

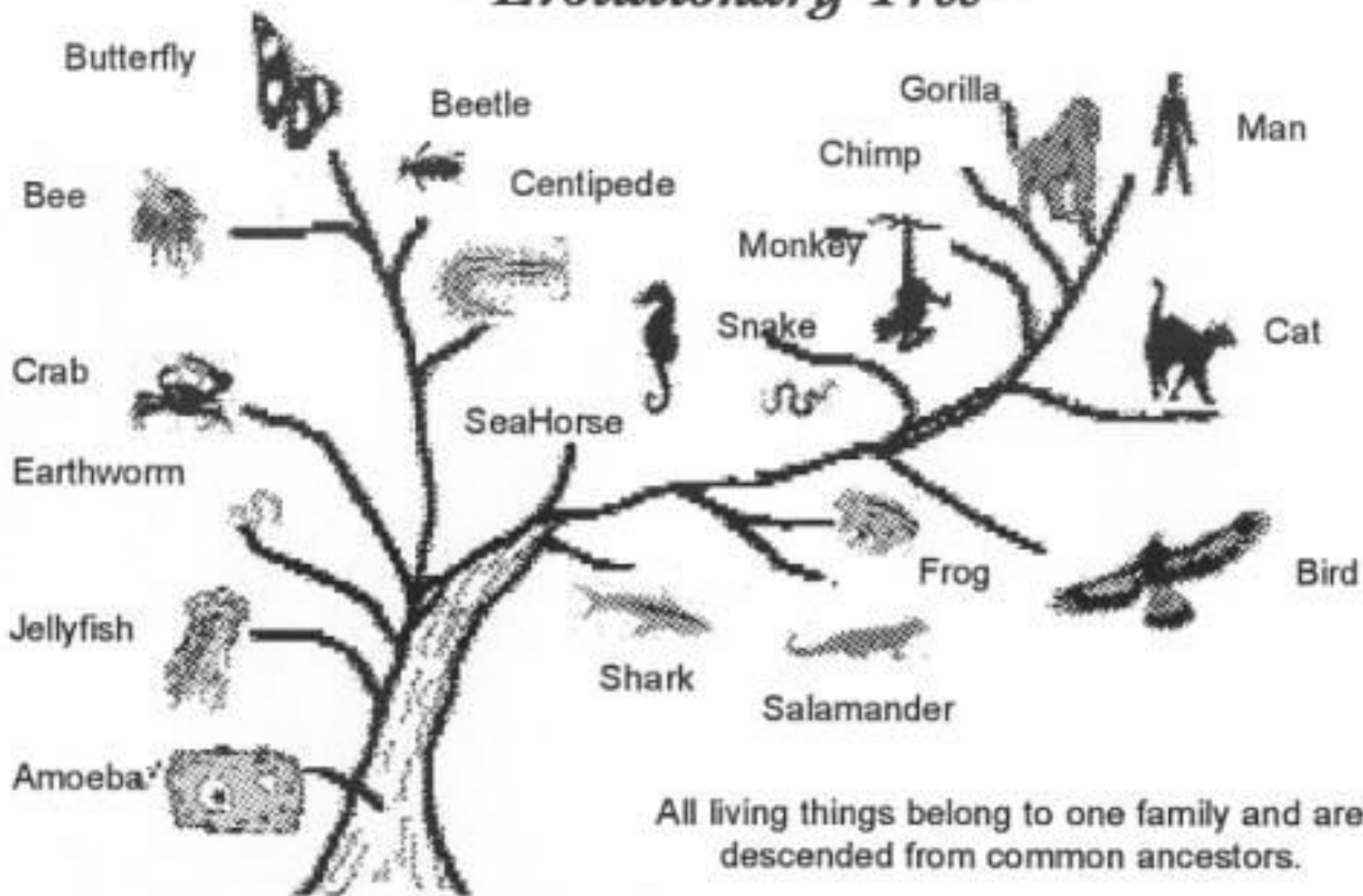
# **Лекция 3**

## **Направления развития искусственного интеллекта**

# Эволюционное моделирование

- Одной из главных характеристик искусственного интеллекта как науки является его междисциплинарность, позволяющая привлекать интересные идеи, теории из других областей знаний, адаптирую и используя готовые разработки для своих задач. Так было с нейронными сетями, с моделированием рассуждений, с компьютерной лингвистикой и т.д. Многие значимые теории науки были так или иначе рассмотрены через призму искусственного интеллекта. Теория Ч. Дарвина (1859 г.) стала отправной точкой для еще одного направления исследований – эволюционного моделирования.

# *“Evolutionary Tree”*



All living things belong to one family and are descended from common ancestors.

- Основной тезис эволюционного моделирования - заменить процесс моделирования сложного объекта моделированием его эволюции. Он направлен на применение механизмов естественной эволюции при синтезе сложных систем обработки информации. Дарвин сформулировал основной закон развития органического мира, охарактеризовав его взаимодействием трех следующих факторов:

- наследственность (потомки сохраняют свойства родителей);
- изменчивость (потомки почти всегда не идентичны);
- естественный отбор (выживают наиболее приспособленные).

Теория Дарвина, дополненная генетическими знаниями, называется синтетической теорией эволюции. Случайное появление новых признаков она объяснила мутациями - изменениями, возникающими в ДНК организмов.

- Понятие «эволюционное моделирование» сформировалось в работах Л. Фогеля, А. Оуэне, М. Уолша. В 1966 году вышла их совместная книга «Искусственный интеллект и эволюционное моделирование». История эволюционных вычислений началась с разработки ряда различных независимых моделей. Основными из них были генетические алгоритмы и классификационные системы Д. Холланда (Holland), опубликованные в начале 60-х годов и получившие всеобщее признание после выхода в свет книги, ставшей классикой в этой области, - "Адаптация в естественных и искусственных системах" ("Adaptation in Natural and Artificial Systems", 1975).

Пусть перед нами стоит задача оптимизации, например:

Задача наилучшего приближения

– Если рассматривать систему  $n$  линейных уравнений с  $m$  неизвестными

$$Ax = b$$

- в случае, когда она переопределена ( $n > m$ ), то иногда оказывается естественной задача о нахождении вектора  $x$ , который "удовлетворяет этой системе наилучшим образом", т. е. из всех "не решений" является лучшим.

## Задача о рационе.

– Пусть имеется  $n$  различных пищевых продуктов, содержащих  $m$  различных питательных веществ. Обозначим через  $a_{ij}$  содержание (долю)  $j$ -го питательного вещества в  $i$ -ом продукте, через  $b_j$  — суточную потребность организма в  $j$ -ом питательном веществе, через  $c_i$  — стоимость единицы  $i$ -го продукта. Требуется составить суточный рацион питания минимальной стоимости, удовлетворяющий потребность во всех питательных веществах



# Транспортная задача.

- Эта задача — классическая задача линейного программирования. К ней сводятся многие оптимизационные задачи. Формулируется она так. На  $m$  складах находится груз, который нужно развезти  $n$  потребителям. Пусть  $a_i$  ( $i = 1, \dots, n$ ) — количество груза на  $i$ -ом складе, а  $b_j$  ( $j = 1, \dots, m$ ) — потребность в грузе  $j$ -го потребителя,  $c_{ij}$  — стоимость перевозки единицы груза с  $i$ -го склада  $j$ -му потребителю. Требуется минимизировать стоимость перевозок.

## Задачи о распределении ресурсов.

- Общий смысл таких задач — распределить ограниченный ресурс между потребителями оптимальным образом. Рассмотрим простейший пример — *задачу о режиме работы энергосистемы*. Пусть  $m$  электростанций питают одну нагрузку мощности  $p$ . Обозначим через  $x_j$  активную мощность, генерируемую  $j$ -ой электростанцией. Техническими условиями определяются возможный минимум  $m_j$  и максимум  $M_j$  вырабатываемой  $j$ -ой электростанцией мощности. Допустим затраты на генерацию мощности  $x$  на  $j$ -ой электростанции равны  $e_j(x)$ . Требуется сгенерировать требуемую мощность  $p$  при минимальных затратах.

- Переформулируем задачу оптимизации как задачу нахождения максимума некоторой функции  $f(x_1, x_2, \dots, x_n)$ , называемой *функцией приспособленности* (fitness function). Она должна принимать неотрицательные значения на ограниченной области определения (для того, чтобы мы могли для каждой особи считать её приспособленность, которая не может быть отрицательной), при этом совершенно не требуются непрерывность и дифференцируемость.
- Каждый параметр функции приспособленности кодируется строкой

Особью будет называться строка,  
являющаяся конкатенацией строк  
упорядоченного набора параметров:

1010 10110 101 ... 10101

| x1 | x2 | x3 | ... | xn |

Универсальность ГА заключается в том, что  
от конкретной задачи зависят только такие  
параметры, как функция  
приспособленности и кодирование  
решений. Остальные шаги для всех задач  
производятся одинаково.

Генетические алгоритмы оперируют совокупностью особей (популяцией), которые представляют собой строки, кодирующие одно из решений задачи. Этим ГА отличается от большинства других алгоритмов оптимизации, которые оперируют лишь с одним решением, улучшая его.

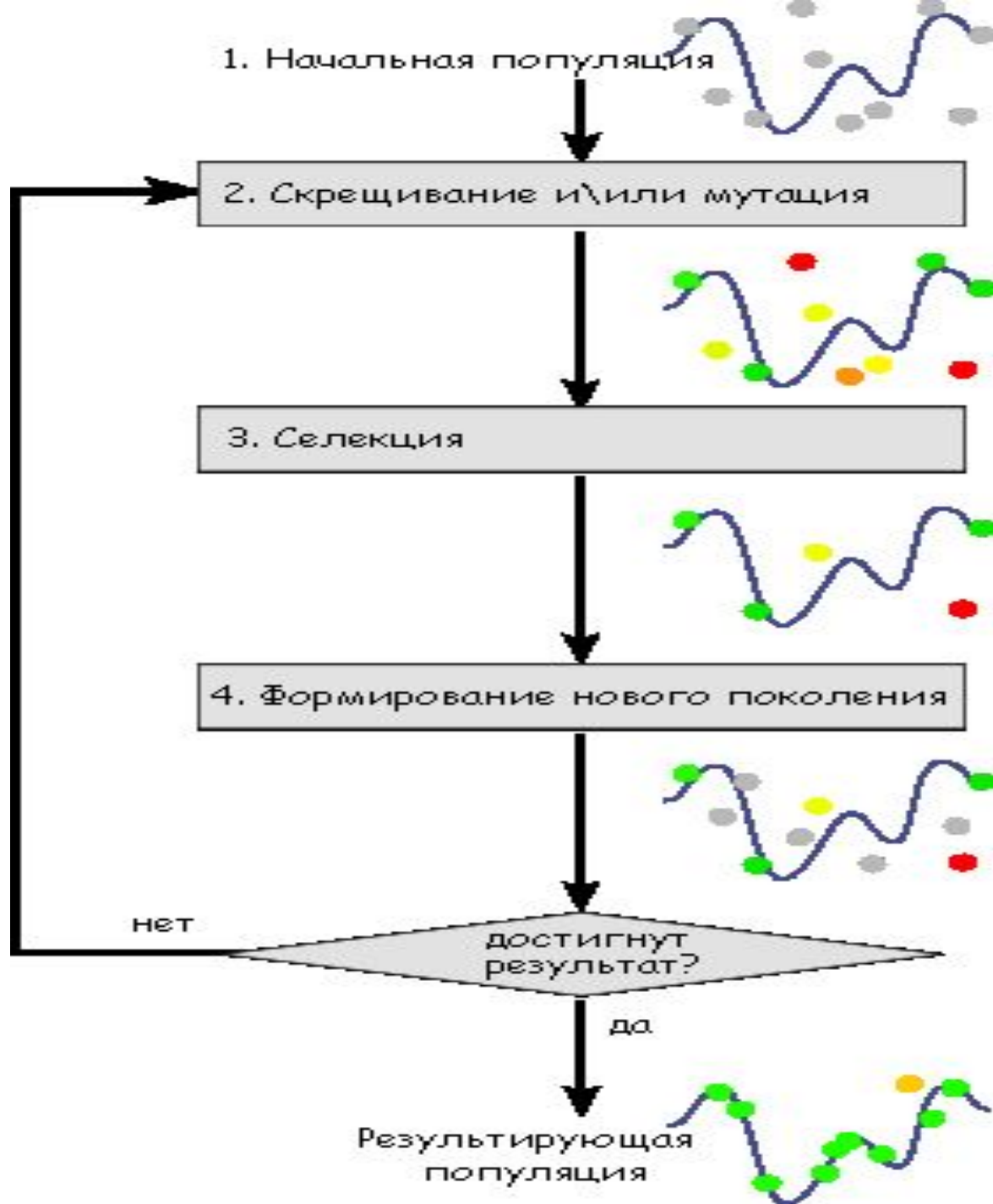
С помощью функции приспособленности среди всех особей популяции выделяют:

- наиболее приспособленные (более подходящие решения), которые получают возможность скрещиваться и давать потомство
- наихудшие (плохие решения), которые удаляются из популяции и не дают потомства

Таким образом, приспособленность нового поколения в среднем выше предыдущего.

