

Темы по ГОС:

понятие информации, общая характеристика процессов сбора, передачи, обработки и накопления информации; технические и программные средства реализации информационных процессов; модели решения функциональных и вычислительных задач; алгоритмизация и программирование; языки программирования высокого уровня; базы данных; программное обеспечение и технологии программирования; локальные и глобальные сети ЭВМ; основы защиты информации и сведений, составляющих государственную тайну; методы защиты информации; компьютерный практикум.

Информация в материальном мире

Сигналы и данные

Все физические объекты, находящиеся вокруг нас, находятся в состоянии непрерывного движения, которое сопровождается обменом энергии. Все виды *энергообмена* сопровождаются появлением сигналов. При взаимодействии сигналов с физическими телами возникают изменения свойств - *регистрация сигналов*, то есть образуются данные. Данные - это зарегистрированные сигналы.

Пример:

Посмотрим на наше сердце. Сокращаясь, оно выбрасывает в сосуды кровь под давлением. Кровь ударяется об стенку артерии и получается щелчок, который мы можем услышать с помощью перископа. Таким образом, мы регистрируем сигналы сердца с помощью какого-то прибора, то есть, получаем некоторые данные о работе сердца.

Данные и методы

Данные несут в себе *информацию* о событиях, произошедших в материальном мире. Для того, чтобы данные давали информацию, необходимо наличие *метода обработки*

Информация - это продукт взаимодействия данных и адекватных им методов.

Пример:

На столе лежит книга. Она содержит какие-то данные. Чтобы получить из этих данных информацию нужно применить методы обработки, т.е. прочитать. Читая книгу, человек получает информацию.

Возьмём опять пример с сердцем. Перископ регистрирует сигналы сердца, но чтобы получить информацию о состоянии сердца нам нужно обработать эти данные, то есть применить методы обработки данных. Человек, который владеет соответствующими методами обработки, сможет получить информацию о работе и состоянии сердца.

Информация и информационный процесс

Процессы получения, передачи, обработки и хранения являются общими для вещества, энергии и информации. Они называются *информационными процессами*. Информационные процессы характерны для живой природы, человека, общества, техники.

Получение	Хранение	Обработка	Передача
Открытие файла Сканирование Просмотр ТВ пере-дач Человек: Зрение, слух, обоняние, осязание, вкус.	Генетический код в каждой клетке организма Годовые кольца на срезе дерева Ноты CD-ROM Оптический диск Ключ от замка Видеофильм Фотография	Решение задачи Исполнение музыки по нотам Перевод иностранного текста Проявление фотоплёнки	Чтение вслух Ввод с клавиатуры Вывод информации на экран Радио ТВ Интернет-связи

Является ли информационным процессом выплавка стали? (Нет)

Таким образом, понятие информации – базовое понятие. Поэтому дать точное определение информации невозможно. Содержание основных базовых понятий в каждой науке должно быть пояснено на примерах или выявлено путём его сопоставления с содержанием других понятий. Понятие информации используется в различных науках: информатике, биологии, социологии, кибернетике. В каждой науке понятие “информация” связано с различными системами понятий.

Понятие информации в биологии связано с понятиями “ген”, “наследственность” и “поведение”.

Понятие информации в кибернетике связано с понятиями “управление”, “прямая связь”, “обратная связь”.

Понятие информации в обществе связано с понятиями “сведения”, “сообщения” “осведомлённость о положении дел”.

Информация с точки зрения науки – это знания.

С информацией можно производить следующие *действия*:

- обмен (*передача и получение*);
- хранение;
- обработка.

Обмен информацией может происходить в *образной* (образное восприятие при непосредственном контакте человека с окружающим миром с помощью зрения, слуха, обоняния, осязания и вкуса и *знаковой* (знаковая система представления информации - это язык) *формах*. Языки бывают разговорными(русский, немецкий, и т.д.), причем в устной форме(фонетика) и в письменной(грамматика) и формальные(в математике - язык формул, в музыке - язык нот, в медицине - латынь).

Хранение информации происходит или в памяти человека, или на внешних носителях. В памяти человека информация может храниться как в *образной форме* (я помню, как пахнет роза), так и в *знаковой* (словесной, формульной). Информацию, хранимую в памяти, называют *оперативной*. Информацию, хранимую на внешних носителях(листе бумаги, диске, пластинке и т.д.), называют *внешней*. Она может быть переведена в разряд оперативной, если будет

"прочитана" человеком.

