**Вопросы к экзамену по дисциплине:**

**«Традиционные и интеллектуальные информационные технологии»**

**Летняя сессия 2022 г.**

1. **Дайте определение понятиям: Алгоритм, Алгоритмизация. Пример алгоритма. Свойства алгоритма.**

Алгоритм – это точный набор инструкций, описывающих порядок действий некоторого исполнителя для достижения результата, решения некоторой задачи за конечное время.

Процесс подготовки решения задачи на ЭВМ часто называют алгоритмизацией.

Общие свойства алгоритмов:

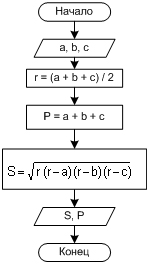
1. Дискретность. Алгоритм – процесс последовательного построения величин, идущих в дискретном времени таким образом, что в начальный момент задается исходная конечная система величин.
2. Детерминированность. Система величин, получаемых в какой-то не начальный момент времени, однозначно определяется системой величин, полученных в предыдущие моменты.
3. Элементарность шагов. Закон получения последующей системы величин из предыдущей должен быть простым и локальным.
4. Направленность. Если способ получения последующей величины из предыдущей не дает результата, должно быть указано, что считать результатом.
5. Массовость. Начальная система величин может выбираться из некоторого бесконечного множества. Иными словами алгоритм служит для решения целого класса задач.
6. Область применения. Наибольшая область начальных данных, на которой алгоритм результативен.
7. Понятность. Чтобы алгоритм можно было выполнить, он должен быть понятен исполнителю.
8. Конечность. Выполнение алгоритма заканчивается после выполнения конечного числа шагов.
9. Элементарность шагов. Закон получения последующей системы величин из предыдущей должен быть простым.
10. Эффективность. Каждый шаг алгоритма должен быть выполнен точно и за конечное время

**ПРИМЕР: Зная длины трех сторон треугольника, вычислить площадь и периметр треугольника.**

 Пусть a, b, c - длины сторон треугольника. Необходимо найти S - площадь треугольника. Для нахождения площади можно воспользоваться формулой Герона:

[https://sites.google.com/site/informvecole/_/rsrc/1467890432493/dla-ucenikov/9-klass/osnovy-algoritmizacii/5.jpg](https://sites.google.com/site/informvecole/dla-ucenikov/9-klass/osnovy-algoritmizacii/5.jpg?attredirects=0), где r - полупериметр, P - периметр.

 Входные данные: a, b, c. Выходные данные: S, P.

 Знак "=" означает не математическое равенство, а операцию присваивания. Переменной, стоящей слева от оператора, присваивается значение, указанное справа**.**

1. **Описание абстрактной машины Поста, система команд, пример программы.**

**Требования к машине:**

1. Она должна быть полностью детерминированной и действовать в соответствии с заданной системой правил.
2. Она должна допускать ввод начальных правил.
3. Заданная система правил и класс решаемых задач должны быть согласованны так, чтобы всегда можно было прочесть результат работы машины.

**Маши́на По́ста** — абстрактная машина Поста состоит из бесконечной ленты, разделенной на равные секции, а также считывающей и записывающей головки. В каждой секции м.б. ничего не записано либо записана метка, такая секция называется отличной. Информация о заполнении метками секции ленты характеризует состояние ленты. Состояние ленты меняется в процессе работы машины. Головка(знак «–» над секцией) может передвигаться вдоль ленты влево и вправо. В неподвижном состоянии она находится над одной из секций. За единицу времени она может сдвинуться влево или вправо на одну секции или остаться на месте. Работа машины Поста заключается в том, что головка передвигается вдоль ленты и печатает или стирает метки в соответствии с заданной программой, которая состоит из отдельных команд, а также распознает, имеется ли метка в секции.

Всего для машины Поста **существует шесть типов команд:**

1. xMy — поставить метку в обозреваемую секцию;
2. xCy — стереть метку в обозреваемой секции;
3. x← y — сдвинуться на одну секцию влево;
4. x→ y — сдвинуться на одну секцию вправо;
5. х → y1|y2 — условная передача управления (если метка отсутствует то управление предается команде y1, если присутствует то команде y2);
6. xCТОПx — конец программы(прекращение работы машины).

Общий вид: хКу – х-номер оператора, К-действие машины, у-номер следующего оператора.

Число(n) на ленте кодируется количеством меток, если n число, то меток подряд на ленте n+1.

Пример: Переход к первой метке(нужно передвинуть головку на первую метку):

I Вариант: Если знаем на каком расстоянии группа меток

**1 → 2**

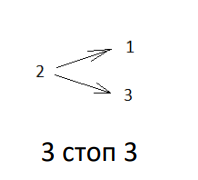
**2 → 3**

**3 → 4**

**4 Стоп 4**

II Вариант: Если не знаем расстояние до группы меток

**1 → 2**



1. **Средства описания алгоритмов: схема алгоритма. Привести пример.**

Схема алгоритма служит как формальный язык для записи содержания и логических связей задачи, а с другой – сравнительно легко преобразуется в программу для ЭВМ на любом ЯП.

Схемы алгоритмов – функционально ориентированные графические изображения, с помощью которых, используя текст и специальные символы, описывают последовательность шагов процесса во времени и в связи, рассматриваемой системы, с внешней средой.

- Краткое текстовое или формальное описание шагов приводится внутри символов

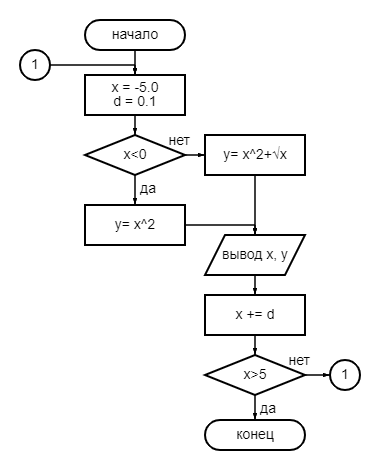
- Сведения о входных и выходных данных некоторых шагов процесса могут даваться в закодированной форме.

