# Формализация информации и Big Data

http//vikchas.ru

## Тема 1. Формализация информации

## Лекция 1 «Данные, информация, базы данных»

Часовских Виктор Петрович
д-р техн. наук, профессор кафедры ШИиКМ

ФГБОУ ВО «Уральский государственный экономический университет»

Екатеринбург 2023

# Данные - определения и интерпретация

Согласно «Информационно-коммуникационные технологии в образовании» ГОСТ Р 52653-2006 , **данные** – представление информации в формализованном виде, пригодном для передачи, интерпретации и обработки.

Согласно «Система стандартов по информации, библиотечному и издательскому делу» ГОСТ 7.0-99 , **данные** – информация, обработанная и представленная в формализованном виде для дальнейшей обработки.

***Словарь Ушакова***

**Данные** - сведения, обстоятельства, служащие для какого-нибудь вывода, решения.

***Современный экономический словарь. 1999***

**Данные**

**1)** факты и характеризующие их числовые, количественные показатели: имена, даты событий сведения об экономических процессах, местах действия;

**2)** сведения, обработанные Специальным образом для принятия решений, информация.

***Словарь экономических терминов***

**Данные**

**1)** факты, не связанные друг с другом: имена, даты, числа;

**2)** сведения, обработанные специальным образом для принятия решений, информация.

***Краткий словарь по вычислительной технике, информатике и метрологии***

**Данные**

сведения, полученные путем измерения, наблюдения, логических или арифметических операций, представленные в форме, пригодной для постоянного хранения, обработки и передачи.

***Полный словарь терминов и понятий мобильной связи***

**Данные**

информация, представленная в формализованном виде, пригодном для автоматизированной обработки.

***Словарь Ожегова***

**Данные**

**1.** Сведения, необходимые для какого-то  вывода, решения.

**2.** Свойства, способности, качества как условия или основания для чего-то.

***Словарь Ефремовой***

**Данные**

Сведения, факты, характеризующие кого-л., что-л., необходимые для каких-л. выводов, решений.

Свойства, способности, качества как условия или основания, необходимые для чего-л.

***Энциклопедия Брокгауза и Ефрона***

**Данные**

— В вопросах математики **Д.** суть величины, значения которых известны или предполагаются известными; зная их, требуется в рассматриваемом вопросе определить искомые неизвестные величины.

**Д.** (Δεδόμενα) есть заглавие одного из сочинений Эвклида, составляющего продолжение его "Элементов".

***Кембриджский словарь***

**Данные**

Это информация, особенно факты и числа, собранные для последующего использования при принятии решений. Данные – это информация в электронной форме, пригодная для хранения и использования компьютером.

**Философия** рассматривает преобразование сведений в данные, данных в информацию, а информации – в знания.

**Кибернетика** – Норберт Винера об информации
«… чем более вероятно сообщение, тем меньше оно содержит информации. Клише, например, имеют меньше смысла, чем великолепные стихи».
«Передача информации возможна лишь как передача альтернатив».
«Идеальная информация не содержит в себе ничего поддающегося измерению, следовательно, доступная измерению информация не может быть идеальной».

«…сообщество простирается лишь до того предела, до которого простирается действительная передача информации. Можно дать некоторую меру сообщества, сравнивая число решений, поступающих в группу извне, с числом решений, принимаемых в группе. Мы измеряем тем самым автономию группы. Мера эффективной величины группы – это тот размер, который она должна иметь, чтобы достичь определенной установленной степени автономии».

«Информация – это обозначение содержания, полученного из внешнего мира в процессе нашего приспособления к нему и приспосабливания к нему наших чувств».

«… подобно тому как энтропия есть мера дезорганизации, информация есть мера организации».



**Информация - это снятая неопределенность** (К. Шеннон)

# Структуры данных

История современных вычислительных машин определяется работами Чарльза Бэббиджа, опубликовавшим в 1822 году описание разностной механической вычислительной машины, а в 1837 году появилось описание аналитической механической вычислительной машины.

Аналитическая машина считается прообразом современных электронных вычислительных машин.

К разработкам Ч. Беббиджа были написаны программы различных вычислений Адой Лавлейс (в настоящее время считается первой программисткой, её день рождения, 10 декабря, объявлен международным днём программиста).

