

# ЛЕКЦИЯ 2

---

Экспертные системы (ЭС). Состав и функции, классификация, назначение ЭС. Основные этапы создания и проектирование ЭС. Среды разработки ЭС.

# Интеллектуальные информационные системы

ИИС особенно эффективны в применении к слабо структурированным задачам, в которых пока отсутствует строгая формализация, где при принятии решений учитываются наряду с экономическими показателями слабо формализуемые факторы — экономические, политические, социальные.

Предназначение ИИС (в области экономики):

- Диагностика состояния предприятия.
- Помощь в антикризисном управлении.
- Выбор оптимальных решений по стратегии развития предприятия и его инвестиционной деятельности.
- Экономический анализ деятельности предприятия.
- Стратегическое планирования.
- Инвестиционный анализа, оценка рисков.
- Формирование портфеля ценных бумаг и т.п.

В основе этих видов деятельности лежит **проблема выбора решений**.

Благодаря наличию средств естественно-языкового интерфейса появляется возможность непосредственного применения ИИС бизнес-пользователем, не владеющим языками программирования, в качестве средств поддержки процессов анализа, оценки и принятия экономических решений.

# Отличительные особенности ИИС по сравнению с обычными ИС

- интерфейс с пользователем на естественном языке с использованием бизнес-понятий, характерных для предметной области пользователя;
- представление модели экономического объекта и его окружения в виде базы знаний и средств дедуктивных и правдоподобных выводов в сочетании с возможностью работы с неполной или неточной информацией;
- решения ИИС обладают "прозрачностью", т.е. могут быть объяснены пользователю на качественном уровне;
- экспертные системы способны пополнять свои знания в ходе взаимодействия с экспертом;
- способность автоматического обнаружения закономерностей бизнеса в ранее накопленных фактах и включения их в базу знаний(т.н. машинное обучение).
- ИИС дает пользователю «готовое» решение, которое по качеству и эффективности не уступает решению эксперта-человека;
- применение специфического компонента – базы знаний.



# База знаний

Служит для представления эвристической и фактологической информации, часто в форме фактов, утверждений и правил вывода.

В базе знаний ИИС интегрируются знания, поступающие от экспертов в конкретной предметной области, а также фундаментальные (энциклопедическими знаниями), составляющие суть общеизвестных научных теорий и моделей.

# База данных

хранит то, что называется форматированными данными, то есть конкретные факты и константы, касающиеся предметной области. Для хранения данных в ИИС используются либо реляционные, либо для интегрированного хранения данных и знаний — объектно-ориентированные базы данных.

# Система интерпретации знаний (Машина вывода)

Это механизм, играющий роль интерпретатора, применяющего знания подходящим образом, чтобы получить результат.

# Система приобретения знаний

Это механизм формализации знаний эксперта, пополнения базы знаний и обучения (самообучения) ИИС. Это реализуется с помощью инженера по знаниям и эксперта, а также машинного обучения.



# Система естественно-языкового интерфейса (СЕЯИ)

Механизм, преобразующий запросы пользователя и выдающий ему решение задачи (результат) на естественном языке. СЕЯИ реализует дружественный к пользователю интерфейс.

**Естественно-языковый интерфейс (СЕЯИ)** – это совокупность программных и аппаратных средств, обеспечивающих общение ИИС с пользователем на ограниченном рамками предметной области естественном языке.

В состав СЕЯИ входят словари, отражающие словарный состав и лексику языка, а также лингвистический процессор, осуществляющий анализ текстов (морфологический, синтаксический, семантический и прагматический) и синтез ответов пользователю.

В идеале СЕЯИ должно удовлетворять двум основным требованиям: обеспечивать вход и выход системы на естественном языке; обеспечивать обработку входных данных и генерацию выходных, основываясь на знании относительно синтаксических, семантических и прагматических аспектов естественного языка.

# Система объяснения решения.

- Механизм протоколирующий работу системы интерпретации знаний и выдающий его пользователю в виде цепочки логических выводов. Это позволяет «объяснить» пользователю найденное решение.

# Проблемы СЕЯИ

- **Проблемы понимания естественного языка**, будь то текст или речь, во многом зависят от знания предметной области. Понимание языка требует знаний о целях говорящего и о контексте. Необходимо также учитывать недосказанность или иносказательность. Например, даже в таком простом предложении «Ваня встретил Машу на поляне с цветами» нам не понятно, кто же был с цветами: Ваня, Маша или поляна? Крылатая фраза знаменитого русского лингвиста, академика Л. В.Щербы «Глокаякуздраштекобудланулабоккра и курдячитбокренка» говорит о том, что такая «непонятная» фраза построена по всем правилам русского языка, не вызывает проблем с грамматическим разбором такого предложения, но вызывает проблемы с пониманием.
- **Проблема «смысл-текст»** как и пример выше. В предложении «Какой завод заказал оборудование для конвертерного цеха в Бельгии?» неясен смысл: был ли сделан заказ в Бельгии или цех находится в Бельгии.
- **Проблема «планирования»** возникает при необходимости вести диалог, например, на тему «Куда Вы хотите лететь?». В этом случае нужно глубокое знание предметной области (номера рейсов, время прилета-отлета, цены и т.д.).
- **Проблема «равнозначности»**. Будут ли равнозначны два предложения «У дома стоит слон» и «У дома стоит существо с хоботом и бивнями»? На первый взгляд нет сомнений в равнозначности этих предложений. А если в базе знаний существо с хоботом и бивнями определено двумя значениями: слон и мамонт, то такие сомнения, наверное, появятся.
- **Проблемы «моделей»** участников общения. У участников общения должны быть сопоставимые модели представления знаний, необходимая глубина понимания, возможность логического вывода, возможность действия.
- **Проблема «эллиптических конструкций»**, то есть опущенных элементов диалога. Например, в пословице «Береги платье снову, а честь - смолоду» вторая часть предложения будет синтаксическим *эллипсисом* (опущен глагол береги).
- **Проблема «временных противоречий»**. Например, в предложении «Я хотел завтра пойти в кино» глагол «хотел» в прошедшей форме сочетается с обстоятельством будущего времени «завтра», что противоречит общепринятой логике.

# Задача создания естественно-языкового интерфейса

может быть рассмотрена как две взаимодействующие подзадачи.

**Первая подзадача** - это взаимодействие интерфейса с пользователем. Здесь непосредственно должен быть реализован алгоритм понимания смысла того, что сообщает пользователь. То есть должно быть реализовано множество выражений, которые пользователь может сказать, а система таким образом должна понять.

