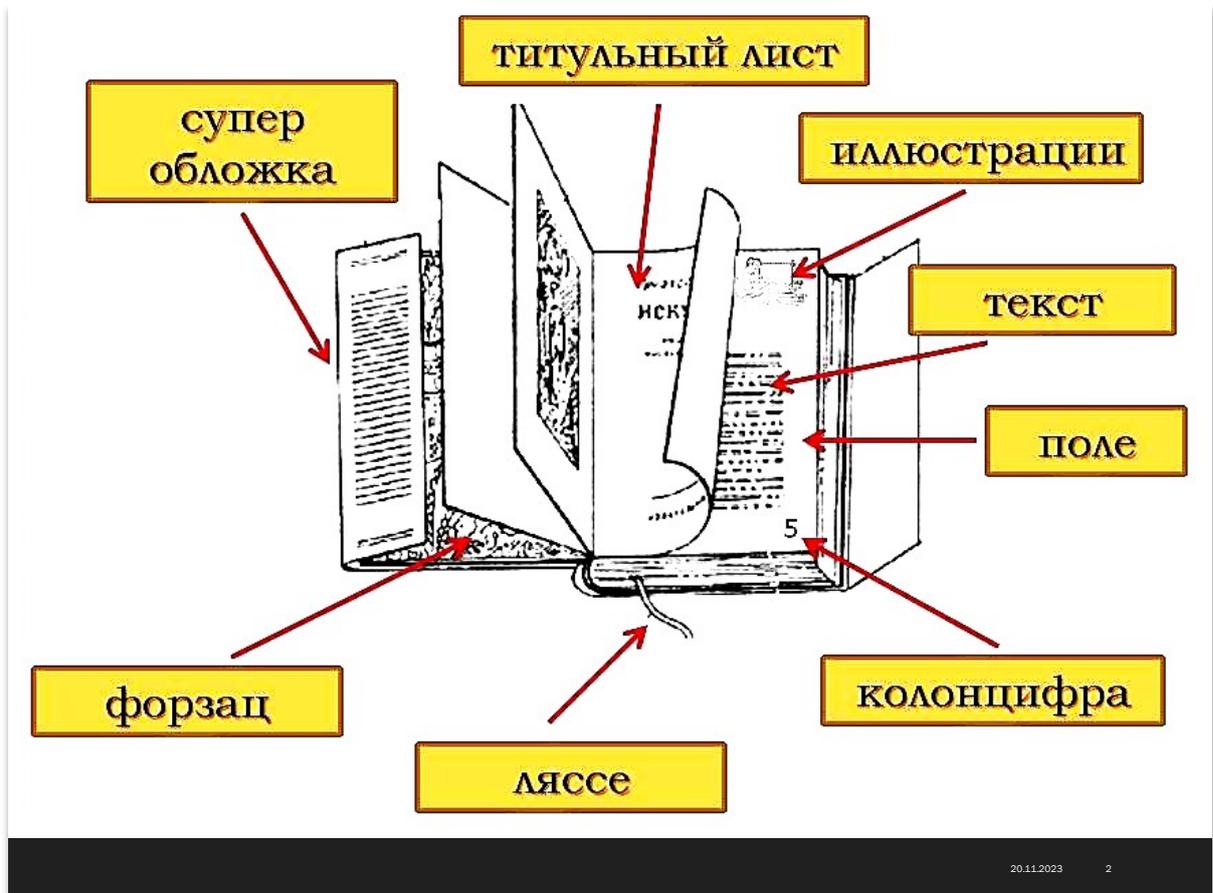


# Информатика

## Лекция 2

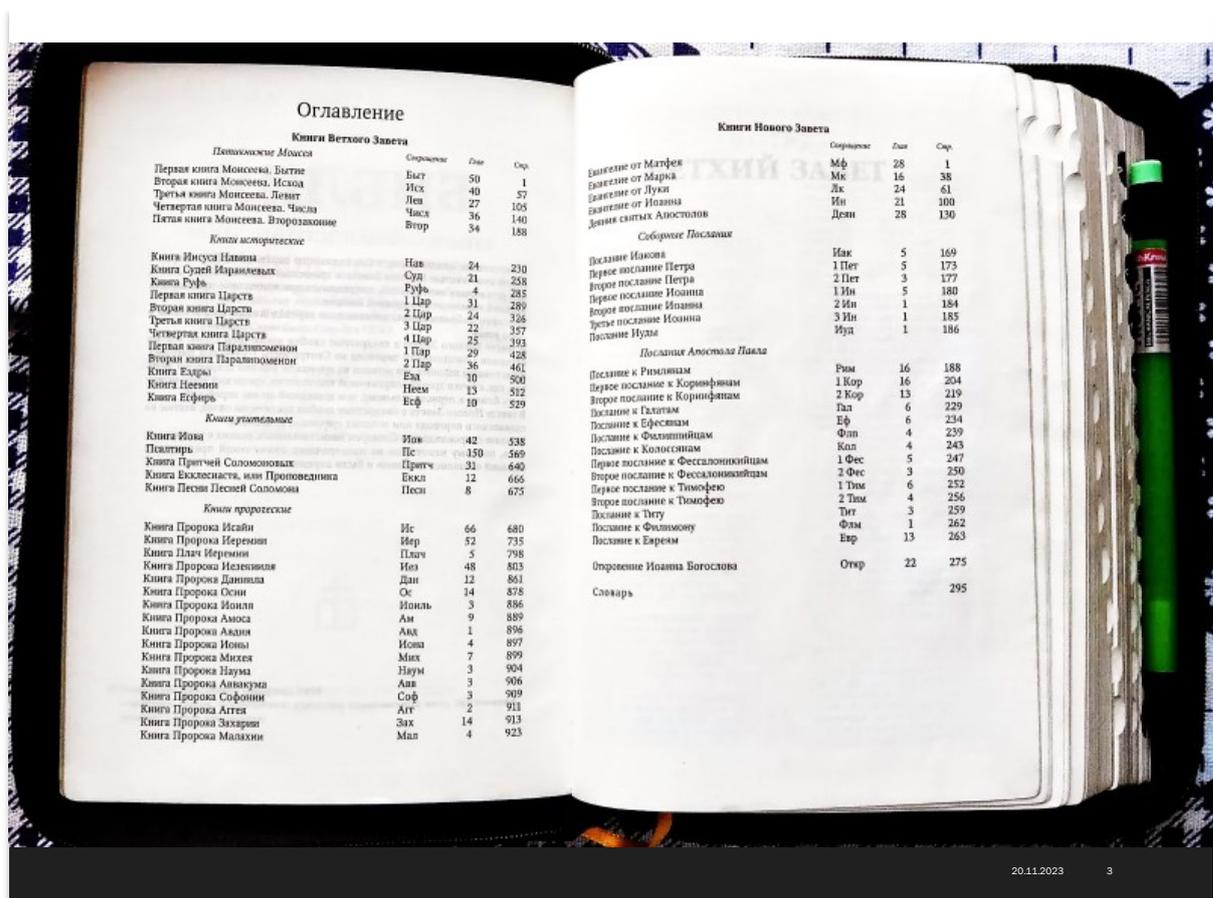
### Основные структуры данных

Работа с большими наборами данных автоматизируется проще, когда данные упорядочены, то есть образуют заданную структуру. Существует три основных типа структур данных: линейная, иерархическая и табличная. Их можно рассмотреть на примере обычной книги.



Если разобрать книгу на отдельные листы и перемешать их, книга потеряет свое назначение. Она по-прежнему будет представлять набор данных, но подобрать адекватный метод для получения из нее информации весьма непросто. (Еще хуже дело будет обстоять, если из книги вырезать каждую букву отдельно, — в этом случае вряд ли вообще найдется адекватный метод для ее прочтения.)

Если же собрать все листы книги в правильной последовательности, мы получим простейшую структуру данных — линейную. Такую книгу уже можно читать, хотя для поиска нужных данных ее придется прочитать подряд, начиная с самого начала, что не всегда удобно.



Для быстрого поиска данных существует иерархическая структура. Так, например, книги разбивают на части, разделы, главы, параграфы и т. п. Элементы структуры более низкого уровня входят в элементы структуры более высокого уровня: разделы состоят из глав, главы из параграфов и т. д.

Для больших массивов поиск данных в иерархической структуре намного проще, чем в линейной, однако и здесь необходима навигация, связанная с необходимостью просмотра. На практике задачу упрощают тем, что в большинстве книг есть вспомогательная перекрестная таблица, связывающая элементы иерархической структуры с элементами линейной структуры, то есть связывающая разделы, главы и параграфы с номерами страниц.

В книгах с простой иерархической структурой, рассчитанных на последовательное чтение, эту таблицу принято называть оглавлением, а в книгах со сложной структурой, допускающей выборочное чтение, ее называют содержанием.

