношений) между ними.

*Семантическая сеть* описывает знания в виде ориентированного графа.

В качестве *вершин* сети выступают понятия, факты, объекты, события и т. п., а в качестве *дуг* сети — отношения, которыми вершины связаны между собой.

Семантическая сеть является представлением структуры памяти человека.

Например, «*программист сел за компьютер и отладил программу*». Объектами являются: программист (A1), компьютер (A2), программа (A3). Объекты связаны отношениями: сел за компьютер (p1), отладил (p2), загружена в компьютер программа (p3).



Основная идея моделирования при помощи семантических моделей заключается в том, что модель представляет данные о реальных объектах и связях между ними прямым способом, что облегчает доступ к знаниям.

Начиная движение от некоторого понятия по дугам отношений, можно достичь других понятий.

Семантическая сеть как модель наиболее часто используется для представления декларативных знаний. С помощью этой модели реализуются такие свойства системы знаний, как интерпретируемость и связность, в том числе по отношениям IS-A и PART-OF.

За счет этих свойств семантическая сеть позволяет снизить объем хранимых данных, обеспечивает вывод умозаключений по ассоциативным связям.

Одной из первых известных моделей, основанных на семантической сети, является **TLC-модель** (Teachable Language Comprehender – доступный механизм понимания языка), разработанная Куиллианом в 1968 году.

Модель использовалась для представления семантических отношений между концептами (словами) с целью описания структуры долговременной памяти человека в психологии.

**Понятиями** обычно выступают абстрактные или конкретные объекты, а **отношения** - это связи типа:

- "это" ("is"),
- "имеет частью" ("has part"),
- "принадлежит",
- "любит".

Характерной особенностью семантических сетей является обязательное наличие трех типов отношений:

- класс - элемент класса;
- свойство - значение;
- пример элемента класса.

Самыми распространенными являются следующие типы отношений:

- **быть** элементом класса, то есть объект входит в состав данного класса (ВАЗ 2106 является автомобилем);
- **иметь** свойства, то есть задаются свойства объектов (жираф имеет длинную шею);
- **иметь** значение, то есть задается значение свойств объектов (человек может иметь двух братьев);
- **является** следствием, то есть отражается причинно-следственная связь (астеническое состояние является следствием перенесенного простудного заболевания).

Более детально можно выделить следующие отношения:

- связи типа "часть-целое" ("класс-подкласс", "элемент-множество" и т.п.);
- функциональные связи (определяемые обычно глаголами "производит", "влияет"...);
- количественные (больше, меньше, равно...);
- пространственные (далеко от, близко от, за, под, над...);
- временные (раньше, позже, в течение...);
- атрибутивные связи (*иметь свойство*, *иметь значение...*);
- логические связи (и, или, не);
- отношения «близости»;
- отношения «сходства-различия»;
- отношения «причина-следствие» и др.

Различают **экстенсиональные** и **интенсиональные** семантические сети.

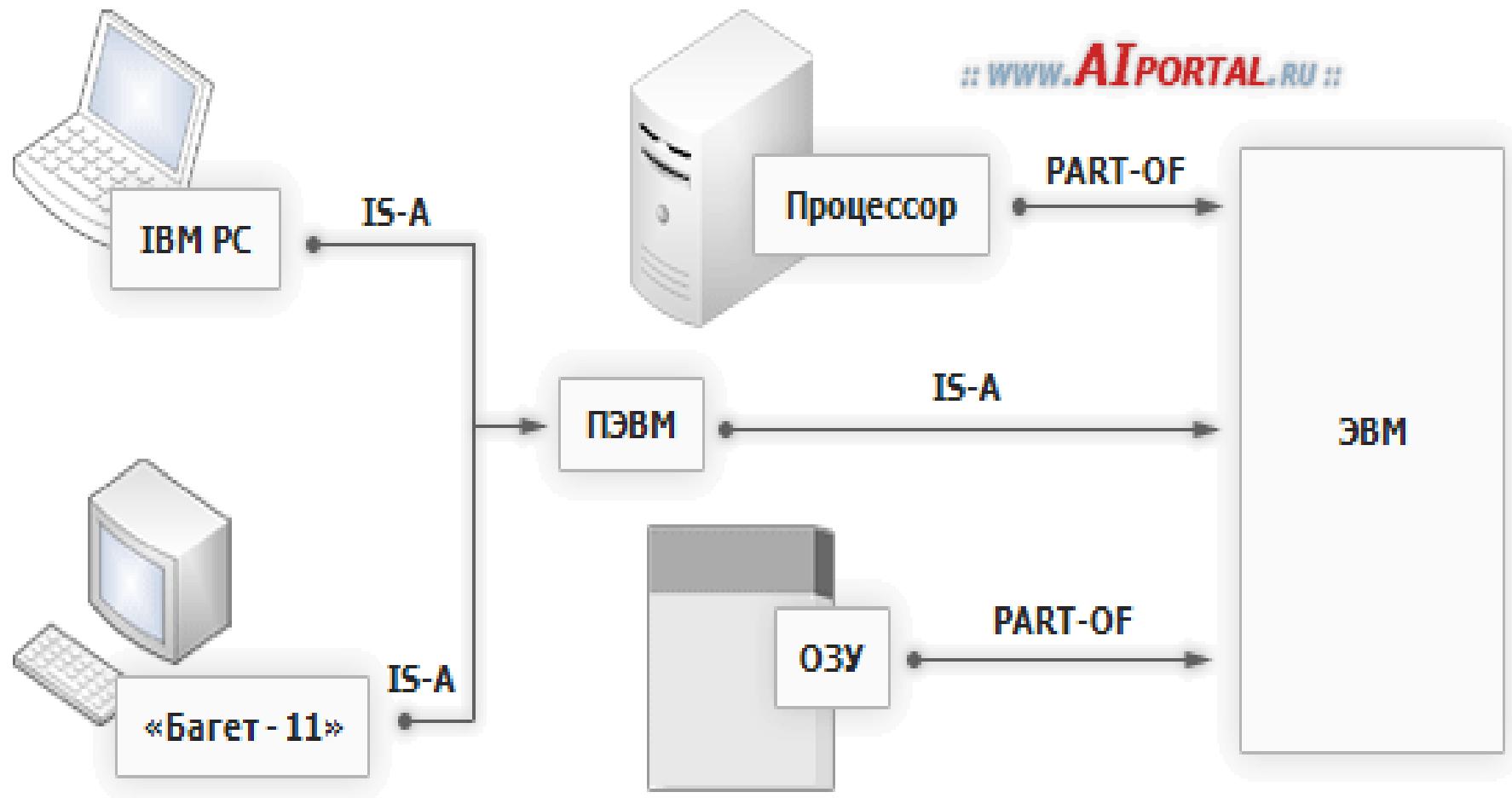
**Экстенсиональная** семантическая сеть описывает конкретные отношения данной ситуации.

