

Структурная организация данных

1. Основные понятия структур данных
2. Структуры хранения данных в оперативной памяти
3. Хранение данных на внешних носителях

1. Основные понятия структур данных

Информация – любые сведения о каком-либо объекте, событии, процессе или явлении, являющиеся объектом операций: восприятия, передачи, преобразования, хранения или использования.

Данные - информация, зафиксированная на носителе в некоторой форме, пригодной для последующей обработки, передачи и хранения.

Самое простое разделение информации на виды — по органам человека, которые ее воспринимают или воспроизводят.

Можно выделить такие виды информации как: звуковую (речь, музыка, шум) и визуальную (текст, рисунки, кинофильмы).

Несколько иначе информация подразделяется в современной вычислительной технике.

Условно приняты следующие основные виды информации (типы данных):

- ▣ **числовая** (вид информации, давший название собственно вычислительной технике);
- ▣ **текстовая** (текст, состоящий из символов — букв, цифр, знаков);
- ▣ **графическая** (графика: изображения, рисунки);
- ▣ **звуковая** (звук);
- ▣ **видеоинформация** (движущееся изображение со звуком).

Под *структурой данных* в общем случае понимают множество элементов данных и множество связей между ними. Такое определение охватывает все возможные подходы к структуризации данных.

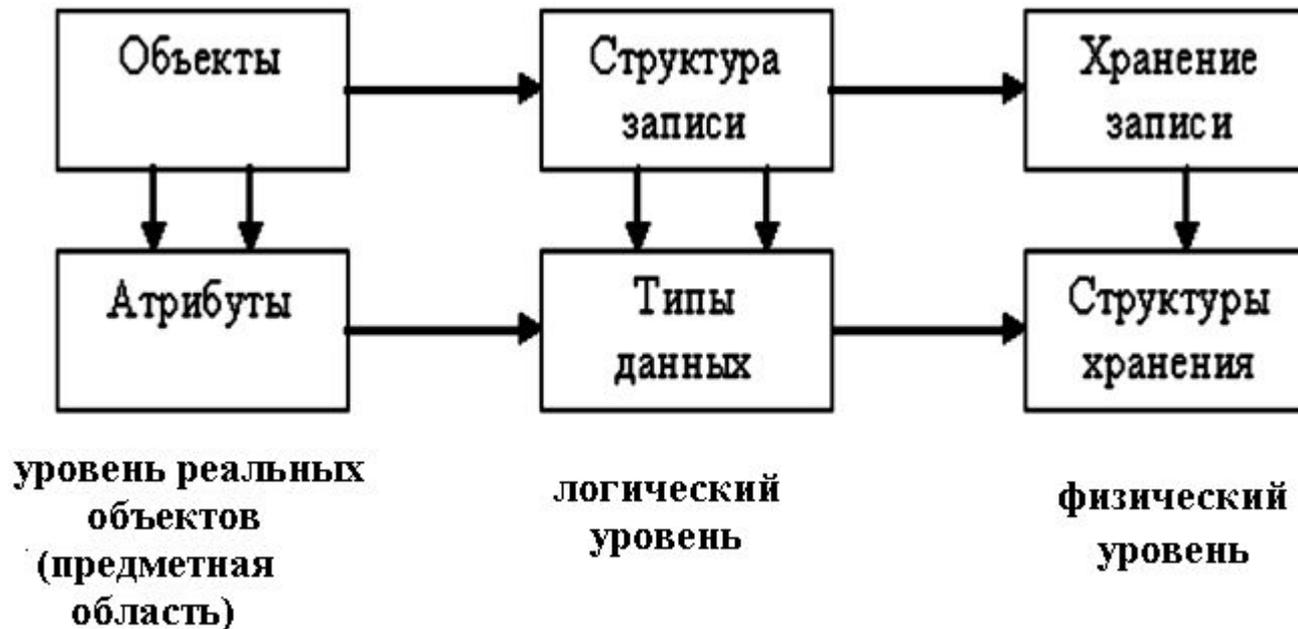
Рассмотрение структуры данных без учета ее представления в машинной памяти называется *абстрактной* или *логической структурой*.

Понятие *физическая структура* данных отражает способ физического представления данных в памяти машины. Другое название - *структура хранения*.

Отображение логической в соответствующую физическую структуру будет зависеть от самой структуры и особенностей физической среды, в которой она должна быть отражена.

Вследствие чего существуют специальные процедуры, осуществляющие отображение логической структуры в физическую и, наоборот, физической структуры в логическую.

С практической точки зрения используют трехуровневую модель представления данных



Предметная область - часть реального мира, подлежащая изучению с целью организации управления и, в конечном итоге, автоматизации.

В составе предметной области можно выделить объекты (факты, лица, события, предметы) и зафиксировать их свойства (атрибуты).