## Линейные структуры (списки данных, векторы данных)

Линейные структуры — это хорошо знакомые нам списки. Список — это простейшая структура данных, отличающаяся тем, что каждый элемент

данных однозначно определяется своим номером в массиве. Проставляя номера на отдельных страницах рассыпанной книги, мы создаем структуру списка. Обычный журнал посещаемости занятий, например, имеет структуру списка, поскольку все студенты группы зарегистрированы в нем под своими уникальными номерами. Мы называем номера уникальными потому, что в одной группе не могут быть зарегистрированы два студента с одним и тем же номером.

**Линейные структуры  
(списки данных, векторы данных)**

№ п/п	Фамилия, Имя, Отчество
1	Аистов Александр Алексеевич
2	Бобров Борис Борисович
3	Воробьева Валентина Владиславовна
.....	
27	Сорокин Сергей Семенович

$a(n-1)$

Аистов Александр Алексеевич \* Бобров Борис Борисович \* Воробьева Валентина Владиславовна \*... \* Сорокин Сергей Семенович

20.11.2023 4

При создании любой структуры данных надо решить два вопроса: как разделять элементы данных между собой и как разыскивать нужные элементы. В журнале посещаемости, например, это решается так: каждый новый элемент списка заносится с новой строки, то есть разделителем является конец строки. Тогда нужный элемент можно разыскать по номеру строки.

№ п/п Фамилия, Имя, Отчество

1 Аистов Александр Алексеевич

2 Бобров Борис Борисович

3 Воробьева Валентина Владиславовна

.....

27 Сорокин Сергей Семенович

Разделителем может быть и какой-нибудь специальный символ. Нам хорошо известны разделители между словами — это пробелы. В русском и во многих европейских языках общепринятым разделителем предложений является точка. В рассмотренном нами классном журнале в качестве разделителя можно использовать любой символ, который не встречается в самих данных, например символ «\*». Тогда список выглядел бы так:

Аистов Александр Алексеевич \* Бобров Борис Борисович \* Воробьева Валентина Владиславовна \*... \* Сорокин Сергей Семенович

В этом случае для розыска элемента с номером  $n$  надо просмотреть список начиная с самого начала и пересчитать встретившиеся разделители. Когда будет отсчитано  $n-1$  разделителей, начнется нужный элемент. Он закончится, когда будет встречен следующий разделитель.

Еще проще можно действовать, если все элементы списка имеют равную длину. В этом случае разделители в списке вообще не нужны. Для розыска элемента с номером  $n$  надо просмотреть список с самого начала и отсчитать  $a(n-1)$  символ, где  $a$  — длина одного элемента. Со следующего символа начнется нужный элемент.

Его длина тоже равна  $a$ , поэтому его конец определить нетрудно. Такие упрощенные списки, состоящие из элементов равной длины, называют векторами данных. Работать с ними особенно удобно.

Таким образом, линейные структуры данных (списки) — это упорядоченные структуры, в которых адрес элемента однозначно определяется его номером.

## **Табличные структуры (таблицы данных, матрицы данных)**

С таблицами данных мы тоже хорошо знакомы, достаточно вспомнить всем известную таблицу умножения. Табличные структуры отличаются от списочных тем, что элементы данных определяются адресом ячейки, который состоит не из одного параметра, как в списках, а из нескольких. Для таблицы умножения, например, адрес ячейки определяется номерами строки и столбца. Нужная ячейка находится на их пересечении, а элемент выбирается из ячейки.

## Табличная структура

Планета	Расстояние до Солнца, а.е.	Относительная масса	Количество спутников
Меркурий	0,39	0,056	0
Венера	0,67	0,88	0
Земля	1,0	1,0	1
Марс	1,51	0,1	2
Юпитер	5,2	318	16

Меркурий\*0,39\*0,056\*0#Венера\*0,67\*0,88\*0#Земля\*1,0\*1,0\*1#Марс\*1,51\*0,1\*2#...

При хранении табличных данных количество разделителей должно быть больше, чем для данных, имеющих структуру списка. Например, когда таблицы печатают в книгах, строки и столбцы разделяют графическими элементами — линиями вертикальной и горизонтальной разметки.

Если нужно сохранить таблицу в виде длинной символьной строки, используют один символ-разделитель между Элементами, принадлежащими одной строке, и другой разделитель для отделения строк, например так:

Меркурий\*0,39\*0,056\*0#Венера\*0,67\*0,88\*0#Земля\*1,0\*1,0\*1#Марс\*1,51\*0,1\*2#...