В классическом ГА:

- начальная популяция формируется случайным образом
- размер популяции (количество особей  $N$ ) фиксируется и не изменяется в течение работы всего алгоритма
- каждая особь генерируется как случайная  $L$ -битная строка, где  $L$  — длина кодировки особи
- длина кодировки для всех особей одинакова



Selection  
(Duplication)

Recombination  
(Crossover)

|          |
|----------|
| String 1 |
| String 2 |
| String 3 |
| String 4 |
| .....    |
| .....    |
|          |
|          |
|          |

|          |
|----------|
| String 1 |
| String 2 |
| String 2 |
| String 4 |
| .....    |
| .....    |
|          |
|          |
|          |



|                     |
|---------------------|
| Offspring-A (1 X 2) |
| Offspring-B (1 X 2) |
| Offspring-A (2 X 4) |
| Offspring-B (2 X 4) |
| .....               |
| .....               |
|                     |
|                     |
|                     |

Current  
Generation  $t$

Intermediate  
Generation  $t$

Next  
Generation  $t + 1$



- Промежуточная популяция — это набор особей, получивших право размножаться. Наиболее приспособленные особи могут быть записаны туда несколько раз, наименее приспособленные с большой вероятностью туда вообще не попадут.
- В классическом ГА вероятность каждой особи попасть в промежуточную популяцию пропорциональна ее приспособленности, т.е. работает *пропорциональный отбор (proportional selection)*.

## Скрещивание

Как известно, в теории эволюции важную роль играет то, каким образом признаки родителей передаются потомкам. В генетических алгоритмах за передачу признаков родителей потомкам отвечает оператор, который называется скрещивание (его также называют кроссовер или кроссинговер). Этот оператор определяет передачу признаков родителей потомкам, к ним применяется вероятностный оператор скрещивания, который строит на их основе новые решения-потомка.

Отобранные особи подвергаются кроссоверу (иногда называемому рекомбинацией) с заданной вероятностью  $P_c$ . Если каждая пара родителей порождает двух потомков, для воспроизводства популяции необходимо скрестить  $m/2$  пары. Для каждой пары с вероятностью  $P_c$  применяется кроссовер. Соответственно, с вероятностью  $1 - P_c$  кроссовер не происходит - и тогда неизмененные особи переходят на следующую стадию (мутации).

- Существует большое количество разновидностей оператора скрещивания. Простейший одноточечный кроссовер работает следующим образом.
- Сначала случайным образом выбирается одна из возможных точек разрыва. (Точка разрыва - участок между соседними битами в строке.) Обе родительские структуры разрываются на два сегмента по этой точке. Затем соответствующие сегменты различных родителей склеиваются и получаются два генотипа потомков.

- Родитель 1 1 1 0 0 1 0 1 1 | 0 1 0 0 1
- Родитель 2 0 1 0 0 0 1 1 | 0 0 1 1 1
- Потомок 1 1 0 0 1 0 1 1 | 0 0 1 1 1
- Потомок 2 0 1 0 0 0 1 1 | 0 1 0 0 1

# Мутация

- Следующий генетический оператор предназначен для того, чтобы поддерживать разнообразие особей с популяции, - это оператор мутации. После того, как закончится стадия кроссовера, потомки могут подвергаться случайным модификациям. В простейшем случае в каждой хромосоме, которая подвергается мутации, каждый бит с вероятностью  $P_m$  изменяется на противоположный (это так называемая одноточечная мутация).

- ^ Особь до мутации: 1 0 0 1 0 1 1 0 0 1 1 1
- Особь после мутации: 1 0 0 1 0 1 0 0 0 1 1  
1

- Более сложной разновидностью мутации являются операторы инверсии и транслокации. Инверсия – это перестановка генов в обратном порядке внутри наугад выбранного участка хромосомы.

- ^ Особь до инверсии: 1 0 0 1 1 1 1 0 0 1 1 1

- Особь после инверсии: 1 0 0 1 0 0 1 1 1 1 1  
1

Транслокация - это перенос какого-либо участка хромосомы в другой сегмент этой же хромосомы.

- ^ Особь до транслокации: 1 0011 1100 111
- Особь после транслокации: 1 110 0011 011  
1

Все перечисленные генетические операторы (одноточечный и многоточечный кроссовер, одноточечная мутация, инверсия, транслокация) имеют схожие биологические аналоги.

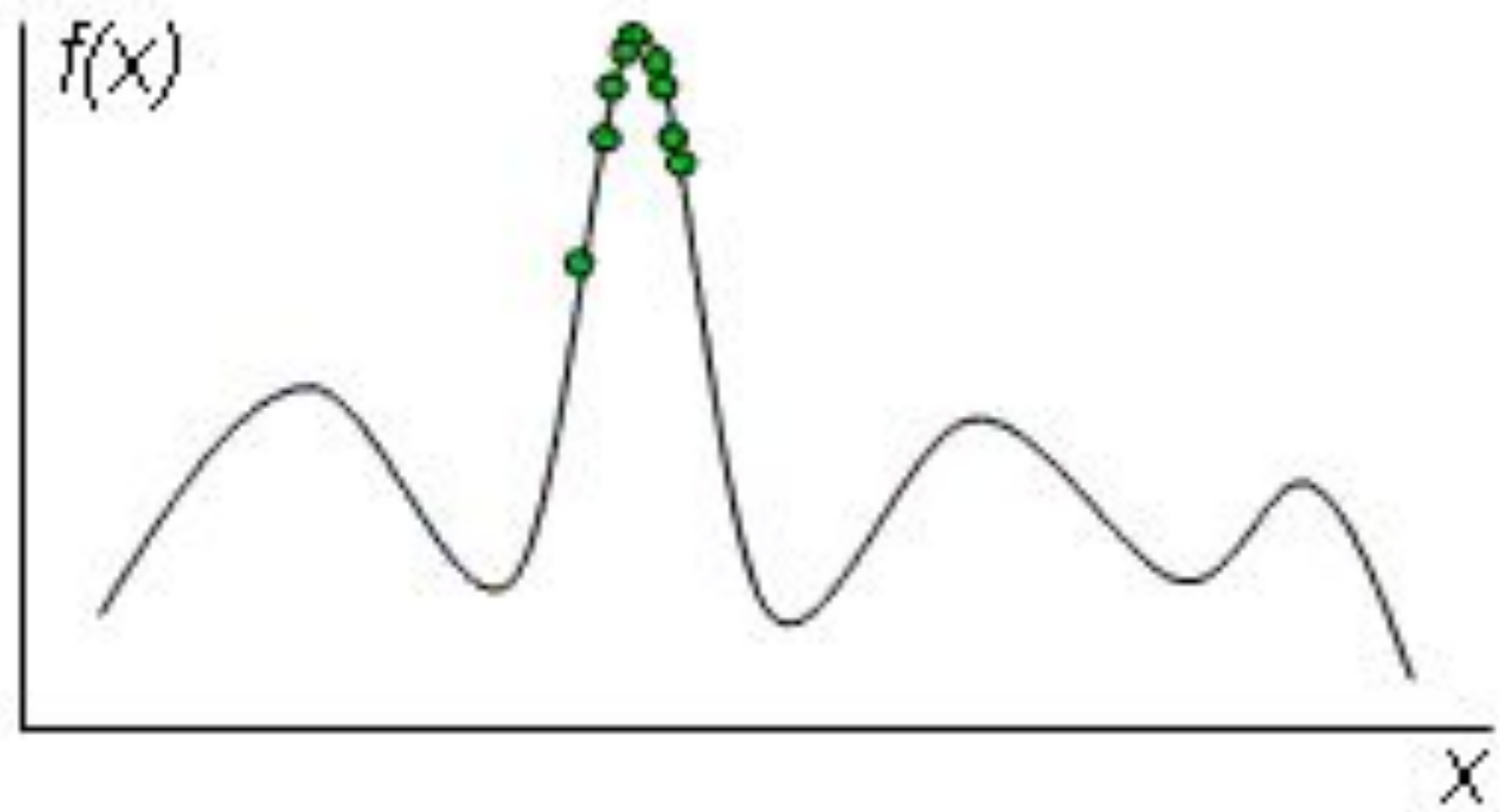


# Формирование нового поколения

После скрещивания и мутации особей необходимо решить проблему: какие из новых особей войдут в следующее поколение, а какие - нет, и что делать с их предками. Есть два наиболее распространенных способа.

- 1. Новые особи (потомки) занимают места своих родителей. После этого наступает следующий этап, в котором потомки оцениваются, отбираются, дают потомство и уступают место своим "детям".
- 2. Следующая популяция включает в себя как родителей, так и их потомков.
- Во втором случае необходимо дополнительно определить, какие из особей родителей и потомков попадут в новое поколение. В простейшем случае, в него после каждого скрещивания включаются две лучших особи из четверки родителей и их потомков. Более эффективным является механизм вытеснения, который реализуется таким образом, что стремится удалять «похожие» хромосомы из популяции и оставлять отличающиеся.

- Такой процесс эволюции, вообще говоря, может продолжаться до бесконечности. Критерием останова может служить заданное количество поколений или *схождение (convergence)* популяции.
- Схождением называется состояние популяции, когда все строки популяции находятся в области некоторого экстремума и почти одинаковы. То есть кроссовер практически никак не изменяет популяции, а мутирующие особи склонны вымирать, так как менее приспособлены. Таким образом, схождение популяции означает, что достигнуто решение близкое к оптимальному.



## ***СНС-алгоритм***

- СНС (Cross generational elitist selection, Heterogenous recombination, Cataclysmic mutation) был предложен Эсхелманом и характеризуется следующими параметрами:
- Для нового поколения выбираются N лучших различных особей среди родителей и детей. Дублирование строк не допускается.
- Для скрещивания выбирается случайная пара, но не допускается, чтобы между родителями было мало хэммингово расстояние или мало расстояние между крайними различающимися битами.
- Для скрещивания используется разновидность однородного кроссовера HUX (Half Uniform Crossover): ребенку переходит ровно половина битов каждого родителя.
- Размер популяции не больше 50 особей. Этим

## *Genitor*

Этот алгоритм был создан Д. Уитли. Genitor-подобные алгоритмы отличаются от классического ГА следующими тремя свойствами:

- На каждом шаге только одна пара случайных родителей создает только одного ребенка.
- Этот ребенок заменяет не родителя, а одну из худших особей популяции (в первоначальном Genitor – самую худшую).
- Отбор особи для замены производится по ее рейтингу, а не по приспособленности.

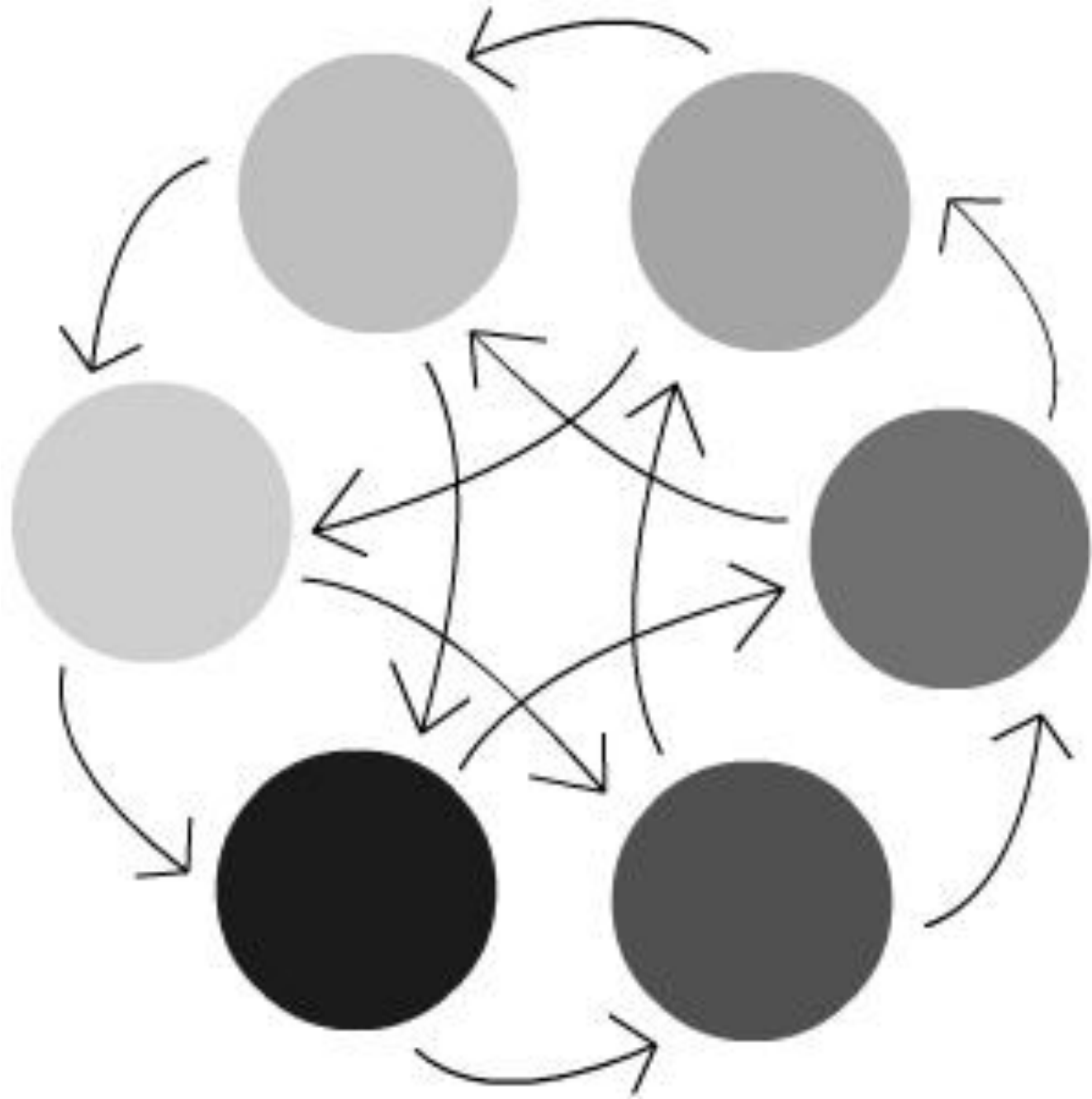
В Genitor сходимость быстрее, чем у классического генетического алгоритма, предложенного Холландом.