Объект — это любой различимый (имеющий имя) элемент предметной области, о котором могут быть собраны данные.

Совокупность однородных объектов образует *класс*.

Объект характеризуется совокупностью свойств – атрибутов.

Атрибут - это наименьшая поименованная единица данных, имеющая смысловое значение для пользователя (другое название - *элементарные данные*).

Каждый атрибут принадлежит определенному объекту и имеет имя - *идентификатор*.

Агрегированные данные - это совокупность атрибутов, которые имеют общее имя и на которые можно ссылаться как на единое целое. Например, агрегированное данное «имя студента» может состоять из следующих элементов данных: «фамилия», «имя», «отчество».

Второй уровень представления данных относится к их логическому описанию (*логический уровень*).

Каждый объект представлен здесь отдельной строкой - *записью* - совокупностью данных (строкой атрибутов), обрабатываемых совместно. Она называется также *логической записью*.

Входящие в запись атрибуты называются - *полями записи* (или просто - *полями*).

Поле имеет *имя* и характеризуется *типом данных*.

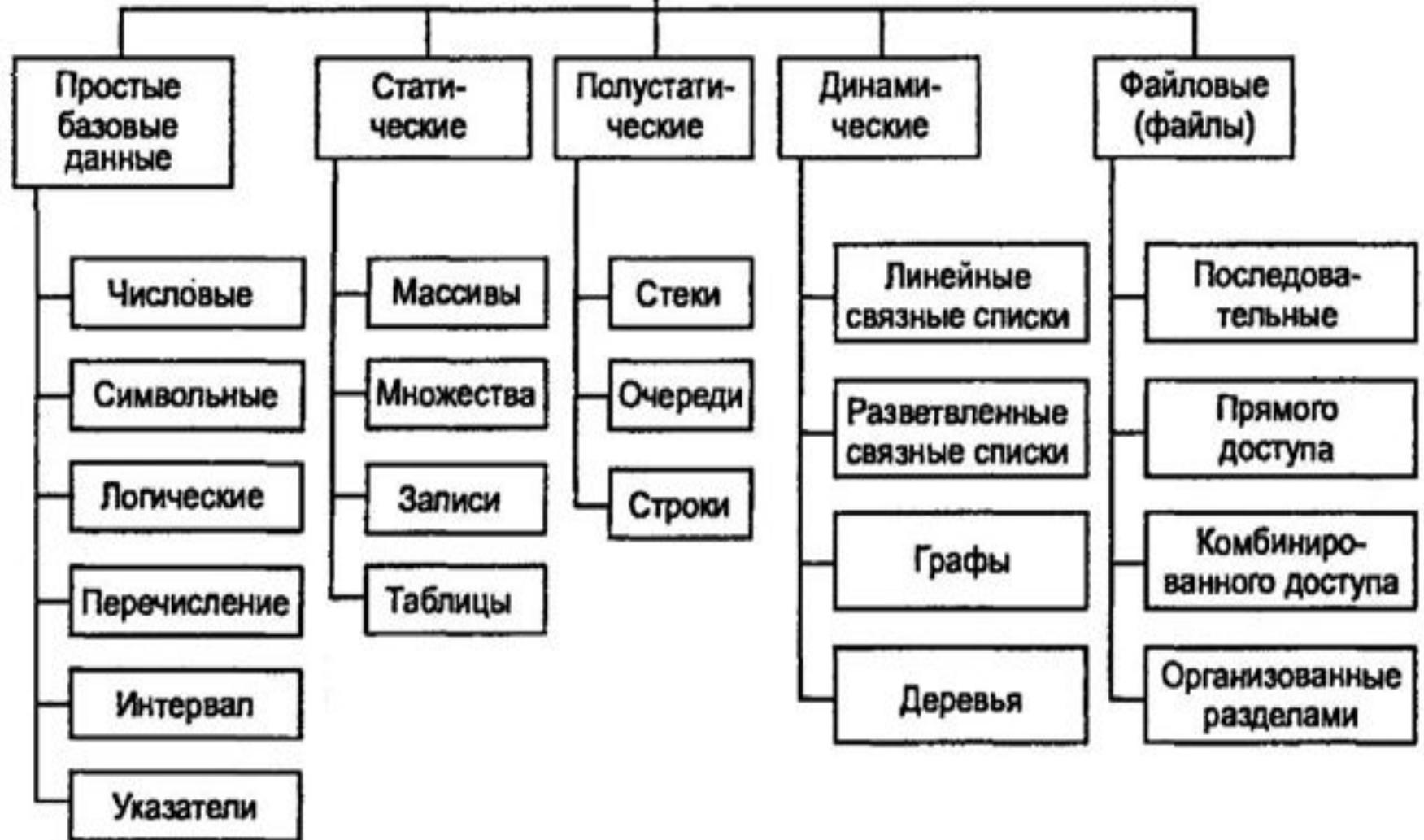
Запись является главной единицей обработки информации на этом уровне.