Таким образом, внешние носители выполняют роль "дополнительной" памяти человека. На них могут храниться звук, тексты, изображения.

Обработка информации производится человеком или в уме, или с помощью каких-либо вспомогательных средств (счеты, калькулятор, компьютер и др.). В результате обработки получается новая информация, которая каким-то образом сохраняется (запоминается, записывается). Обработка информации производится по каким-то определенным правилам (алгоритмам). Сами эти правила могут также подвергаться обработке (дополняться, исправляться, уточняться).

Количество информации как мера уменьшения неопределённости знания

Уменьшение неопределённости знания.

Подход к информации как к мере уменьшения неопределённости знания позволяет количественно измерять информацию. Пусть у нас имеется монета, которую мы бросаем на ровную поверхность. С равной вероятностью произойдёт одно из двух возможных событий – монета окажется в одном из двух положений: "орёл" или "решка". События равновероятны, если при возрастающем числе опытов число выпадений "орла" и "решки" постепенно сближаются. Перед броском существует неопределённость нашего знания (возможны 2 события), и как упадёт монета – предсказать невозможно. После броска наступает полная определённость, т.к. мы видим, что монета в данный момент находится в определённом положении. Это сообщение приводит к уменьшению неопределённости нашего знания в 2 раза, т.к. из двух возможных равновероятных событий реализовалось одно. В окружающей действительности часто встречаются ситуации, когда может произойти большее, чем 2, число равновероятных событий. Чем больше начальное число возможных равновероятных событий, тем больше начальная неопределённость нашего знания и тем большее количество информации будет содержать сообщение о результатах опыта.

Единица измерения количества информации

За *единицу количества информации* принято такое количество информации, которое содержит сообщение уменьшающее неопределённость знания в 2 раза. Такая единица названа *бит*.

Компьютер оперирует числами в двоичной системе счисления, поэтому в кратных единицах измерения количества информации используется коэффициент 2^n .

Следующей по величине единицей измерения количества информации является *байт*, причём

$$1 \text{ байт} = 2^3 \text{ бит} = 8 \text{ бит.}$$

Кратные байту единицы измерения количества информации вводятся следующим образом:

$$1 \text{ Кбайт} = 2^{10} \text{ байт} = 1024 \text{ байт}$$

$$1 \text{ Мбайт} = 2^{10} \text{ Кбайт} = 1024 \text{ Кбайт} = 2^{20} \text{ байт}$$

$$1 \text{ Гбайт} = 2^{10} \text{ Мбайт} = 1024 \text{ Мбайт} = 2^{20} \text{ Кбайт} = 2^{30} \text{ байт.}$$

Количество возможных событий и количество информации

Существует формула, которая связывает между собой количество возможных событий N и количество информации I :

$$N=2^I$$

Формула Шеннона

Формулу для вычисления количества информации для событий с различными вероятностями предложил К. Шеннон. В этом случае количество информации определяется по формуле:

$$I = - \sum_{i=1}^N p_i \log_2 p_i$$

Где I – количество информации,
 N – количество возможных событий,
 p_i – вероятности отдельных событий.

Количество информации, которое мы получаем, достигает *максимального значения*, если события *равновероятны*.

Логические основы компьютера

Алгебра логики — это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Логическое высказывание — это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Так, предложения "*6 — четное число*", "*Рим — столица Франции*" являются высказываниями. А предложения "*ученик десятого класса*" и "*информатика — интересный предмет*" не являются высказываниями.

Алгебра логики рассматривает любое высказывание только с одной точки зрения — является ли оно истинным или ложным.

Употребляемые в обычной речи слова и словосочетания "**не**", "**и**", "**или**", "**если... , то**", "**тогда и только тогда**" и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются **логическими связками**.

Высказывания, образованные из других высказываний с помощью логических связок, называются **составными**. Высказывания, не являющиеся составными, называются **элементарными**.

Чтобы обращаться к логическим высказываниям, им назначают имена. Пусть через **A** обозначено высказывание "*Тимур поедет летом на море*", а через **B** — высказывание "*Тимур летом отправится в горы*". Тогда составное высказывание "*Тимур летом побывает и на море, и в горах*" можно кратко записать как **A и B**. Здесь "**и**" — логическая связка, **A**, **B** — логические переменные, которые могут принимать только два значения — "истина" или "ложь", обозначаемые, соответственно, "1" и "0".

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение:

НЕ Операция, выражаемая словом "не", называется **отрицанием** и Высказывание \bar{A} истинно, когда A ложно, и ложно, когда A истинно.