# ***Параллельные генетические алгоритмы***

- Генетические алгоритмы можно организовать как несколько параллельно выполняющихся процессов, это увеличит их производительность.
- Рассмотрим переход от классического генетического алгоритма к параллельному. Для этого будем использовать турнирный отбор. Заведем  $N / 2$  процесса (здесь и далее процесс подразумевается как некоторая машина, процессор, который может работать независимо). Каждый из них будет выбирать случайно из популяции 4 особи, проводить 2 турнира и скрещивать победителей. Полученные дети будут записываться в новое поколение. Таким образом, за один цикл работы одного процесса будет сменяться целое поколение.

- Островная модель (island model, рис. 17) – это тоже модель параллельного генетического алгоритма. Она заключается в следующем: пусть у нас есть 16 процессов и 1600 особей. Разобьем их на 16 подпопуляций по 100 особей. Каждая из них будет развиваться отдельно с помощью некоего генетического алгоритма. Таким образом, можно сказать, что мы расселили особи по 16-ти изолированным островам.

Изредка (например, каждые 5 поколений) процессы (или острова) будут обмениваться несколькими хорошими особями. Этот процесс называется миграцией. Миграция позволяет островам обмениваться генетическим материалом





# КОГНИТИВНОЕ МОДЕЛИРОВАНИЕ

Одно из наиболее продуктивных решений проблем, возникающих в области управления и организации, состоит в применении **когнитивного анализа**.

- Методология когнитивного моделирования, предназначенная для анализа и принятия решений в плохо определенных ситуациях, была предложена американским исследователем Р. Аксельродом.
- Изначально когнитивный анализ сформировался в рамках социальной психологии, а именно – когнитивизма, занимающегося изучением процессов восприятия и познания.
- Применения разработок социальной психологии в теории управления привело к формированию особой отрасли знаний – когнитологии, концентрирующейся на исследовании проблем управления и принятия решений.

# ***Этапы когнитивного анализа***

1. Формулировка цели и задач исследования.
2. Изучение сложной ситуации с позиций поставленной цели: сбор, систематизация, анализ существующей статистической и качественной информации относительно объекта управления и его внешней среды, определение присущих исследуемой ситуации требований, условий и ограничений.
3. Выделение основных факторов,

4. Определение взаимосвязи между факторами путем рассмотрения причинно-следственных цепочек (построение когнитивной карты в виде ориентированного графа).
5. Изучение силы взаимовлияния разных факторов. Для этого используются как математические модели, описывающие некоторые точно выявленные количественные зависимости между факторами, так и субъективные представления эксперта относительно неформализуемых качественных

- (В результате прохождения этапов 3 – 5 строится, в конечном итоге, когнитивная модель ситуации (системы), которая отображается в виде функционального графа. Поэтому можно сказать, что этапы 3 – 5 представляют собой когнитивное моделирование. Более подробно все эти стадии и основные понятия когнитивного моделирования будут рассмотрены ниже).
- 6. Проверка адекватности когнитивной модели реальной ситуации (верификация когнитивной модели).

**7. Сценарное моделирование:** Определение с помощью когнитивной модели возможных вариантов развития ситуации (системы), обнаружение путей, механизмов воздействия на ситуацию с целью достижения желаемых результатов, предотвращения нежелательных последствий, то есть выработка стратегии управления. Задание целевых, желаемых направлений и силы изменения тенденций процессов в ситуации. Выбор комплекса мероприятий (совокупности управляющих факторов), определение их возможной и желаемой силы и направленности воздействия на ситуацию (конкретно-практическое применение когнитивной модели).

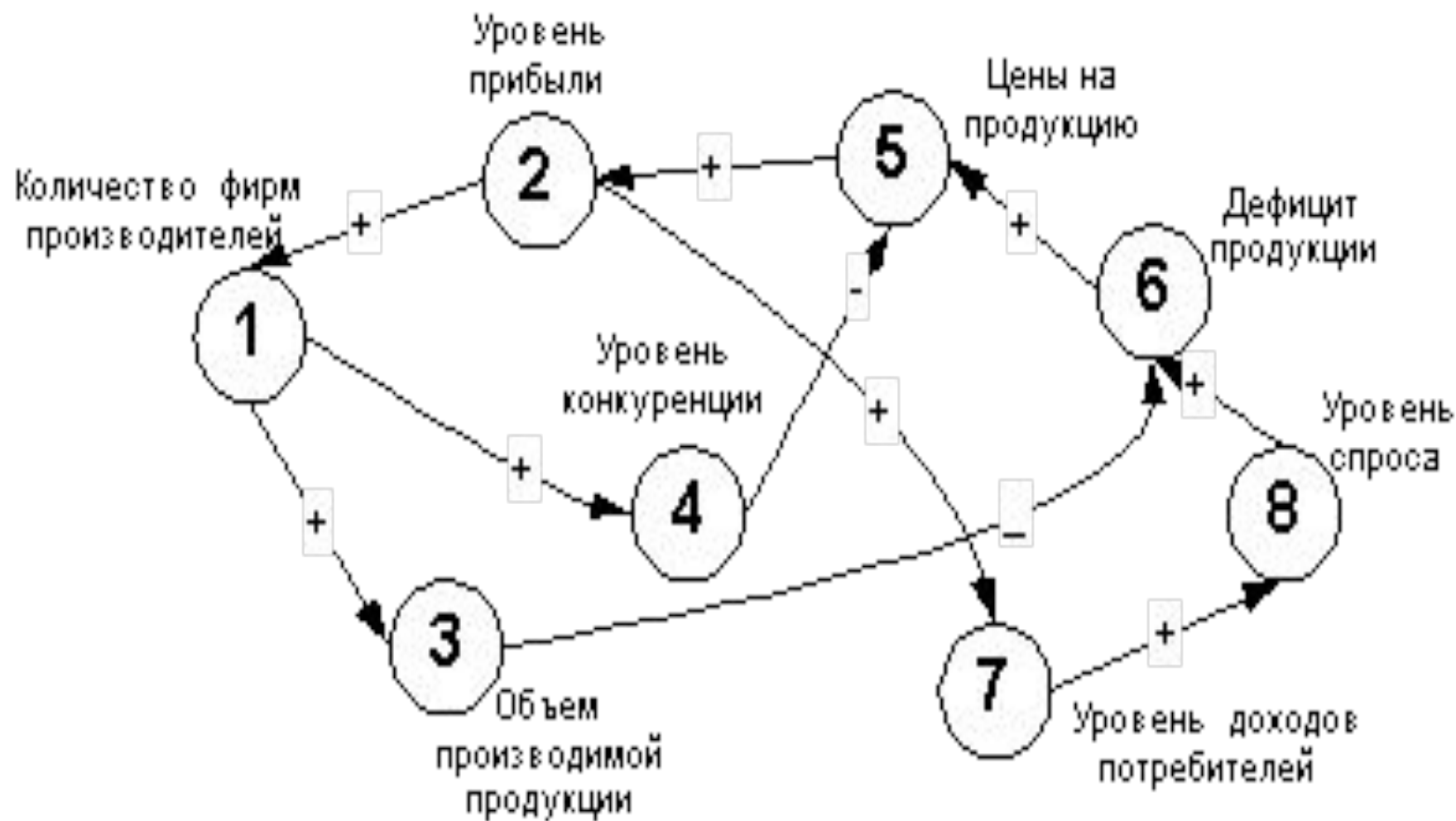
# ***Этапы когнитивного моделирования***

- Выявление факторов, характеризующих проблемную ситуацию, развитие системы (среды). Например, суть проблемы неплатежей налогов можно сформулировать в факторах «Неплатежи налогов», «Собираемость налогов», «Доходы бюджета», «Расходы бюджета», «Дефицит бюджета» и др.
- Выявление связей между факторами. Определение направления влияний и взаимовлияний между факторами. Например, фактор «Уровень налогового бремени» повлияет на «Неплатежи налогов»

- Определение характера влияния (положительное, отрицательное, +\-) Например, увеличение (уменьшение) фактора «Уровень налогового бремени» увеличивает (уменьшает) «Неплатежи налогов» - положительное влияние; а увеличение (уменьшение) фактора «Собираемость налогов» уменьшает (увеличивает) «Неплатежи налогов» - отрицательное влияние. (На этом этапе осуществляется построение когнитивной карты в виде ориентированного графа.)
- Определение силы влияния и взаимовлияния факторов (слабо, сильно). Например, увеличение (уменьшение) фактора «Уровень налогового бремени» «значительно» увеличивает (уменьшает) «Неплатежи налогов» (Окончательное построение когнитивной модели в виде функционального

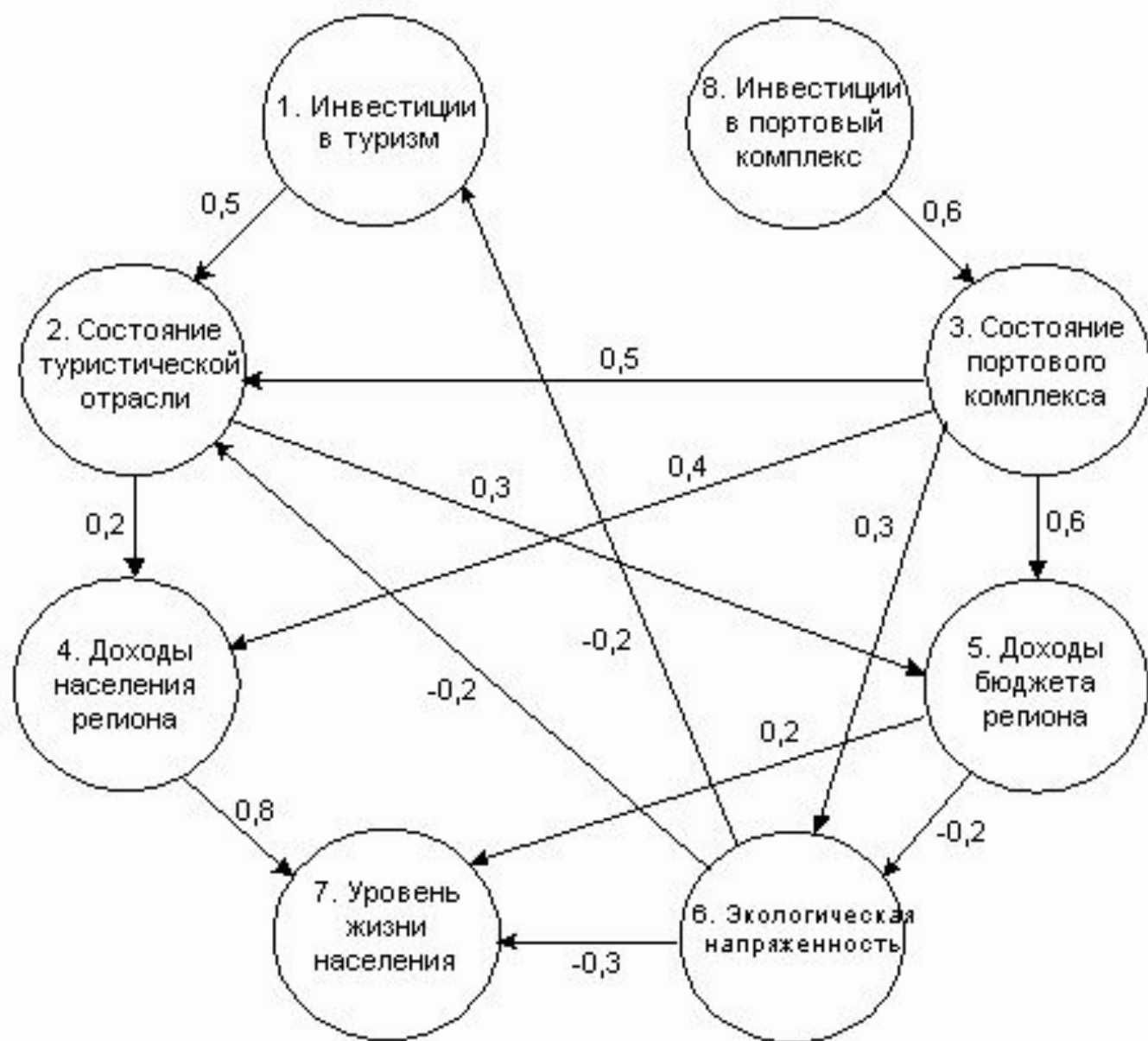
- Для того чтобы понять и проанализировать поведение сложной системы, строят структурную схему причинно-следственных связей элементов системы (факторов ситуации). Два элемента системы А и В, изображаются на схеме в виде отдельных точек (вершин), соединённых ориентированной дугой, если элемент А связан с элементом В причинно-следственной связью:  $A \rightarrow B$ , где: А - причина, В - следствие.
- Факторы могут влиять друг на друга, причем такое влияние, как уже указывалось, может быть положительным, когда увеличение (уменьшение) одного фактора приводит к увеличению (уменьшению) другого фактора, и отрицательным, когда увеличение (уменьшение) одного фактора приводит к уменьшению (увеличению) другого фактора. Причём, влияние может иметь и переменный знак в зависимости от возможных дополнительных условий.
- Подобные схемы представления причинно-следственных связей широко используются для анализа сложных систем в





Когнитивная карта отображает лишь факт наличия влияний факторов друг на друга. В ней не отражается ни детальный характер этих влияний, ни динамика изменения влияний в зависимости от изменения ситуации, ни временные изменения самих факторов. Учет всех этих обстоятельств требует перехода на следующий уровень структуризации информации, то есть к когнитивной модели.