**Вторая подзадача** - интерфейс должен конвертировать смысл сказанной фразы в какое-либо внутреннее представление.

- Это может быть уже существующий язык, например, Пролог, VBScript и др.
- Это может быть какое-либо свое внутреннее представление. Представление во внутренних кодах нужно для понимания смысла запроса пользователя и трансляции этого смысла в запросы SQL. После формирования запроса во внутреннем представлении необходимо представить информацию на языке, понятном для базы данных. Таким образом, происходит трансляция в SQL. Возврат информации осуществляется в обратном порядке. База данных на запрос отвечает множеством записей из базы данных. Результат транслируется во внутреннее представление, затем во фразы естественного языка и выдается пользователю как ответ. Затем пользователь снова обращается и так далее.

# Направления реализации естественно-языкового диалогового интерфейса

Многие IT-компании занимаются созданием, так называемых, голосовых пользовательских интерфейсов (sounduserinterface) или SUI, в основе которых лежит решение проблемы понимания смысла пользовательской фразы.

Термин — понимание смысла — интерпретируется как процесс трансляции фразы пользователя в некоторое адекватное действие, совершаемое системой.

# Системы понимания естественной речи или NLU-систем

**1. Командные системы (commandcontrol).** Этот тип систем интерпретирует единичную фразу пользователя в набор конечных команд, которые выполняются компьютером. С помощью таких систем строится голосовое управление компьютерными приложениями, например, MicrosoftWord.

Эти системы полностью содержат в себе команды управления конкретным приложением и транслируют голосовые команды пользователя в команды управления приложением.

# Пример работы системы commandcontrol

ПОЛЬЗОВАТЕЛЬ: Открыть MicrosoftWord.

КОМПЬЮТЕР: Запускает приложение MicrosoftWord.

ПОЛЬЗОВАТЕЛЬ: Открыть File.

КОМПЬЮТЕР: Открывает диалоговое окно открытия файла.

ПОЛЬЗОВАТЕЛЬ: Изменить путь.

КОМПЬЮТЕР: Изображает дерево изменения директорий.

ПОЛЬЗОВАТЕЛЬ: Выйти.

КОМПЬЮТЕР: Закрывает приложение Microsoft Word.

Система commandcontrol понимает фразы естественного языка только лишь как набор команд. Поэтому в ее задачи не входит понимание смысла фразы пользователя. Система такого типа должна лишь сопоставить фразу пользователя с тем множеством фраз, которые заложены внутри нее, и при совпадении выполнить требуемое действие. Конечно, в этом случае можно говорить лишь об управлении с использованием ограниченного числа команд, представленных на естественном языке.

# Системы понимания естественной речи или NLU-систем

**2. Диалоговые системы (dialogsystem).** Эти системы более широко интерпретируют понятие естественного языкового общения и в своей основе содержат диалог пользователя с компьютером для выяснения конечной цели, которую хочет достичь пользователь и получения от него необходимых данных для решения поставленной задачи.

Пример:

ПОЛЬЗОВАТЕЛЬ: Я хотел бы поехать в Киев.

DS: Каким транспортом вы бы хотели воспользоваться?

ПОЛЬЗОВАТЕЛЬ: Конечно, лучше поездом.

DS: Каким транспортом вы бы хотели воспользоваться?

ПОЛЬЗОВАТЕЛЬ: Конечно, лучше поездом.

DS: Когда вы хотите быть в Киеве?

ПОЛЬЗОВАТЕЛЬ: 25 декабря. 23

DS: Перечисляет возможные рейсы и просит выбрать нужный

ПОЛЬЗОВАТЕЛЬ: Сообщает нужный рейс.

...

DS: После того как получена полная информация, бронирует билет.

Как видно из данного примера, что в течение диалога, DS может работать не с одним, но несколькими приложениями, как базой данных движения поездов, наличием мест, бронированием билетов и т.д.

DS используется абсолютно иной подход. Здесь система нацелена на понимание смысла сказанного. Это возможно благодаря использованию некоторой логической структуры, на которую отображается речь пользователя.



# Методы рассуждения в интеллектуальных информационных системах

При реализации рассуждений в ИИС различают два подхода: **логический** метод рассуждения и **эвристический** метод рассуждения.

**Логический метод** рассуждений основан на применении логики. Логика это наука о правильных способах рассуждений. В классическом варианте состоит из учения о понятиях, учения о суждениях и учения об умозаклучениях. Важно отметить, что логика есть наука о мышлении в понятиях, а не о познании мира посредством мышления о понятиях.

# Различают следующие виды логических рассуждений

1) **Дедукция** - рассуждение от сложного к простому, то есть получение частного правила на основе общего правила. Например, правила решений квадратных уравнений в общем случае выглядит так: Необходимо определить дискриминант и в зависимости от того, какое значение примет этот дискриминант ( $>0, =0, <0$ ) делается вывод о том, что квадратное уравнение имеет 2 корня, один корень или не имеет ни одного, а частные правила для решения квадратного уравнения можно считать по теореме Виета. Она применима только для уравнений, у которых коэффициент при  $x^2$  равен единице.

2) **Индукция** - рассуждение от простого к сложному, то есть когда на основе частных примеров синтезируются общие правила. Например, метод математической индукции из школьного курса математики гласит, что если формула верна при  $n=1, 2, \dots, k$ , то она будет верна и для  $n=k+1$ , то она верна для всех  $n$ .

3) **Аналогия** - рассуждение на основе прошлого опыта, т.е. скрытая (неявная) закономерность, которая присуща многим на первый взгляд внешне различающимся объектам. Например, было у меня три собаки, одна из них прожила 8 лет, другая - 10 лет и можно сделать вывод, что собака живет от 8 до 15 лет; собаки преданные друзья.

Не монотонность логического вывода означает, что вывод может никогда не закончиться и не дать ответа. Другими словами есть не разрешимые проблемы в принципе, либо разрешимы, для некоторого конкретного метода. Не разрешимые проблемы в принципе - это такие проблемы, которые нельзя решить ни одним из существующим методом, но также ни одним из методом, которые когда либо будет создан.

# Эвристический метод рассуждения основан

на применении правдоподобных рассуждений, которые называют эвристиками.

**Эвристика** - это алгоритм или таблица решений, которая базируется на опыте - причем качественном, прошлом и обширном, а не на научных данных или логическом выводе.

Эвристика отражает особенности того, как такие задачи решает человек, когда он не пользуется строго формальными приемами.

Если эти человеческие способы решения удастся запрограммировать, то такие программы называются эвристическими. Эвристика часто используются при программировании игр, имитации творческих процессов и т. п.