**Преимущества схем алгоритмов:**

1. Не требуется специальных знаний для понимания.
2. Для удобства чтения схем алгоритмов в ряде стран разработаны стандартные графические изображения операторов.

**Задание: построить схему алгоритма**

Протабулировать функцию для x в интервале [-5,5] с шагом 0.1.



1. **Дайте определения конечного автомата и автономного автомата.**

Время обычно считается непрерывным и предполагается изменяющимся от настоящего к будущему.

Однако при рассмотрении ряда устройств удобно вводить **дискретное представление времени**.

Пусть полуось времени (0, ∞) каким-то образом разбита на бесконечное число отрезков произвольной длины. Моменты времени 0, 1, 2, …, **i** назовём **тактами**.

**Текущий такт**, соответствующий настоящему моменту времени, обозначим через i, а предшествующие ему и последующие такты - соответственно через ( i - 1, i - 2, …) и (i+1, i+2, …).

Будем рассматривать функционирование в **дискретном времени** динамических системы, описываемых координатами, каждая из которых изменяется на конечном множестве ( С ).

Всякая динамическая система получается ряд внешних возмущений ( P ).Будем считать, что их число также конечно и каждое из возмущений задано на конечном множестве.

Динамические системы, удовлетворяющие вышеуказанными требования, назовём **конечными динамическими системами**.

Рассмотрим множества

{C} = {C1, C2, …, Ck};

{P} = {P1, P2, …, Pr},

Состоящие из соответственно k и r символом.

Набор **C - состояния динамической системы**, а набор **P - входами динамической системы**.

Время обычно считается непрерывным и предполагается изменяющимся от настоящего к будущему.

Однако при рассмотрении ряда устройств удобно вводить **дискретное представление времени**.

Пусть полуось времени (0, ∞) каким-то образом разбита на бесконечное число отрезков произвольной длины. Моменты времени 0, 1, 2, …, **i** назовём **тактами**.

**Текущий такт**, соответствующий настоящему моменту времени, обозначим через i, а предшествующие ему и последующие такты - соответственно через ( i - 1, i - 2, …) и (i+1, i+2, …).

Будем рассматривать функционирование в **дискретном времени** динамических системы, описываемых координатами, каждая из которых изменяется на конечном множестве (С).

Всякая динамическая система получается ряд внешних возмущений (P).Будем считать, что их число также конечно и каждое из возмущений задано на конечном множестве.

Динамические системы, удовлетворяющие вышеуказанными требования, назовём **конечными динамическими системами**.

Рассмотрим множества

{C} = {C1, C2, …, Ck};

{P} = {P1, P2, …, Pr},

Состоящие из соответственно k и r символом.

Набор **C - состояния динамической системы**, а набор **P - входами динамической системы**.

1. **Метасинтаксический язык Бэкуса.**

Метасинтаксический язык, предложенный Бэкусом и Науром, впервые использовался для описания синтаксиса реального языка программирования Алгол. Наряду с новыми обозначениями метасимволов, в нем использовались содержательные обозначения нетерминалов. Это сделало описание языка нагляднее и позволило в дальнейшем широко использовать данную нотацию для описания реальных языков программирования.

Язык Бэкуса – это метасинтаксический язык, который состоит из конечного числа предложений называемых металингвистическими формулами.

1) символ " ::= " отделяет левую часть правила от правой(читается как «по определению есть»);

2) символ | - читается ка «или»

Остальные метасимволы(нетерминалы) выбираются произвольно разработчиком формального языка и заключаются в угловые скобки "<" и ">";

Начало и конец формулы ничем не обозначены, поэтому принято делать отступы.

**Терминалы** - это символы, используемые в описываемом языке;

Каждое правило определяет порождение нескольких альтернативных цепочек, отделяемых друг от друга символом вертикальной черты "|".

Формы Бекуса-Наура используются с двумя основными целями:

1. они являются металингвистическим языком, стандартным для описания языков программирования;
2. они описывают правила построения текстов или конструкций.

Таким образом, множество терминальных символов, множество нетерминалов с выделенными начальными символами и множество правил вывода входят в формальное определение порождающей грамматики. Язык, порождаемый такими грамматиками, это множество терминальных цепочек, которые можно вывести из начального символа.

**Пример:**

**Формальный язык для представления чётных натуральных чисел с помощью формулы Бэкуса.**

<ненулевая цифра> ::= 1|2|3|4|5|6|7|8|9

<цифра> ::= 0 | <ненулевая цифра>

<чётная цифра> ::= 0|2|4|6|8

<начало> ::= <ненулевая цифра>|<начало><цифра>

<чётное число> ::=<чётная цифра>|<начало><чётная цифра>

1. **Нотации Бэкуса-Наура.**

Для описания формального языка нужен другой язык, описываемый язык называют **языком объекта**, а язык, используемый для описания, называется **мета(внешний) язык.**

Внешний язык должен обладать возможностями для описания как структуры, так и смысла предложения языка программирования.

**Мета(внешний) язык**

1. Мета семантический - описание смысла.
2. Мета синтаксический - описание структуры.

При описании структуры формального языка обычно указывают его алфавит и приводят правило для построения предложений.

Язык Бэкуса – это мета синтаксический язык, который состоит из конечного числа предложений, называемых **мета лингвистическими формулами**.

Для их построения используется **два мета символа**:

1. ::= (По определению есть)
2. | (Или)

Остальные мета символы выбираются произвольно разработчиком формального языка, заключаются в угловые скобки <>.

Начало и конец формулы ничем не обозначен, поэтому принято делать отступ.

**Пример 1:**

Формальный язык для представления чётных натуральных чисел с помощью формулы Бэкуса.