И Операция, выражаемая связкой "и", называется **конъюнкцией** (лат. conjunctio — соединение) или логическим умножением и обозначается точкой (может также обозначаться знаками \wedge или $\&$). Высказывание $A \cdot B$ истинно тогда и только тогда, когда оба высказывания A и B истинны.

ИЛИ Операция, выражаемая связкой "или" (в неисключающем смысле этого слова), называется **дизъюнкцией** или логическим сложением и обозначается знаком \vee (или плюсом). Высказывание $A \vee B$ ложно тогда и только тогда, когда оба высказывания A и B ложны.

ЕСЛИ-ТО Операция, выражаемая связками "если ..., то", "из ... следует", "... влечет ...", называется **импликацией** (лат. *implico* — тесно связаны) и обозначается знаком \rightarrow . Высказывание $A \rightarrow B$ ложно тогда и только тогда, когда A истинно, а B ложно.

Импликацию можно выразить через дизъюнкцию и отрицание: $A \rightarrow B = \bar{A} \vee B$.

РАВНОСИЛЬНО Операция, выражаемая связками "тогда и только тогда", "необходимо и достаточно", "... равносильно ...", называется **эквиваленцией** или двойной импликацией и обозначается знаком \leftrightarrow

Высказывание $A \leftrightarrow B$ истинно тогда и только тогда, когда значения A и B совпадают.

Эквиваленцию можно выразить через отрицание, дизъюнкцию и конъюнкцию:

$$A \leftrightarrow B = (\bar{A} \vee B) \cdot (\bar{B} \vee A).$$

Порядок выполнения логических операций задается круглыми скобками. Но для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицания ("не"), затем конъюнкция ("и"), после конъюнкции — дизъюнкция ("или") и в последнюю очередь — импликация.

С помощью логических переменных и символов логических операций любое высказывание можно формализовать, то есть заменить логической формулой.

Определение **логической формулы**:

1. Всякая логическая переменная и символы "истина" ("1") и "ложь" ("0") — формулы.
2. Если A и B — формулы, то \bar{A} , $A \cdot B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ — формулы.
3. Никаких других формул в алгебре логики нет.

Формулы называются тождественно истинными формулами или **тавтологиями**, если они истинны при всех значениях переменных. Высказывания, которые формализуются тавтологиями, называются логически истинными высказываниями.

Формулы называются тождественно ложными формулами или **противоречиями**, если ложны при всех значениях переменных. Высказывания, которые формализуются противоречиями, называются логически ложными высказываниями.

Если две формулы А и В одновременно, то есть при одинаковых наборах значений входящих в них переменных, принимают одинаковые значения, то они называются **равносильными**.

Равносильность двух формул алгебры логики обозначается символом "=" или символом "≡". Замена формулы другой, ей равносильной, называется равносильным преобразованием данной формулы.

Связь между алгеброй логики и двоичным кодированием

Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: "1" и "0".

Из этого следует два вывода:

1. одни и те же устройства компьютера могут применяться для обработки и хранения как числовой информации, представленной в двоичной системе счисления, так и логических переменных;
2. на этапе конструирования аппаратных средств алгебра логики позволяет значительно упростить логические функции, описывающие функционирование схем компьютера, и, следовательно, уменьшить число элементарных логических элементов, из десятков тысяч которых состоят основные узлы компьютера.

Данные и команды представляются в виде двоичных последовательностей различной структуры и длины. Существуют различные физические способы кодирования двоичной информации. В электронных устройствах компьютера двоичные единицы чаще всего кодируются более высоким уровнем напряжения, чем двоичные нули (или наоборот), например:



Логические элементы компьютера

Логический элемент компьютера — это часть электронной логической схемы, которая реализует элементарную логическую функцию.

Логическими элементами компьютеров являются электронные схемы **И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ** и другие (называемые также **вентильями**), а также **триггер**.

С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно у вентилей бывает от двух до восьми входов и один или два выхода.

Чтобы представить два логических состояния — "1" и "0" в вентильях, соответствующие им входные и выходные сигналы имеют один из двух установленных уровней напряжения. Например, +5 вольт и 0 вольт.

Высокий уровень обычно соответствует значению "истина" ("1"), а низкий —

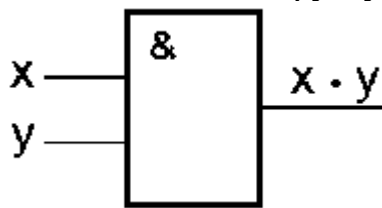
значению “ложь” (“0”).

Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Это упрощает запись и понимание сложных логических схем. Работу логических элементов описывают с помощью таблиц истинности.

Таблица истинности это табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

С х е м а И

Схема **И** реализует **конъюнкцию** двух или более логических значений. Условное обозначение на структурных схемах схемы **И** с двумя входами:



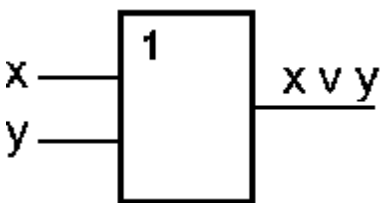
x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Единица на выходе схемы **И** будет тогда и только тогда, когда на всех входах будут единицы. Когда хотя бы на одном входе будет ноль, на выходе также будет ноль.

С х е м а ИЛИ

Схема **ИЛИ** реализует **дизъюнкцию** двух или более логических значений. Когда хотя бы на одном входе схемы **ИЛИ** будет единица, на её выходе также будет единица.

Условное обозначение на структурных схемах схемы **ИЛИ** с двумя входами представлено на рисунке ниже. Знак "1" на схеме — от устаревшего обозначения дизъюнкции как " ≥ 1 " (т.е. значение дизъюнкции равно единице, если сумма значений операндов больше или равна 1). Связь между выходом z этой схемы и входами x и y описывается соотношением: $z = x \vee y$ (читается как "x или y").



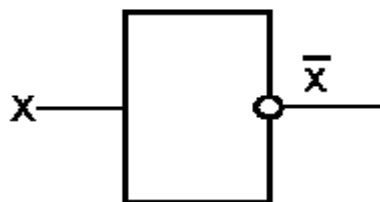
x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

С х е м а НЕ

Схема **НЕ** (инвертор) реализует операцию **отрицания**. Связь между входом x этой

схемы и выходом z можно записать соотношением $z = \overline{x}$, где \overline{x} читается как "не x " или "инверсия x ".

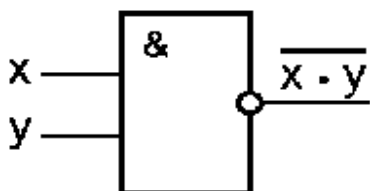
Если на входе схемы 0 , то на выходе 1 . Когда на входе 1 , на выходе 0 . Условное обозначение на структурных схемах инвертора — на рисунке 5.3



x	\overline{x}
0	1
1	0

Схема И—НЕ

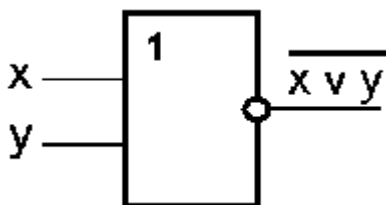
Схема **И—НЕ** состоит из элемента **И** и инвертора и осуществляет отрицание результата схемы **И**. Связь между выходом z и входами x и y схемы записывают следующим образом: $z = \overline{x \cdot y}$, где $\overline{x \cdot y}$ читается как "инверсия x и y ".



x	y	$\overline{x \cdot y}$
0	0	1
0	1	1
1	0	1
1	1	0

Схема ИЛИ—НЕ

Схема **ИЛИ—НЕ** состоит из элемента **ИЛИ** и инвертора и осуществляет отрицание результата схемы **ИЛИ**. Связь между выходом z и входами x и y схемы записывают следующим образом: $z = \overline{x \vee y}$, где $\overline{x \vee y}$ читается как "инверсия x или y ". Условное обозначение на структурных схемах схемы **ИЛИ—НЕ** с двумя входами:



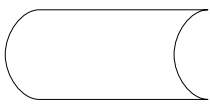

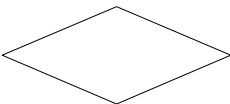
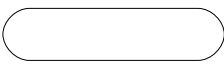


x	y	$\overline{x \vee y}$
0	0	1
0	1	0
1	0	0
1	1	0

Алгоритмы. Блок-схемы. Алгоритмические структуры

Алгоритм – строго детерминированная последовательность команд описывающая процесс по преобразованию объекта из начального состояния в конечное, написанная на понятном исполнителю языке.

Блок-схема — графическое представление программы или алгоритма с использованием стандартных графических элементов (прямоугольников, ромбов, трапеций и др.), обозначающих команды, действия, данные и т. п.