На этом уровне каждая связь между факторами когнитивной карты раскрывается соответствующими зависимостями, каждая из которых может содержать как количественные (измеряемые) переменные, так и качественные (не измеряемые) переменные. При этом количественные переменные представляются естественным образом в виде их численных значений. Каждой же качественной переменной ставится в соответствие совокупность лингвистических переменных, отображающих различные состояния этой качественной переменной (например, покупательский спрос может быть «слабым», «умеренным», «ажиотажным» и т.п.), а каждой лингвистической переменной соответствует определяющий числовой эквивалент.



# Обучение и самообучение.

Включает модели, методы и алгоритмы, ориентированные на автоматическое накопление и формирование знаний с использованием процедур анализа и обобщения данных. К данному направлению относятся системы добычи данных (Data mining) и системы поиска закономерностей в компьютерных базах данных.

# Понятие Data Mining

- **Data Mining – это процесс поддержки принятия решений, основанный на поиске в данных скрытых закономерностей (шаблонов информации) .**
- **Технологию Data Mining достаточно точно определяет Григорий Пиатецкий-Шапиро (*Gregory Piatetsky-Shapiro*) – один из основателей этого направления:**

**Data Mining – это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.**

- **Неочевидных** – это значит, что найденные закономерности не обнаруживаются стандартными методами обработки информации или экспертным путем.
- **Объективных** – это значит, что обнаруженные закономерности будут полностью соответствовать действительности, в отличие от экспертного мнения, которое всегда является субъективным.
- **Практически полезных** – это значит, что выводы имеют конкретное значение, которому можно найти практическое применение.

- Традиционные методы анализа данных (статистические методы) и OLAP в основном ориентированы на проверку заранее сформулированных гипотез (verification-driven data mining) и на «грубый» разведочный анализ, составляющий основу оперативной аналитической обработки данных (online analytical processing, OLAP), в то время как одно из основных положений Data Mining – поиск **неочевидных закономерностей**.
- Инструменты Data Mining могут находить такие закономерности самостоятельно и также **самостоятельно строить гипотезы о взаимосвязях**.

# Мультидисциплинарность





# Задачи Data Mining

- Классификация
- Кластеризация
- Прогнозирование
- Ассоциация
- Визуализация
- анализ и обнаружение отклонений
- Оценивание
- Анализ связей
- Подведение итогов

# Методы Data Mining. Технологические методы.

- Непосредственное использование данных, или сохранение данных:  
кластерный анализ, метод ближайшего соседа, метод k-ближайшего соседа, рассуждение по аналогии
- Выявление и использование формализованных закономерностей, или дистилляция шаблонов:  
логические методы; методы визуализации; методы кросс-табуляции; методы, основанные на уравнениях

# Методы Data Mining. Статистические методы.

- Дескриптивный анализ и описание исходных данных.
- Анализ связей (корреляционный и регрессионный анализ, факторный анализ, дисперсионный анализ).
- Многомерный статистический анализ (компонентный анализ, дискриминантный анализ, многомерный регрессионный анализ, канонические корреляции и др.).
- Анализ временных рядов (динамические модели и прогнозирование).

# Методы Data Mining. Кибернетические методы.

- Искусственные нейронные сети (распознавание, кластеризация, прогноз);
- Эволюционное программирование (в т.ч. алгоритмы метода группового учета аргументов);
- Генетические алгоритмы (оптимизация);
- Ассоциативная память (поиск аналогов, прототипов);
- Нечеткая логика;
- Деревья решений; **этот метод будет рассмотрен подробнее.**
- Системы обработки экспертных знаний.

# Визуализация инструментов Data Mining.

- Для деревьев решений - визуализатор дерева решений, список правил, таблица сопряженности.
- Для нейронных сетей - в зависимости от инструмента это может быть топология сети, график изменения величины ошибки, демонстрирующий процесс обучения.
- Для карт Кохонена: карты входов, выходов, другие специфические карты.
- Для линейной регрессии - линия регрессии.
- Для кластеризации: дендрограммы, диаграммы рассеивания.

# Проблемы и вопросы

- Data Mining не может заменить аналитика!
- Сложность разработки и эксплуатации приложения Data Mining. Основные аспекты:
  - Квалификация пользователя
  - Сложность подготовки данных
  - Большой процент ложных, недостоверных или бессмысленных результатов
  - Высокая стоимость
  - Наличие достаточного количества репрезентативных данных

# Области применения Data mining

- ***Database marketers*** - Рыночная сегментация, идентификация целевых групп, построение профиля клиента
- ***Банковское дело*** - Анализ кредитных рисков, привлечение и удержание клиентов, управление ресурсами
- ***Кредитные компании*** - Детекция подлогов, формирование "типичного поведения" обладателя кредитки, анализ достоверности клиентских счетов, cross-selling программы
- ***Страховые компании*** - Привлечение и удержание клиентов, прогнозирование финансовых показателей
- ***Розничная торговля*** - Анализ деятельности торговых точек, построение профиля покупателя, управление ресурсами

# Области применения Data mining.

## Продолжение.

- **Телекоммуникация и энергетика** - Привлечение клиентов, ценовая политика, анализ отказов, предсказание пиковых нагрузок, прогнозирование поступления средств
- **Налоговые службы и аудиторы** - Детекция подлогов, прогнозирование поступлений в бюджет
- **Фармацевтические компании** - Предсказание результатов будущего тестирования препаратов, программы испытания
- **Медицина** - Диагностика, выбор лечебных воздействий, прогнозирование исхода хирургического вмешательства
- **Управление производством** - Контроль качества, материально-техническое обеспечение, оптимизация технологического процесса
- **Ученые и инженеры** - Построение эмпирических моделей, основанных на анализе данных, решение научно-технических задач



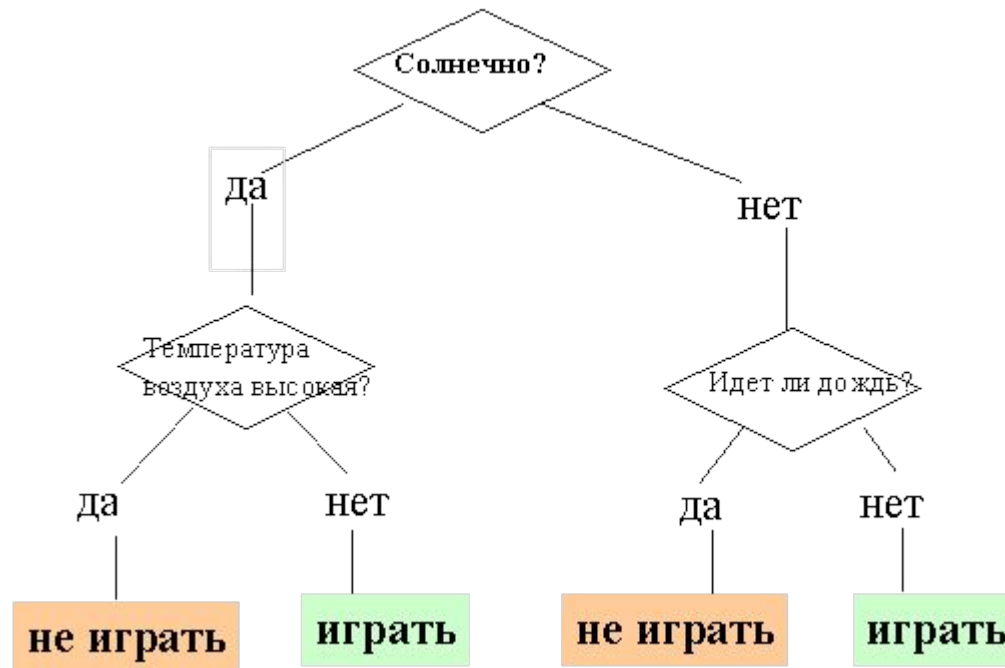
# Перспективы технологии Data Mining.

- выделение типов предметных областей с соответствующими им эвристиками
- создание формальных языков и логических средств, с помощью которых будет формализованы рассуждения
- создание методов Data Mining, способных не только извлекать из данных закономерности, но и формировать некие теории, опирающиеся на эмпирические данные;
- преодоление существенного отставания возможностей инструментальных средств Data Mining от теоретических достижений в этой области.

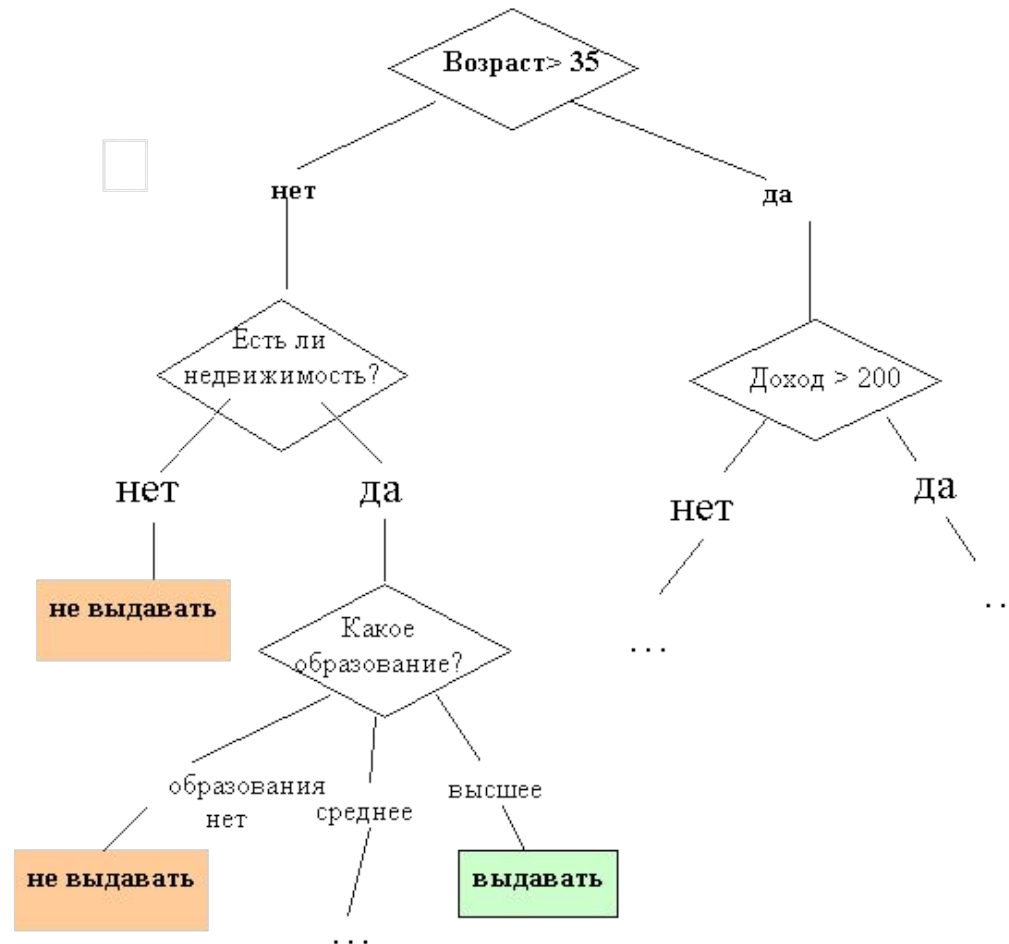
# Деревья решений. История и основные понятия.