В экспертных системах при формализации профессиональных знаний человека, касающихся способов решения задач в той или иной проблемной области, широко используются те эвристики, которыми руководствуются профессионалы-эксперты.

# Инструментальные средства разработки ИИС

- Начать со стр. 29 из учебника **А.Н. Козлов**  
**ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ** – краткий обзор эволюции и развития средств разработки и примеры ИИС

# Данные, знания и представление знаний

**Данные** — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства. При обработке на ЭВМ данные трансформируются, условно проходя следующие этапы:

D1 — данные как результат измерений и наблюдений;

D2 — данные на материальных носителях информации (таблицы, протоколы, справочники);

D3 — модели (структуры) данных в виде диаграмм, графиков, функций;

D4 — данные в компьютере на языке описания данных;

D5 — базы данных на машинных носителях информации.

**Знания** — это закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области.

# Данные, знания и представление знаний

Знания основаны на данных, полученных эмпирическим путем. Они представляют собой результат мыслительной деятельности человека, направленной на обобщение его опыта, подученного в результате практической деятельности. С информационной точки зрения можно дать следующее определение знаний.

*Знания - совокупность сведений, образующих целостное описание, соответствующее некоторому уровню осведомленности об описываемом вопросе, предмете, проблеме и т.д.*

При обработке на ЭВМ знания трансформируются аналогично данным.

Z1 — знания в памяти человека как результат мышления;

Z2 — материальные носители знаний (учебники, методические пособия);

Z3 — *поле знаний* — условное описание основных объектов предметной области, их атрибутов и закономерностей, их связывающих;

Z4 — знания, описанные на языках представления знаний (продукционные языки, семантические сети, фреймы — см. далее);

Z5 — *база знаний на машинных носителях информации.*

Часто используется еще такое определение знаний.

**Знания**— это хорошо структурированные данные, или данные о данных, или мета

**Особенности знаний.** Знания состоят из понятий, которые имеют имя, структуру и набор признаков, связей между понятиями и утверждений о свойствах понятий.

# Типы и виды знаний

**Поверхностные** — знания о видимых взаимосвязях между отдельными событиями и фактами в предметной области.

Пример, поверхностные знания: «Если нажать на кнопку звонка, раздастся звук. Если болит голова, то следует принять аспирин».

# Типы и виды знаний

**Глубинные знания** – это абстракции, аналогии, схемы, отображающие структуру и природу процессов, протекающих в предметной области. Эти знания объясняют явления и могут использоваться для прогнозирования поведения объектов.

Пример глубинных знаний:

*«Принципиальная электрическая схема звонка и проводки.  
Знания физиологов и врачей высокой квалификации о причинах, видах головных болей и методах их лечения».*

Современные экспертные системы работают в основном с поверхностными знаниями. Это связано с тем, что на данный момент нет универсальных методик, позволяющих выявлять глубинные структуры знаний и работать с ними.



# Типы и виды знаний

**Процедурные знания** описывают последовательности действий, которые могут использоваться при решении задач. Это, например, программы для ЭВМ, словесные записи *алгоритмов*, инструкция по сборке некоторого изделия.

**Декларативные знания** — это все знания, не являющиеся процедурными, например, статьи в толковых словарях и энциклопедиях, формулировки законов в физике, химии и других науках, собрание историчес

В отличие от процедурных знаний, отвечающих на вопрос: "Как сделать  $X$ ", декларативные знания отвечают, скорее, на вопросы: "Что есть  $X$ " или "Какие связи имеются между  $X$  и  $Y$ ", "Почему  $X$ " и т. д. ких фактов и т. п.

# Типы и виды знаний

**Знания о предметной области** - совокупность сведений о предметной области, хранящихся в базе знаний интеллектуальной системы. В эти знания входят факты, относящиеся к предметной области, закономерности, характерные для нее, гипотезы о возможных связях между явлениями, процессами и фактами в ней, процедуры для решения типовых задач в данной проблемной области. Знания о предметной области вводятся в базу знаний инженером по знаниям. В процессе функционирования интеллектуальной системы они могут пополняться.

# Типы и виды знаний

**Знания прагматические** - знания о способах решения задач в определенной предметной области.

**Знания эвристические** - знания, накапливаемые интеллектуальной системой в процессе ее функционирования, а также знания, заложенные в ней априорно, но не имеющие статуса абсолютной истинности в данной проблемной области. Часто знания эвристические связаны с отражением в базе знаний человеческого (неформального) опыта решения задач.

# Типы и виды знаний

**Знания экспертные** - знания, которыми располагает специалист в некоторой предметной области.

**База знаний.** Принципиальное отличие систем искусственного интеллекта состоит в том, что для такого рода систем программист не готовит конкретные программы для исполнения. Человек лишь дает машине нужное задание, а программу, выполняющую это задание, система должна построить сама. Для этого нужны знания как о предметной области, к которой относится задание, так и о том, как строятся программы.

Все эти знания хранятся в ИИС в специальном блоке, называемом базой знаний. База знаний основа любой ИИС.

Знания, хранящиеся в базе знаний, записываются в специальной формализованной форме. В базе знаний могут реализоваться процедуры обобщения и корректировки хранимых знаний, а также процедуры, создающие новые знания на основании тех, которые уже там имеются.

Знания, хранимые в базе, отличаются от данных, хранящихся в *базе данных*, несколькими особенностями.

Во-первых, структура знаний намного сложнее структуры данных.

Во-вторых, знания обладают свойством внутренней активности. Чтобы пояснить это принципиальное свойство знаний, вспомним, что при обычной работе с компьютером данные всегда играют пассивную роль.

# База знаний

*совокупность программных средств, обеспечивающих поиск, хранение, преобразование и запись в памяти ЭВМ сложно структурированных информационных единиц - знаний.*

**База знаний замкнутая** - база знаний, содержимое которой в процессе функционирования не изменяется.

Логический вывод в такой базе эквивалентен выводу в формальной системе и обладает свойством монотонности, т.е. ранее выведенные утверждения остаются верными на весь период функционирования базы знаний.

**База знаний открытая** - база знаний, позволяющая в процессе ее функционирования пополнять содержимое базы и убирать знания из базы. Свойство открытости приводит к тому, что вывод в такой базе является немонотонным, т.е. истинность выведенных в ней утверждений может меняться в процессе работы системы с такой базой.

# Требования к моделям знаний

Представление знаний – важная проблема, т.к. она влияет на свойства и характеристики ИИС.