Ч. Беббидж и Ада Лавлейс рассматривали представления исходных данных, промежуточные вычисления и итоги в виде таблиц и отдельных столбцов этих таблиц.

**Это первая формализация и первое определение структур данных для вычислительных машин**.

В настоящее время практически все программы для ЭВМ обычно оперируют с таблицами информации.

В большинстве случаев это не просто аморфные массы числовых величин: в таблицах присутствуют важные структурные отношения между элементами данных. В простейшей форме таблица может быть линейным списком элементов. В более сложных ситуациях таблица может быть двумерным массивом (т. е. матрицей, иногда называемой сеткой, имеющей структуру строк и столбцов), либо может быть n-мерным массивом при весьма больших значениях и, либо она может иметь структуру дерева,

представляющего отношения иерархии или ветвления, либо это может быть сложная много связанная структура с множеством взаимных соединений, такая, например, которую можно найти в человеческом мозгу.

Рассмотрим наиболее важные факты (формализацию), касающиеся информационных структур: статические и динамические свойства разного рода структур; средства распределения памяти и представления структурных данных; эффективные алгоритмы для создания, изменения, разрушения структурной информации и доступа к ней.

# ЛИНЕЙНЫЕ СПИСКИ

## Стеки, очереди и деки

**Линейный список** — это множество, состоящее из n ≥ 0 узлов X[1], X [2], ..., X [n], структурные свойства которого по сути ограничиваются лишь линейным (одномерным) относительным положением узлов, т. е. теми условиями, что если n > 0, то X [1] является первым узлом; если 1 < k < n, то k-му узлу X [k] предшествует X [k — 1] и за ним следует X [k + 1]; X [n] является последним узлом.

**Операции**, которые мы можем выполнять с линейными списками, включают, например, следующие:

1. Получить доступ к k-му узлу списка, чтобы проанализировать и/или изменить содержимое его полей.
2. Включить новый узел непосредственно перед k-м узлом,
3. Исключить k-й узел.
4. Объединить два (или более) линейных списка в один список,
5. Разбить линейный список на два (или более) списка.
6. Сделать копию линейного списка.
7. Определить количество узлов в списке.
8. Выполнить сортировку узлов списка в возрастающем порядке по некоторым полям в узлах.
9. Найти в списке узел с заданным значением в некотором поле

Очень часто встречаются **линейные списки**, в которых включение, исключение или доступ к значениям почти всегда производятся в первом или последнем узлах, и мы дадим им специальные названия:

**Стек** — линейный список, в котором все включения и исключения (и обычно всякий доступ) делаются в одном конце списка.

**Очередь** — линейный список, в котором все включения производятся на одном конце списка, а все исключения (и обычно всякий доступ) делаются на другом его конце.

**Дек** (очередь с двумя концами) — линейный список, в котором все включения и исключения (и обычно всякий доступ) делаются на обоих концах списка. Следовательно, дек обладает большей общностью, чем стек или очередь; он имеет некоторые общие свойства с колодой карт (в английском языке эти слова созвучны). Мы будем различать деки с ограниченным выходом или ограниченным входом; в таких деках соответственно исключение или включение допускается только на одном конце

В некоторых разделах математики слово “**очередь**” используют в более широком смысле, обозначая любой сорт списка, в котором производятся включения и исключения; указанные выше специальные случаи называются тогда “очередями с различными дисциплинами”.

**Стеки** очень часто встречаются в практике. Простым примером может служить ситуация, когда мы просматриваем множество данных и составляем список особых состояний или объектов, которые должны обрабатываться позднее; когда первоначальное множество обработано, мы возвращаемся к этому списку и выполняем последующую обработку, удаляя элементы из списка, пока список не станет пустым.

## Массивы и ортогональные списки.

Одним из простейших обобщений линейных списков являются двумерные массивы или массивы более высокой размерности.

В качестве примера рассмотрим матрицу размера m x n

$$\begin{bmatrix} A[1,1] & A[1,2] & \ldots . A[1,n] \\ A[2,1] & A[2,2] & \ldots . A[2,n] \\ \vdots & & \\ A[m,1] & A[m,2] & \ldots . A[m,n] \end{bmatrix}$$

В таком двумерном массиве, каждый узел A [j, k] принадлежит двум линейным спискам: списку “строки j” A [j, 1], A [j, 2], ... ..., A [j, n] и списку “столбца k” A [1, k], A [2, k], ..., A [m, k].