**Интенсиональная** – имена классов объектов, а не индивидуальные имена объектов. Связи в интенсиональной сети отражают те отношения, которые всегда присущи объектам данного класса.

Пример семантической сети. В качестве вершин понятия: Человек, Иванов, Волга. Автомобиль, Вид транспорта. Двигатель.



# Фрагмент описания вычислительной техники

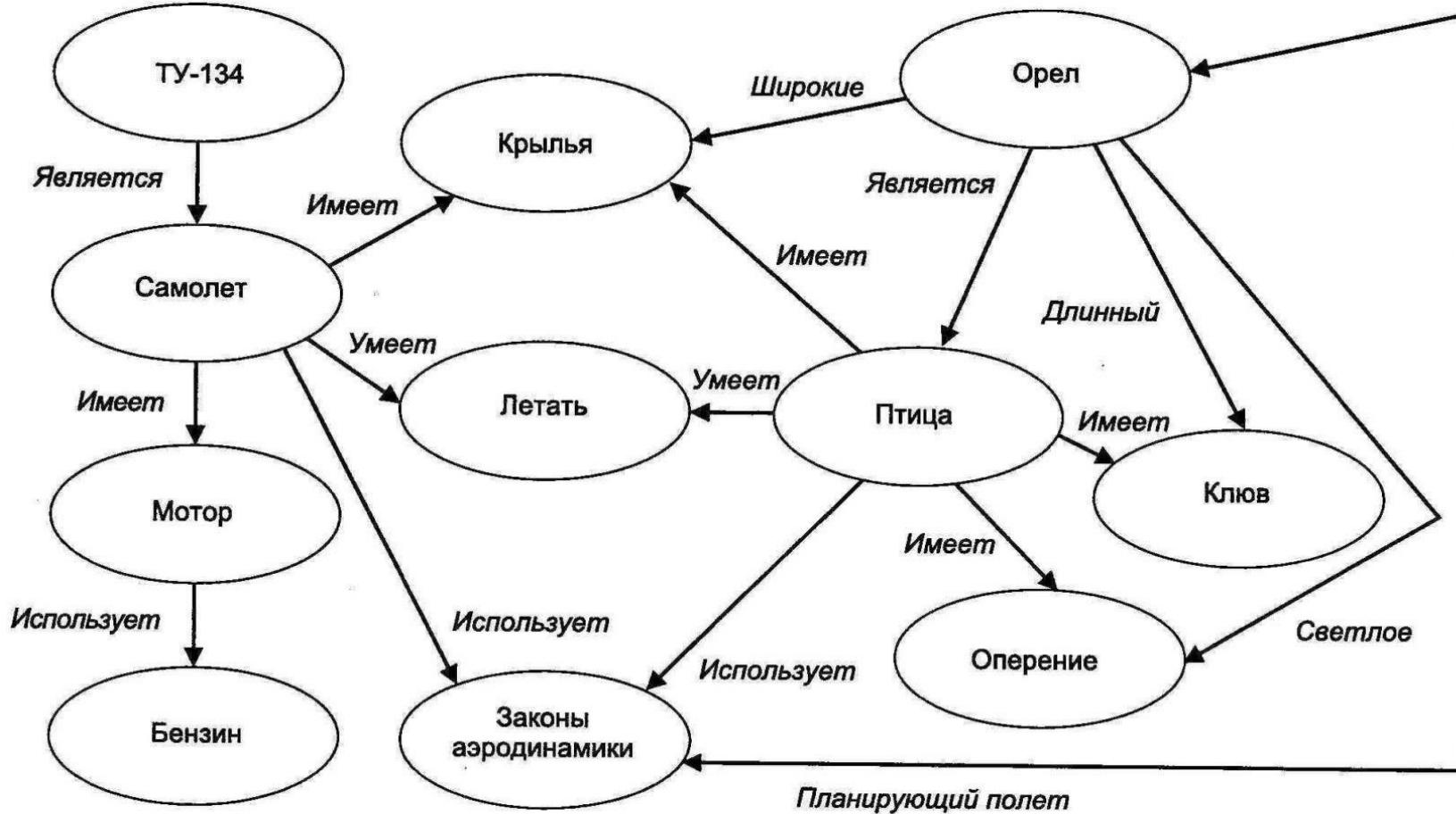


Преимущество этой модели - в соответствии с современным представлениям об организации долговременной памяти человека.