Основные символы

	Данные Данные, носитель данных не определен.
	Запоминаемые данные Хранимые данные в виде, пригодном для обработки, носитель данных не определен.
	Процесс Функция обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации)
	Подготовка Модификация команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы).
	Решение решение или функция переключательного типа, имеющая один вход и ряд альтернативных выходов, из которых только один может быть активизирован после вычисления условий, определенных внутри этого символа.
	Терминатор Выход во внешнюю среду и вход из внешней среды (начало или конец схемы программы, внешнее использование и источник или пункт назначения данных)
	Соединитель выход в часть схемы и вход из другой части этой схемы; обрыв линии и продолжение ее в другом месте
	Комментарий описательные комментарии или пояснительные записи в целях объяснения или примечания
	Линия. поток данных или управления.
	Пунктирная линия.

Алгоритмические структуры

1. Линейный алгоритм

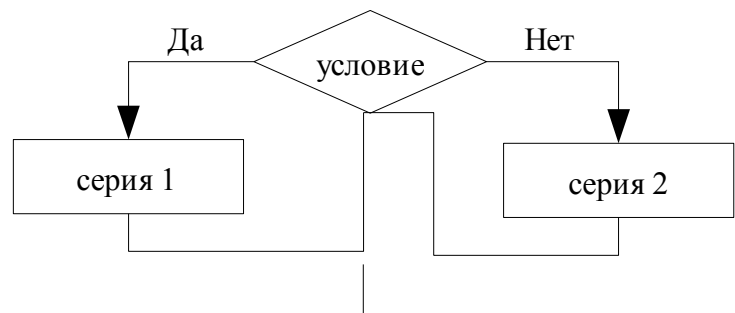
Команды в линейном алгоритме выполняются последовательно одна за другой. Такие последовательности называют сериями.



2. Структура «ветвление»

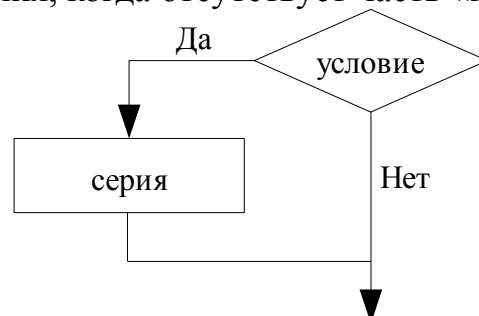
В зависимости от истинности или ложности условия выполняется та или иная серия команд:

если <условие>, то <серия1>
иначе <серия2>
конец ветвления



Возможен вариант неполного ветвления, когда отсутствует часть «иначе»:

если <условие> то <серия>
конец ветвления



3. Алгоритмическая структура «выбор»

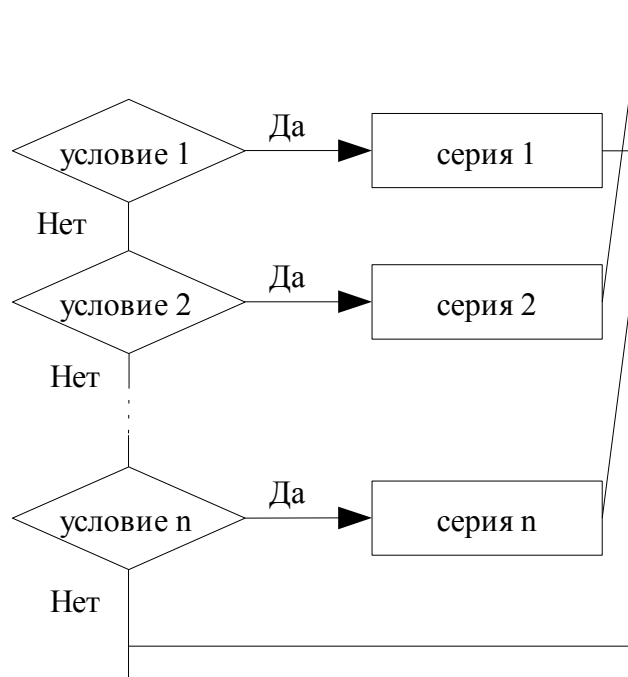
выбор

при <условие 1> : <серия 1>

при <условие 2> : <серия 2>

при <условие n> : <серия n>

конец выбора



4. Алгоритмическая структура «цикл»

Многочетное выполнение серии команд (тело цикла).