- Возникновение - 50-е годы (Ховиленд и Хант (Noveland, Hunt) )
- Метод также называют деревьями решающих правил, деревьями классификации и регрессии
- *Это способ представления правил в иерархической, последовательной структуре*

# Деревья решений. Пример 1.



# Деревья решений. Пример 2.

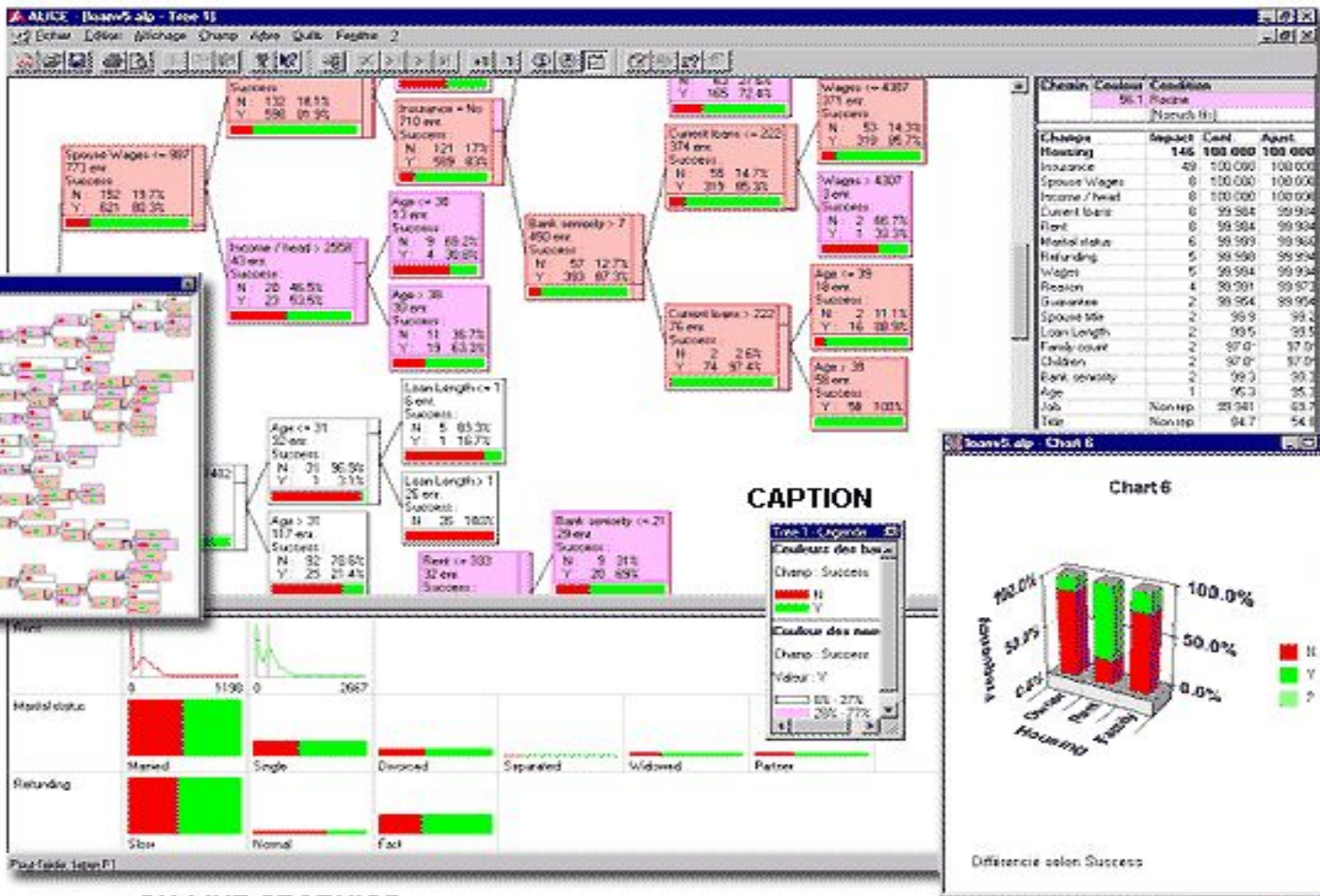


# Деревья решений. Преимущества метода.

- **Интуитивность деревьев решений**
- **Возможность извлекать правила из базы данных на естественном языке**
- **Не требует от пользователя выбора входных атрибутов**
- **Точность моделей**
- **Разработан ряд масштабируемых алгоритмов**
- **Быстрый процесс обучения**
- **Обработка пропущенных значений**
- **Работа и с числовыми, и с категориальными типами данных**

# Data mining

## DECISION TREE



3D CHART

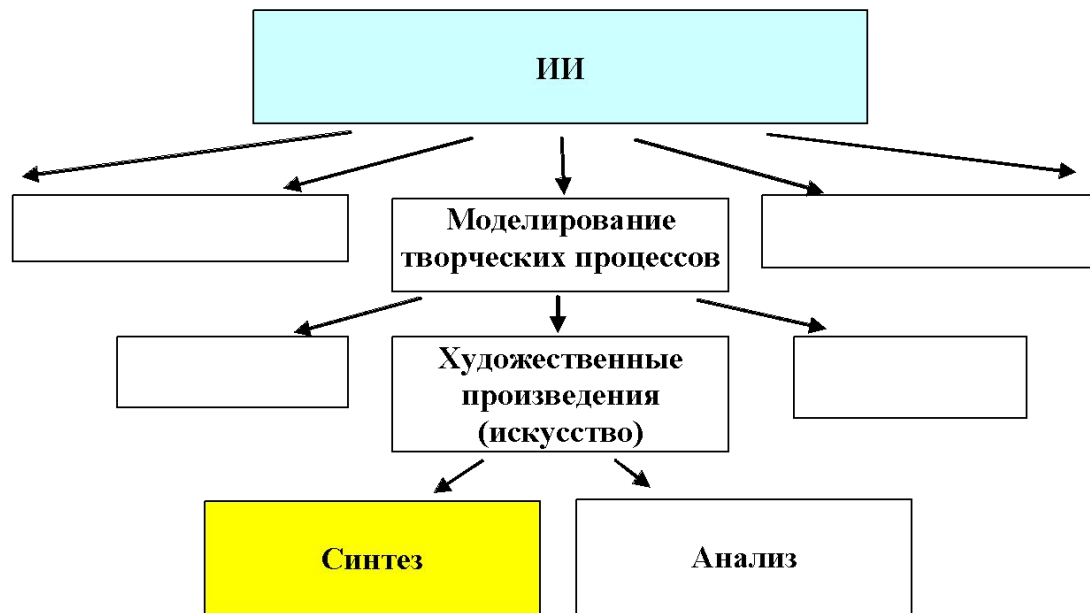
# Игры и машинное творчество.

Охватывает создание компьютерной музыки, стихов, интеллектуальные системы для изобретения новых объектов, создание интеллектуальных компьютерных игр

- В середине 50-х годов в США (Л.Хиллер и Л.Айзексон), а несколько позже в СССР (Р.Х.Зарипов) были сделаны первые попытки сочинения музыкальных произведений с помощью программ, реализуемых на ЭВМ. В основе этих программ лежала идея об использовании генераторов случайных чисел, интерпретируемых как нотные знаки со всеми присущими им параметрами, для порождения музыкального произведения за счет отбора из генерируемого потока нот лишь тех, которые удовлетворяли бы определенным правилам.
- Эти правила заимствовались из музыковедческой литературы и отражали специфику восприятия музыки человеком. Таким образом, в этих программах, как и в некоторых программах автоматизированного реферирования или доказательства теорем, использовался случайный процесс, детерминируемый системой ограничивающих модельных правил



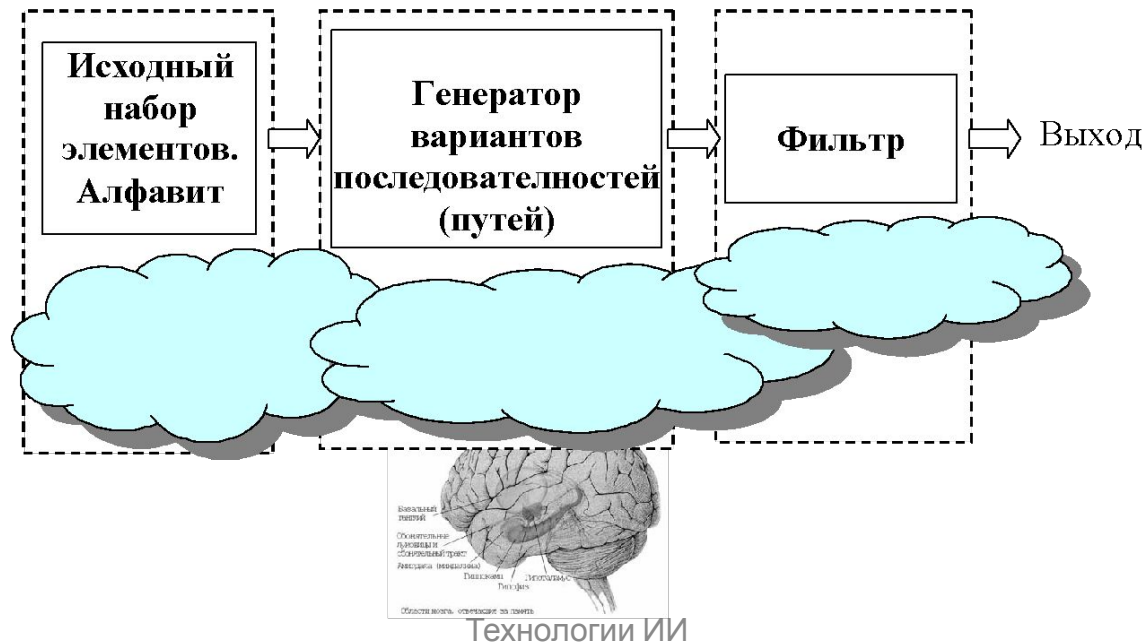
# Машинное творчество и ИИ



- Начальный период развития ИИ: машинное творчество – одно из основных направлений ИИ
- Период зрелости: потеря интереса - синтез художественных произведений свелся к внешней имитации творческой деятельности
- Сегодня: рост интереса к проблематике

# Общая структура творческого процесса

- Моделирование творческих процессов и лабиринтная модель мышления:  
Гипотеза: *решение любой творческой задачи есть по своей сути поиск в некотором лабиринте возможностей.*
- В.Н. Пушкин: Творческий процесс есть суть соотнесение структурированных описаний двух ситуаций: начальной и целевой. Суть творчества в построении пути не в готовом, заранее заданном лабиринте, а в формировании той его части, в которой этот путь существует.

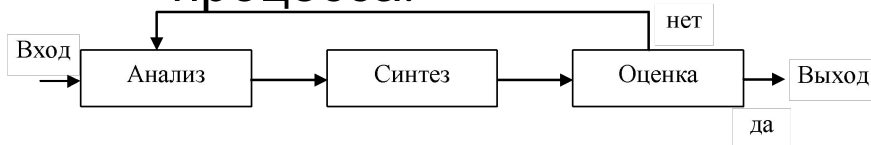


# Музыка

Моцарт. «Инструкция по сочинению вальсов с помощью двух игральных костей без малейшего знания музыки композиции».

- Специальные таблицы - «Таблица цифр», определяющая выбор очередного такта создаваемой пьесы и «Таблица музыки».

Р.Х.Зарипов. Машинный поиск вариантов при моделировании творческого процесса.



**INSTRUCTION**  
pour composer des Valses par le moyen de 2 Dèz  
sans avoir la moindre connaissance  
de la Musique ou de la Composition

„Таблица цифр“

|    | A   | B   | C   | D   | E   | F   | G   | H   |
|----|-----|-----|-----|-----|-----|-----|-----|-----|
| 2  | 96  | 22  | 141 | 41  | 105 | 122 | 11  | 30  |
| 3  | 32  | 8   | 128 | 83  | 146 | 46  | 134 | 81  |
| 4  | 69  | 95  | 158 | 13  | 153 | 56  | 110 | 24  |
| 5  | 40  | 17  | 113 | 85  | 161 | 2   | 159 | 100 |
| 6  | 148 | 74  | 163 | 45  | 80  | 97  | 36  | 107 |
| 7  | 104 | 157 | 27  | 167 | 184 | 68  | 118 | 91  |
| 8  | 152 | 90  | 171 | 93  | 99  | 133 | 21  | 127 |
| 9  | 119 | 84  | 114 | 50  | 140 | 86  | 169 | 94  |
| 10 | 98  | 142 | 42  | 156 | 75  | 129 | 62  | 123 |
| 11 | 3   | 87  | 165 | 81  | 135 | 47  | 147 | 33  |
| 12 | 54  | 130 | 10  | 103 | 28  | 37  | 106 | 5   |

1. Les Lettres A-H qui sont placées au dessus des 8 Colonnes des Tables de nombres, montrent les 8 Mesures de chaque partie du Walzer. Par Exemple: A. la première, B. la seconde, C. la troisième, etc et les nombres dans la Colonnes dessous les lettres démontrent le nombre de la mesure, dans les notes.

2. Les nombres de 2 jusqu'à 12 montrent la somme du nombre qu'on peut jeter.

3. On jette donc par exemple pour la première Mesure de la première partie du Walzer avec deux dèz 6 et cherche près du nombre 6 dans la Colonne A. le nombre de la mesure 148 dans la Musique. L'on met cette mesure sur le papier et voila ce qui fait le commencement du Walzer. Après cela on jette pour la seconde Mesure p. e. 9. on cherche près de 9 sous B. et on trouve N°. 81 de la table de musique. L'on met cette mesure a côté de la première et l'on continue ainsi jusqu'après avoir jeté les dèz huit fois, et alors on a achevé la première partie du Walzer: ensuite on fait le signe de répétition et commence 2e partie. Veut on avoir un Walzer plus long. on recommence de la même manière, et ainsi cela va a l'infini.

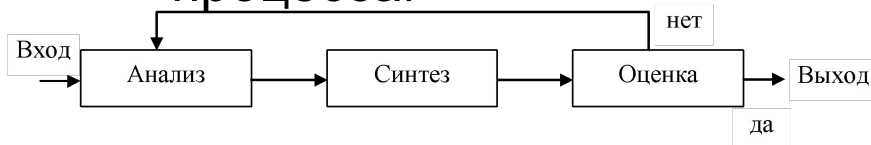
„ТАБЛИЦА МУЗЫКИ“

# Музыка

Моцарт. «Инструкция по сочинению вальсов с помощью двух игральных костей без малейшего знания музыки композиции».