*Представление знаний – это формализация знаний для их ввода в базу знаний.*

Действия над знаниями осуществляются программным путём, поэтому знания должны быть представлены формальными моделями, к которым предъявляются два основных требования:

1. **ОДНОРОДНОСТЬ** (единообразие) - дает возможность упростить механизм управления логическим выводом и знанием;

2. **ПОНЯТНОСТЬ** для экспертов и пользователей. Иначе затрудняется процесс приобретения знаний и оценка их качества.

# Основные типы моделей представления знаний

1. Логическая модель представляет собой формальную систему в виде логического исчисления, как правило исчисление предикатов первого порядка. Все знания о предметной области описываются в виде формул этого исчисления или правил вывода. Описание в виде формул дает возможность представить декларативные знания, а правила вывода — процедурные знания.

# Основные типы моделей представления знаний

2. Продукционная модель (модель правил) - это модель, основанная на правилах, которая позволяет представить знания в виде предложений типа «**Если** (условие), **то**(действие)». Знания представлены совокупностью правил «если-то».

Различают два противоположных типа продукционной модели:

- а) с прямыми выводами – для решения задач диагностического характера;
- б) с обратными выводами – для решения задач проектирования и прогнозирования.



# Основные типы моделей представления знаний

## 3. Сетевые модели:

а) *Семантическая сеть* – это модель, в которой структура знаний предметной области формализуется в виде ориентированного графа с помеченными вершинами и дугами. Вершины графа обозначают понятия различных категорий: объекты, события, свойства, операции, а дуги – отношения между ними.

б) *Фреймовая модель* – это модель, в которой структура знаний предметной области формализуется в виде совокупности взаимосвязанных фреймов, описывающих объекты, а свойства этих объектов и факты, относящиеся к ним, описываются в структурных элементах фрейма.

**Более подробно обо всех моделях и примерах со стр. 57 учебника ИИС А.Н. Козлов**

# Назначение экспертных систем

**Экспертная система** — компьютерная система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации.

В начале восьмидесятых годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС).

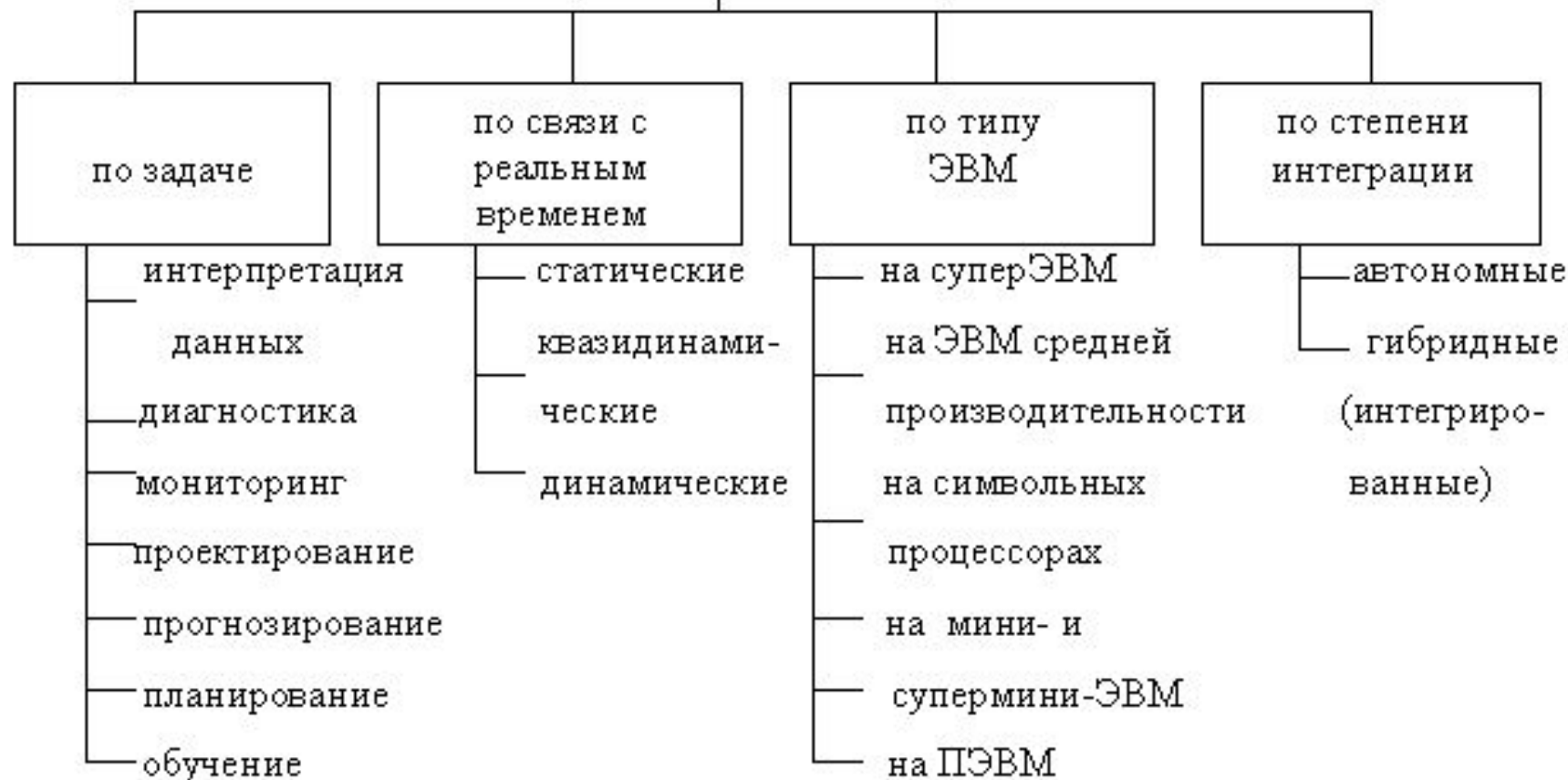
Цель исследований по ЭС состоит в разработке программ, которые при решении задач, трудных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом.

Исследователи в области ЭС для названия своей дисциплины часто используют также термин "инженерия знаний", введенный Е.Фейгенбаумом как "привнесение принципов и инструментария исследований из области искусственного интеллекта в решение трудных прикладных проблем, требующих знаний экспертов".