Разделяют следующие типы циклов:

- цикл с параметром (со счетчиком);
- цикл с условием:
 - цикл с предусловием;
 - цикл с постусловием;

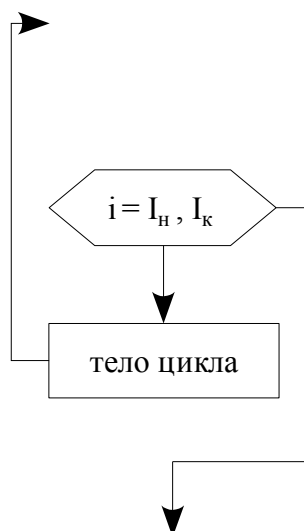
Цикл с параметром

Выполнение тела цикла для всех значений некоторой переменной (параметра цикла, счетчика) в заданном диапазоне.

для i от I_n до I_k

<тело цикла>

конец цикла

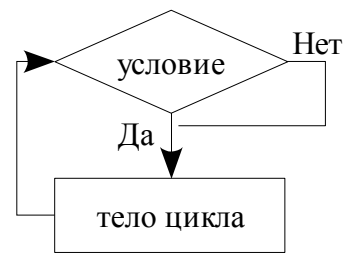


Цикл с условием

Выполнение тела цикла, пока истинно условие цикла.

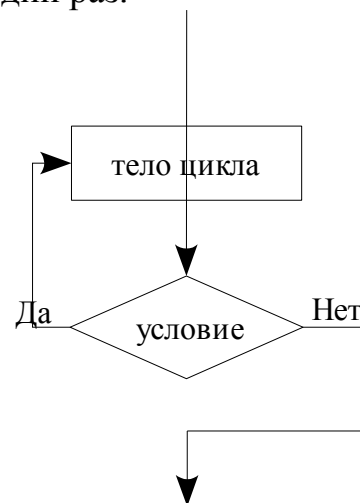
Цикл с предусловием. Условие предшествует телу цикла. Тело цикла может не выполниться ни разу, если условие цикла изначально неверно

пока <условие> выполнять
 <тело цикла>
конец цикла



Цикл с постусловием. Проверка условия происходит после выполнения тела цикла. Поэтому тело цикла выполниться хотя бы один раз.

начало цикла
 <тело цикла>
пока <условие>



Моделирование

Моделирование – метод познания, состоящий в создании и исследовании моделей.

Модель – некий новый объект, отражающий существенные особенности изучаемого объекта, явления или процесса. Один и тот же объект может иметь множество моделей, разные объекты могут описываться одной и той же моделью.

Построение модели включает следующие этапы:

1. **подбор структуры модели** – представление логики развития объекта изучения в форме математической модели, пригодной для дальнейших исследований. Это выбор и анализ состава переменных и выбор класса аппроксимирующих функций (линейная и др.).
2. **Идентификация параметров модели** (определение значений параметров) – оценка параметров модели, выявленных спецификаций с помощью определенных методов (например метод наименьших квадратов, обобщенный метод наименьших квадратов).
3. **Верификация модели** — проверка ее истинности, адекватности:

сопоставление результатов расчетов по модели с соответствующими данными действительности.

Модели:

- 1) **материальные(предметные)** – воспроизводят физические свойства объектов в материальной форме (глобус, модели кристаллических решеток и т.д.)
- 2) **информационные** – представляют объекты и процессы в образной(рисунок, фотография) или знаковой форме (формула, таблица(таблица Менделеева), программа на языке программирования)

Процесс построения информационных моделей с помощью формальных языков называется **формализацией**. С помощью формальных языков строятся формальные информационные модели (математические ,логические)

Система – совокупность взаимосвязанных объектов, которые называются элементами системы.

Модели систем

Наши представления о реальных системах носят приближенный, модельный характер. Описывая в какой-либо форме реальную систему, мы создаем ее информационную модель. Рассмотрим три разновидности информационных моделей систем:

1. модель черного ящика;
2. модель состава;
3. структурная модель.

Модель "черного ящика". отображает только связи системы со средой, в виде перечня "входов" и "выходов". **Вход системы** — это воздействие, на систему со стороны внешней среды, а **выход** — это воздействие, оказываемое системой на окружающую среду. Такое представление о системе называется моделью "черного ящика" (см. рисунок).



Модель "черного ящика"

Модель "черного ящика" используется в тех случаях, когда внутреннее устройство системы недоступно или не представляет интереса, но важно описать ее внешние взаимодействия. Например, в любой инструкции по использованию бытовой техники (телевизор, магнитофон, стиральная машина и пр.) дается описание работы с ней на уровне входов и выходов: как включить, как регулировать работу, что получим на выходе. Такого представления может быть вполне достаточно для пользователя данной техникой, но не достаточно для специалиста по ее ремонту.