<ненулевая цифра> ::= 1|2|3|4|5|6|7|8|9

<цифра> ::= 0 | <ненулевая цифра>

<чётная цифра> ::= 0|2|4|6|8

<начало> ::= <ненулевая цифра>|<начало><цифра>

<чётное число> ::= <чётная цифра>|<начало><чётная цифра>

**Пример 2: Описание идентификатора на языке Бэкуса**

<буквы> ::= A|B|C…|a|b|c|z.

<цифра> ::= 0|1|2|3|4|5|6|7|8|9

<идентификатор> ::= <буква>|<идентификатор><буква>|<идентификатор> <цифра>

**Расширенные Бэкуса-Наура формы(РБНФ)**

**РБНФ** — формальная система определения синтаксиса, в которой одни синтаксические категории последовательно определяются через другие.

1. [ ] - заключённая в них конструкция может отсутствовать.
2. { } - заключённая конструкция может повторяться возможно 0 раз.
3. {/ /} - повторение 1 и более раз.
4. ( ) - используются для ограничения альтернативных конструкций.
5. Нетерминальный символ, состоящий из нескольких слов, пишется слитно. (Чётное\_число, ЧётноеЧисло).
6. Терминальные символы изображаются словами, заключёнными в кавычки. ( "if")
7. Синтаксические правила начинаются со знака $.
8. Левая часть отделяется от правой знаком =.

$буква = "A" | "B" | … | "a" | "b" | … | "z" .

$цифра = "0" | "1" | … | | "9".

$идентификатор = буква {буква | цифра}.

1. **Подходы к построению алгоритмических языков: Основные этапы.**

Алгоритмический язык является средством точного формулирования вычислительных процессов для их последующей реализации на вычислительных машинах.

ЯП🡪Библиотеки🡪Фреймворки   
(рост абстракций)

**3 основные задачи:**

1. Алгоритмы численного анализа (расчёты по формулам и т.п.)
2. Процессы обработки данных различных типов и структур (обработка таблиц)
3. Переработка символьной информации (обработка текстов, аналитические выкладки, моделирование интеллекта и т.п.)

**Этапы разработки системы программирования:**

1. Отбор изобразительных средств языка
2. Описание языка
3. Разработки транслятора

**Свойства "хорошего" языка**

1. **Ясность, простота и единообразие понятий.** ЯП обеспечивает как систему понятий для обдумывания алгоритмов, так и средства выражения этих алгоритмов.

Идея: минимальное количество понятий с простыми и систематизированными правилами комбинирования.

Акцент: Удобочитаемость.

1. **Естественность для приложений.** Язык должен иметь такой синтаксис, который при правильном использовании отражает в коде реализуемые алгоритмические структуры.

Различные алгоритмические структуры для последовательных, параллельных, логических алгоритмов.

1. **Ортогональность.** Любые возможные комбинации различных языковых конструкций будут осмысленными.  
   <выражение> и <условный оператор><выражение>  
   Преобразование типов.
2. **Поддержка абстракций.** Важной частью работы программиста является разработка конкретных абстракций для решения задачи и их реализация. В идеале, язык должен позволять определять структуры данных, типы данных и операции и поддерживать их как самодостаточные абстракции.
3. **Удобство верификации программы.** Методы верификации программы позволяют без выполнения проверить на наличие ошибок.   
   Основной фактор - простота семантики и синтаксических структур.

**Переносимость программ**  
Кроссплатформенные языки, Языки под различные аппаратные архитектуры.

1. **Использование формальных языков для поиска.**

**Информационный поиск**

**Информационно-логические задачи –** задачи обработки информации, библиографического поиска и анализа фактографических данных. Для них характерны логические обработки больших объемов информации и ее хранение.

**Дескрипторный метод поиска**

Содержание документа в общих чертах может быть представлена набором характерных для данного текста слов. Их называют **ключевыми словами**. Для построения поисковой системы из всего многообразия ключевых слов, выбранных из ряда характерных для данной тематики текстов составляется стандартный набор терминов без синонимов со строго фиксированными значениями. Эти термины называются **дескрипторами**, а полный набор таких терминов – **словарём дескрипторов.** Для каждого документа составляется свой набор дескрипторов, называемый **поисковым образом документа.** Массив дескрипторных наборов документа может быть построен двумя разными способами: **прямым** и **инверсным**.

**Прямой:** в память машины последовательно записываются номера документов и за каждым из них указываются дескрипторы или их коды. Процесс поиска заключается в последовательном сравнении для каждого документа всех дескрипторов запроса с дескрипторами документа и выделение тех документов, для которых выполняется критерий соответствия.

**Инверсный:** за основные позиции принимают не документы, а дескрипторы. Для каждого дескриптора записываются все номера документов, которые имеют в своих поисковых образах этот дескриптор. Поиск нужных документов осуществляется в словаре дескрипторов путём обращения к тем дескрипторам, которые имеются в запросе, и путём выбора всех номеров документов этих дескрипторов. Отобранными считаются те документы, номера которых оказались общими для всех дескрипторов, указанных в запросе.

1. **Дайте определение понятиям: Базы данных и Банки данных.**

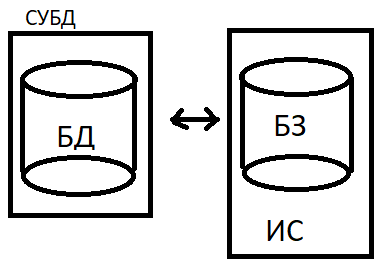
Идея создания БД применяется, когда для решения задач исходные данные и промежуточные результаты хранятся отдельно от исполняемых программ.

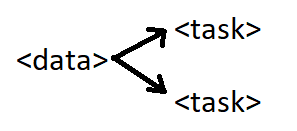
**БД** — совокупность взаимосвязанной информации, организованной по определенным правилам.

**Банк данных (БнД)** – это система специально организованных данных, предназначенных для централизованного накопления и коллективного многоцелевого использования.