Для розыска элемента, имеющего адрес ячейки (m, n), надо просмотреть набор данных с самого начала и пересчитать внешние разделители. Когда будет отсчитан m-1 разделитель, надо пересчитывать внутренние разделители. После того как будет найден n-1 разделитель, начнется нужный элемент. Он закончится, когда будет встречен любой очередной разделитель.

Еще проще можно действовать, если все элементы таблицы имеют равную длину.

Такие таблицы называют матрицами. В данном случае разделители не нужны, поскольку все элементы имеют равную длину и количество их известно. Для розыска элемента с адресом (m, n) в матрице, имеющей M строк и N столбцов, надо просмотреть ее с самого начала и отсчитать а  $[N(m-1) +$

(и- 1)] символ, где  $a$  — длина одного элемента. Со следующего символа начнется нужный элемент. Его длина тоже равна  $a$ , поэтому его конец определить нетрудно.

Таким образом, табличные структуры данных (матрицы) — это упорядоченные структуры, в которых адрес элемента определяется номером строки и номером столбца, на пересечении которых находится ячейка, содержащая искомый элемент.

## Многомерные таблицы

---

Номер факультета:	3
Номер курса (на факультете):	2
Номер специальности (на курсе):	2
Номер группы в потоке одной специальности:	1
Номер учащегося в группе:	19

23.11.2023 6

Многомерные таблицы. Выше мы рассмотрели пример таблицы, имеющей два измерения (строка и столбец), но в жизни нередко приходится иметь дело с таблицами, у которых количество измерений больше. Вот пример таблицы, с помощью которой может быть организован учет учащихся.

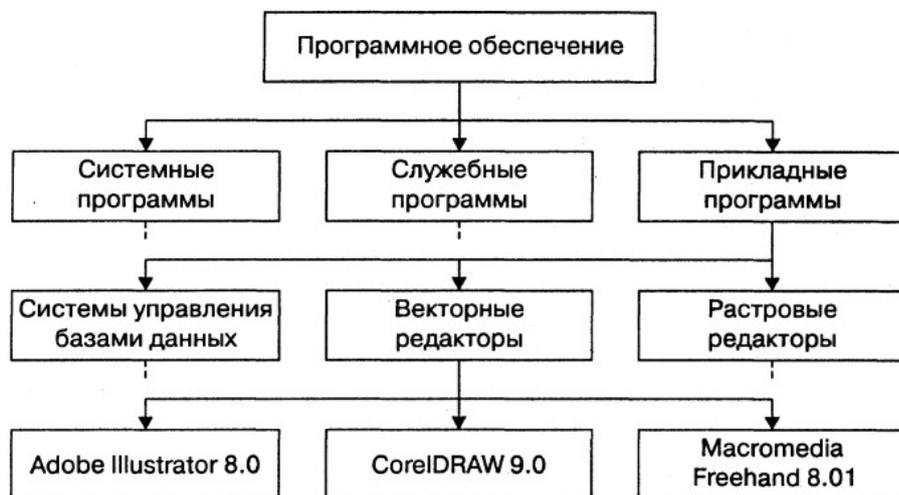
Номер факультета:	3
Номер курса (на факультете):	2
Номер специальности (на курсе):	2
Номер группы в потоке одной специальности:	1
Номер учащегося в группе:	19

Размерность такой таблицы равна пяти, и для однозначного отыскания данных об учащемся в подобной структуре надо знать все пять параметров (координат).

## Иерархические структуры данных

Нерегулярные данные, которые трудно представить в виде списка или таблицы, часто представляют в виде иерархических структур. С подобными структурами мы очень хорошо знакомы по обыденной жизни. Иерархическую структуру имеет \* система почтовых адресов. Подобные структуры также широко применяют в научных систематизациях и всевозможных классификациях.

### Пример иерархической структуры данных



23.11.2023 7

В иерархической структуре адрес каждого элемента определяется путем доступа (маршрутом), ведущим от вершины структуры к данному элементу. Вот, например, как выглядит путь доступа к команде, запускающей программу Калькулятор (стандартная программа компьютеров, работающих в операционной системе Windows):

Пуск ► Программы ► Стандартные ► Калькулятор.