Модель "черного ящика", как и любая другая, строится в соответствии с целью моделирования, учитывая лишь те входы и выходы системы, которые существенны

с точки зрения цели моделирования, назначения создаваемой модели.

Трудность построения модели "черного ящика" состоит в том, что надо решить, какие из многочисленных реальных связей включать, а какие не включать в состав модели. Кроме того, всегда существуют и такие связи, которые нам неизвестны, но они-то и могут оказаться существенными.

Модель состава системы дает описание входящих в нее элементов и подсистем, но не рассматривает связей между ними. Очевидно, что и модель состава компьютера может иметь разные варианты в зависимости от отражаемой в ней точки зрения на систему. Например:

Вариант 1: системный блок, клавиатура, монитор, принтер, мышь.

Вариант 2: оперативная память, внешняя память, центральный процессор, устройства ввода, устройства вывода.

Вариант 3: центральный процессор, ОЗУ, ПЗУ, жесткий диск, флоппи-диск, лазерный диск, информационная магистраль, клавиатура, монитор, контроллеры внешних устройств и пр.

Главная трудность в построении модели состава заключается в том, что разделение целостной системы на части является относительным, условным, зависящим от целей моделирования (это относится не только к границам между частями системы, но и к границам самой системы). Кроме того, относительным является и выбор уровня дробления, определение самой малой части – элемента. Для удобства обычно применяют иерархические модели состава системы, состоящие из нескольких уровней детализации.

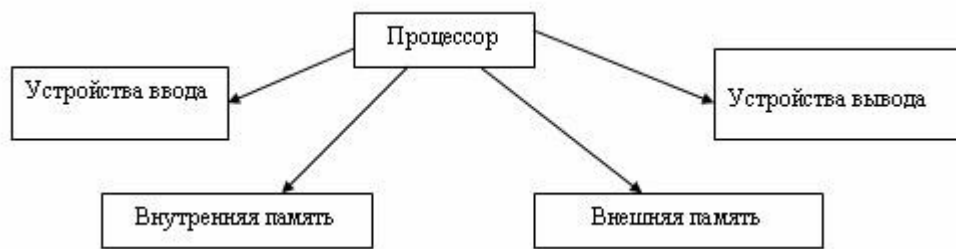
Структурную модель системы еще называют **структурной схемой**. На структурной схеме отражается состав системы и ее внутренние связи. Наряду с термином "связь" нередко употребляют термин "отношение".

Наглядным способом описания структурной модели системы являются графы. На рисунке в виде ориентированного графа приведена структурная модель компьютера.



Здесь стрелки обозначают информационные связи между элементами системы. Направление стрелок указывает на направление передачи информации.

Однако если нас интересуют связи по управлению, то получится следующая граф-модель компьютера:



Здесь стрелка обозначает направление управляющего воздействия. Смысл схемы заключается в том, что процессор управляет работой всех остальных устройств компьютера.

Следовательно, структурная модель одной и той же системы может быть разной. Все определяется целями моделирования.

База данных - организованная совокупность данных, предназначенная для длительного хранения и постоянного обновления и использования. БД представляет собой информационную модель определенной предметной области. Базы данных строятся на основе информационных моделей.

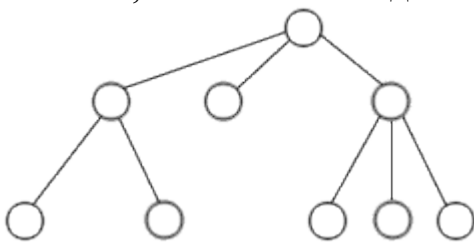
Типы информационных моделей.

иерархическая;

- сетевая;
- реляционная.

Иерархическая модель данных

Строится по принципу иерархии типов объектов, т. е. один тип объекта является главным, а остальные подчиненными.:

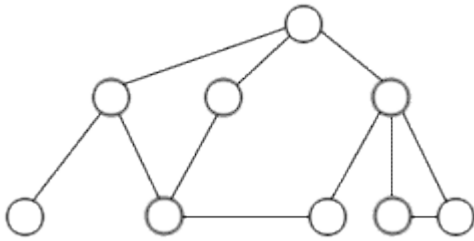


Узел дерева - это совокупность атрибутов, описывающих объект.

Между вершинами дерева существуют связи "один ко многим". Каждая вершина содержит данные и несколько связей с подчиненными вершинами. Все вершины, кроме корня, могут иметь несколько подчиненных и одну главную вершину.