Большинство сложных автоматизированных систем требуют не только использования специфических процессов для подготовки исходных данных, решения отдельных задач, но и обмена данными с другими задачами и поддержания информационного процесса для всей системы.

База данных должна быть независима от используемого ее ПО.



1. **Основные требования к Базам данных.**
2. Целостность БД, т.е. требования полноты и непротиворечивости данных.
3. Многократное использование данных.  
   ****
4. Уменьшение избыточности данных.
5. Быстрый поиск и получение информации по запросам пользователей.
6. Простота обновления данных(поддержка актуальности БД).
7. Защита данных от несанкционированного доступа, искажение и уничтожение(должны быть бэкапы БД).
8. **Этапы проектирования БД: концептуальная модель.**

**Этапы проектирования БД:**

1. **Концептуальное проектирование**(идея) – происходит сбор, анализ и редактирование требований к данным. В результате создаётся концептуальная модель базы данных.
2. **Логическое проектирование**(алгоритм) – на основе концептуальной модели создаётся структура данных.
3. **Физическое проектирование**(кодирование) – определение особенностей хранения данных, методов доступа.

**Проектирование БД –** это процесс создания проекта (БД), предназначенного для поддержки функционирования объекта и способствующего достижению его целей.

**Концептуальная модель** — это модель, представленная множеством понятий и связей между ними, определяющих смысловую структуру рассматриваемой предметной области или её конкретного объекта.

1. **Модель “Сущность-Связь” (ER-диаграмма).**

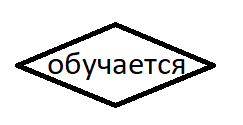
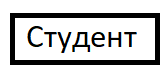
Средством моделирования предметной области на этапе концептуального проектирования является модель "сущность–связь". Часто ее называют ER-моделью (Entity – сущность, Relation – связь). В ней моделирование структуры данных предметной области базируется на использовании графических средств – ER-диаграмм (диаграмм "сущность–связь"). В наглядном виде они представляют связи между сущностями.

**Основные понятия ER-диаграммы**: сущность, атрибут, связь.

**Сущность** - ϶ᴛᴏ некоторый объект реального мира, который может существовать независимо. Сущность имеет экземпляры, отличающиеся друг от друга значениями атрибутов и допускающие однозначную идентификацию.

**Атрибут** - ϶ᴛᴏ свойство сущности. Атрибут, который уникальным образом идентифицирует экземпляры сущности называется **ключом**. Ключ может быть составной, когда представляет собой комбинацию нескольких атрибутов. К примеру, сущность КНИГА характеризуется такими атрибутами, как автор, наименование, цена, издательство, тираж, количество страниц. Конкретные книги являются экземплярами сущности КНИГА

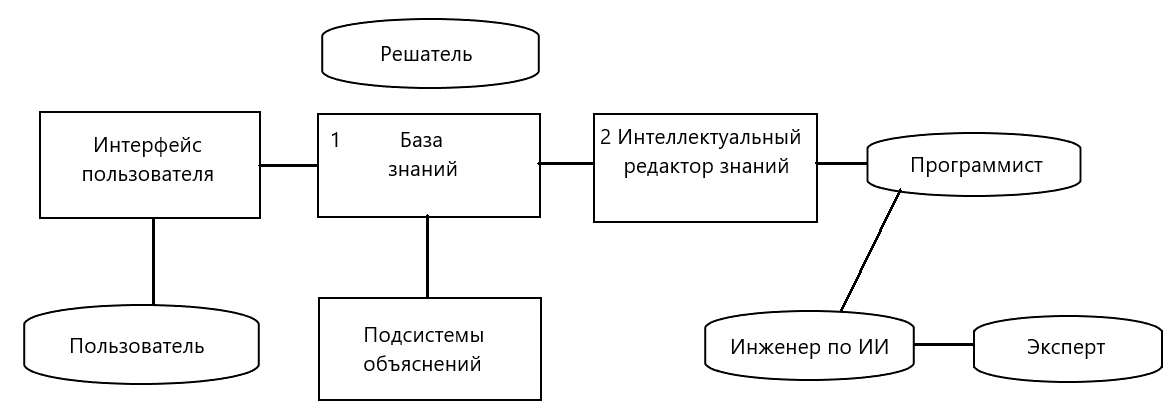
На ER-диаграмме сущность изображается прямоугольником, в котором указывается ее имя, а связь изображается ромбом.

1. **Этапы проектирования БД: логическая модель.**
2. **Этапы проектирования БД: физическая модель.**
3. **Дайте определения понятиям: Знания и База знаний. Назовите отличия данных от знаний.**
4. **Модели представления знаний: Продукционная модель. Привести пример.**
5. **Модели представления знаний: Сеть Фреймов. Привести пример.**
6. **Модели представления знаний: Семантическая сеть. Привести пример.**
7. **Определение экспертной системы. Блок-схема экспертной системы.**
8. **Определение экспертной системы. Технология разработки экспертной системы (схема этапов и стадии).**

**Экспертные системы –** программные комплексы, аккумулирующие знания в конкретных предметных областях и тиражирующие эти знания для консультации другими пользователями.

**Схема экспертной системы:**



**База знаний** – ядро системы, совокупность знаний предметной области, записанная в машинной форме.

**Интеллектуальный редактор базы знаний –** программа, предоставляющая возможность создавать базу знаний или обучать системы.

**Решатель** – программа, моделирующая ход рассуждения эксперта на основании знаний.

**Подсистема объяснений –** программа, позволяющая пользователю получать ответы на вопросы: как было принято данное решение и почему.

**Интерфейс пользователя –** комплекс программ, реализующий диалог пользователя с экспертной системы.

Процесс разработки экспертных систем часто используют прототипный подход.