# Классификация ЭС

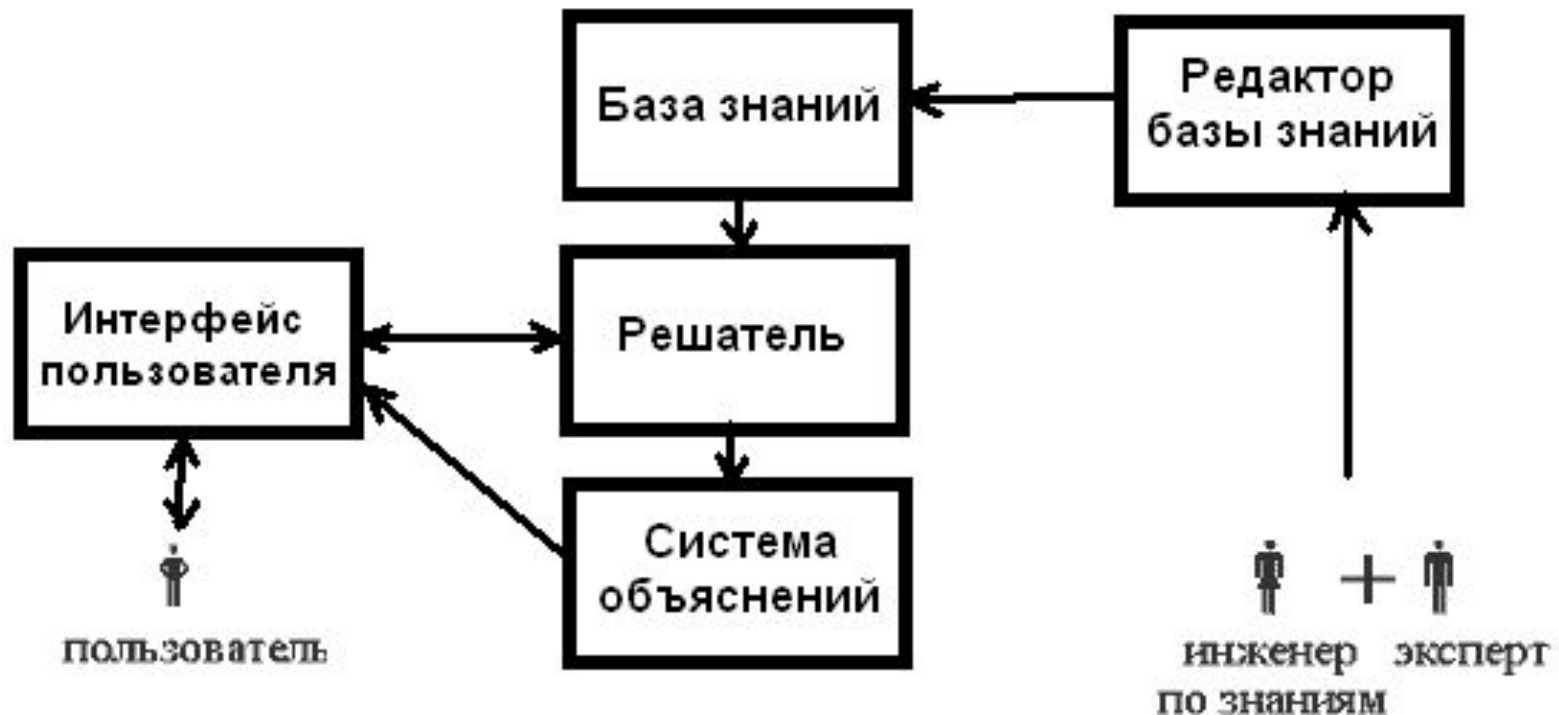
Класс "экспертные системы" сегодня объединяет несколько тысяч различных программных комплексов, которые можно классифицировать по различным признакам.

# Экспертные системы



# Обобщенная структура ЭС

ЭС - это сложный программный комплекс, аккумулирующий знания специалистов в конкретных предметных областях и тиражирующий этот эмпирический опыт для консультаций менее квалифицированных пользователей. Реальные экспертные системы могут иметь более сложную структуру, однако блоки, изображенные на рисунке, непременно присутствуют в любой действительно экспертной системе.



# Состав и назначение элементов ЭС

- *Интерфейс пользователя* - комплекс программ, реализующих диалог пользователя с ЭС как на стадии ввода информации, получения результатов и «объяснения» решения.
- *База знаний(БЗ)* - ядро ЭС, совокупность формализованных знаний предметной области, записанная на машинный носитель в форме, понятной эксперту и инженеру по знаниям. Параллельно такому представлению существует БЗ во внутреннем "машинном" представлении.
- *Решатель(Машина вывода, Интерпретатор)* - программа, моделирующая ход рассуждений эксперта на основании формализованных знаний, имеющихся в БЗ и исходных данных (фактах), получаемых от пользователя.

# Состав и назначение элементов ЭС

- *Подсистема объяснений* - программа, протоколирующая работу решателя в виде «цепочки логических выводов». Она позволяет пользователю получить ответы на вопросы; "Как была получена та или иная рекомендация?" и "Почему система приняла такое решение?" Ответ на вопрос "как" - это трассировка всего процесса получения решения с указанием использованных фрагментов БЗ, т.е. всех шагов цепи умозаключений. Ответ на вопрос "почему"- ссылка на умозаключение, непосредственно предшествовавшее полученному решению, т.е. отход на один шаг назад.
- *Редактор БЗ*- программа, представляющая инженеру по знаниям возможность создавать и пополнять БЗ в диалоговом режиме. Осуществляет ввод формализованных знаний. Например, правил продукционной модели представления знаний. Включает в себя систему вложенных меню, шаблонов языка представления знаний, подсказок ("help" - режим) и других сервисных средств, облегчающих работу с базой знаний.

# Состав и назначение элементов ЭС

В разработке и эксплуатации ЭС участвуют как минимум четыре человека:

- эксперт;
- инженер по знаниям;
- программист (на схеме не показан);
- пользователь.

Возглавляет коллектив инженер по знаниям, это ключевая фигура при разработке систем, основанных на знаниях. Синонимы: когнитолог, инженер-интерпретатор, аналитик.



# Статическая и динамическая ЭС

**Статическая экспертная** система используются в тех случаях, когда можно не учитывать изменения окружающего мира, происходящие за время решения задачи.

Экспертная система работает в двух режимах:

- в режиме приобретения знаний;
- в режиме решения задачи (называемом также режимом консультации или режимом использования ЭС).

**В режиме приобретения знаний** общение с ЭС осуществляет (через посредничество инженера по знаниям) эксперт. В этом режиме эксперт, используя компонент приобретения знаний (редактор БЗ), наполняет систему знаниями, которые позволяют ЭС в режиме решения самостоятельно (без эксперта) решать задачи из проблемной области. Эксперт описывает проблемную область в виде совокупности данных и правил. Данные определяют объекты, их характеристики и значения, существующие в области экспертизы. Правила определяют способы манипулирования с данными, характерные для рассматриваемой области.