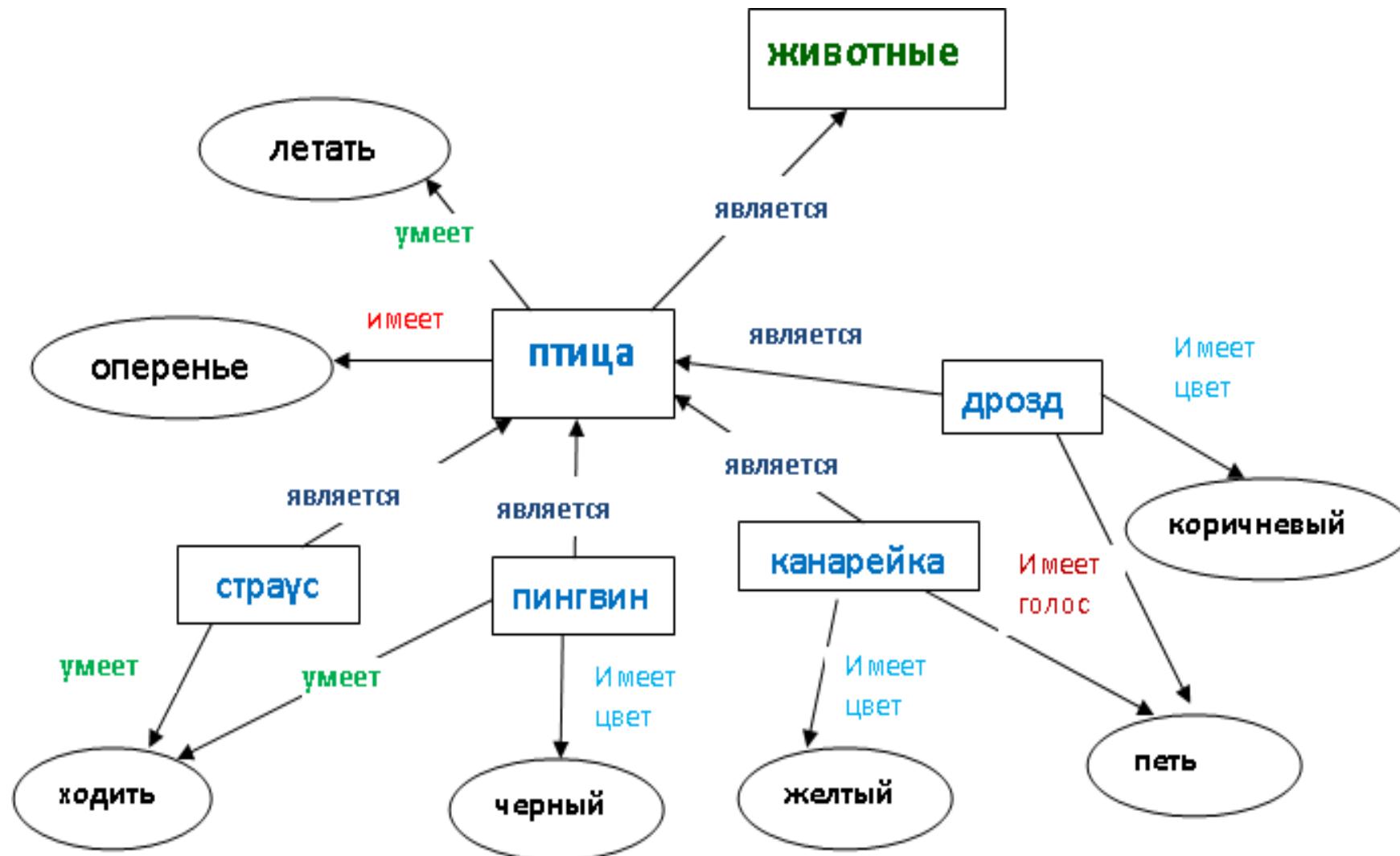
Недостаток модели - сложность поиска вывода на семантической сети.

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, который соответствует некоторой подсети, соответствующей поставленному вопросу.

# Семантическая сеть, показывающая взаимоотношения птиц и самолета



# Фрагмент семантической сети, описывающей птиц



Для реализации семантических сетей существуют специальные сетевые языки, например NET и др. Широко известны экспертные системы, использующие семантические сети в качестве языка представления знаний - PROSPECTOR, CASNBT, TORUS.

# Фреймовая модель

Фреймовая модель основана на концепции Марвина Мински (Marvin Minsky) – профессора Массачусетского технологического института, основателя лаборатории искусственного интеллекта, автора ряда фундаментальных работ. Фреймовая модель представляет собой систематизированную психологическую модель памяти человека и его сознания.

**Фреймом** (англ. frame – рамка, каркас) называется структура данных для представления некоторого концептуального объекта.

Информация, относящаяся к фрейму, содержится в составляющих его слотах.

**Слот** (англ. slot – щель, прорезь) может быть терминальным (листом иерархии) или представлять собой фрейм нижнего уровня.

Фрейм имеет имя, служащее для идентификации описываемого им понятия, и содержит ряд описаний – слотов, с помощью которых определяются основные структурные элементы этого понятия. Слот может содержать не только конкретное значение, но и имя процедуры, вычисляющей это значение по заданному алгоритму.

Фреймовые системы подразделяются на *статические* и *динамические*. Динамические допускают изменение фреймов в процессе решения задачи.

**Каждый фрейм, состоит из произвольного числа слотов**



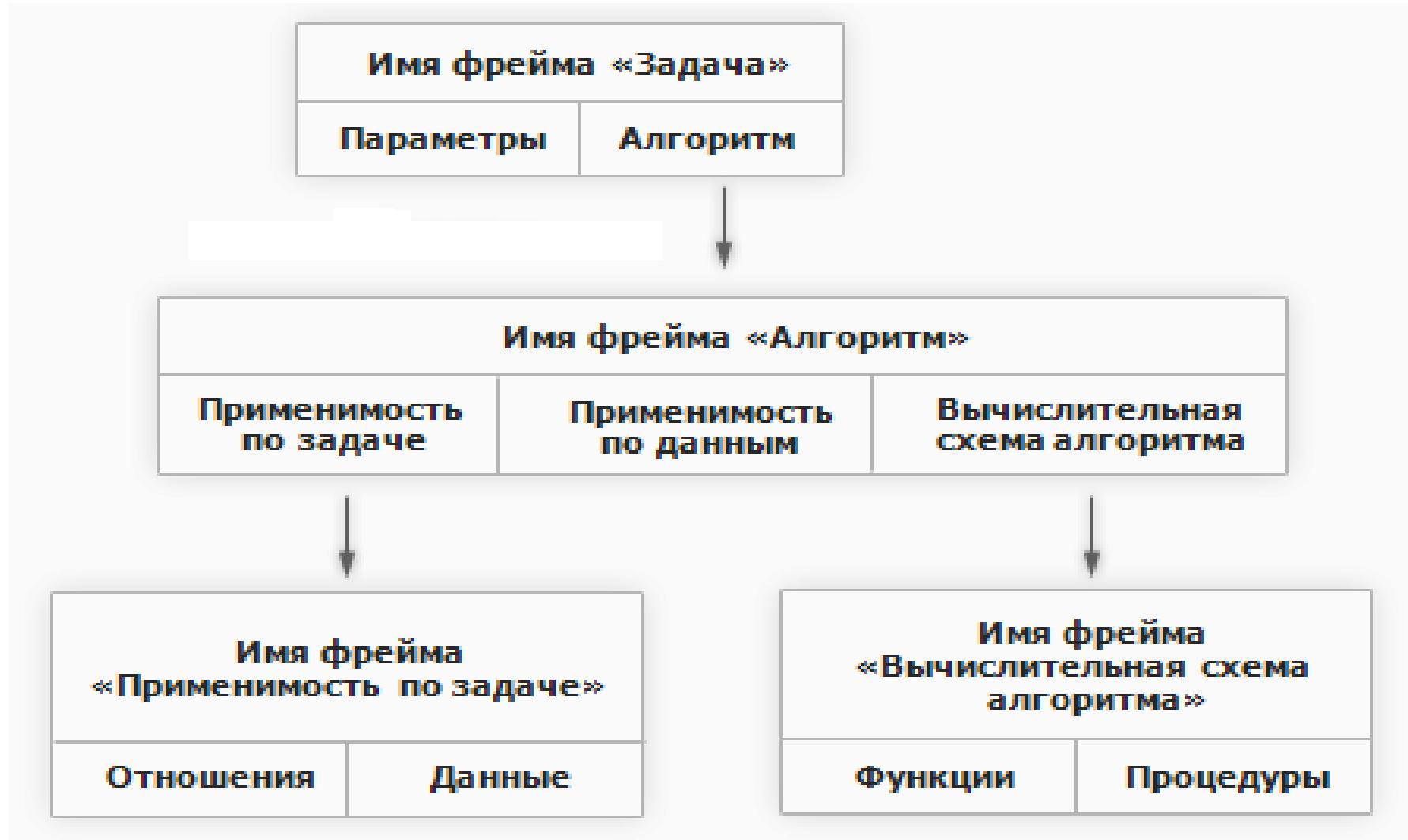
- **имя фрейма** – это идентификатор, присваиваемый фрейму. Фрейм должен иметь имя, единственное в данной фреймовой модели (уникальное имя);
- **имя слота** – это идентификатор, присваиваемый слоту. Слот должен иметь уникальное имя во фрейме, к которому он принадлежит. Обычно имя слота не несет никакой смысловой нагрузки и является лишь идентификатором данного слота, но в некоторых случаях оно может иметь специфический смысл;
- **указатель наследования** – только для фреймовых моделей иерархического типа; они показывают, какую информацию об атрибутах слотов во фрейме верхнего уровня наследуют слоты с такими же именами во фрейме нижнего уровня;
- **указатель атрибутов** – указатель типа данных слота. К таким типам относятся: FRAME (указатель), INTEGER (целое), REAL (вещественное), BOOL (булево), LISP (присоединенная процедура), TEXT (текст), LIST (список), TABLE (таблица), EXPRESSION (выражение) и другие;
- **значение слота** – значение, соответствующее типу данных слота и удовлетворяющее условиям наследования;
- **демон** – процедура, автоматически запускаемая при выполнении некоторого условия. Демоны запускаются при обращении к конкретному слоту фреймовой модели.