Третий уровень представления данных связан с проблемой хранения данных в памяти компьютера. Физические структуры хранения должны полностью отображать логическую структуру записей.

Поскольку процесс обработки и хранения данных требует использования как оперативной, так и внешней памяти, необходимо также определить оптимальные способы размещения записей в каждой из этих видов памяти.

Таким образом, *структура хранения* включает описание способов доступа к данным и средств манипулирования ими.

Структуры данных



Над всеми структурами данных могут выполняться четыре операции:

- создание;
- уничтожение;
- выбор (доступ);
- обновление.

Операция создания заключается в выделении памяти для структуры данных.

Память может выделяться

- в процессе выполнения программы при первом появлении имени переменной в исходной программе
- или на этапе компиляции.

В ряде языков для структурированных данных, конструируемых программистом, операция создания включает в себя также установку начальных значений параметров создаваемой структуры.

Операция уничтожения структур данных противоположна по своему действию операции создания.

Операция уничтожения помогает эффективно использовать память за счет ее освобождения.

Операция выбора (доступа) используется для доступа к данным внутри самой структуры и зависит от типа структуры данных, к которой осуществляется обращение.

Метод доступа — одно из наиболее важных свойств структур, особенно в связи с тем, что это свойство имеет непосредственное отношение к выбору и использованию конкретной структуры данных.

Операция обновления позволяет изменить значения данных в структуре данных.

Примером операции обновления является операция присваивания, или, более сложная форма — передача параметров.

2. Структуры хранения данных в оперативной памяти

Для размещения логических записей в оперативной памяти используются

- линейные и
- нелинейные структуры хранения данных.

В *линейных* структурах хранения все элементы равноправны.

В *нелинейных* связь между элементами определяется отношениями подчинения или какими-либо логическими условиями.

К линейным структурам относятся

- массив,
- множество,
- стек,
- очередь,
- таблица.

Массив — структура данных с **фиксированным числом элементов одного и того же типа**, идентифицируемых по индексу.

Индекс - это целое число, которое определяет позицию соответствующего элемента в массиве и используется для осуществления доступа к нему.

В массивах, как правило, отсутствуют операции добавления и удаления элементов - они могут лишь модифицироваться.

Размер массива – это число элементов, хранящихся в массиве.

Размерность массива – количество индексов.

Одномерный массив



Одномерный массив называется вектором, двухмерный - матрицей (каждый элемент в матрице определяется двумя индексами).

Под массив выделяется непрерывный блок оперативной памяти.

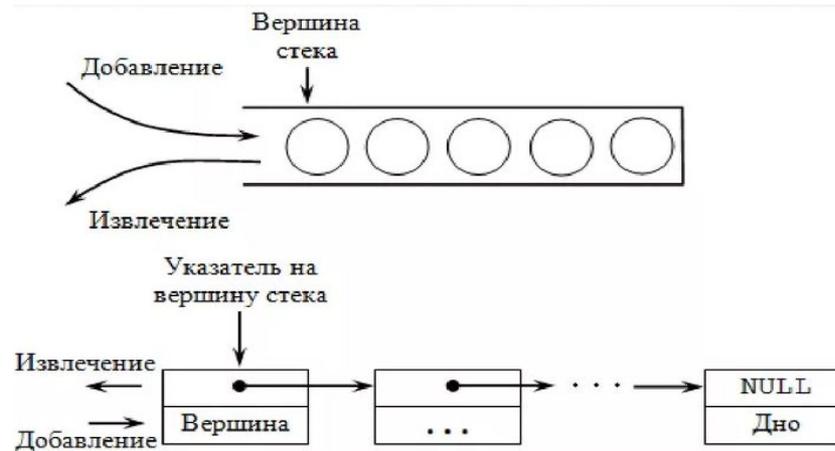
Ряд языков программирования поддерживает динамические массивы, длина которых может изменяться по ходу работы программы.

Множество — структура, представляющая собой набор **неповторяющихся данных одного и того же типа**.

Допускает добавление и исключение элементов, и выполнение стандартных операций – объединения, пересечения и нахождения разности множеств.

Стек - линейная структура **переменного размера**. Объем данных в нем может изменяться (расти и сокращаться) во время выполнения программы.

Доступ к его элементам возможен только с одного края структуры - с *вершины стека* и данные обрабатываются по принципу «последним пришел - первым ушел» (LIFO).



Стек - это структура с ограниченным доступом поскольку невозможно непосредственное обращение к произвольному элементу.

Очередь, как и стек, является линейной структурой переменного размера.

Включение элементов производится только с ее конца, а исключение элементов допускается лишь с ее начала.

Т.е. информация в очередях обрабатывается по принципу «первым пришел - первым ушел» (FIFO).