Демонстрационный прототип решает часть требуемых задач демонстрируя жизнеспособность выбранного метода:

* + - 1. Демонстрационный прототип
      2. Исследовательская программа
      3. Действующий прототип
      4. Промышленные экспертные системы
      5. Коммерческие экспертные системы

Система надежно решает все задачи, многие решения сложных задач, может потребоваться много ресурсов (время, память). Промышленная экспертная система обеспечивает высокое качество решения задач при менее затрачиваемых ресурсах.

Коммерческая экспертная система отличается от промышленной тем, что может подаваться различным потребителям.

**Этапы разработки экспертной системы:**

* 1. Идентификация
  2. Концептуализация
  3. Формулизация
  4. Выполнение (реализация)
  5. Тестирование
  6. Опытная эксплуатация.

1. **Моделирование: определение, цели, виды.**

Моделирование можно рассматривать, как замещение исследуемого объекта (оригинала) его условным образом, описанием или другим объектом – моделью.

**Основная задача моделирования**: познание свойств оригинала, через исследование модели.

Чем сложнее объект, тем важнее роль моделирования в его изучении в его изучении и создании.

Реальная польза от моделей может быть получена при выполнении главных условий:

Модель должна быть **адекватной**: точно отображать исследуемые характеристики оригинала.

Модель должна устранять проблемы с физическим измерением характеристик оригинала (т. е. достаточной).

**Цели моделирования:**

1. Изучение объекта – познавательная цель. Механизмов функционирования, свойств и характеристик (и как следствие – прогнозирование состояний).
2. Управление объектами и системами с целью выработки оптимальных управляющих воздействий и характеристик системы.

(Адаптивное управление/настройка)

**Виды моделей:**

1. Физические модели.

Реальное воплощение физических свойств оригинала. Часто называют макетами (меньшие масштабы реального объекта).

1. Математические модели

Представляют собой формализованное описание объекта или системы с помощью некоторого языка, например совокупности математических соотношений и формул (или схемы алгоритма). Стохастическая (вероятностная) модель

1. Имитационная модель

Сложная система представляется в виде алгоритма или логических действий (блоков), в которых также используется аппарат математического моделирования, статистики и теории вероятности.

1. Модель ИИ. Это “архитектура” (НС, Решающее дерево, Регрессии, Кластеры, Композиции, Ансамбли) реализующая интересующие характеристики объекта (процесса), параметры которой настроены методами машинного обучения.

По своей структуре они могут напоминать математические модели (например, линейная регрессия)

1. **Моделирование: свойства, этапы разработки модели, погрешности моделирования.**

**Свойства моделей**:

1. Адекватность – это степень соответствия модели исследуемому объекту.

На практике модель считают адекватной, если она с удовлетворительной точностью позволяет достичь целей исследования.

Для настройки модели применяют верификацию. //Для ИИ-обучения.

2)Простота. Чем большее количество свойств объекта описывает модель, тем выше ее адекватность (или полезность). Надо стремиться найти более простую модель, позволяющую достичь требуемых результатов.

1. **Имитационное моделирование: определение, свойства, этапы создания, требования к моделям.**

**Задачи моделирования**

Для каждой моделируемой ситуации известна цель (или несколько целей), достижение которой (которых) считается желательным.

В прикладных исследованиях приходится иметь дело со сложными системами, в которых имеется множество целевых функций, и количественное выражение этих функций задача затруднительная.

Основные сферы применения компьютерного моделирования:

-решение оптимизационных задач -исследование сложных систем -прогнозирование их будущих состояний в зависимости от избираемых стратегий управления.

**Общие свойства имитационного моделирования**

Процесс функционирования сложной системы представляется в виде определенного алгоритма, то есть логических действий, которые и реализуется на компьютере. По результатам реализации могут быть сделаны те или иные выводы относительно исходного процесса.

Имитационное моделирование предоставляет возможность преодолеть аналитические трудности и найти ответ на поставленные вопросы о поведении системы.

**Этапы создания имитационной модели:**

1. Формулирует вопросы о поведении сложной системы.
2. Декомпозиция системы.
3. Формулируют Гипотезы (Н).
4. Задание системного времени.
5. Задание свойств системы.
6. Случайные параметры заменяются машинной реализацией.
7. **Моделирование случайных процессов с использованием случайных чисел. Структура случайного процесса.**

Под статистическим моделированием понимается воспроизведение с помощью ЭВМ функционирования вероятностной модели некоторого объекта.

Задачи статистического моделирования состоят в том, чтобы научиться воспроизводить с помощью ЭВМ поведение таких моделей, например:

- с помощью специальных методов и средств вырабатывать программы реализации случайных чисел;

- с помощью этих чисел получать реализацию случайных величин или случайных процессов с более сложными законами распределения;

- с помощью полученных реализации вычислять значения величин, характеризующих модель, и производить обработку результатов экспериментов;

Устанавливать связь алгоритмов моделирования с алгоритмами решения задач вычислительной математики с помощью метода Монте-Карло и строить так называемые «фиктивные» модели, т.е. модели, не имеющие связи с объектом моделирования, но удобные в вычислительном отношении и позволяющие вычислять нужные нам характеристики объекта.

Моделирование случайных процессов строится на основе базовых распределений случайных величин.

Структура случайного процесса: Структуру случайного процесса можно установить по корреляционной функции или по известной плотности распределения. В зависимости от типа законов распределения можно выделить нормальные, равномерные, релеевские, пуассоновские и другие случайные процессы.

1. **Основные характеристики случайных величин.**

**Математическое ожидание:** число, вокруг которого сосредоточены значения случайной величины.



Pi=вероятность этого значения

Xi=возможное значение случайной величины

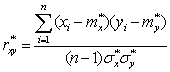
**Дисперсия:** определяет рассеяние случайной величины около ее мат. ожидания



**Среднеквадратичное отношение:**



**Коэффициент корреляции:** характеризует степень линейной связи между величинами



1. **Генерация псевдослучайных чисел: равномерное распределение, нормальное распределение.**

Для метода статистического моделирования на ЭВМ большая доля машинного времени расходуется на действия со случайными числами. Результаты статистического моделирования существенно зависят от качества исходных последовательностей случайных чисел. Поэтому наличие простых и экономичных способов формирования последовательностей случайных чисел требуемого качества во многом определяет возможности практического использования машинного моделирования систем.