*Отметим, что режиму приобретения знаний в традиционном подходе к разработке программ соответствуют этапы алгоритмизации, программирования и отладки, выполняемые программистом. Таким образом, в отличие от традиционного подхода в случае ЭС разработку программ осуществляет не программист, а эксперт, не владеющий программированием.*

**В режиме консультации** общение с ЭС осуществляет пользователь, которого интересует результат и (или) способ его получения. Необходимо отметить, что в зависимости от назначения ЭС пользователь может не быть специалистом в данной проблемной области (в этом случае он обращается к ЭС за результатом, не умея получить его сам), или быть специалистом (в этом случае пользователь может сам получить результат, но он обращается к ЭС с целью либо ускорить процесс получения результата, проконсультироваться, либо возложить на ЭС рутинную работу). В режиме консультации исходные данные (факты) о задаче от пользователя поступают через интерфейс в решатель. Решатель на основе входных данных, общих данных о проблемной области и правил из БЗ формирует решение задачи. ЭС при решении задачи не только исполняет предписанную последовательность операции, но и предварительно формирует ее. Если реакция системы не понятна пользователю, то он может потребовать объяснения, которое выдается из системы объяснений.

# Статическая и динамическая ЭС

**Динамическая экспертная** система используются в тех случаях, когда необходимо учитывать изменения окружающего мира, происходящие за время решения задачи

По сравнению со статической ЭС вводятся дополнительных два компонента:

- система сопряжения;
- источники внешних динамических данных (датчики).

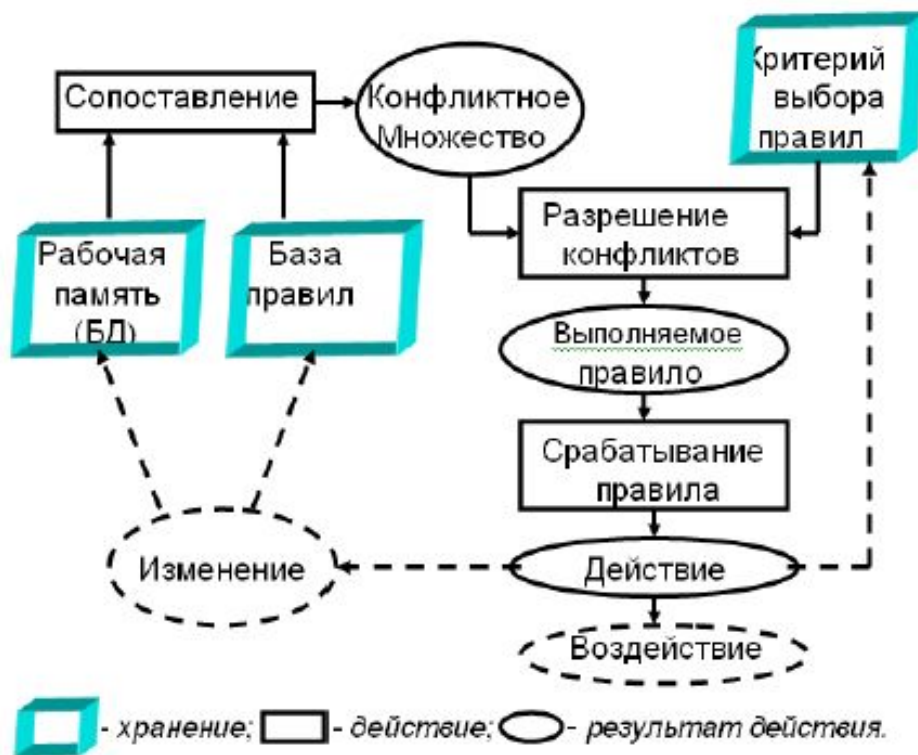
Система сопряжения осуществляет связи с внешним миром через систему датчиков и контроллеров. Кроме того, традиционные компоненты статической ЭС (база знаний и машина вывода) претерпевают существенные изменения, чтобы отразить временную логику происходящих в реальном мире событий.

# Машина вывода экспертных систем

Несмотря на все недостатки, наибольшее распространение получила продукционная модель представления знаний. При использовании продукционной модели база знаний состоит из набора правил.

**Машина вывода** (интерпретатор) – это программа, управляющая перебором правил в базе знаний. Машина вывода (Интерпретатор продукции) работает циклически. Цикл работы интерпретатора схематически изображен справа.

Более подробно со стр.92 учебника А.Н. Козлов ИИС  
Обязательно про обратный и прямой вывод

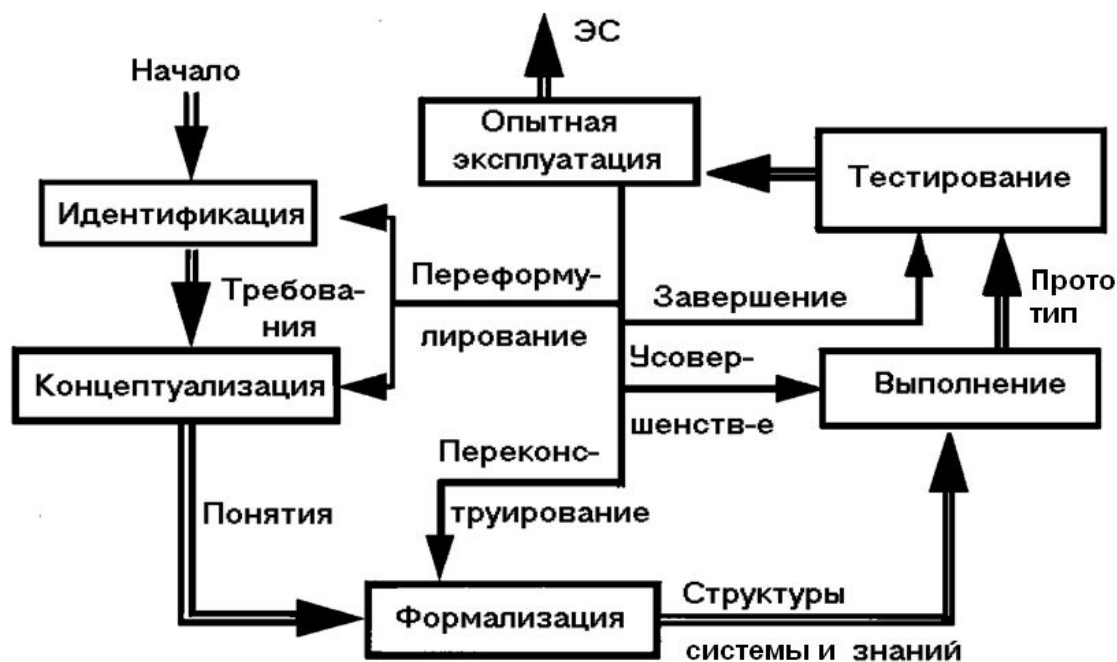


# Этапы разработки экспертных систем

В настоящее время сложилась определенная технология разработки ЭС, которая включает следующие шесть этапов:

- идентификация,
- концептуализация,
- формализация,
- выполнение,
- тестирование
- опытная эксплуатация.