## Путь доступа к команде, запускающей программу Калькулятор в WINDOWS

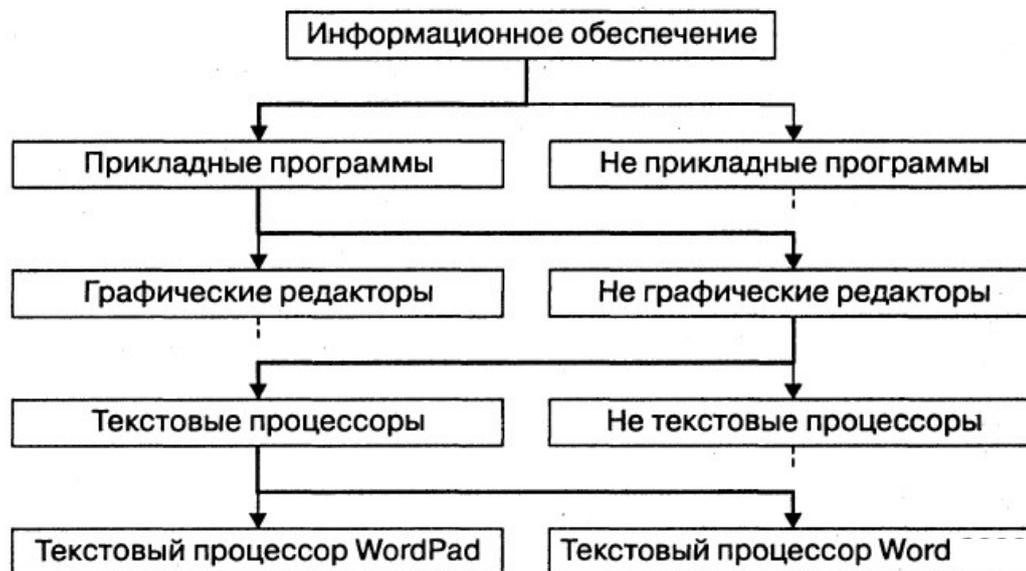
---

Пуск ► Программы ► Стандартные ► Калькулятор

23.11.2023 8

Дихотомия данных. Основным недостатком иерархических структур данных является увеличенный размер пути доступа. Очень часто бывает так, что длина маршрута оказывается больше, чем длина самих данных, к которым он ведет. Поэтому в информатике применяют методы для регуляризации иерархических структур с тем, чтобы сделать путь доступа компактным. Один из методов получил название дихотомии. Его суть понятна из примера, представленного на слайде.

Пример, поясняющий принцип действия метода дихотомии



1010

23.11.2023 9

В иерархической структуре, построенной методом дихотомии, путь доступа к любому элементу можно представить как путь через рациональный лабиринт с поворотами налево (0) или направо (1) и, таким образом, выразить путь доступа в виде компактной двоичной записи. В нашем примере путь доступа к текстовому процессору Word выразится следующим двоичным числом: 1010.

#### Упорядочение структур данных

Списочные и табличные структуры являются простыми. Ими легко пользоваться, поскольку адрес каждого элемента задается числом (для списка), двумя числами (для двумерной таблицы) или несколькими числами для многомерной таблицы. Они также легко упорядочиваются. Основным методом упорядочения является сортировка. Данные можно сортировать по любому избранному критерию, например: по алфавиту, по возрастанию порядкового номера или по возрастанию какого-либо параметра.

Несмотря на многочисленные удобства, у простых структур данных есть и недостаток — их трудно обновлять. Если, например, перевести студента из одной группы в другую, изменения надо вносить сразу в два журнала посещаемости; при этом в обоих журналах будет нарушена списочная структура. Если переведенного студента вписать в конец списка группы, нарушится упорядочение по алфавиту, а если его вписать в соответствии с

алфавитом, то изменятся порядковые номера всех студентов, которые следуют за ним.

Таким образом, при добавлении произвольного элемента в упорядоченную структуру списка может происходить изменение адресных данных у других элементов. В журналах успеваемости это пережить нетрудно, но в системах, выполняющих автоматическую обработку данных, нужны специальные методы для решения этой проблемы.

Иерархические структуры данных по форме сложнее, чем линейные и табличные, но они не создают проблем с обновлением данных. Их легко развивать путем создания новых уровней. Даже если в учебном заведении будет создан новый факультет, это никак не отразится на пути доступа к сведениям об учащихся прочих факультетов.

Недостатком иерархических структур является относительная трудоемкость записи адреса элемента данных и сложность упорядочения. Часто методы упорядочения в таких структурах основываются на предварительной индексации, которая заключается в том, что каждому элементу данных присваивается свой уникальный индекс, который можно использовать при поиске, сортировке и т. п. Ранее рассмотренный принцип дихотомии на самом деле является одним из методов индексации данных в иерархических структурах. После такой индексации данные легко разыскиваются по двоичному коду связанного с ними индекса.