**Равномерное распределение:**



Метод серединных квадратов.

1.Выбирается число разрядов n (целое)

2.Задается 2n разрядное число x0=a1, a2… a2n, меньше 1

3.Текущее число возводится в квадрат: (x0)2= b1, b2… b4n

4.Определяется новое значение искомого числа x1=bn+1, bn+2 …b3n, путем выделения 2n средних разрядов из квадрата исходного числа.

5.x0=x1, переход к пункту 3.

Рекурсивный метод.

Основан на формуле 

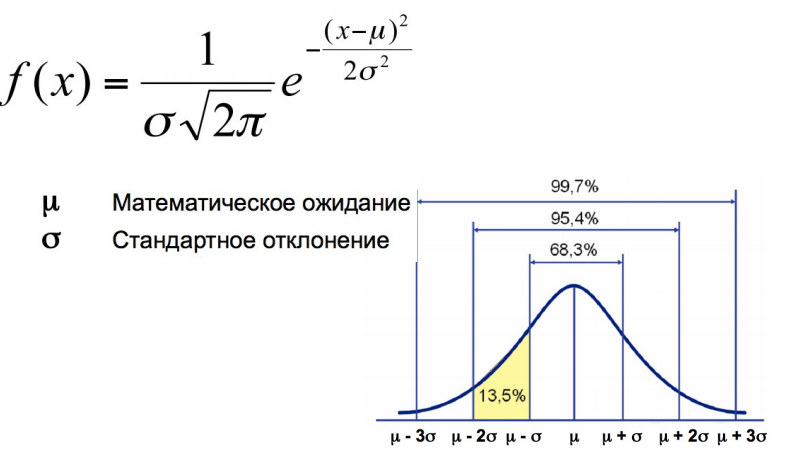
где a0, a1 – целые положительные коэффициенты.

M – целое положительное число, модуль

Zi – рассчитываемые целые положительные числа

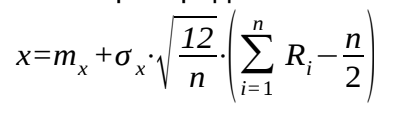
Xi – рассчитываемые квазиравномерные числа

**Нормальное распределение**



Алгоритм имитации.

Основан на: сумма независимых случайных величин имеет асимптотически гауссовское распределение



Ri – случайные квазиравномерные числа

n – положительное число

1. **Генерация псевдослучайных чисел: равномерное распределение, экспоненциальное распределение.**

Для метода статистического моделирования на ЭВМ большая доля машинного времени расходуется на действия со случайными числами. Результаты статистического моделирования существенно зависят от качества исходных последовательностей случайных чисел. Поэтому наличие простых и экономичных способов формирования последовательностей случайных чисел требуемого качества во многом определяет возможности практического использования машинного моделирования систем.

**Равномерное распределение:**



Метод серединных квадратов.

1.Выбирается число разрядов n (целое)

2.Задается 2n разрядное число x0=a1, a2… a2n, меньше 1

3.Текущее число возводится в квадрат: (x0)2= b1, b2… b4n

4.Определяется новое значение искомого числа x1=bn+1, bn+2 …b3n, путем выделения 2n средних разрядов из квадрата исходного числа.

5.x0=x1, переход к пункту 3.

Рекурсивный метод.

Основан на формуле 

где a0, a1 – целые положительные коэффициенты.

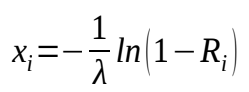
M – целое положительное число, модуль

Zi – рассчитываемые целые положительные числа

Xi – рассчитываемые квазиравномерные числа

**Экспоненциальное распределение**

Алгоритм имитации:



Ri –квазиравномерные случайные числа.

1. **Метод случайного поиска и его применение для решения задач.**

Случайными или стохастическими методами поиска называют методы, в которых для определения оптимума целевой функции или направления движения к оптимуму сознательно используются случайные числа. Необходимым условием применения данного метода является получение случайных чисел с заданными вероятностными характеристиками. Эта задача далеко не простая. Поэтому на практике почти всегда вместо случайных чисел используются так называемые псевдослучайные (ПСЧ). Доказано, что если ПСЧ удовлетворяют определенным требованиям, то они с достаточной для практики точностью могут заменить случайные числа.

Под поиском понимается процесс отыскания такого значения критерия оптимизации (целевой функции), которое близко к оптимальному и в то же время удовлетворяет всем ограничениям.

Различают 3 вида случайного писка:

1.Ненапиравленный случайный поиск.

2.Направленный случайный поиск

3.Направленный поиск с самообучением (если изменение параметра привело к улучшению, то изменяем параметр в том же направлении)

Алгоритм случайного поиска:

1.Задаем случайную точку x0, объявляем ее текущей и вычисляем в ней значение целевой функции Q(x0).

2.Текущей точке придаем приращение Δx, и вычисляется значение целевой функцииQ` = Q(xi+ Δx)

3.Если значение целевой функции улучшилось (Q`<Q), то данную точку делаем текущей.

4.Проверка условия останова, если выполняется, тогда поиск закончен, иначе п2.

Случайный поиск используется при:

1.Отсутствии или не желательном использовании аналитических и численных методов

2.Есть возможность создания модели функционирования системы (процесса)

3.Наличие большого числа случайных факторов в исследуемой системе(процессе).

1. **Задача принятия решения. Постановка задачи, виды задач, типовые задачи.**

**Механизмы принятия решений**

Процессы принятия решений в различных сферах деятельности во многом аналогичны. Поэтому необходим универсальный метод поддержки принятия решений.