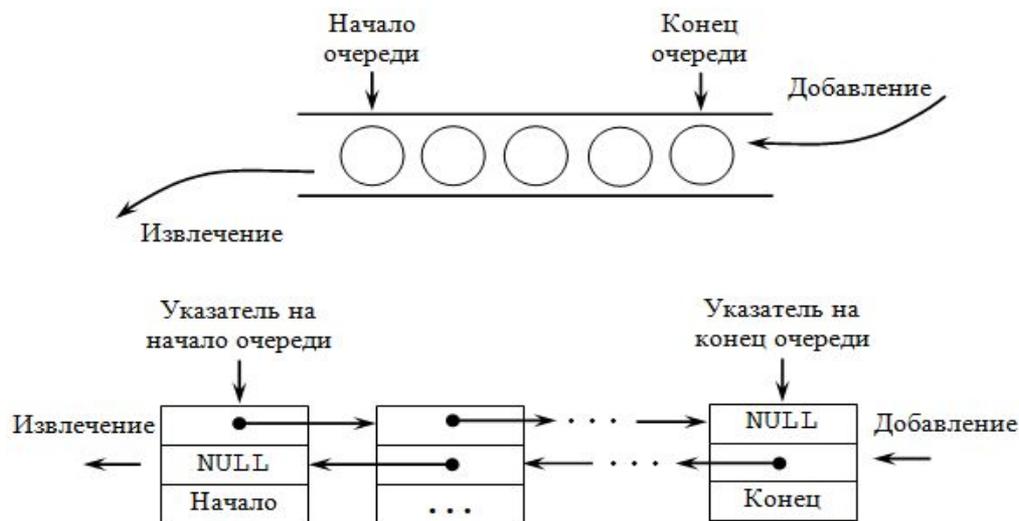


Таблица представляет собой линейную структуру **переменного размера**.

Элементами таблицы являются **строки (записи)**, включающие фиксированный набор атрибутов (полей), задаваемых при создании структуры.

В таблице имеется возможность непосредственного обращения к любой из ее строк по значению так называемого *ключа (ключевого поля)*.

Добавлять и удалять можно только **строку целиком**.

В большинстве случаев не допускается наличие пустых полей.

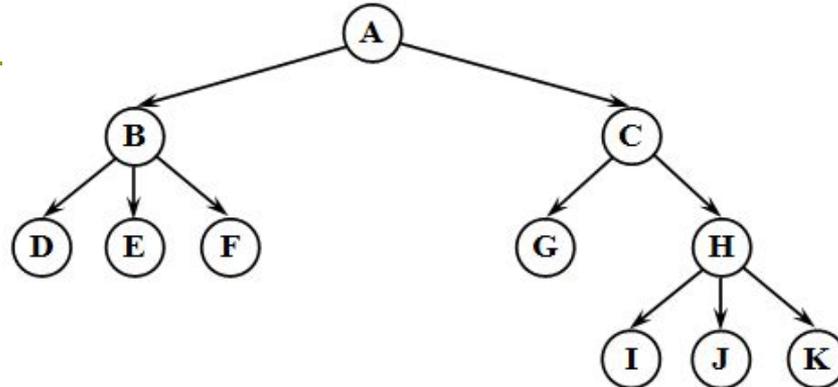
Отношения между объектами предметной области часто носят нелинейный характер.

Это могут быть отношения, определяемые логическими условиями, а также отношения типа «один ко многим» или «многие ко многим».

К нелинейным структурам принадлежат

- деревья,
- графы и
- списковые структуры.

Дерево – это структура данных, представляющая собой совокупность элементов и отношений, образующих иерархическую структуру.



Т.е. древовидные структуры используются для описания отношения «один ко многим».

Каждый элемент дерева называется *вершиной* (узлом) дерева. Вершины дерева соединены направленными дугами, которые называют *ветвями* дерева. Вершины относятся к определенному уровню. Начальный узел дерева называют *корнем* дерева, ему соответствует нулевой уровень. *Листьями* дерева называют вершины, в которые входит одна ветвь и не выходит ни одной ветви.

Каждое дерево обладает следующими свойствами:

- существует узел, в который не входит ни одной дуги (корень);
- в каждую вершину, кроме корня, входит одна дуга. Все вершины, в которые входят ветви, исходящие из одной общей вершины, называются *потомками*, а сама вершина – *предком*.

Для каждого предка может быть выделено несколько потомков.

Уровень потомка на единицу превосходит уровень его предка.

Корень дерева не имеет предка, а листья дерева не имеют потомков.

Высота (глубина) дерева определяется номером уровня, на котором располагаются его листья.

Степенью вершины в дереве называется количество дуг, которое из нее выходит.

Степень дерева равна максимальной степени вершины, входящей в состав дерева. При этом листьями в дереве являются вершины, имеющие степень нуль.