Адресные данные. Если данные хранятся не как попало, а в организованной структуре (причем любой), то каждый элемент данных приобретает новое свойство (параметр), который можно назвать адресом. Конечно, работать с упорядоченными данными удобнее, но за это приходится платить их размножением, поскольку адреса элементов данных — это тоже данные и их тоже надо хранить и обрабатывать.

## **Файлы и файловая структура**

### **Единицы представления данных**

Существует множество систем представления данных. С одной из них, принятой в информатике и вычислительной технике, двоичным кодом, мы познакомились выше. Наименьшей единицей такого представления является бит (двоичный разряд).

Совокупность двоичных разрядов, выражающих числовые или иные данные, образует некий битовый рисунок. Практика показывает, что с битовым представлением удобнее работать, если этот рисунок имеет регулярную форму. В настоящее время в качестве таких форм используются группы из восьми битов, которые называются байтами.

# Байт

Десятичное число	Двоичное число	Байт
1	1	0000 0001
2	10	0000 0010
255	11111111	11111111

23.11.2023 10

Понятие о байте как группе взаимосвязанных битов появилось вместе с первыми образцами электронной вычислительной техники. Долгое время оно было машинно-зависимым, то есть для разных вычислительных машин длина байта была разной. Только в конце 60-х годов понятие байта стало универсальным и машинно-независимым.

Выше мы видели, что во многих случаях целесообразно использовать не восьмиразрядное кодирование, а 16-разрядное, 24-разрядное, 32-разрядное и более.

Группа из 16 взаимосвязанных бит (двух взаимосвязанных байтов) в информатике называется словом. Соответственно, группы из четырех взаимосвязанных байтов (32 разряда) называются удвоенным словом, а группы из восьми байтов (64 разряда) — учетверенным словом. Пока, на сегодняшний день, такой системы обозначения достаточно.

## Единицы измерения данных

Существует много различных систем и единиц измерения данных. Каждая научная дисциплина и каждая область человеческой деятельности может использовать свои, наиболее удобные или традиционно устоявшиеся единицы.

В информатике для измерения данных используют тот факт, что разные типы данных имеют универсальное двоичное представление и потому вводят свои единицы данных, основанные на нем.

Наименьшей единицей измерения является байт. Поскольку одним байтом, как правило, кодируется один символ текстовой информации, то для текстовых документов размер в байтах соответствует лексическому объему в символах.

Более крупная единица измерения — килобайт (Кбайт). Условно можно считать, что 1 Кбайт примерно равен 1000 байт. Условность связана с тем, что для вычислительной техники, работающей с двоичными числами, более удобно представление чисел в виде степени двойки и потому на самом деле 1 Кбайт равен  $2^{10}$  байт (1024 байт). Однако всюду, где это не принципиально, с инженерной погрешностью (до 3 %) «забывают» о «лишних» байтах.

В килобайтах измеряют сравнительно небольшие объемы данных. Условно можно считать, что одна страница неформатированного машинописного текста составляет около 2 Кбайт.

Более крупные единицы измерения данных образуются добавлением кратных префиксов мега-, гига- тера-; в более крупных единицах пока нет практической надобности.