# Пример фрейма РУКОВОДИТЕЛЬ

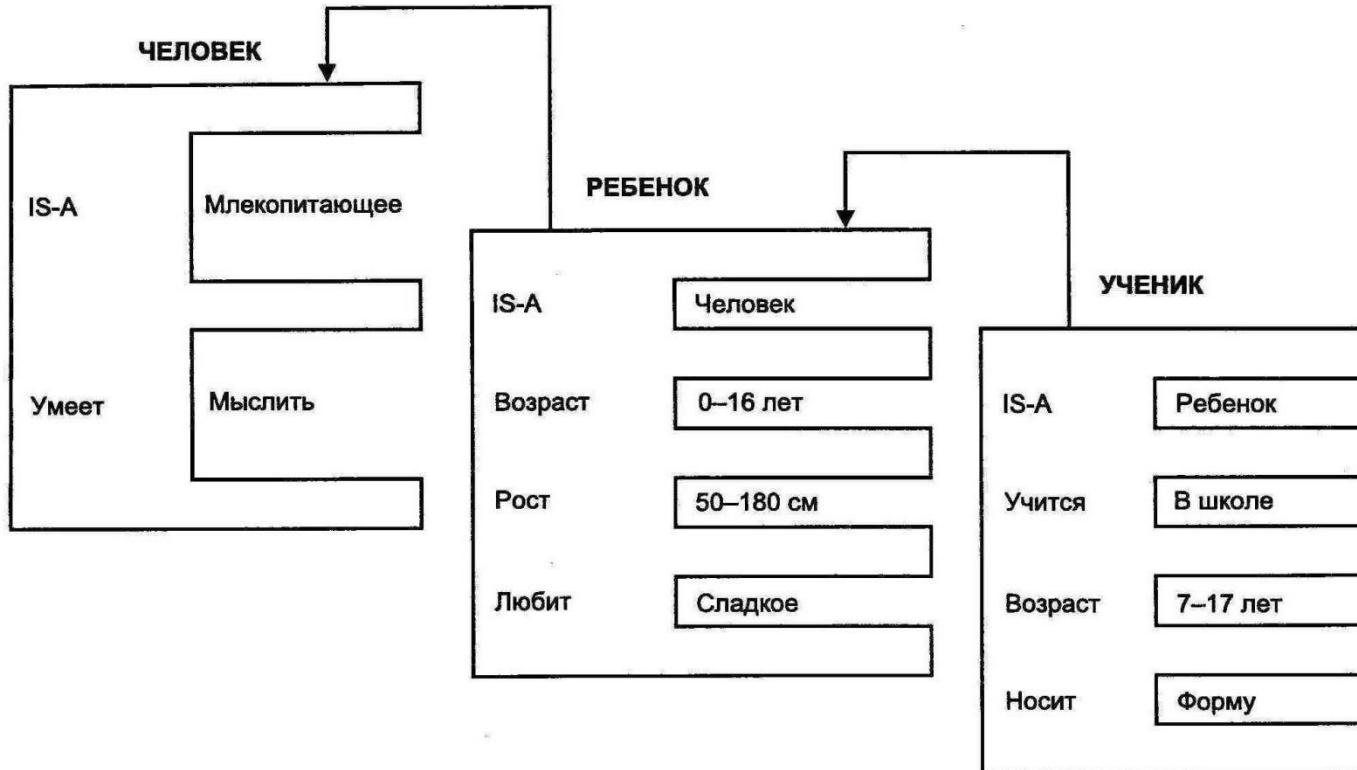
Имя слота	Значение слота	Тип значения слота
Имя	Иванов И.И.	Строка символов
Дата_рождения	01.01.1965	Дата
Возраст	Age(дата, дата_рождения)	Процедура
Специальность	Юрист	Строка символов
Отдел	Отдел кадров	Строка символов
Зарплата	50000	Число
Адрес	Дом_адрес	Фрейм

Фреймы образуют иерархию. Иерархия во фреймовых моделях порождает единую многоуровневую структуру, описывающую либо объект, если слоты описывают только свойства объекта, либо ситуацию или процесс, если отдельные слоты являются именами процедур, присоединенных к фрейму и вызываемых при его актуализации.

