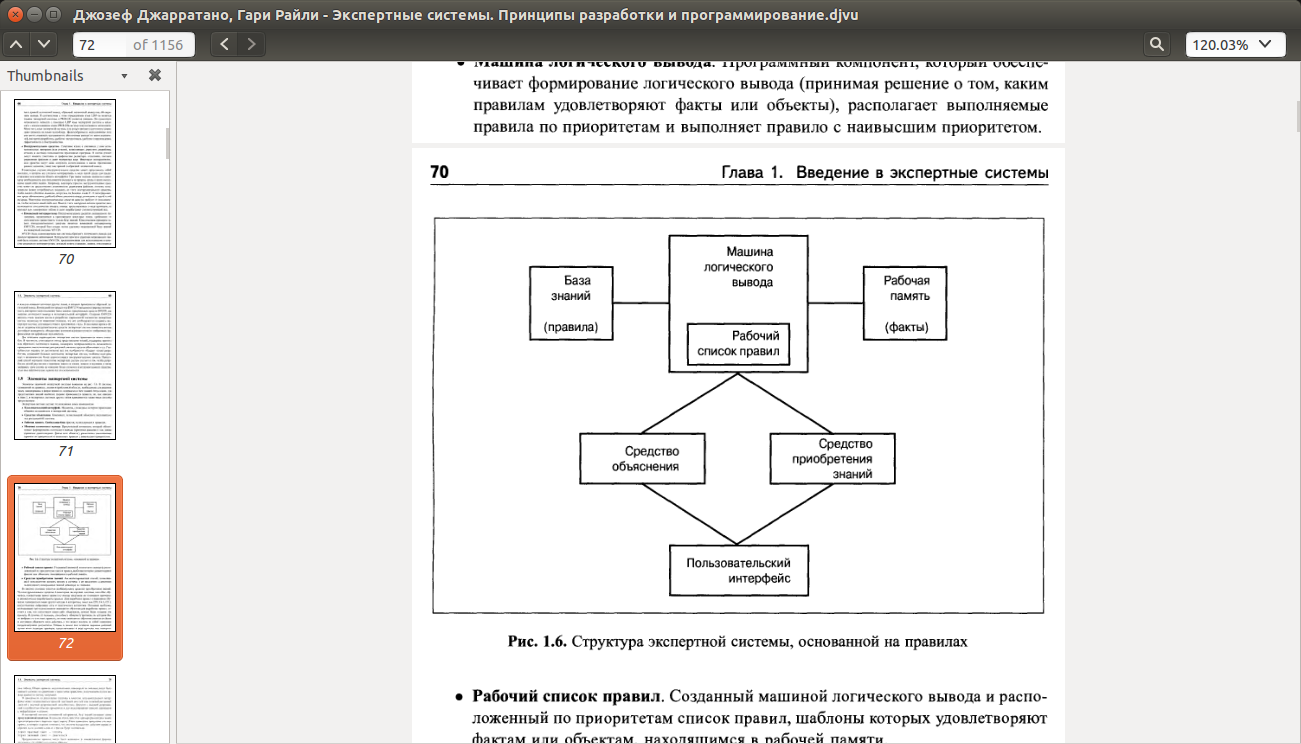
# Экспертные системы

## Структура экспертной системы

Экспертную систему можно представить в виде:



База знаний - набор задокументированных правил. Правила могут быть представлены таким образом:

***IF*** *условие/паттерн/шаблон/антецедент/LHS*

***THEN*** *заключение/вывод/действие/консеквент/RHS* (это всё прямой вывод)

Машина логического вывода - программный компонент, формирует логический вывод. Она принимает решение о том, каким правилам удовлетворяют факты (или объекты):

берёт из рабочей памяти факт и сравнивает его с правилами в базе знаний; если совпадение, то добавляет в рабочий список правил. Машина также сортирует правила по приоритетам (!) и выполняет правило с наивысшим приоритетом!

Рабочий список правил - созданный машиной список правил рассортированных по приоритетам, шаблоны которых удовлетворяют фактам или объектам в рабочей памяти.

Рабочая память - глобальная база фактов, используемых в правилах.

Запуск правила -> выполнить его действие (правая часть правила).