- Специальные таблицы - «Таблица цифр», определяющая выбор очередного такта создаваемой пьесы и «Таблица музыки».

Р.Х.Зарипов. Машинный поиск вариантов при моделировании творческого процесса.



**INSTRUCTION**  
pour composer des Valses par le moyen de 2 Dèz sans avoir la moindre connaissance de la Musique ou de la Composition

„Таблица цифр“

|    | A   | B   | C   | D   | E   | F   | G   | H   |
|----|-----|-----|-----|-----|-----|-----|-----|-----|
| 2  | 96  | 22  | 141 | 41  | 105 | 122 | 11  | 30  |
| 3  | 32  | 8   | 128 | 83  | 146 | 46  | 134 | 81  |
| 4  | 69  | 95  | 158 | 13  | 153 | 96  | 110 | 24  |
| 5  | 40  | 17  | 113 | 85  | 161 | 2   | 159 | 100 |
| 6  | 148 | 74  | 163 | 45  | 80  | 97  | 36  | 107 |
| 7  | 104 | 157 | 27  | 167 | 184 | 68  | 118 | 91  |
| 8  | 152 | 90  | 171 | 93  | 99  | 133 | 21  | 127 |
| 9  | 119 | 84  | 114 | 50  | 140 | 86  | 199 | 94  |
| 10 | 98  | 142 | 42  | 156 | 75  | 129 | 62  | 123 |
| 11 | 3   | 87  | 165 | 81  | 135 | 47  | 147 | 33  |
| 12 | 54  | 130 | 10  | 103 | 28  | 37  | 106 | 5   |

1. Les Lettres A-H qui sont placées au dessus des 8 Colonnes des Tables de nombres, montrent les 8 Mesures de chaque partie du Walzer. Par Exemple: A. la première, B. la seconde, C. la troisième, etc et les nombres dans la Colonnes dessous les lettres démontrent le nombre de la mesure, dans les notes.

2. Les nombres de 2 jusqu'à 12 montrent la somme du nombre qu'on peut jeter.

3. On jette donc par exemple pour la première Mesure de la première partie du Walzer avec deux dèz 6 et cherche près du nombre 6 dans la Colonne A. le nombre de la mesure 148 dans la Musique. L'on met cette mesure sur le papier et voila ce qui fait le commencement du Walzer. Après cela on jette pour la seconde Mesure p. e. 9. on cherche près de 9 sous B. et on trouve N°. 81 de la table de musique. L'on met cette mesure a côté de la première et l'on continue ainsi jusqu'après avoir jeté les dèz huit fois, et alors on a achevé la première partie du Walzer: ensuite on fait le signe de répétition et commence 2e partie. Veut on avoir un Walzer plus long. on recommence de la même manière, et ainsi cela va à l'infini.

„ТАБЛИЦА МУЗЫКИ“

# Компьютерные игры

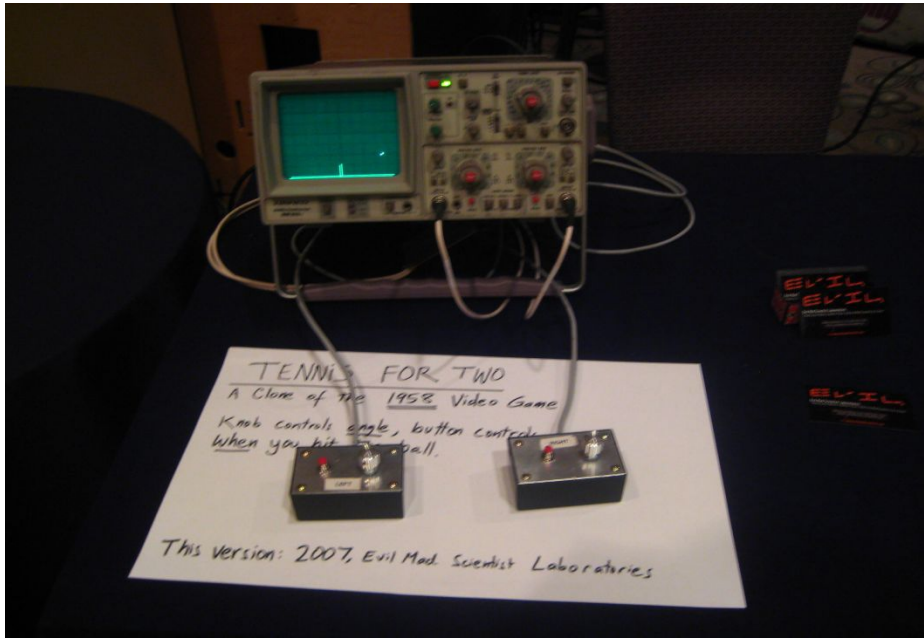
- История компьютерных игр началась в далеком 1947 году, когда Томас Т. Голдсмит-младший и Эстл Рей Манн, создали ракетный симулятор, который представлял устройство на электронно-лучевой трубке, это была первая из известных электронных интерактивных игр. Были использованы не цифровые, а аналоговые цепи, для положения точек на экране и контроля ЭЛТ пучка. По виду он напоминал аппарат(радар) времён Второй мировой войны. Наложения на экран были использованы для прицеливания, однако графика была крайне слаба для того времени. В массы аппарат так и не поступил.



- В 1948 году Алан Тьюринг и Д.Г. Чемпернаут написали алгоритм шахматной игры. Для запуска алгоритма, было недостаточно мощности компьютерного оборудования, того времени, алгоритм выиграл и проиграл один раз. Вскоре в марте 1950 года Клод Шеннон разработал шахматную программу, которая вышла в свет в статье «Программирование шахматных игр для компьютера», она была опубликованна в Philosophical Magazine. Статья затрагивавада проблему компьютерных

- В отрезке времени с 1951 по 1960 гг. изобретения в направлении компьютерных игр приписывают тройке лиц, а именно, Ральфу Баеру, инженеру, подавшему идею интерактивного телевидения. А. С. Дугласу, написавшему в 1952 крестики-нолики на компьютере (OXO) и Уильяму Хигинботаму, который в 1958 игру создал игру «Tennis for Two».



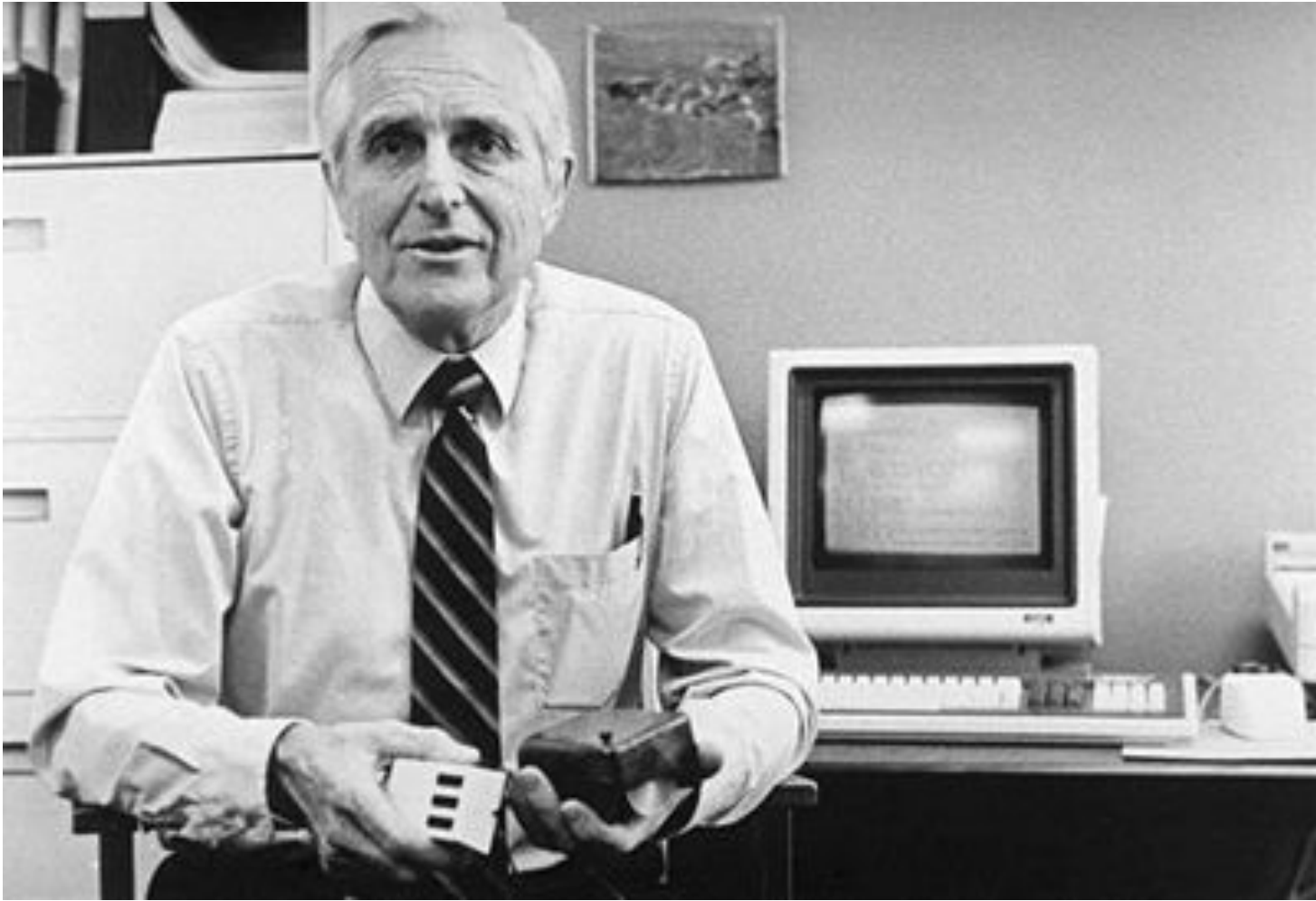


- Апрель 1962, одна из первых известных цифровых компьютерных игр Spacewar! Была очень популярной игрой в 1960-е годы и была портирована на большинство платформ DEC, таких как PDP-10 или PDP-11, или различные CDC-машины. Ранние системы микроЭВМ также поддерживают Spacewar!. Существовала версия «The Cromemco Dazzler», для ECD Micromind. отсутствовал дисплей с высоким растровым разрешением, из-за высокой стоимости памяти



Copyright (c) 2007 Nik Clayton

- В 1970 году Дуглас Энгельбарт запатентовал «систему X-Y индикации на мониторе»( Компьютерная мышь)
- С 1971 года разработка компьютерных игр ведётся очень бурно.
- Ральфу Баеру 22 марта был выдан 1-ый патент на «обучающую и телевизионную игровую аппаратуру» в Патентном бюро США.



- 1972 год
- Грегори Йоб написал (Охота на Вампуса) «*Hunt The Wumpus*» — первую текстовую игру в жанре квест, написана на языке BASIC для мейнфреймов.
- 29 ноября Atari выпускает первую аркаду — Pong.
- 24 мая компания Magnavox на съезде в Барлинггейме, представила игровую приставку Odyssey и начинает её продажу через

Hunt the Wumpus

Original BASIC version (1972) by Gregory Yob.

Inform port (1999) by Magnus Olsson <rebulon@pobox.com>.

Release 1 / Serial number 991216 / Inform v6.21

Type 1 to read the instructions, 2 to read the implementation of  
the game, or 4 to quit.

3

You can choose between the following caves:

1: The Dodecahedron

2: The Möbius Strip

3: The String of Beads

4: The Dendrite

5: The One-way Lattice

Which cave (1-5)? 1

OK, using the Dodecahedron

Bats nearby!

You are in room 1

Tunnels lead to 2 3 5

Shoot, Move or Quit (S-M-Q)?

- 1973 год
- Atari выпускает Gotcha! — игру в жанре лабиринт для аркадных автоматов.
- Mazewar разработана для миникомпьютера Imlac PDS-1, пожалуй, это самый первый шутер от первого лица и один из ранних примеров сетевой игры.
- 19 марта Кагемаса Козуки, владелец Konami, занимающейся производством и ремонтом музыкальных автоматов, начинает производство аркадных автоматов.