Со стр. 97 учебника А. Н. Козлов ИИС все этапы описание



# Коллектив разработчиков ЭС

Под коллективом разработчиков понимается группа специалистов, ответственных за создание ЭС. В разработке ЭС участвуют представители следующих специальностей.

**Эксперт** определяет знания (данные и правила), характеризующие проблемную область, обеспечивает полноту и правильность введенных в ЭС знаний.

**Инженер по знаниям** помогает эксперту выявить и структурировать знания, необходимые для работы ЭС; осуществляет выбор того инструментального средства, которое наиболее подходит для данной проблемной области, и определяет способ представления знаний. Руководит коллективом.

**Программист(ы)** осуществляет программную реализацию ЭС, сопрягает все основные компоненты ЭС.

**Пользователь** осуществляет проверку удобства работы с ЭС на заключительных этапах разработки. Конечно же его нельзя отнести к специалистам, однако его роль достаточно важна.

Таким образом, минимальный состав коллектива включает четыре человека; реально же он разрастается до 8-10 человек.

# Основы разработки экспертной системы

## Постановка задачи.

Прежде всего, поставим задачу, для решения которой будет разрабатываться ЭС.

Подходящей задачей, при решении которой можно использовать обратную цепочку рассуждений, может быть задача, вытекающая из следующей ситуации: **к директору крупной технической фирмы пришел человек, желающий устроиться на работу.**

Директор располагает сведениями о потребностях фирмы в специалистах и общем положении дел в фирме.

Ему нужно решить, какую должность в фирме может занять посетитель.

Для этого необходимо задать посетителю такие вопросы, ответы на которые дадут возможность сделать правильный выбор должности.

# Основы разработки экспертной системы

На первый взгляд задача не очень сложная, но на решение директора влияет много факторов. Допустим, претендент работает в данной области недавно, но уже сделал важное открытие или он закончил учебное заведение с посредственными оценками, но несколько лет работал по специальности.

В данной ситуации люди ведут себя по-разному, и хотя для того, чтобы получить работу необходимо, удовлетворять определенным критериям, в биографии претендента могут быть самые различные факты, анализ которых поможет подобрать для него соответствующую должность.

Поскольку в задаче надо выбрать один из нескольких возможных вариантов (должностей), для ее решения можно воспользоваться обратной цепочкой рассуждений.

Таким образом, необходимо разработать экспертную систему, которая определит подходящую должность посетителю. Экспертная система будет содержать экспертные знания директора и заменит его при решении описанной задачи. Такой системой может пользоваться как сам посетитель, так и менее квалифицированный (не эксперт - директор) сотрудник фирмы.

Итак, задача поставлена. Теперь нужно наглядно ее представить. Для описания подобных задач обычно используются диаграммы, которые называются **деревьями решений**. Деревья решений дают необходимую наглядность и позволяют проследить ход рассуждений.