По величине степени дерева различают два типа деревьев:

- двоичные – степень дерева не более двух;
- сильноветвящиеся – степень дерева произвольная.

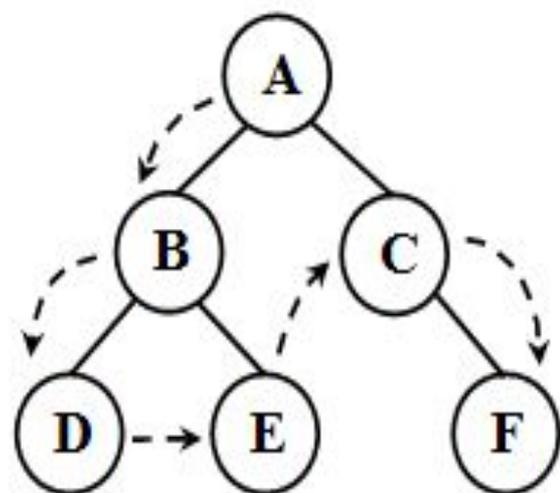
Обход дерева – это упорядоченная последовательность вершин дерева, в которой каждая вершина встречается только один раз.

При обходе все вершины дерева должны посещаться в определенном порядке.

Существует несколько способов обхода всех вершин дерева:

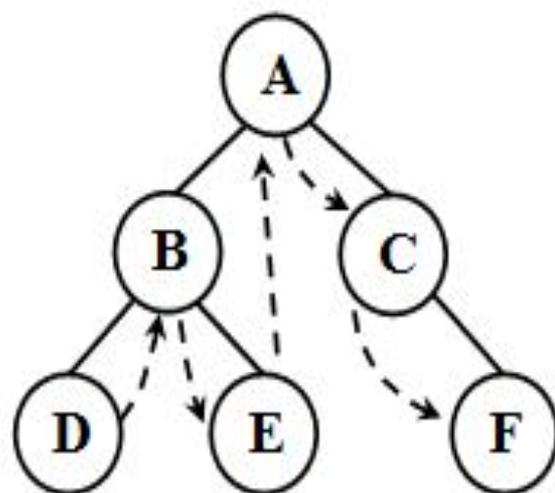
- прямой (нисходящий);
- симметричный (комбинированный);
- обратный (восходящий).