## Единицы измерения

---

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

$$1 \text{ Гбайт} = 1024 \text{ Мбайт} = 10^{30} \text{ байт}$$

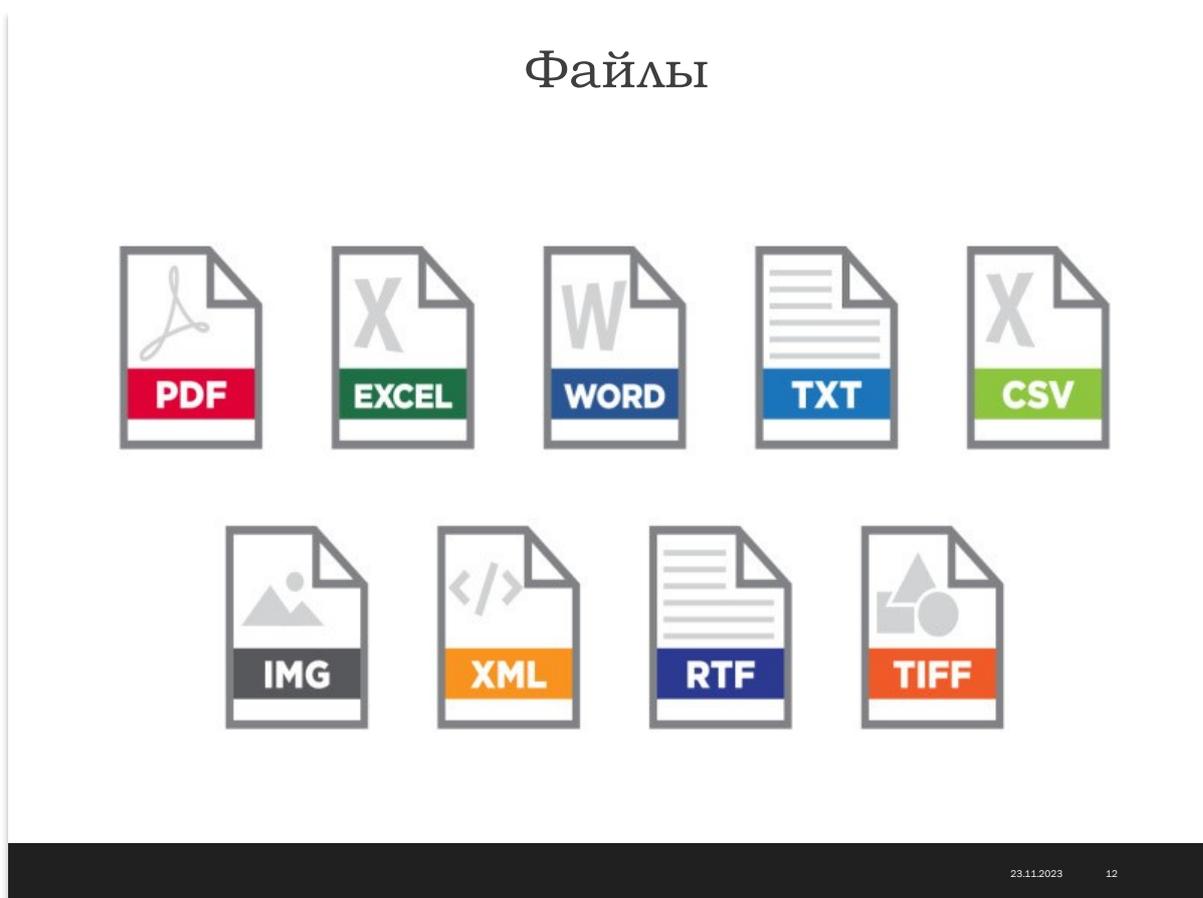
$$1 \text{ Тбайт} = 1024 \text{ Гбайт} = 10^{40} \text{ байт}$$

Особо обратим внимание на то, что при переходе к более крупным единицам «инженерная» погрешность, связанная с округлением, накапливается и становится недопустимой, поэтому на старших единицах измерения округление производится реже.

## Единицы хранения данных

При хранении данных решаются две проблемы: как сохранить данные в наиболее компактном виде и как обеспечить к ним удобный и быстрый доступ (если доступ не обеспечен, то это не хранение). Для обеспечения доступа необходимо, чтобы данные имели упорядоченную структуру, а при этом, как мы уже знаем, образуется «паразитная нагрузка» в виде адресных данных. Без них нельзя получить доступ к нужным элементам данных, входящих в структуру.

Поскольку адресные данные тоже имеют размер и тоже подлежат хранению, хранить данные в виде мелких единиц, таких как байты, неудобно. Их неудобно хранить и в более крупных единицах (килобайтах, мегабайтах и т. п.), поскольку неполное заполнение одной единицы хранения приводит к неэффективности хранения.



В качестве единицы хранения данных принят объект переменной длины, называемый файлом. Файл — это последовательность произвольного числа байтов, обладающая уникальным собственным именем. Обычно в отдельном

файле хранят данные, относящиеся к одному типу. В этом случае тип данных определяет тип файла.

Проще всего представить себе файл в виде безразмерного канцелярского досье, в которое можно по желанию добавлять содержимое или извлекать его оттуда. Поскольку в определении файла нет ограничений на размер, можно представить себе файл, имеющий 0 байтов (пустой файл), и файл, имеющий любое число байтов.

В определении файла особое внимание уделяется имени. Оно фактически несет в себе адресные данные, без которых данные, хранящиеся в файле, не станут информацией из-за отсутствия метода доступа к ним. Кроме функций, связанных с адресацией, имя файла может хранить и сведения о типе данных, заключенных в нем. Для автоматических средств работы с данными это важно, поскольку по имени файла они могут автоматически определить адекватный метод извлечения информации из файла.

#### Понятие о файловой структуре

Требование уникальности имени файла очевидно — без этого невозможно гарантировать однозначность доступа к данным. В средствах вычислительной техники требование уникальности имени обеспечивается автоматически — создать файл с именем, тождественным уже имеющемуся, не может ни пользователь, ни автоматика.