- Mattel выпускает Missile Attack, первую портативную игру на ЖК-дисплее.
- Cinematronics выпускает Space Wars — первый игровой автомат с векторной графикой.
- Nintendo дарит миру Color TV Game 6, из шести вариаций консоли с игрой Light Tennis (клон Pong). Mitsubishi, партнер Nintendo, производит большую часть системных компонентов.
- RCA Corporation выпускает в январе игровую приставку RCA Studio II.
- В октябре Atari выпускает в продажу игровую приставку Video Computer System, более известную, как Atari 2600

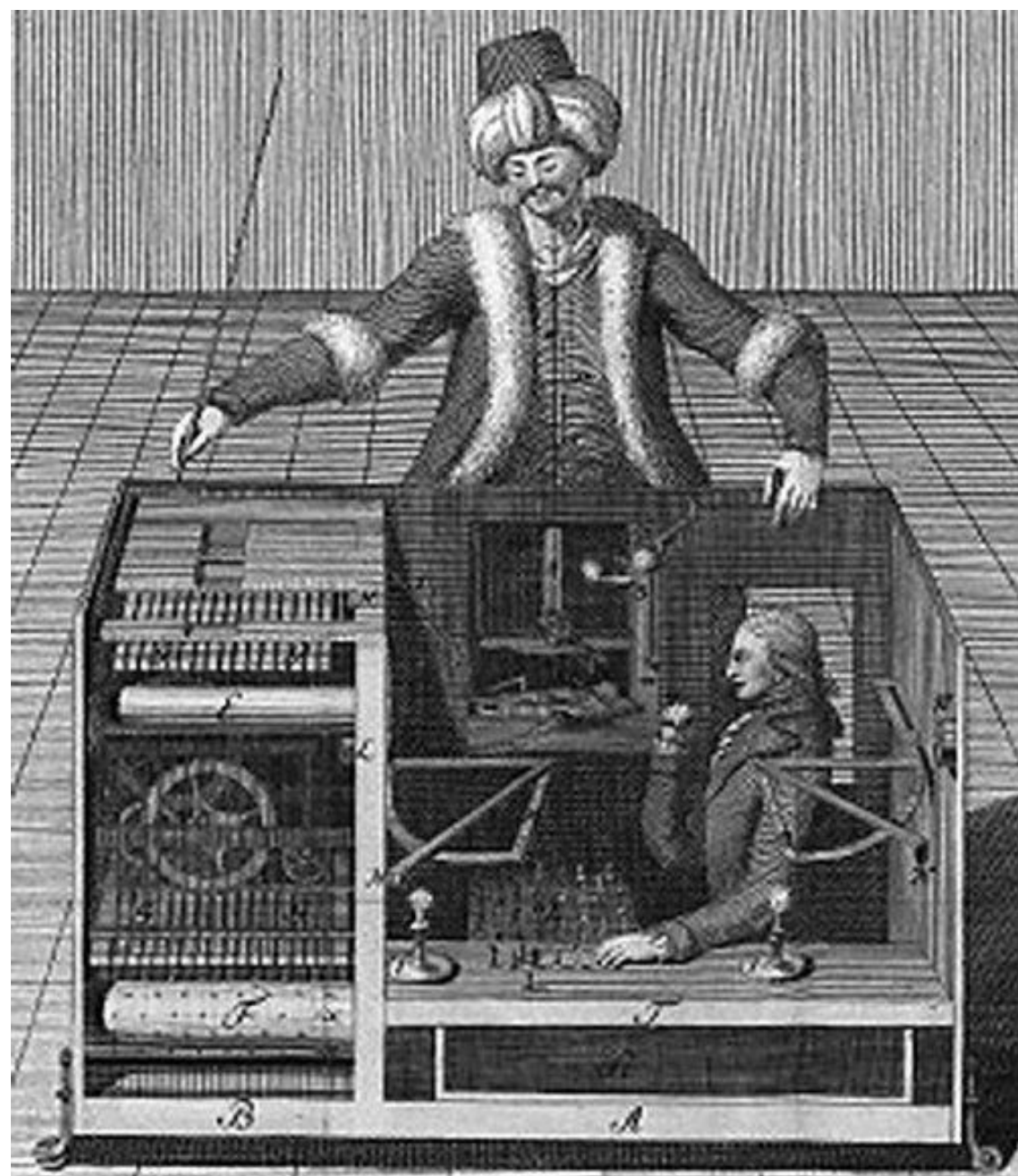


[www.atari.com](http://www.atari.com)



- Всё что происходит далее, похоже на гонку компаний, их перепродажу и появление более крупных и мощных организаций, это так называемая битва реальности, битва разработок, появление новых игровых консолей и видеоигр, новые движки и усиление графики всё больше приближают нас к качественному восприятию и удобству управления.

- История шахматных машин старше, чем история компьютеров. Идея создать машину, играющую в шахматы, датируется ещё восемнадцатым веком. Около 1769 года появился шахматный автомат «Механический турок». Он был предназначен для развлечения королевы Марии-Терезии. Машина действительно неплохо играла — внутри неё находился сильный шахматист, который и делал ходы.



- Создание механических шахматных автоматов прекратилось с появлением цифровых компьютеров в середине XX века. В 1951 году Алан Тьюринг написал алгоритм, с помощью которого машина могла бы играть в шахматы, только в роли машины выступал сам изобретатель. Этот нонсенс даже получил название — «бумажная машина Тьюринга». Человеку требовалось более получаса, чтобы сделать один ход. Алгоритм был довольно условный, и сохранилась даже запись партии, где «бумажная машина» Тьюринга проиграла одному из его коллег. За отсутствием доступа к компьютеру, программа ни разу не проверялась в работе.

- Примерно в это же время, в 1951 году, математик Клод Шеннон написал свою первую статью о шахматном программировании. Он писал: «Хотя, возможно, это и не имеет никакого практического значения, сам вопрос представляется теоретически интересным, и будем надеяться, что решение этой задачи послужит толчком для решения других задач аналогичной природы и большего значения». Шеннон также отметил теоретическое существование лучшего хода в шахматах и практическую невозможность его найти.



Следующим шагом в развитии шахматного программирования стала разработка в ядерной лаборатории Лос-Аламоса в 1952 году на компьютере Maniac 1 (тактовая частота 11 кГц) шахматной программы для игры на доске 6x6, без участия слонов. Известно, что этот компьютер сыграл одну партию против сильного шахматиста, она продолжалась 10 часов и закончилась победой шахматиста. Ещё одна партия была сыграна против девушки, которая недавно научилась играть в шахматы. Машина победила на 23-м ходу. Сейчас это выглядит смешно, но для своего времени это было большое достижение.

В 1957 году Алексом Бернстейном была создана первая программа для игры на стандартной шахматной доске и при участии всех фигур

- Важное событие для компьютерных шахмат произошло в 1958 году, когда Аллен Ньюэлл, Клифф Шоу и Герберт Саймон разработали алгоритм уменьшения дерева поиска, названный Альфа-бета отсечение, на основе которого построены функции поиска всех сильных современных программ.
- Первой же машиной, которая достигла уровня шахматного мастера, была Belle, законченная в 1983 г. Джо Кондоном и Кеном Томпсоном. «Belle» был первым компьютером, спроектированным только для игры в шахматы. Его официальный рейтинг Эло был 2250, таким образом, это была самая сильная шахматная машина своего времени.

- В 1994 Гарри Каспаров проиграл программе Fritz 3 турнирную блиц-партию в Мюнхене. Программа также выиграла у Вишванатана Ананда, Бориса Гельфанд и Владимира Крамника. Гроссмейстер Роберт Хюбнер отказывался играть против программы и автоматически проиграл. Каспаров сыграл второй матч с Fritz и победил с 4 выигрышами и 2 ничьими.

- В феврале 1996 года Гарри Каспаров победил шахматный суперкомпьютер Deep Blue со счетом 4-2. Этот матч выдающийся тем, что первую партию выиграл Deep Blue, автоматически став первым компьютером, победившим чемпиона мира по шахматам в турнирных условиях. Deep Blue вычислял 50 миллиардов позиций каждые три минуты, в то время как Каспаров 10 позиций за это же время. В Deep Blue было 200 процессоров. С тех пор шахматные энтузиасты и компьютерные инженеры создали много шахматных машин и компьютерных программ.



- Компьютерные шахматные программы рассматривают шахматные ходы как игровое дерево. Теоретически, они должны оценивать все позиции, которые возникнут после всех возможных ходов, затем все возможные ходы после этих ходов и т. д. Каждый ход одного игрока называется «узел». Перебор ходов продолжается, пока программа не достигает максимальной глубины поиска или определяет, что достигнуто конечная позиция (например мат или пат). Уже на основании оценки позиции выбирает оптимальную стратегию. В каждой позиции количество возможных ходов игрока примерно равно 35. Для полного анализа четырёх ходов (по два хода каждого игрока) нужно исследовать около полутора миллиона возможностей, для шести — почти два миллиарда. Анализ на хода вперед — очень мало для хорошей игры.
- Программисты пытаются по-разному ограничить массу ходов, которые надо перебрать (обрезание дерева поиска — game tree pruning). Самым популярным является альфа-бета отсечение, в котором не рассматриваются позиции, имеющие меньшую оценку, чем уже оценённые.

Вторым распространенным методом является итерационное заглублиение. Сначала перебирается дерево игры до определенной глубины, после чего выделяется несколько лучших ходов. Затем программа оценивает эти ходы применительно к большей глубине, чтобы узнать больше об их последствиях. Эта операция повторяется до наилучшего с точки зрения программы хода. Такой подход позволяет быстро отбросить немалый процент неперспективных вариантов игры. Например, не имеет смысла исследовать, что произойдет, когда обменять ферзя на пешку, если в позиции есть лучшие ходы.

# **Системы когнитивной графики.**

Ориентированы на общение с пользователем ИИС посредством графических образов, которые генерируются в соответствии с изменениями параметров моделируемых или наблюдаемых процессов.