1а



# Пример иерархии фреймов



# Фрагмент фрейма, описывающий гостиничный номер



## Пример фреймов для базы знаний о птицах

name: bird

isa: animal

properties: flies  
feathers

default:

name: animal

isa: animate

properties: eats  
skin

default:

name: canary

isa: bird

properties: color(yellow)  
sound(sing)

default: size(small)

name: tweety

isa: canary

properties:

default: color(white)

Фреймы подразделяются на:

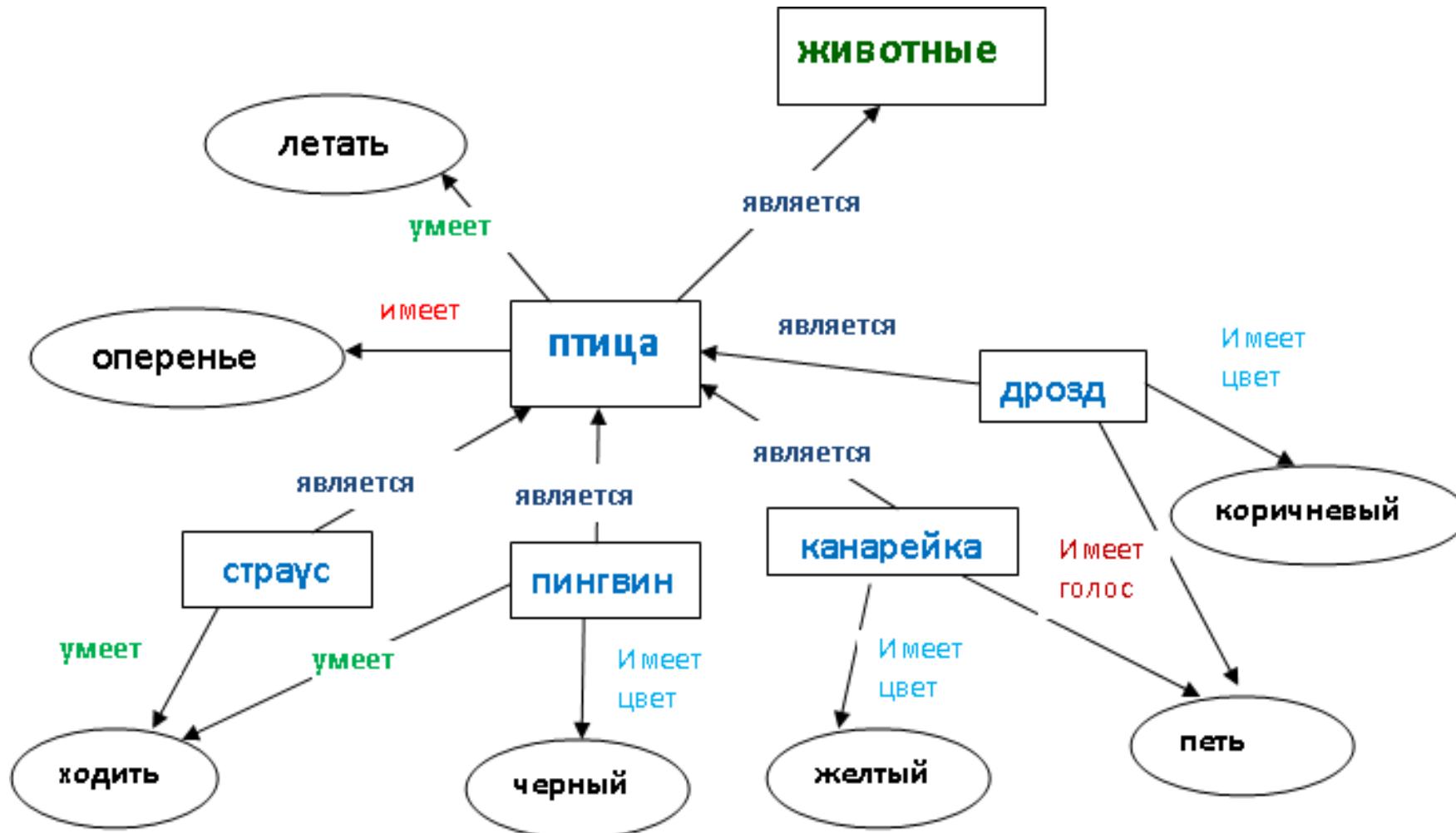
- **фрейм-экземпляр** – конкретная реализация фрейма, описывающая текущее состояние в предметной области;
- **фрейм-образец** – шаблон для описания объектов или допустимых ситуаций предметной области;
- **фрейм-класс** – фрейм верхнего уровня для представления совокупности фреймов образцов.

Состав фреймов и слотов в каждой конкретной фреймовой модели может быть разный, однако в рамках одной системы целесообразно единое представление для устранения лишнего усложнения.

- Разнотипные объекты или объекты, соответствующие концепции «множественности миров», заключающейся, к примеру, в том, что лошадь – животное бескрылое для одного (реального) мира и одновременно крылатое (Пегас в мифическом мире) для другого, могут описываться отличающимися друг от друга фреймами.
- В целом фреймовая модель допускает представление всех свойств декларативных и процедурных знаний. Глубина вложенности слотов во фрейме (число уровней) зависит от предметной области и языка, реализующего модель.

# Реализация моделей данных на языке ПРОЛОГ

# Фрагмент семантической сети, описывающей птиц



## Обозначения:

Объекты- птица, страус, пингвин, канарейка, дрозд.

- 1) Отношение «является» - связывает классы различных уровней иерархии.
- 2) Отношение «имеет свойство» - описывает свойство объектов:
  - имеет,
  - имеет цвет,
  - имеет голос,
  - умеет.

Опишем предикаты:

1) Отношение «является».

**является(объект, класс)** – указывает, что объект является подтипов класса.

Например:

является(канарейка, птица).

является(страус, птица).

является(пингвин, птица).

является(дрозд, птица).

Является(птица, животные).

2) Отношение «имеет свойство» - описывает свойство объектов: **имеет**, **имеет цвет**, **имеет голос**, **умеет**.

**имеет\_свойство(объект, свойство, значение)** –  
описывает свойство объекта,  
где объект и значение – узлы сети,  
свойство- имя объединяющих их связи.

Например:

имеет\_свойство(канарейка, цвет, желтый).  
имеет\_свойство(канарейка, голос, пение).  
имеет\_свойство(птица, покров, оперенье).

Можно рекурсивно определить, обладает ли некий объект семантической сети указанным свойством.