Эти **списки** ортогональных строк и столбцов по существу и определяют двумерную структуру матрицы. Аналогичные замечания применимы и к информационным массивам более высокой размерности.

# Деревья

Рассмотрим деревья, наиболее важные нелинейные структуры, встречающихся в вычислительных алгоритмах.

Структура дерева означает “разветвление”, такое отношение между “узлами”, как и в обычных деревьях.

Определим формально дерево как конечное множество $T$, состоящее из одного или более узлов, таких, что

1. Имеется один специально обозначенный узел, называемый корнем данного дерева.

2. Остальные узлы (исключая корень) содержатся в $m \geq 0$ попарно не пересекающихся множествах $T_1$ ... $T_m$, каждое из которых в свою очередь является деревом. Деревья $T_1$ ... $T_m$ поддеревьями данного корня.

Это определение «дерева» является **рекурсивным**, т. е. мы определили дерево в терминах самих же деревьев.

Разумеется, никакого порочного круга здесь не возникает, поскольку деревья с n > 1 узлами определяются с использованием понятия дерева с количеством узлов, меньшим чем n; следовательно, наше определение дает понятия деревьев с двумя узлами, тремя узлами, в конечном итоге с любым числом узлов.

В нашем определение каждый узел дерева является корнем некоторого поддерева, которое содержится в этом дереве. Число поддеревьев данного узла называется степенью этого узла.

Узел с нулевой степенью называется концевым узлом; иногда его называют листом. Не концевые узлы часто называют узлами разветвления.

Уровень узла по отношению к дереву $T$ определяется следующим образом: говорят, что корень имеет уровень 1, а другие узлы имеют уровень на единицу выше их уровня относительно содержащего их поддерева $T_j$ этого корня.

**Лес** — это множество (обычно упорядоченное), состоящее из некоторого (быть может, равного нулю) числа непересекающихся деревьев.

Пункт 2 нашего определения дерева можно было бы сформулировать иначе, сказав, что узлы дерева, за исключением корня, образуют лес. (Некоторые авторы для леса из $n$ деревьев пользуются термином “дерево с $n$-кратным корнем”).

Между нашими абстрактными лесами и деревьями лишь небольшое различие; если мы удалим у дерева корень, то получим лес, и наоборот, если к любому лесу добавить всего один узел, то получим дерево.

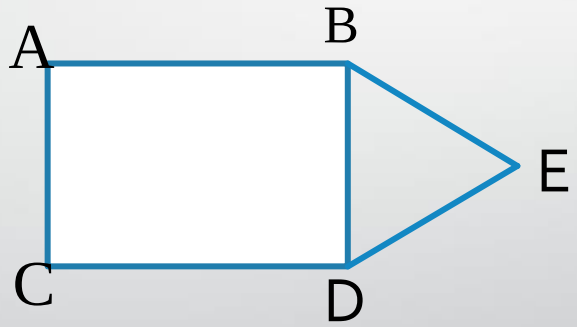
## Основные математические свойства деревьев

Структуры типа дерева стали объектом математических исследований очень давно, задолго до появления компьютеров, и за эти годы было установлено много интересных фактов относительно этих структур.

Математическая теория деревьев, которая находит важные применения в вычислительных алгоритмах определяется большим разделом математики, известного как теория графов.