ИС: DSS, OLAP

Технологии:DM, KDD

- Зачастую перед ЛПР (лицо принимающее решение) стоит задача выбора между различными альтернативами.

- ЛПР руководствуется своим опытом и интуицией (неопределенный характер решений, субъективное решение).

- С целью придания ясности процесс подготовки принятия решения сопровождается количественным выражением категорий («важнее»)

Имеются:

1. несколько альтернатив (объектов, действий и т. п.),
2. главный критерий (главная цель) сравнения альтернатив (функционал для оптимизации).
3. Несколько групп факторов (частных критериев, объектов, действий и т. п.), влияющих на отбор альтернатив.

f(w, x) = y` ; f – альтернатива

Q (y`) -> opt; Q – оценка альтернативы

Требуется каждой альтернативе поставить в соответствие приоритет – получить рейтинг альтернатив.

Причем чем более предпочтительна альтернатива по избранному критерию, тем больше ее приоритет.

Типовая задача принятия решения:

1. Рассматриваются несколько вариантов решения,
2. Задан критерий, по которому определяется в какой мере то или иное решение является подходящим,

Виды задач принятия решений

1. Задача выбора.

Выбрать или отвергнуть несколько вариантов из группы возможных

1. Задача распределения ресурсов.

- Множество вариантов

1. **Задача принятия решения. Сопутствующие задачи и проблемы.**

**Типовые задачи:**

1. Работа с клиентами.

Какой из клиентов чаще покупает мои товары? Кто из потенциальных клиентов является наиболее перспективным?

Планирование рекламных компаний?

Рекомендательные системы

CRM-системы для сбора данных

1. Анализ рисков.

Анализ рисков инвестирования, составление портфеля активов.

Анализ в Страховании.

Выдача кредитов.

1. Распределение ресурсов

Пример. Руководство предприятия рассматривает перспективные проекты развития. Для них создается рейтинговая модель. В итоге каждому проекту приписывается доля от единицы. Эти доли показывают, какой процент от имеющихся ресурсов надо вложить в каждый проект.

1. Планирование от достигнутого.

Исходя из имеющихся: основных фондов кадров, сырья, инфраструктуры, партнеров, конкурентов и пр. составляется рейтинг возможных положений предприятия через год (месяц, квартал).

Моделирование в стрессовых ситуациях.

1. Планирование

- Анализ перспективных стратегий развития.

6. Анализ эффективность-стоимость. (цена-качество)

Выбор направления деятельности (затрат) с максимальной отдачей при минимальных потерях.

7.Поиск существенных факторов.

Задача определения существенных факторов особенно актуальна при решении масштабных проблем и проблем и проблем стратегического планирования.

8.Диагностика.

Анализ текущего состояния объекта.

9.Построение зависимостей.

Комплексная задача, основная на поиске существенных факторов.

Попытка расширить область задачи и данных.

(Рекомендательные системы)

**Сопутствующие задачи**

1. Выявить наиболее неясные и противоречивые этапы создания модели для поддержки принятия решения

- является ли рассматриваемый набор решений полным

-учтены ли все группы факторов, влияющих на выбор наиболее приоритетного решения.

-известны ли сравнительные оценки того, как сильно влияют главный критерий, факторы и альтернативы друг на друга, имеются ли противоречия в оценках.

-имеются ли альтернативные мнения по рассматриваемой проблеме.

2) Разбить большую задачу о принятии решения на ряд малых задач.

Позволяет распределить работу по подготовке принятия решения. Представить в понятной форме схему взаимодействия факторов, влияющих на формирование приоритетов решений, и самих решений.

3)Оценить и минимизировать противоречивость данных, использующихся для определения приоритетов рассматриваемых решений (очистка данных)

4)Установить предварительно условия, при которых по найденному рейтингу приоритетов возможных решений можно сделать выбор лучшего решения.

5)Проверить факторы и варианты решений на избыточность (отбор параметров – feature selection, отбор стратегий).

6)Оценить устойчивость результатов, полученных в результате применения метода.

Невозможно гарантировать что данные и модели будут абсолютно точными.

Как изменения в параметрах повлияет на изменение приоритета выбора решения (устойчивость).

**Сопутствующие проблемы**

1. Полный набор решений неизвестен (и факторов).

Так же факторы могут иметь свою динамику: С течением времени малозначительные на данный момент факторы и альтернативы могут стать важнейшими, и наоборот, то что важно сейчас может оказаться незначительным впоследствии.

1. Большой набор альтернатив и факторов, влияющих на их отбор.

Необходимы способы разбиения совокупности рассматриваемых решений и факторов, которые потенциально определяют приоритеты решений, на достаточно малые группы – кластеры.

1. В реальных задачах влияния различных факторов на выбор оптимального решения сложны и запутаны.

- нелинейные зависимости

- рекурсивные зависимости (факторы, влияющие на решение, сами зависят от решения)

4) Нет точной количественной информации, необходимой для решения задачи.

- проблема добычи данных

5) Имеющиеся данные противоречивы.

- поиск противоречий

- устранение противоречий

6) Нет четкой и универсальной методики составления рейтинга рассматриваемых решений.

- эксперты предлагают разные системы оценок, оценки разнятся.

- как сформировать целевую функцию (Q -> min)?

Для поддержки принятия решений с помощью IT, включая анализ и выработку альтернатив, в СППР используются следующие технологии: информационный поиск, интеллектуальный анализ данных, извлечение знаний в базах данных, рассуждение на основе прецедентов, имитационное моделирование, генетические алгоритмы, искусственные нейронные сети.

1. **Методы поддержки принятия решений: информационный поиск. Основные определения, виды, методы поиска и оценки эффективности.**

Информационный поиск – процесс поиска неструктурированной документальной

информации и наука об этом поиске.