Прямой



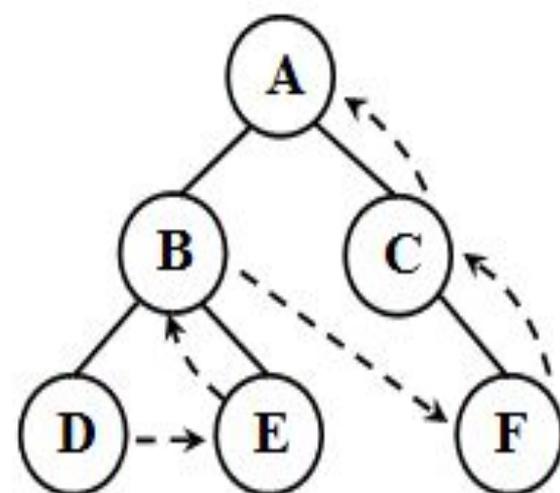
ABDECF

Симметричный



DBEACF

Обратный

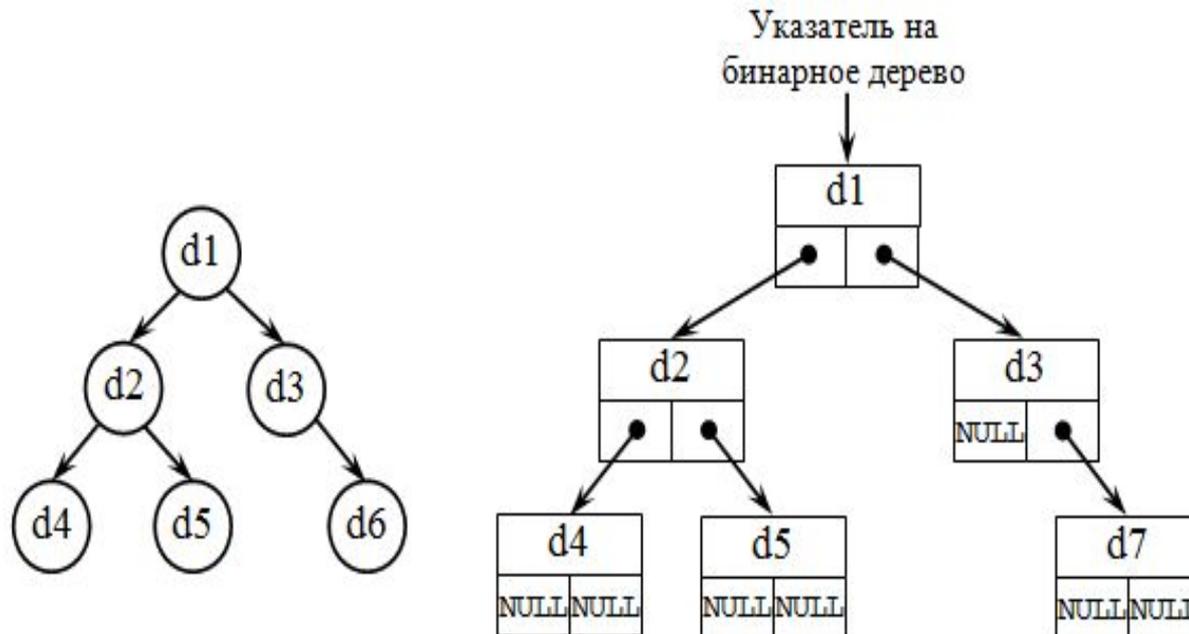


DEBFCA

Бинарное (двоичное) дерево – это структура данных, представляющее собой дерево, в котором каждая вершина имеет не более двух потомков.

Таким образом, бинарное дерево состоит из элементов, каждый из которых содержит информационное поле и не более двух ссылок на различные бинарные поддеревья.

На каждый элемент дерева имеется ровно одна ссылка.

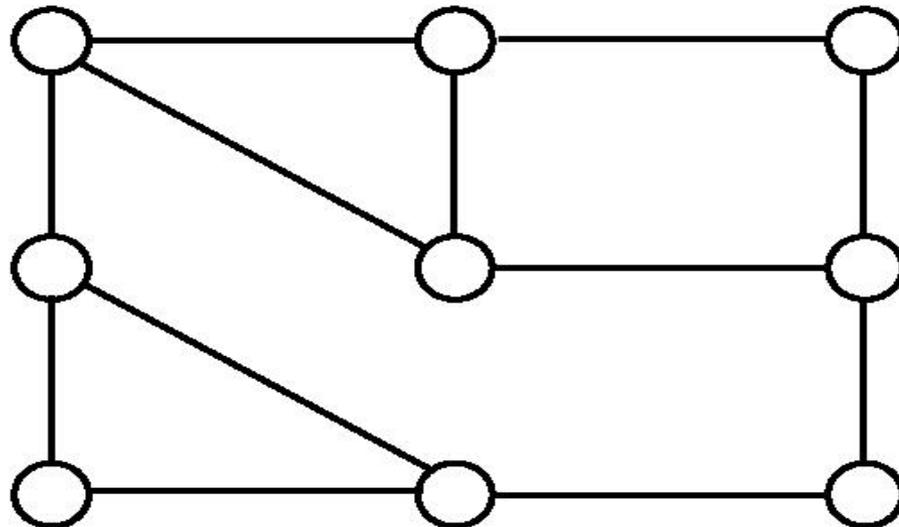


Отношения «многие ко многим» носят более универсальный характер и описываются *структурой графа*.

Граф общего вида состоит из ряда вершин и ребер, связывающих пары вершин.

Вершины графа описывают определенные объекты, а ребра соответствуют отношениям между ними.

Модель данных, имеющую вид графа, называют сетью.