Активизированное правило - это правило, все шаблоны которого удовлетворены.

Работа машины логического вывода:

**WHILE** работа не окончена

**Разрешение конфликтов:** если имеются активизированные правила, то выбираем с наивысшим приоритетом, иначе - конец работы.

**Действие:** последовательно осуществляем действия в правой части выбранного активизированного правила. Только что запущенное правило удаляется из рабочего списка правил.

**Согласование:** обновить рабочий список путём pattern matching: сопоставляем левую часть правила с фактами. Если ОК, тогда активизировать соответствующее правило. Удалить активизированные правила, если левая часть соотв. не выполняется.

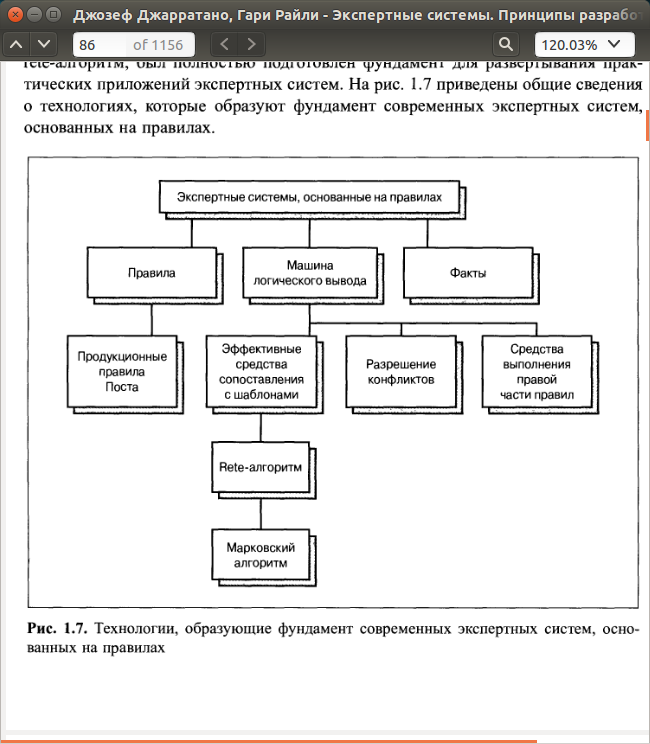
**Проверка условия остановки:** конец если был break.

**END-WHILE**

# Технологии лежащие в основе экспертных систем основанных на правилах

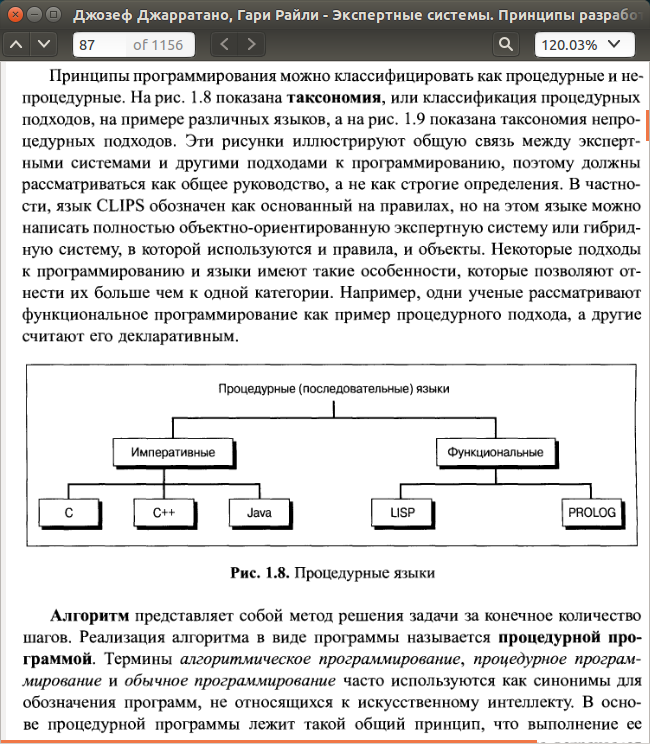
Продукционные системы:

1. Система Поста: продукционная система Поста основана на постановке правил: IF антецедент THEN консеквент  
   Преимущества: простота  
   Недостатки: отстутствие стратегии управления, которая позволяла бы регламентировать применение правил.
2. Алгоритмы Маркова - продукционная система основанная на применении правил с высшими приоритетами и подстановках.
3. Rete-алгоритм - функционирует как сеть (предназнач. для хранения большого объёма информации), обеспечивает быстродействие при запуске правил, алгоритм основан на динамической структуре данных.