Например, свойство «летать» относится к классу «птица» и всем его подклассам. Исключения размещаются на отдельном уровне исключений.

Введем предикат и опишем правило:

**иметь\_свойства(Объект, Свойство, Значение):-**  
**имеет\_свойство(Объект, Свойство, Значение).**

**иметь\_свойства(Объект, Свойство, Значение):-**  
**является(Объект, Класс),**  
**иметь\_свойства(Класс, Свойство, Значение).**

## Задания для самостоятельного решения

- 1) Описать на языке Пролог фрагмент семантической сети о птицах, проверить работу программы.
  
- 2) Описать на языке Пролог семантическую сеть, описывающую самолет и птиц . Проверить работу программы.
  
- 3) Привести свой пример семантической сети (графическое представление, программная реализация)

# Фреймы

Семантические сети можно разбивать на части, добавляя к описаниям узлов дополнительную информацию, получая фреймовую структуру.

Представим пример с птицами в виде фрейма.

Каждый фрейм- это набор отношений семантической сети.

- 1 йй слот – содержит имя узла,
- 2 ой слот - определяет отношение наследования между узлом и его родителями,
- 3 ий слот- содержит список свойств, описывающих этот узел,
- 4 ый слот может содержать сведения исключения (в виде списка).

# Пример фреймовой структуры

имя:	<b>птица</b>
является:	<b>животное</b>
свойства:	<b>летать,</b> <b>оперенье</b>
отличия:	

имя:	<b>канарейка</b>
является:	<b>птица</b>
свойства:	<b>цвет желтый,</b> <b>имеет голос</b>
отличия:	<b>маленький</b> <b>размер</b>

Введем предикат **фрейм**, который связывает имена слотов, списки свойств и принимаемые по умолчанию значения.

**фрейм(имя (птица),  
является(животное),  
[летать, оперенье],  
[] ).**

**фрейм(имя (канарейка),  
является(птица),  
[цвет\_желтый, имеет\_голос],  
[маленький\_размер] ).**

Для определения всех свойств объекта, опишем рекурсивную процедуру.

**определить(Свойство, Объект):-**

    фрейм(имя(Объект), \_\_, Список\_свойств, \_\_),  
    запомнить(Свойство, Список\_свойств).

**определить(Свойство, Объект):-**

    фрейм(имя(Объект), \_\_, \_\_, Список\_исключений),  
    запомнить(Свойство, Список\_исключений).

**определить(Свойство, Объект):-**

    фрейм(имя(Объект), является(Класс), \_\_, \_\_),  
    определить(Свойство, Класс).

## Задания для самостоятельного решения

- 1) Дополнить фреймовое описание о птицах. Описать на языке Пролог фреймовое представление о птицах, проверить работу программы.
  
- 2) Привести свой пример фрейма (графическое представление, программная реализация)

# Логическая модель

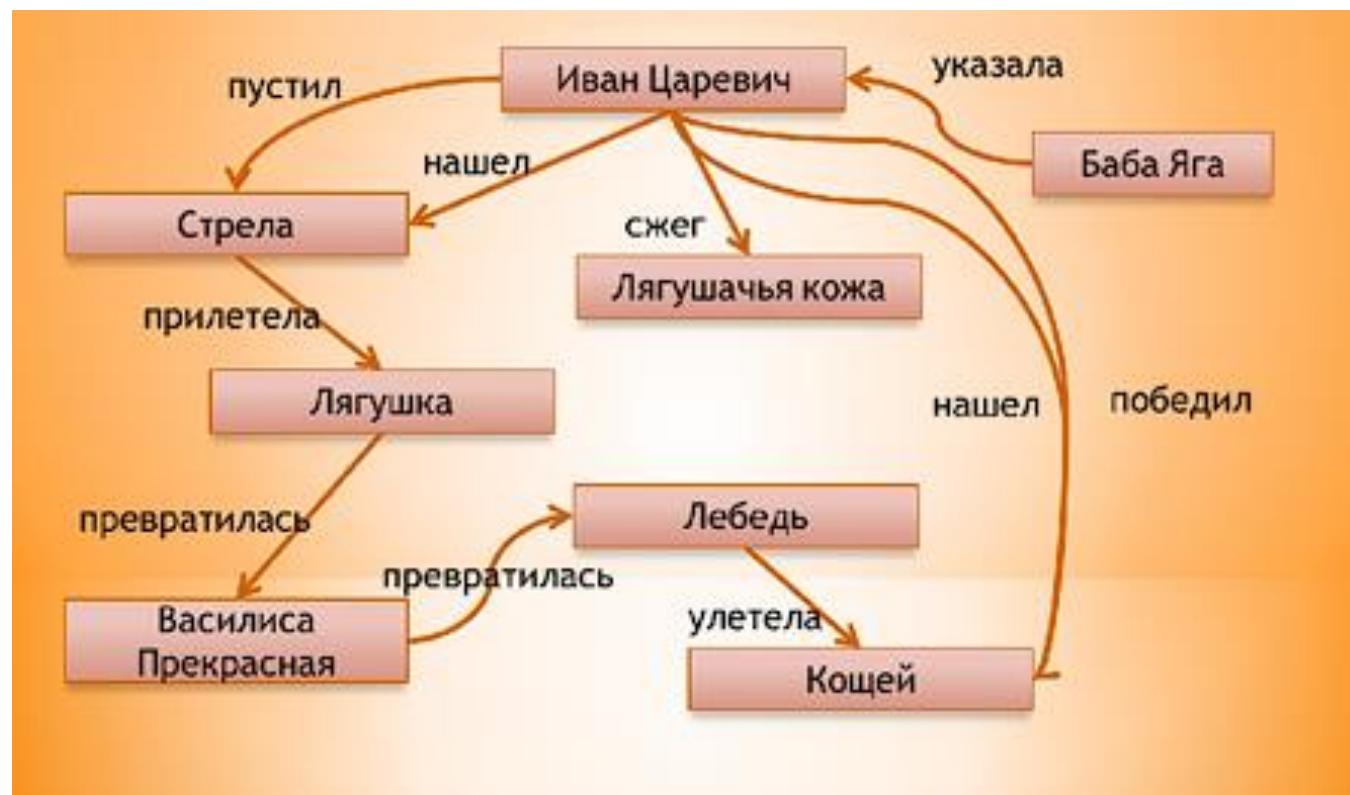
**Логическая модель** основана на системе исчисления предикатов первого порядка.