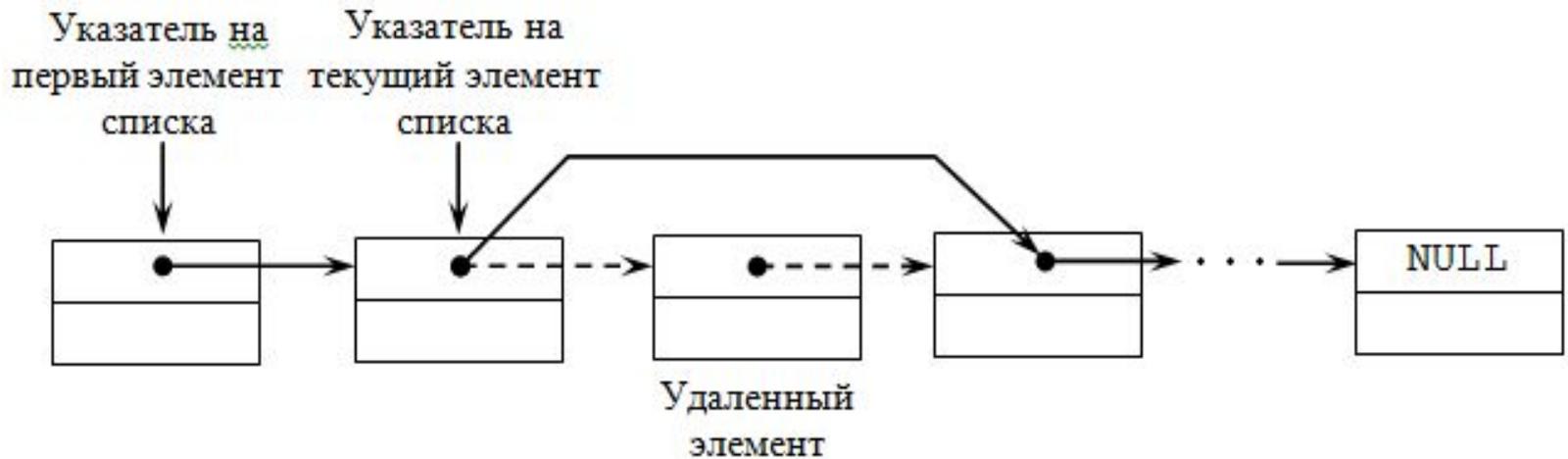
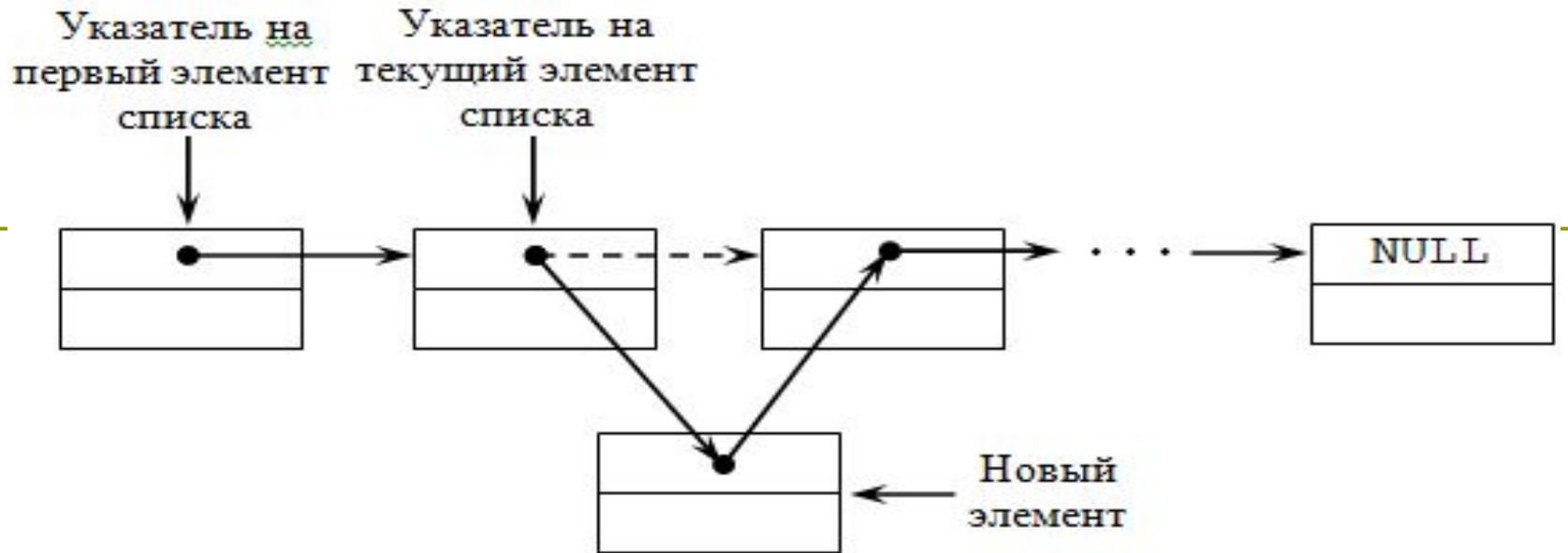
Каждый список имеет особый элемент, называемый *указателем начала списка*, который обычно по содержанию отличен от остальных элементов.

В поле указателя последнего элемента списка находится специальный признак NULL, свидетельствующий о конце списка.

Длина списка равна числу элементов, содержащихся в списке, список нулевой длины называется *пустым списком*.

Из всего многообразия связанных списков можно выделить следующие основные:

- однонаправленные (односвязные) списки;
- двунаправленные (двусвязные) списки;
- циклические (кольцевые) списки.



Вставка и удаление элемента списка

3. Хранение данных на внешних носителях

При размещении информации на внешних носителях (физическом уровне ее хранения) единицей обработки и хранения информации является *файл*.

Под *файлом* понимается именованная часть внешней памяти, предназначенная для хранения совокупности данных.

Файл состоит из *записей (физических записей)*. Запись характеризуется длиной, выраженной в байтах.

Запись состоит из данных, которые передаются между оперативной и внешней памятью за одну операцию чтения или записи данных.

Чтение данных – это передача данных из внешней памяти (файла) в оперативную память.