Сетевая модель данных

Понятие главного и подчиненного объекта несколько расширено. Любой объект может быть главным и подчиненным. Каждый объект может участвовать в любом числе взаимодействий:



Реляционная (табличная) модель данных

В реляционной модели данных объекты и взаимодействия между ними представляются с помощью таблиц. Каждая запись содержит информацию об отдельном объекте системы (книге из библиотеки, сотруднике предприятия), а каждое поле – определенную характеристику этого объекта. Каждая таблица должна иметь первичный ключ - поле или комбинацию полей, которая единственным образом идентифицирует каждую строку таблицы.

СУБД (система управления базами данных) – программное обеспечение для работы с базами данных.

Между таблицами возможны 4 типа отношений:

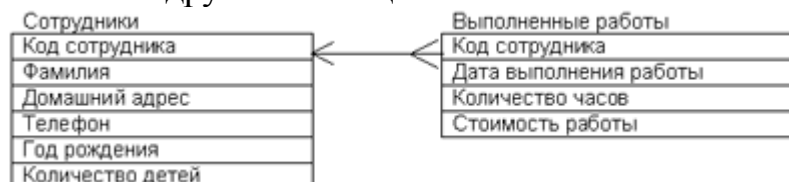
- "один к одному"
- "один ко многим"
- "многие к одному"
- "многие ко многим"

"один к одному": каждая запись одной таблицы соответствует одной записи в другой таблице:



Связь осуществляется по совпадающему полю. В примере это Код сотрудника. Отношение "один к одному" целесообразно использовать, если часть данных, например данные об образовании используются нечасто. Использование данного отношения позволит увеличить скорость обработки данных.

"один ко многим": каждой записи в одной таблице соответствует несколько записей в другой таблице:

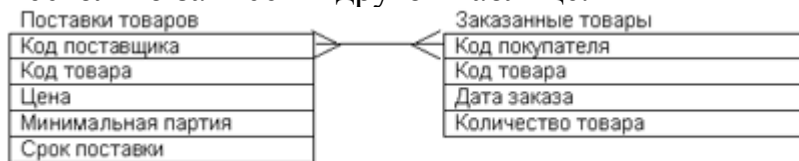


Одному сотруднику может соответствовать несколько выполненных работ. Отношение "один ко многим" используется очень часто, поддерживается всеми СУБД.

"многие к одному": нескольким записям в одной таблице соответствует одна запись в другой таблице.

Отношение "многие к одному" аналогично отношению "один ко многим". Тип отношения зависит от Вашей точки зрения. Например, если Вы будете рассматривать отношение между выполненной работой и сотрудниками, то получите отношение "многие к одному".

"многие ко многим": нескольким записям в одной таблицы соответствуют несколько записей в другой таблице.



Между таблицами Поставки товаров и Заказанные товары существует отношение многие ко многим. Т.к. на каждый товар может быть более одного заказа, аналогично каждый поставляемый товар может производиться более чем одним предприятием.

Компьютерные сети

Локальная сеть объединяет несколько компьютеров и дает возможность пользователям совместно использовать ресурсы компьютеров, а также подключенных к сети периферийных устройств.

В **одноранговой сети** все компьютеры равноправны: нет иерархии среди компьютеров и нет выделенного (dedicated) сервера. Как правило, каждый компьютер функционирует и как клиент, и как сервер. Все пользователи самостоятельно решают, какие данные на своем компьютере сделать общедоступными по сети. Для небольшой группы пользователей подобные сети легко обеспечивают разделение данных и периферийных устройств. Вместе с тем, поскольку администрирование в одноранговых сетях нецентрализованное, обеспечить развитую защиту данных трудно.

Сети на основе сервера. Если к сети подключено более 30 пользователей, то одноранговая сеть, где компьютеры выступают в роли и клиентов, и серверов, может оказаться недостаточно производительной. Поэтому большинство сетей использует выделенные серверы. Выделенным называется такой сервер, который функционирует только как сервер (исключая функции клиента или рабочей станции). Они специально оптимизированы для быстрой обработки запросов от сетевых клиентов и для управления защитой файлов и каталогов. Сети на основе сервера наиболее эффективны в том случае, когда совместно используется большое количество ресурсов и данных. Администратор может управлять защитой данных, наблюдая за функционированием сети. В таких сетях может быть один или несколько серверов, в зависимости от объема сетевого трафика, количества периферийных устройств и т.п.