**Высказывание** – это предложение, смысл которого можно выразить значениями истина и ложь.

Исчисление высказываний позволяет формализовать лишь малую часть множества рассуждений, т.к. этот аппарат не позволяет учитывать внутреннюю структуру высказывания, которая существует в естественных языках.



**Семантические сети** представляют собой специализированный вид графовой структуры данных, в которой информация о мире организована с учетом ее семантики, то есть смысла и значения. В семантических сетях знания описываются с использованием узлов (nodes) и дуг (edges), где узлы представляют концепты, объекты или сущности, а дуги обозначают отношения между ними.



# Семантические сети - представление знаний

- Элементарной единицей знаний в семантической сети служит *триплет*, в котором имя предиката представляет помеченную дугу между двумя узлами графа, соответствующими двум связанным объектам.

# Важнейшими типизированными отношениями объектов являются:

- "Род" - "Вид",
- "Целое" - "Часть",
- "Причина" - "Следствие",
- "Средство" - "Цель",
- "Аргумент" - "Функция",
- "Ситуация" - "Действие".

**Узлы (Nodes или вершины)** : Узлы в семантических сетях представляют собой сущности или концепты, которые нужно описать или классифицировать. Например, узел может представлять собой концепт "Человек", "Автомобиль", "Город" и так далее.

**Дуги (Edges или ребра):** Дуги в семантических сетях обозначают отношения между узлами и определяют, какие связи существуют между концептами. Примеры отношений включают "Является частью", "Принадлежит", "Имеет атрибут" и многие другие.

**Семантика:** Основное отличие семантических сетей от других структур данных заключается в том, что они учитывают смысл и значение отношений между узлами. Это позволяет компьютерам и системам более глубоко понимать и обрабатывать информацию, а также делать выводы на основе контекста.

**Таксономии и онтологии:** Важным аспектом семантических сетей является использование таксономий и онтологий для организации знаний. Таксономии представляют собой иерархические структуры, позволяющие классифицировать концепты, в то время как онтологии определяют формальные определения и отношения между ними.



Рисунок – Триплет RDF

# Значение семантических сетей в интеллектуальных информационных системах

**Более глубокое понимание информации:** Семантические сети позволяют информационным системам понимать смысл и значение данных, а не просто обрабатывать символы или слова. Это способствует более точному анализу и интерпретации текстовых, графических и аудио-данных.

**Улучшенный поиск информации:** В ИИС семантический поиск позволяет пользователям искать информацию на основе ее смысла, а не просто наличия ключевых слов. Это уменьшает шум в результатах поиска и улучшает точность результатов.

**Интеграция данных:** Семантические сети позволяют объединять данные из разных источников и их разнородных форматов. Это особенно полезно в больших организациях, где данные могут храниться в различных системах и базах данных.

**Автоматический вывод и рекомендации:** Семантические сети позволяют информационным системам делать автоматические выводы на основе семантических отношений. Например, они могут автоматически выдвигать рекомендации для пользователей, основанные на их предпочтениях и контексте.

**Семантический анализ текста и данных:** Использование семантических сетей позволяет проводить более глубокий анализ текстов и данных, выявляя скрытые связи, контекст и смысловую нагрузку. Это полезно в областях, таких как анализ тональности, распознавание именованных сущностей и многих других.

**Интеллектуальные агенты и автономные системы:** Семантические сети играют важную роль в разработке интеллектуальных агентов и автономных систем, способных анализировать и принимать решения на основе семантических данных.

**Семантический веб и интероперабельность:** В рамках семантического веба семантические сети помогают обеспечить интероперабельность между различными системами и источниками данных в Интернете, что способствует обмену и интеграции информации.

**Исследования и инновации:** Семантические сети предоставляют основу для исследований в области искусственного интеллекта и обработки естественного языка. Они также поддерживают развитие новых технологий и приложений, таких как умный поиск, автономные автомобили и многие другие.

## Таблица 1 - Основные виды отношений в семантических сетях.

Тип	Описание
являться наследником (a-kind-of)	задает иерархические связи между классами
являться экземпляром	определяет значение, описывает конкретный объект, понятие
(is-a, например)	
это (are, есть)	может использоваться вместо связи a-kind-of в отношениях подразумевающих равенство или эквивалентность
являться частью (has-part)	определяет структурные связи, описывает части или целые объекты.
Функциональные	определяются обычно глаголами, отражают различные отношения (учить, владеть и т.д.).
Количественные	отображают количественные соотношения между вершинами (больше, меньше и т.д.)
Пространственные	отображают пространственные отношения между вершинами (близко, далеко и т.д.)
Временные	описывают временные связи между вершинами (скоро, долго, сейчас и т.д.)
Атрибутивные	описывают свойства объектов, понятий
Логические	описывают логические связи между вершинами (и, или, не)

Таблица 2 - Типы семантических сетей

Тип	Описание
<i>По типу знания</i>	
экстенсиональные	описывает конкретные отношения данной ситуации.
интенсиональные	описывают имена классов объектов, а не индивидуальные имена объектов, связи отражают те отношения, которые всегда присущи объектам данного класса.
<i>По типу ограничений на дуги и вершины</i>	
Простые	вершины сети не обладают внутренней структурой
иерархические	вершины обладают внутренней структурой, в иерархической сети есть возможность разделять сеть на подсети и устанавливать отношения не только между вершинами, но и между подсетями (различные подсети, существующие в сети, могут быть упорядочены в виде дерева подсетей, вершины которого—подсети, а дуги — отношения видимости)
динамические (сценарии)	сети с событиями
<i>По количеству типов отношений</i>	
Однородные	обладают только одним типом отношений
Неоднородные	количество типов отношений больше двух
<i>По арности отношений</i>	
Бинарные	все отношения в графе связывают ровно два понятия
N-арные	в сети есть отношения, связывающие более двух объектов

# Пример задачи

Построить сетевую модель представления знаний в предметной области «Ресторан» (посещение ресторана).

Для построения сетевой модели представления знаний необходимо выполнить следующие шаги:

- 1) Определить абстрактные объекты и понятия предметной области, необходимые для решения поставленной задачи. Оформить их в виде вершин.
- 2) Задать свойства для выделенных вершин, оформив их в виде вершин, связанных с исходными вершинами атрибутивными отношениями.
- 3) Задать связи между этими вершинами, используя функциональные, пространственные, количественные, логические, временные, атрибутивные отношения, а также отношения типа «являться наследником» и «являться частью».
- 4) Добавить конкретные объекты и понятия, описывающие решаемую задачу.