Запись данных – это передача данных из оперативной памяти во внешнюю память (файл).
К файловым структурам возможны два варианта доступа - *последовательный* или *произвольный (прямой)*.

При последовательном доступе (режиме обработки) записи файла передаются с внешнего запоминающего устройства в оперативную память в том порядке, в котором они размещены на носителе.

В режиме произвольного доступа записи могут извлекаться из файла так, как этого требует конкретная прикладная программа.

Для размещения данных на внешних носителях используют следующие *типы файловых структур*:

- последовательные;
- прямые;
- индексно-последовательные;
- библиотечные.

При формировании *последовательного* файла записи располагаются на носителе в порядке их поступления.

Каждая очередная запись размещается на свободном месте сразу же за последней записью файла (добавление в конец файла).

Удаление записи можно выполнить только путем перезаписи нужных данных на новое место.

Поиск в таких файлах осуществляется последовательным считыванием файла с начала и сравнением «всего» с искомым. Так же и обращение к определённому участку файла каждый раз требует чтения файла с начала.

К записям последовательного файла возможен только последовательный доступ, осуществляемый через *указатель текущей записи*, указывающий на ту запись файла, которая читалась или записывалась последней.

Момент завершения просмотра файла обнаруживается по специальному признаку конца файла (eof).

Все типы внешних запоминающих устройств обеспечивают возможность последовательной организации файла.

Прямые файлы (файлы прямого доступа) хранят информацию в структурированном (для поиска и обращения) виде.

Дисковое пространство, занимаемое таким файлом, поделено на одинаковые участки (записи), имеющие одинаковую структуру полей. При этом существует непосредственная связь между адресом (ключом) записи и ее местоположением на носителе.

Поиск в таких файлах осуществляется в области адресов (ключей) и завершается обращением непосредственно к искомому участку.

В произвольном режиме обработки (режиме прямого доступа) значение ключа преобразуется в адрес записи и, в зависимости от характера обработки, запись создается, удаляется, модифицируется, извлекается по этому адресу.

Файл, преобразованный (отсортированный) по какому-либо ключевому полю файл называется *инвертированным*.

При обработке файла по нескольким ключам приходится создать соответствующее количество инвертированных файлов.

Для рационализации обработки данных стали использовать *индексно-последовательные файлы* - совокупность файла данных и одного или нескольких индексных файлов.

В последних хранятся не сами исходные данные, а только номера (индексы) записей исходного файла, определяющие порядок его обработки по определенному полю (ключу).

Индексный файл обрабатывается в последовательном режиме, а файл данных - в режиме прямого доступа.

N записи			Ключ	Ссылка на номер записи
1	10-44/01	Иванов	01-30/01	7
2	11-44/01	Степанов	02-40/02	8
3	20-44/02	Кустов	05-40/00	5
4	45-32/01	Чернов	06-40/00	6
5	06-40/00	Трусов	06-50/01	9
6	05-40/00	Терпухов	07-35/00	10
7	01-30/01	Павлов	10-44/01	1
8	02-40/02	Мухина	11-44/01	2
9	06-50/01	Артемьев	15-40/01	11
10	07-35/00	Васильев	17-40/02	12
11	15-40/01	Васильева	20-44/02	3
12	17-40/02	Уваров	45-32/01	4
13	48-40/03	Шишкин	48-40/03	13
14	50-44/00	Удалов	50-44/00	14

Основная область
Файл данных

Индексная область

Операция добавления осуществляет запись в конец файла данных.

Индексный файл при этом перестраивается (создается заново), чтобы не нарушать его упорядоченности.

При удалении записи возникает следующая последовательность действий:

- запись в файле данных помечается как удаленная,
 - индексный файл перестраиваются - соответствующий индекс уничтожается физически
- Количество *физических записей* в файле данных не изменяется, а количество *активных записей* в индексном файле уменьшается на одну.

Файл с библиотечной организацией (с разделами) состоит из последовательно организованных разделов, каждый из которых имеет имя и содержит одну или несколько записей.

В начале файла размещается специальный служебный раздел, называемый *справочником* или *оглавлением* файла.

Оглавление состоит из элементов, каждый элемент содержит *имя раздела* и его *адрес* (ссылку на первую запись каждого раздела).

К каждому разделу библиотечного файла осуществляется прямой доступ через оглавление. Записи в разделах просматриваются последовательно в порядке их физического размещения на носителе.

Корректировка библиотечного файла заключается в добавлении и удалении разделов.

Новые разделы размещаются в конце файла на свободном участке памяти, и сведения о них заносятся в оглавление.

При удалении раздела удаляется соответствующий элемент оглавления, так что доступ к записям раздела становится невозможным.

Обслуживание библиотек, т.е. помещение в них новых разделов, удаление ранее записанных, переименование и т.п., производится специальной программой под названием «Библиотекарь» (library).