# Принципы программирования

## Процедурные подходы



Процедурные (последовательные):

1) Императивные - последовательные, нисходящее проектирование (последовательное уточнение управляющей системы), сложные структуры.

2) Функциональные - последовательные, восходящее проектирование, обеспечение ссылочной прозрачности, в основе обычно лежат *лямбда-исчисления*.

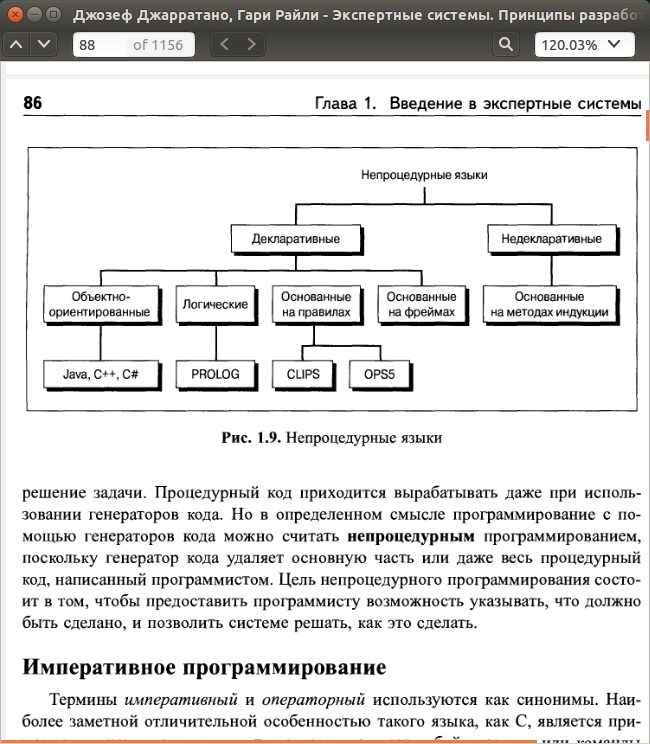
Ведение разработки объекта последовательно от общих черт к детальным называется *нисходящим проектированием*. Его результатом будут требования к отдельным частям и узлам.

Возможен ход разработки от частного к общему, что образует процесс *восходящего проектирования*. Такое проектирование встречается, если одна или несколько частей уже являются готовыми (покупными или уже разработанными) изделиями.

Императивное программирование - это парадигма, которая описывает процесс вычисления в виде инструкций, изменяющих состояния данных.

Функциональное программирование - это парадигма программирования, которая описывает вычисление значений функций, в математ. смысле последних.

## Непроцедурные подходы



Декларативное программирование - это парадигма программирования, которая предусматривает описание того, **что** из себя представляет проблема и результат (в общем, всё означает спецификацию).

В противоположность декларативному программированию ставиться императивное - то **как** должна решаться задача и **как** должен представляться результат.

Непроцедурные:

1) Декларативные (путь достижения цели от методов):

- объектно-ориентированные - и императивный и декларативный:

Декларативный в ООП: мы пытаемся представить данные в виде объектов.

Императивный в ООП: мы пытаемся решить эту проблему определяя операции для этих самых объектов.

- логические

- основанные на правилах

- основанные на фреймах

2) Недекларативные (основанные на индукции)

Примечание: функциональное программирование можно считать декларативным, а структурное и ООП - всё-таки императивным.

# Логические выводы

1. Прямой логический вывод
2. Обратный логический вывод

Прямой логический вывод - это метод формирования рассуждений от ФАКТОВ к ЗАКЛЮЧЕНИЯМ, которые следуют из фактов.

IF идёт на улице идёт дождь -> факт

THEN нужно взять зонтик ----> заключение

Обратный логический вывод - это метод формирования рассуждений от ГИПОТЕЗЫ к ФАКТАМ, которые подтверждают гипотезу.