Оформить их в виде вершин, связанных с уже существующими отношениями типа «являться экземпляром», «есть».

- 5) Проверить правильность установленных отношений (вершины и само отношение при правильном построении образуют предложение, например «Двигатель является частью автомобиля»).

# Шаг 1

Ключевые понятия данной предметной области

- ресторан,
- тот, кто посещает ресторан (клиент)
- те, кто его обслуживают (повара, метрдотели, официанты, для простоты ограничимся только официантами).

У обслуживающего персонала и клиентов есть общие характеристики, поэтому целесообразно выделить общее абстрактное понятие – человек. Продукцией ресторана являются блюда, которые заказывают клиенты.

Исходя из этого, вершины графа будут следующими: «Ресторан», «Человек», «Официант», «Клиент», «Заказ» и «Блюдо».

## Шаг 2

У этих объектов есть определенные свойства и атрибуты.

Например, рестораны располагаются по определенным адресам, каждое блюдо из меню имеет свою цену.

Поэтому добавим вершины «Адрес» и «Цена».

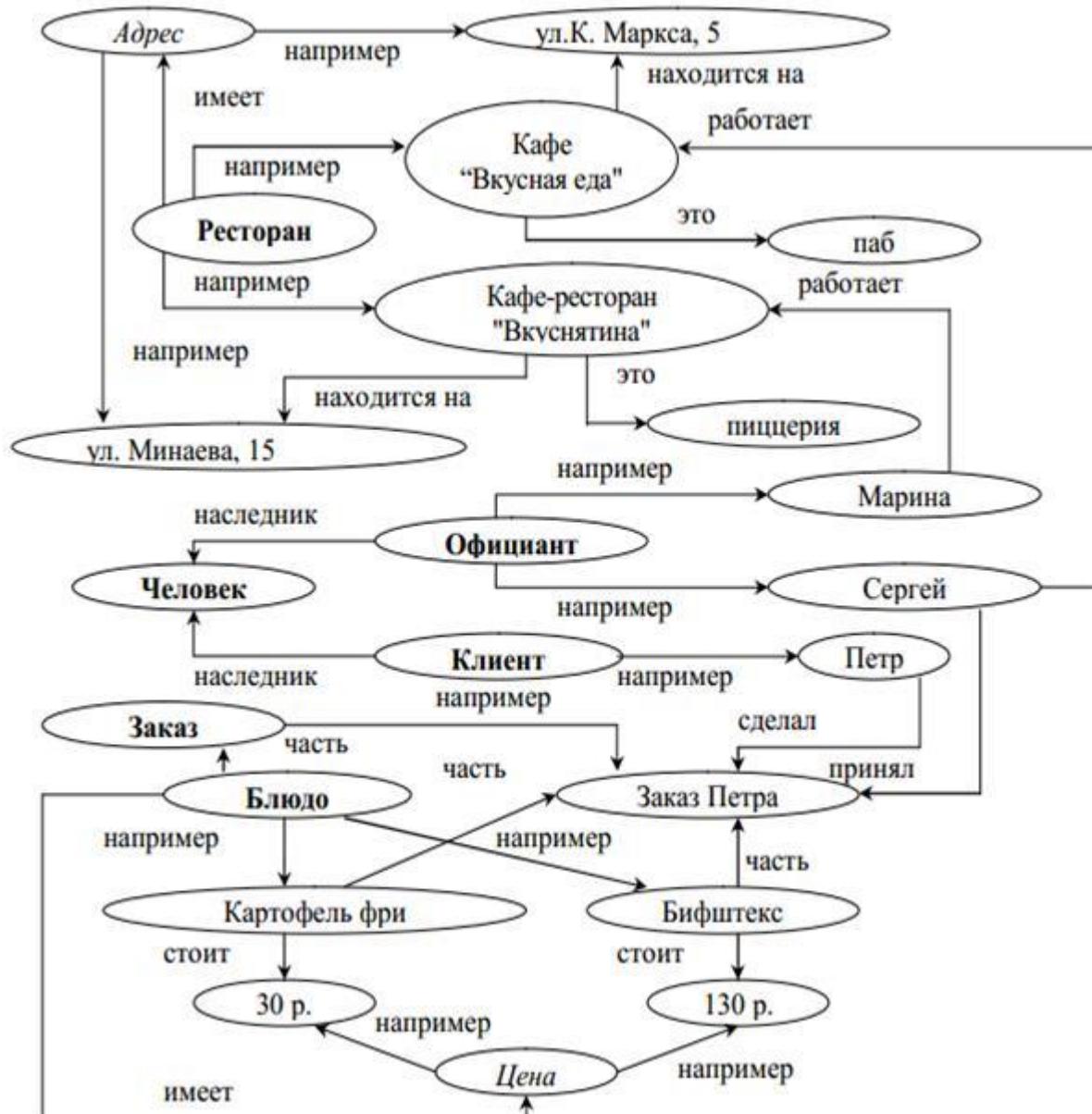
## Шаг 3

Определим для имеющихся вершин отношения и их типы, используя таблицу 1.

## Шаг 4

Добавим знание о конкретных фактах решаемой задачи.

Пусть имеется два ресторана: «Вкуснятина» и «Вкусная еда», в первом работает официантка Марина, а во втором официант Сергей. Пётр решил пойти в ресторан «Вкусная еда» и сделал заказ официанту на 2 блюда: картофель фри за 30 р., бифштекс за 130 р. Также известны адреса этих ресторанов и их специфика. Исходя из этого, добавим соответствующие вершины в граф и соединим их функциональными отношениями и отношениями типа «например или являться экземпляром».



## Шаг 5

Осуществим проверку установленных связей. Например, возьмем вершину «Блюдо» и пройдем по установленным связям. Получаем следующую информацию: блюдо является частью заказа, примерами блюд могут служить картофель фри и бифштекс.

Для получения ответа на какой-либо вопрос по этой задачи, необходимо найти соответствующий участок сети и, используя связи, получить результат. Например, вопрос «Какова цена заказа Петра (сколько Петр заплатил за заказ)?» Из запроса понятно, что необходимо найти следующие вершины: «Цена», «Петр» и «Заказ» или «Заказ Петра». Часть семантической сети, находящаяся между этими вершинами, содержит ответ, а именно, частью заказа Петра являются картофель фри и бифштекс, которые стоят 30 и 130 р. соответственно. Больше информации о заказе Петра в модели нет, поэтому делаем вывод – Петр заплатил 160 р.