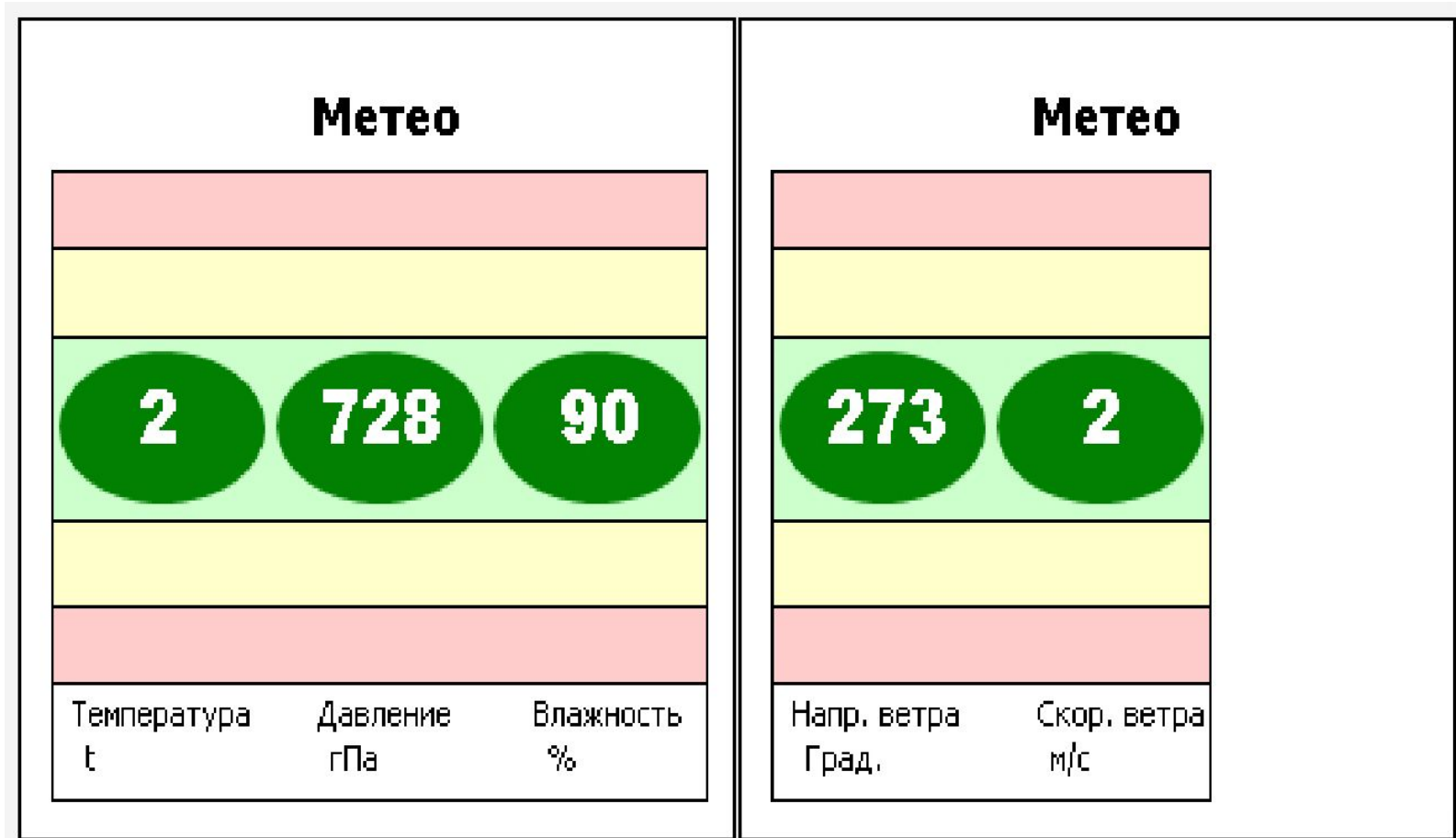


# Системы когнитивной графики.

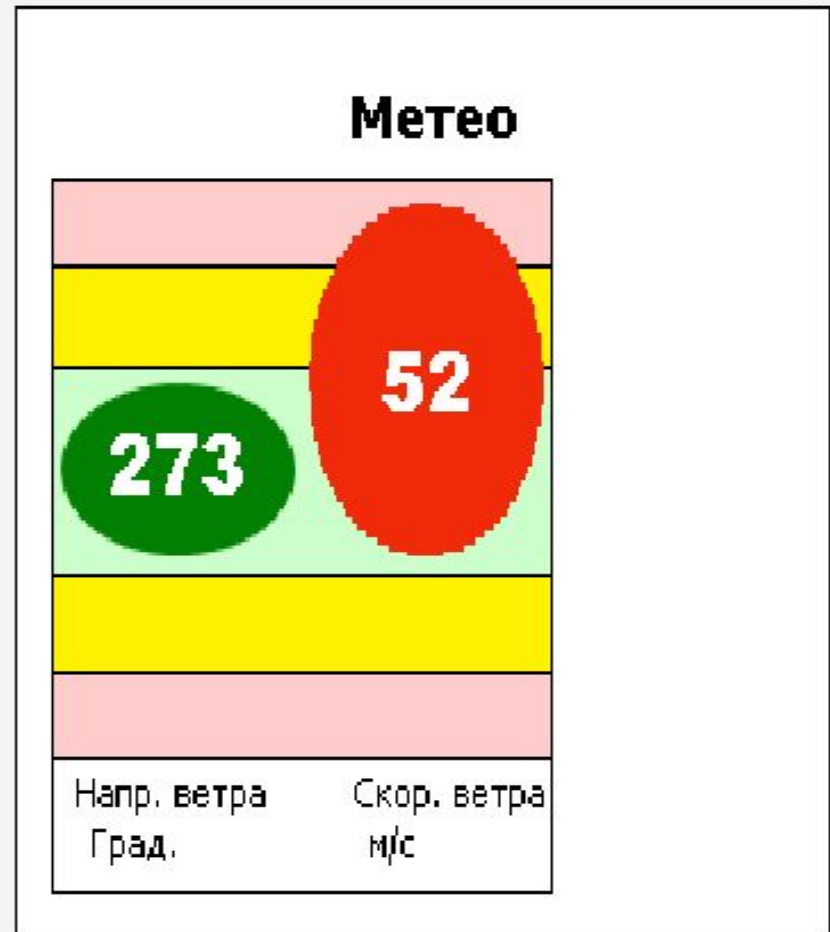
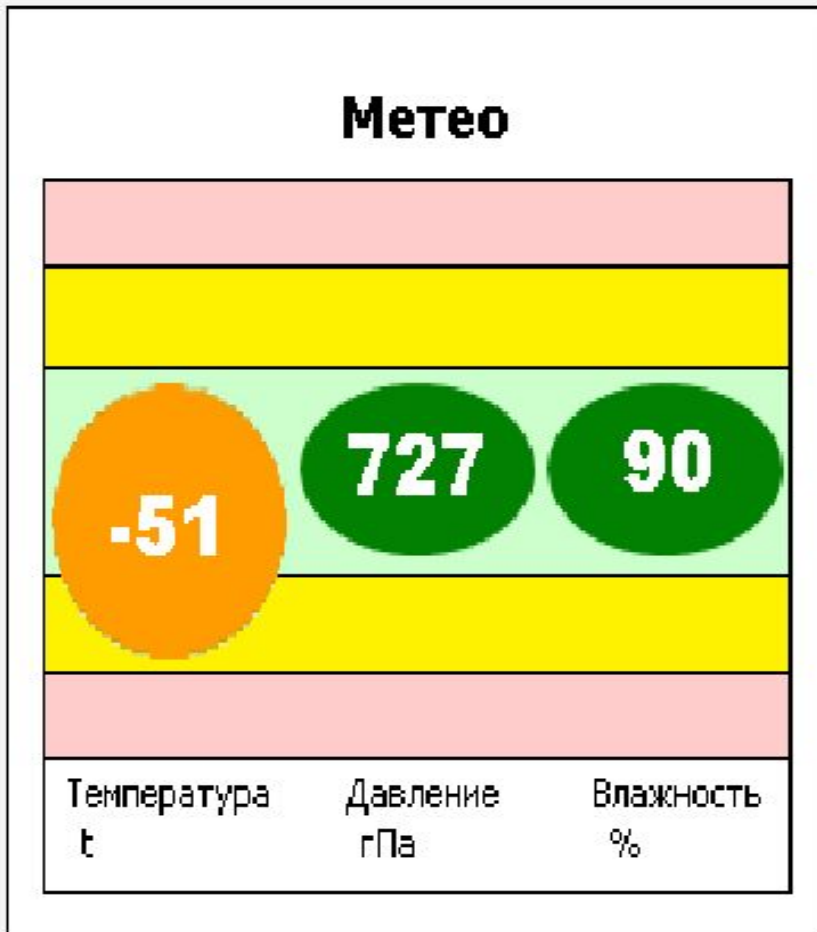
## Когнитивная графика

позволяет в наглядном и выразительном виде представить множество параметров, характеризующих изучаемое явление, освобождает пользователя от анализа тривиальных ситуаций, способствует быстрому освоению программных средств.

# Системы когнитивной графики.



# Системы когнитивной графики.



# Системы контекстной помощи.

В них пользователь описывает проблему, а система на основе дополнительного диалога конкретизирует ее и выполняет поиск относящихся к ситуации рекомендаций. В обычных гипертекстовых системах, наоборот, компьютерные приложения навязывают пользователю схему поиска требуемой информации.

# Программное обеспечение систем ИИ

- Языки программирования, ориентированные на обработку символьной информации (<LISP, SMALLTALK, РЕФАЛ),
- языки логического программирования (PROLOG), языки представления знаний (OPS 5, KRL, FRL),

# Программное обеспечение систем ИИ

- интегрированные- программные среды, содержащие арсенал инструментальных средств для создания систем ИИ (КБ, ARTS, GURU, G2),
- оболочки экспертных систем (BUILD, EXSYS Professional, ЭКСПЕРТ),

# Признаки ИИС

- *коммуникативные способности* — способ взаимодействия конечного пользователя с системой;
- *решение сложных плохо формализуемых задач*, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, характеризующейся неопределенностью и динамичностью исходных данных и знаний;

# Признаки ИИС

- *способность к самообучению* — умение системы автоматически извлекать знания из накопленного опыта и применять их для решения задач;
- *адаптивность* — способность системы к развитию в соответствии с объективными изменениями области знаний.



# Модели представления знаний

*Знания о некоторой ПрО* представляют собой совокупность сведений об объектах этой ПрО, их существенных свойствах и связывающих их отношениях, процессах, протекающих в данной ПрО, а также методах анализа возникающих в ней ситуаций и способах разрешения ассоциируемых с ними проблем.

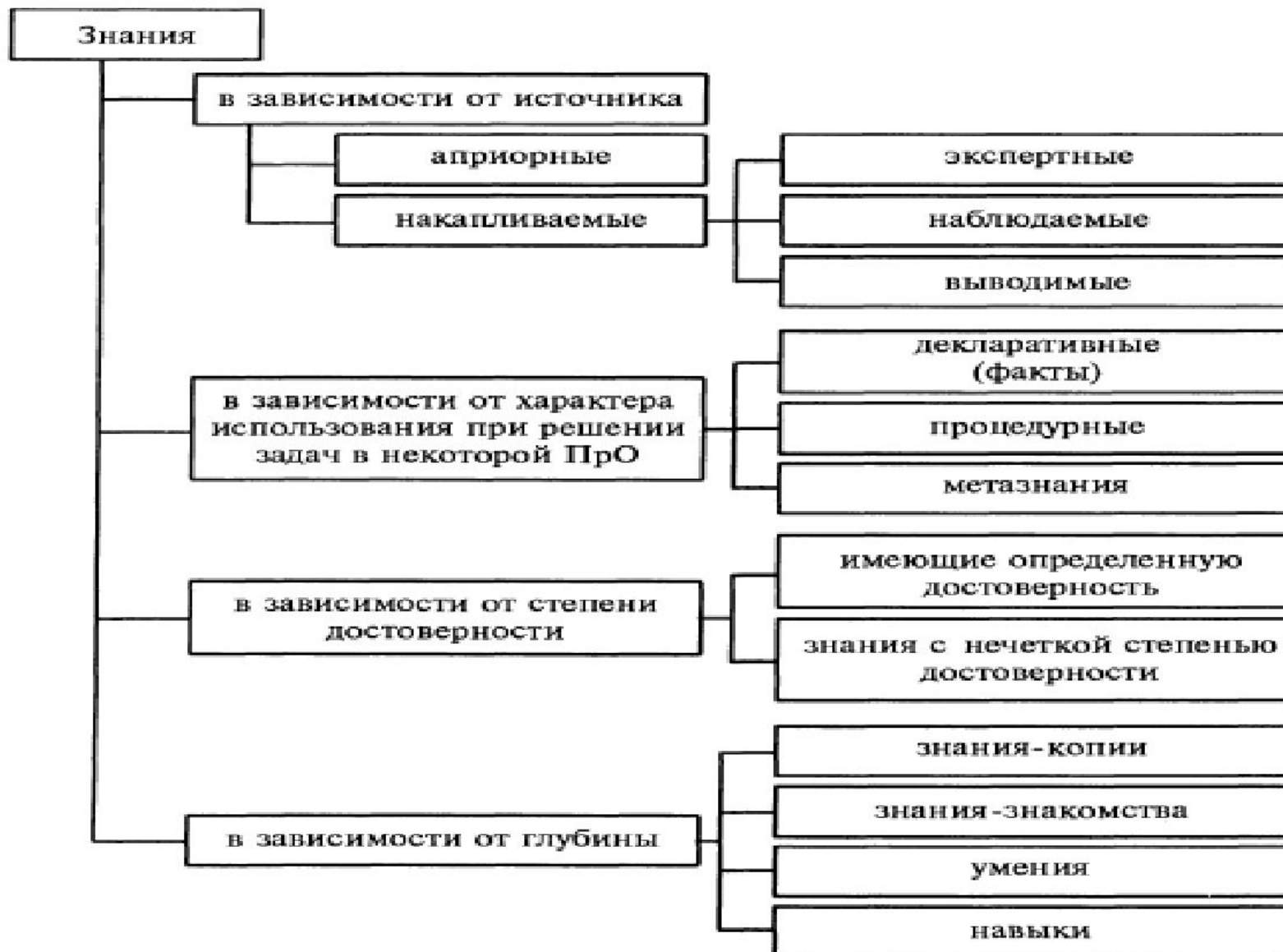
# Трактовки знаний

- *психологическая*: психические образы, мысленные модели;
- *интеллектуальная*: совокупность сведений о некоторой ПрО, включающих факты об объектах данной Про, свойствах этих объектов и связывающих их отношениях, описания процессов, протекающих в данной ПрО, а также информацию о способах решения типовых (в рамках этой ПрО) задач

# Трактовки знаний

- *формально-логическая*: формализованная информация о некоторой ПрО, используемая для получения (вывода) новых знаний об этой ПрО с помощью специализированных процедур;
- *информационно-технологическая*: структурированная информация, хранящаяся в памяти ЭВМ и используемая при работе интеллектуальных программ

# Классификация знаний



# Классификация знаний

- *декларативные знания*- факты, сведения описательного характера;
- *процедурные знания* - информация о способах решения типовых задач в некоторой ПрО);
- *метазнания* чаще всего определяются как «знания о знаниях» и содержат общие сведения о принципах использования знаний. К уровню метазнаний также относят стратегии управления выбором и применением процедурных знаний.

# Классификация знаний

- Знания, имеющих определенную степень достоверности: «Следующим днем календаря после 31 мая является 1 июня» и «Для кипячения воды при нормальном давлении требуется ее нагрев до 100 °С».
- Знания с нечеткой степенью достоверности: «Завтра в Москве будет дождь» и «При игре в шахматы не следует располагать коня на краю доски».

# Концептуальные свойства знаний

- 1) внутренняя интерпретация;
- 2) наличие внутренней структуры связей;
- 3) наличие внешней структуры связей;
- 4) шкалирование;
- 5) погружение в пространство с семантической метрикой;
- 6) наличие активности.

# Внутренняя интерпретация знаний

Позволяет соотнести данные, хранящиеся в памяти ЭВМ, с их смысловым содержанием. Например, пусть в оперативном запоминающем устройстве ЭВМ записано число «4». Очевидно, что этот факт сам по себе мало что говорит, так как непонятно, что конкретно обозначает число «4».



# Внутренняя интерпретация знаний

По иному обстоят дела, если информация представлена выражением: «Оценка студента Иванова на экзамене 4». Поскольку оценка на экзамене — целое число, не большее 5 и не меньшее 2, такое представление накладывает ограничения на данные, заносимые в поле оценки.

# Наличие внутренней и внешней структур связей

Основываются на структурном подходе к представлению ПрО, согласно которому в объекте ПрО могут быть выделены его части (элементы). Отношения между объектом-целым и его составляющими называются отношениями типа целое-часть (включение) и часть-целое (вхождение).

# Шкалирование знаний

Позволяет сопоставлять и упорядочивать качественно одинаковые, но различающиеся в количественном плане свойства и отношения объектов ПрО. Мера этого различия называется *интенсивностью* свойства или отношения.

# Шкалирование знаний

| Вид шкалы                                 | Параметры, задающие шкалу |                      |                   | Пример   | Схема |
|---|---------------------------|----------------------|-------------------|--|-------|
|   | направление               | точка начала отсчета | единичный отрезок |  |       |
| Метрическая абсолютная                    | +                         | +                    | +                 | Вчера днем в Москве температура составила 20 °С. Сегодня днем в Москве температура 22 °С |       |
| Порядковая с фиксированной точкой отсчета | +                         | +                    |                   | Вчера днем в Москве температура составила 20 °С. Сегодня днем в Москве теплее            |       |
| Метрическая относительная                 | +                         |                      | +                 | Сегодня днем в Москве температура на 2 °С выше, чем вчера                                |       |
| Порядковая                                | +                         |                      |                   | Сегодня днем в Москве теплее, чем вчера  |       |