Гипотеза - это потенциальное заключение, которое должно быть доказано.

IF кто-то вошел в дом с влажными ботинками и зонтиком -> факт

THEN идёт дождь -> приняли гипотезу

Для подтверждения гипотезы нужно спросить у человека, идет ли дождь.

Если да, то гипотеза истинная, то она становится фактом.

# Представление знаний

Эвристика - опыт, который можно использовать для решения некоторой задачи.

Эпистемология (часто синоним *гносеологии*) - (фил-методическая дисциплина) наука о знаниях, изучает характер, структуру и происхождение знаний.

Гносеоло́гия — философская дисциплина, занимающаяся исследованиями, критикой и теориями познания.

Тенденция к разграничению этих двух понятий характерна для неклассической (современной) философии, и при таком подходе:

**Эпистемология** — изучает отношение *«объект–знание».*

**Гносеология** — изучает отношение *«субъект–объект»*;

Прецендент - уже встречавшийся ранее случай.

Эпистемология состоит из:

1. Философская теория
2. Априорные знания - невозможно опровергнуть (мат законы, логические знания)
3. Апостериорные знания - полученные с помощью органов чувств. Они могут быть опровергнуты с помощью новых знаний

Знания могут делится на:

* процедурные: знание о процессе чего-либо
* декларативные: знания о том, является утверждение истинным или ложным
* неявные - **подсознательные**: не могут быть выражены с помощью языка

Алгоритмы + Структуры данных = Программы

Знания + Логический вывод = Экспертные системы

Экспертные системы - это **формирование логических выводов**

Человеческие размышления - это **рассуждения**.

Данные (Алгоритм получения информации)-> Информация -> Знания -> Метазнания ->Мудрость

Знания - это правило.

Экспертные знания - это неявные знания и навыки эксперта!

Метазнания - это знания об обычных и экспертных знаниях.

Мудрость - это метазнания, позволяющие определять наилучшие цели в жизни и находить пути их достижения.

Методы представления знаний (модели):

1. Правила и продукционные правила
2. Семантические сети
3. Фреймы
4. Сценарии
5. Логика
6. Концептуальные схемы

## Продукции (продукционные правила)

Для определения продукций используется форма Бэкуса-Наура (BNF).

ФБН - это система обозначений, **метаязык**, применимый для определения **синтаксиса**(формы) любого языка.

Продукционное правило для формулировки простого правила:

<sentence>::= <subject> <verb> <end-mark>

(предложение определяется (состоит из) подлежащим, сказуемом и маркером конца предложения)

<>, := - это всё символы метаязыка.

::= - “определено как”

Термины в <> - это нетерминальные символы (nonterminal).

nonterminal символ - это переменная, представляющая другой терм (нетерминальный символ или терминальный (константа)).

Нетерминальный символ <sentence> относится к **начальным символам**.

Например, в программных языках таким символом является <program>.

Это продукционное правило раскрывает значение нетерминальных символов указанием терминальных символов через “или” - символ   
“|”:

<subject> -> I | You | We

<verb> -> left | came

<end-mark> -> . | ? | !

Продукции - это все возможные предложения языка.

Получить продукции можно путём замены всех нетерминальных символов терминальными или нетерминальными, пока не будут устранены все нетерминальные символы:

I left.

I left?

I left!

………………..

Строка символов языка - ряд терминальных символов.

Грамматика - полный набор продукционных правил, который однозначно определяет язык.

## Семантические сети (пропозициональные сети)

Семантическая сеть - это помеченный ориентированный граф.

Высказывания - либо истинные либо ложные - они всегда атомартные, то есть их истиностные значения не подлежат дальнейшей декомпозиции.

Узлы сс именуются объектами, а дуги - связями.

Главным в семантических сетях являются отношения.

Особо полезными отношениями являются отношения: IS-A и A-KIND-OF (а еще есть HAS-A).

IS-A - связь между экземпляром и классом, только для конкретных объектов. Пример: автомобиль “Nissan” is-a семейный автомобиль.

A-KIND-OF - между классами. Пример: Семейный автомобиль a kind of грузовой автомобиль.

Триплет: объект-атрибут-значение

Пара: атрибут-значение (если используется один объект)