Этапы поиска:

1) определение (уточнение) информационной потребности и формулировка информационного запроса;

2) установление совокупности возможных держателей информационных массивов (источников);

3) извлечение информации из выявленных информационных массивов;

4) ознакомление с полученной информацией и оценка результатов поиска.

Различают поиск: полнотекстовый – по всему содержимому документа; по метаданным – по неким атрибутам документа, поддерживаемым системой (название документа, дата создания, размер, автор и т.д.); по изображению – по содержанию изображения.

Информационный поиск осуществляется следующими методами: адресный – процесс поиска документов по чисто формальным признакам, указанным в запросе; семантический – процесс поиска документов по их содержанию; документальный – процесс поиска в хранилище информационно-поисковой системы первичных документов или в базе данных вторичных документов, соответствующих запросу пользователя; фактографический – процесс поиска фактов, соответствующих информационному запросу.

Оценка эффективности.

Точность – отношение числа релевантных документов, найденных ИП, к общему числу найденных документов.

Полнота - отношение числа найденных релевантных документов, к общему числу релевантных документов в базе.

Выпадение – характеризует вероятность нахождения нерелевантного ресурса и определяется, как отношение числа найденных нерелевантных документов к общему числу нерелевантных документов в базе

Мера (F)

F = 2\*(точность) \* (полнота) / (точность + полнота).

1. **Методы поддержки принятия решений: интеллектуальный анализ данных. Определения, цели, решаемые задачи, основные этапы.**

**Интеллектуальный анализ данных (Data Mining)**

1. Метод ближайшего соседа (NN – Nearest Neighbor)

С целью определения степени сходства на множестве параметров, используемых для описания прецедентов и текущей ситуации, вводится определенная метрика.

Основными преимуществами данного метода являются простота реализации и универсальность.

К существенным недостаткам метода можно отнести сложность выбора метрики для определения степени сходства и требовательность к ресурсам (для больших БП)

kNN – k ближайших соседей

Термин Data Mining получил свое название из двух понятий: поиска ценной информации в большом наборе информации (data) и добычи полезных ископаемых (mining).

Эти процессы требуют или просеивания огромного количества сырого материала, или разумного исследования и поиска искомых ценностей.

Термин «Data Mining» введен Григорием Пятецким-Шапиро в 1989 году.

Интеллектуальный анализ данных (Data Mining) – выявление скрытых закономерностей или взаимосвязей между переменными в больших массивах необработанных данных.

Data Mining – мультидисциплинарная область, возникшая и развивающаяся на базе таких наук как прикладная статистика, распознавание образов, искусственный интеллект, теория баз данных и др.

Иногда используют термин – Инженерия знаний (рус. Вариант)

К Data Mining относятся: теория БД, машинное обучение, алгоритмизация, искусственный интеллект, распознавание образов, визуализация, статистика, другие дисциплины.

Суть и цель технологии Data Mining можно охарактеризовать так: это технология, которая предназначена для поиска в больших объемах данных неочевидных, объективных и полезных на практике закономерностей.

1. **Методы поддержки принятия решений: извлечение знаний из баз данных (KDD).**

**Последовательность операций KDD**

1. Консолидация данных – процесс их извлечения из различных источников (online-систем, СУБД, файлов, Интернета и т. д.) и загрузка в централизованное хранилище данных.
2. Подготовка анализируемых выборок данных (в том числе, обучающих), загрузка их из хранилища данных или других источников в аналитическое приложение.
3. Очистка данных от факторов, мешающих их корректному анализу (шум, аномальные значения, дубликаты, противоречия, пропуски, фиктивные значения и т. д.)
4. Трансформация – оптимизация данных для решения определенной задачи. Обычно на данном этапе выполняется исключение незначащих факторов, снижения размерности входных данных, нормализация, обогащение (дополнение-augmentation).
5. Анализ данных – применение методов и технологий Data Mining: построение и обучение моделей (например, нейронных сетей, деревьев решений и др.), решение задач классификации, регрессии, кластеризации, прогнозирования и т. д.
6. Интерпретация и визуализация результатов анализа, их применение в бизнес-приложениях.

**Характеристика KDD**

Knowledge Discovery in Databases не задает набор методов обработки или пригодные для анализа алгоритмы, он определяет последовательность действий, которую необходимо выполнить для того, чтобы из исходных данных получить знания. Данный подход универсальный и не зависит от предметной области.

**Data engineering**

Организация ETL/ELT pipeline, DB, Data Lake, Data Warehouse

Big Data: Apache Hadoop, Spark, Airlow

**Рассуждения на основе прецедентов**

Case-Based Reasoning / CBR

Прецедент – случай, имевший место ранее и служащий примером или оправданием для последующих случаев подобного рода.

Вывод на основе прецедентов является подходом, позволяющим решить новую задачу, используя или адаптируя решение уже известной задачи.

Преимущества:

1. Возможность напрямую использовать опыт, накопленный системой без интенсивного привлечения эксперта в той или иной предметной области.
2. Отсутствует необходимость полного и углубленного рассмотрения знаний о конкретной предметной области.
3. Возможно применение эвристик, повышающих эффективность решения задач.

Недостатки:

1. При описании прецедентов обычно ограничиваются поверхностными знаниями о предметной области.
2. Большое количество прецедентов (большой размер базы прецедентов) может привести к снижению производительности системы.
3. Выбор критериев для индексации и сравнения прецедентов является сложной задачей.
4. Невозможность получения решения задач, для которых нет прецедентов ил степень их сходства (подобия) меньше заданного порогового значения.
5. **Методы поддержки принятия решений: рассуждения на основе прецедентов. Основные методы извлечения прецедентов**

**Основные методы извлечения прецедентов**

1. Метод ближайшего соседа (NN – Nearest Neighbor)

С целью определения степени сходства на множестве параметров, используемых для описания прецедентов и текущей ситуации, вводится определенная метрика.