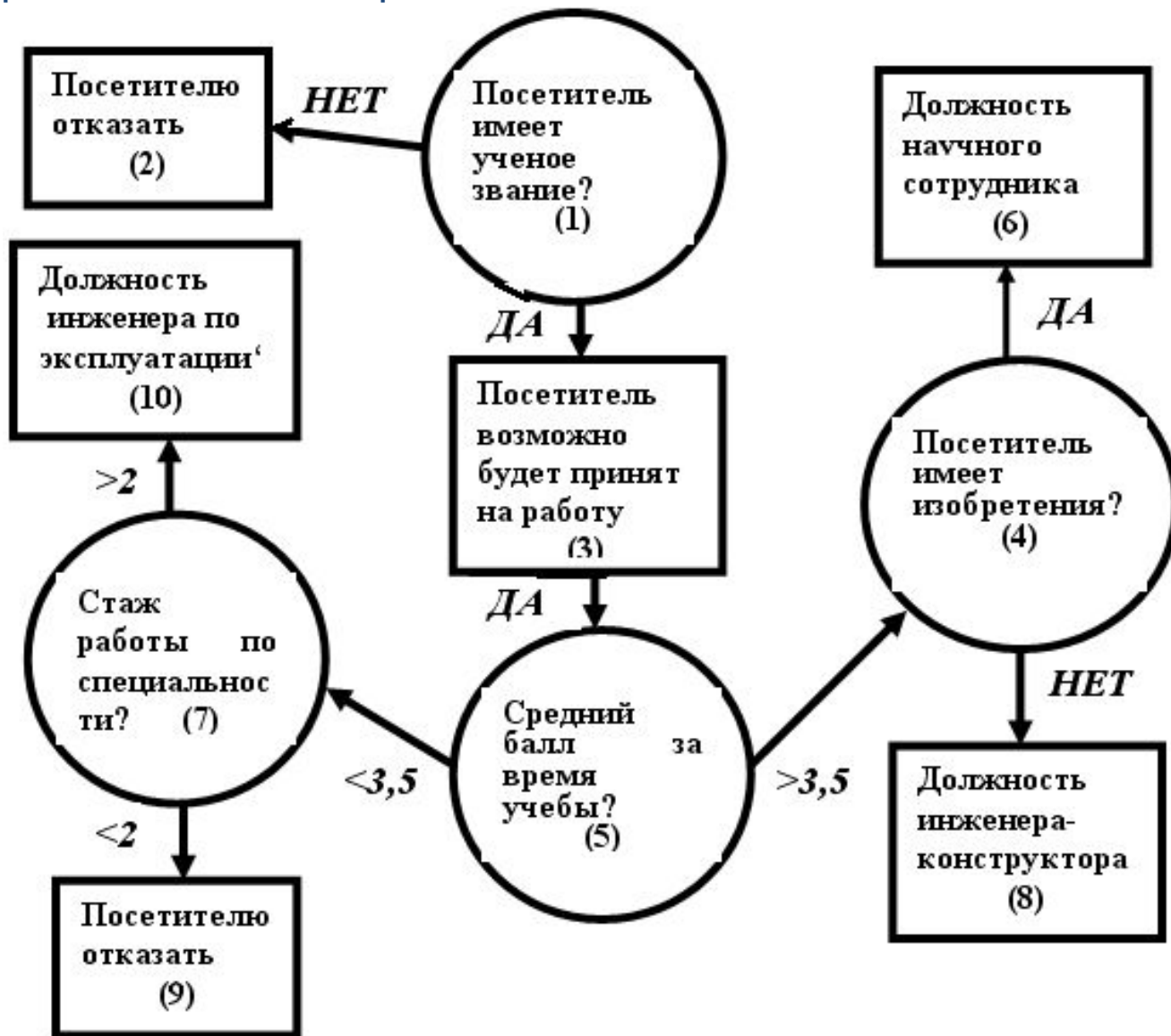
# Разработка дерева решений

*Дерево решений – это ориентированный граф, вершинами которого являются условия и выводы, а дугами результат выполнения (проверки) условий.*

Диаграммы называются деревьями решений потому, что, подобно настоящему дереву, имеют ветви. Ветви деревьев решений заканчиваются логическими выводами. Для рассматриваемого примера вывод заключается в том, предложит ли директор должность поступающему на работу, и если да, то какую. Многие задачи сложны, и их непросто представить (или для их решения не собираются использовать экспертную систему). Дерево решений помогает преодолеть эти трудности.



## Дерево решений для выбора должности



# Основы разработки экспертной системы

На рис предыдущего слайда показано дерево решений для рассматриваемого примера приема на работу. Видно, что диаграмма состоит из кружков и прямоугольников, которые называются вершинами. Каждой вершине присваивается номер. На вершины можно ссылаться по этим номерам. Номера вершин можно выбрать произвольно, т.к. они и служат только для удобства идентификации, за исключением первой вершины. Линии, соединяющие вершины, называются дугами. Совокупность вершин и дуг называется ветвями.

Кружки, содержащие вопросы, называются вершинами условий.

Прямоугольники содержат логические выводы.

Линии (стрелки) показывают направление диаграммы.

Подписи возле линий это ответы на вопрос, содержащийся в вершине условия.

Вершины условий могут иметь сразу по несколько выходящих линий (стрелок), связывающих их с другими вершинами. В этом случае каждая линия (стрелка) должна быть четко определена.

Не может быть две линии, у которых подписи одинаковые, например, подпись «Да».

Выбор выходящей из вершины ветви определяется проверкой условия (вопроса), содержащегося в вершине.

В программе под каждую вершину отводится переменная, а затем ей присваивается значение (ответ посетителя). Можно сказать, что вершины содержат переменные, а пути - это условия, в соответствии с которыми переменным присваиваются значения.

В дереве решений могут быть локальные (частные) выводы или цели.

Для рассматриваемого примера локальным выводом может быть содержащийся в прямоугольнике 3 ответ на вопрос, будет ли посетителю предложена должность. Однако эта вершина имеет и исходящие ветви, и, следовательно, через нее может проходить путь к следующему логическому выводу. В последнем случае, поскольку исходящая ветвь не содержит условия и она только одна, говорят, что вершина содержит локальный вывод для другой цели. Локальный вывод - это также составляющая условной части (ЕСЛИ) правила.

# Преобразование дерева решений в правила

Как уже говорилось, правило «ЕСЛИ-ТО» состоит из двух частей. Часть ЕСЛИ может включать несколько условий, которые связываются между собой логическими операторами И, ИЛИ и НЕ. Часть ТО правила включается в работу только в том случае, если истинны все условия в условной части. В дереве решений обеим частям правила соответствуют связанные между собой вершина(ы) логического условия(ий) (кружки) и вершина логического вывода (прямоугольник). Условная часть содержит все вершины условия, находящиеся на пути к логическому выводу, т.е. каждая вершина решения на пути к выводу - это одно условие части ЕСЛИ, например, вершины 1 и 4. Вывод же составляет часть ТО правила, в данном примере вершины 6, 8 и т.д.

# Порядок формирования правил

Выбрать из дерева решений вершину вывода (прямоугольник) и зафиксировать её.

В обратном направлении линии (стрелки) найти вершину условия (кружок) и зафиксировать её.

Повторять шаг 2 до тех пор, пока не будут исчерпаны все вершины условия, расположенные в обратном направлении стрелок от зафиксированной вершины вывода, или не встретится вершина локального вывода. Если встретилась вершина локального вывода, то её надо зафиксировать и прекратить выполнение шага 2.

Каждая вершина условия (кружок), составляющая путь, - это одна из переменных части ЕСЛИ правила. Эти вершины объединяются логическим оператором И.

Выбранный на шаге 1 логический вывод перенести в часть ТО правила.

**Пример создание правила.** В качестве примера рассмотрим путь 6, 4, 5, 3. Создание правила начинается с вывода (вершина 6) и дерево решения просматривается в обратную сторону. Просмотр данной ветви (пути) заканчивается на вершине 3, которая является локальным выводом. Если бы вершины 3 не было в дереве решений, то путь закончился бы на вершине 1.

Применив полученный путь, запишем правило в следующем виде:

**ЕСЛИ** посетитель, возможно, будет принят на работу = да

**И** средний балл за время учебы  $\geq 3,5$

**И** посетитель имеет изобретения = да,

**ТО** предложенная должность = научный сотрудник.

Для каждой вершины логического вывода определяется путь и записывается правило. В рассматриваемом дереве решений имеется 6 вершин логического вывода (прямоугольников), в том числе и локальный вывод в вершине 3, значит будет составлено 6 правил.

Длинную фразу —посетитель, возможно, будет принят на работу || можно заменить переменной, принимающей значения —да или —нет. Список имен переменных, текст, который они заменяют, и номера вершин пути сводят в таблицу (**со стр. 113 учебника А.Н. Козлов ИИС пример таблицы**). Использование переменных вместо полного текста упрощает формирование и запись правил.

# Структуры данных экспертной системы

При создании экспертной системы для упрощения ответа на вопросы и решения поставленной задачи в систему включается ряд полезных таблиц или структур данных. Структуры данных нужны для работы с базой знаний. После определения метода решения выбранного круга задач можно приступить к разработке системы **(со стр. 115 учебника А.Н. Козлов ИИС)**.

**Список логических выводов** - это структура данных, содержащая упорядоченный список возможных логических выводов.

**Спасибо за внимание!**