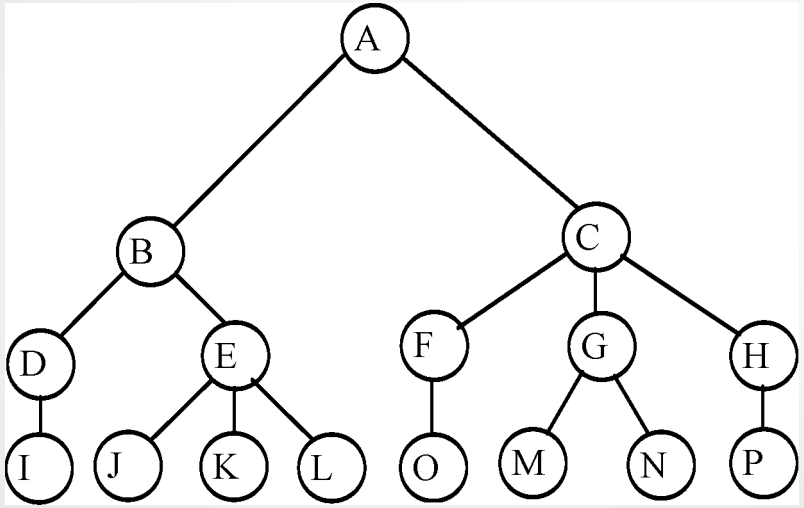
**Граф** обычно определяется как некоторое множество точек (называемых вершинами) и некоторое множество линий (называемых ребрами), соединяющих определенные пары вершин. Каждая пара вершин соединяется не больше чем одним ребром. Две вершины называются смежными, если существует ребро, соединяющее их.

Пусть V и V' — вершины, и пусть n > 0; говорят, что $(V_0, V_1, .. V_n)$ — путь длины $n$ от V до V’, если V = $V_0$, вершина $V_k$ смежна с $V_{r+1}$ при 0 ≤ k < n, а $V_n = V'$. Путь ***прост***, если вершины $V_0, V_1, \ldots V_{n-1}$ все различны между собой, а также различны и все вершины $V_1, V_2, \ldots V_n$. Граф называется ***связным***, если имеется путь между каждыми двумя вершинами этого графа. Циклом называется простой путь длины не менее трех от какой-либо вершины до нее самой. Такой граф имеет следующий вид:




У графа, пять вершин и шесть ребер. Вершина C смежна с A, но не смежна с B; от B до C имеются два пути длины два, а именно (B, A, C) и (B, D, C). В этом графе есть несколько циклов, например (B, D, E, B).

**Свободное дерево**, или “дерево без корня”, — это связный граф, не имеющий циклов. Данное определение относится как к бесконечным, так и к конечным графам, хотя при работе с компьютерами нас, естественно, интересуют главным образом конечные деревья.




Имеется много эквивалентных способов определения свободного дерева; некоторые из них отражены в следующей хорошо известной теореме:

**Теорема А**.

Пусть G — граф. Следующие утверждения эквивалентны:

a) G — свободное дерево.

b) G — связный граф, но если убрать любое из ребер, то получающийся в результате граф уже не будет связным.

c) Если $V$ и $V'$ — различные вершины графа G, то от $V$ до $V'$ имеется точно один простой путь. Кроме того, если граф G конечен и содержит точно $n$ вершин, следующие свойства также эквивалентны свойствам (a) — (c).

d) G не содержит циклов и имеет $n$ — 1 ребер.

ъ) граф G связен и имеет $n$ — 1 ребер.

Практика применения ЭВМ для решения любых задач показала, чтобы правильно использовать ЭВМ, важно добиться хорошего понимания структурных отношений, существующих между данными, способов представления таких структур в машине и методов работы с ними.

Мы рассмотрели только основные концепции данных (информации) для ЭВМ. Полное и математически строгое рассмотрение формализации данных для любых ЭВМ и любых программ приведены в фундаментальных монографиях (первое издание на русском языке вышло в 1976 году):

***Дональд Э. Кнут. Искусство программирования для ЭВМ. Том 1 - 7.***


# Информация об авторах

***Часовских Виктор Петрович*** — профессор кафедры шахматного искусства и компьютерной математики УрГЭУ, доктор технических наук, профессор
e-mail: u2007u@ya.ru

***Воронов Михаил Петрович*** — доцент кафедры шахматного искусства и компьютерной математики УрГЭУ, кандидат технических наук, доцент
e-mail: mstrk@yandex.ru

***Лабунец Валерий Григорьевич*** — профессор кафедры шахматного искусства и компьютерной математики УрГЭУ, доктор технических наук, профессор
e-mail: vlabunets05@yahoo.com

***Стариков Евгений Николаевич*** — заведующий кафедрой шахматного искусства и компьютерной математики УрГЭУ, кандидат экономических наук, доцент
e-mail: starikov_en@usue.ru

***Иванов Игорь Владимирович*** — старший преподаватель кафедры шахматного искусства и компьютерной математики УрГЭУ
e-mail: igor_v_ivanov@mail.ru

# Благодарю за внимание!