Хранение файлов организуется в иерархической структуре, которая в данном случае называется файловой структурой. В качестве вершины структуры служит имя носителя, на котором сохраняются файлы. Далее файлы группируются в каталоги (папки), внутри которых могут быть созданы вложенные каталоги (папки). Путь доступа к файлу начинается с имени устройства и включает все имена каталогов (папок), через которые проходит. В качестве разделителя используется символ «\» (обратная косая черта).

Уникальность имени файла обеспечивается тем, что полным именем файла считается собственное имя файла вместе с путем доступа к нему. Понятно, что в этом случае на одном носителе не может быть двух файлов с тождественными полными именами.

Пример записи полного имени файла:

<имя носителя>\<имя каталога-1>\...\<имя каталога-N>\<собственное имя файла>

Вот пример записи двух файлов, имеющих одинаковое собственное имя и размещённых на одном носителе, но отличающихся путем доступа, то есть полным именем.

## Примеры полных имён

---

C:\АВТОМАТИЧЕСКИЕ  
АППАРАТЫ\ВЕНЕРА\АТМОСФЕРА\Результаты  
исследований.docx

C:\РАДИОЛОКАЦИЯ\ВЕНЕРА\РЕЛЬЕФ\Результаты  
исследований.docx

**Информатика**

**Предмет и задачи информатики**

**Информатика** — это техническая наука, систематизирующая приемы создания, хранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими.

— ОПРЕДЕЛЕНИЕ

Информатика — это техническая наука, систематизирующая приемы создания, хранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими.

Из этого определения видно, что информатика очень близка к технологии, поэтому ее предмет нередко называют информационной технологией.

Предмет информатики составляют следующие понятия:

- Аппаратное обеспечение средств вычислительной техники;
- Программное обеспечение средств вычислительной техники; .
- средства взаимодействия аппаратного и программного обеспечения;
- Средства взаимодействия человека с аппаратными и программными средствами.

23.11.2023 14

Предмет информатики составляют следующие понятия:

- аппаратное обеспечение средств вычислительной техники;
- программное обеспечение средств вычислительной техники; .
- средства взаимодействия аппаратного и программного обеспечения;
- средства взаимодействия человека с аппаратными и программными средствами.

Как видно из этого списка, в информатике особое внимание уделяется вопросам взаимодействия. Для этого даже есть специальное понятие — интерфейс.

Методы и средства взаимодействия человека с аппаратными и программными средствами называют пользовательским интерфейсом. Соответственно, существуют аппаратные интерфейсы, программные интерфейсы и аппаратно-программные интерфейсы.

Основной задачей информатики является систематизация приемов и методов работы с аппаратными и программными средствами вычислительной техники.

– ЗАДАЧА

Основной задачей информатики является систематизация приемов и методов работы с аппаратными и программными средствами вычислительной техники. Цель систематизации состоит в выделении, внедрении и развитии передовых, наиболее эффективных технологий, в автоматизации этапов работы с данными, а также в методическом обеспечении новых технологических исследований.

Информатика — практическая наука. Ее достижения должны проходить подтверждение практикой и приниматься в тех случаях, когда они соответствуют критерию повышения эффективности. В составе основной задачи информатики сегодня можно выделить следующие направления для практических приложений:

- архитектура вычислительных систем (приемы и методы построения систем, предназначенных для автоматической обработки данных);
- интерфейсы вычислительных систем (приемы и методы управления аппаратным и программным обеспечением);
- программирование (приемы, методы и средства разработки компьютерных программ);
- преобразование данных (приемы и методы преобразования структур данных);
- защита информации (обобщение приемов, разработка методов и средств защиты данных);
- автоматизация (функционирование программно-аппаратных средств без участия человека);

- стандартизация (обеспечение совместимости между аппаратными и программными средствами, а также между форматами представления данных, относящихся к различным типам вычислительных систем).

На всех этапах технического обеспечения информационных процессов для информатики ключевым понятием является эффективность. Для аппаратных средств под эффективностью понимают отношение производительности оборудования к его стоимости (с учетом стоимости эксплуатации и обслуживания). Для программного обеспечения под эффективностью понимают производительность лиц, работающих с ними (пользователей). В программировании под эффективностью понимают объем программного кода, создаваемого программистами в единицу времени.

В информатике все жестко ориентировано на эффективность. Вопрос, как сделать ту или иную операцию, для информатики является важным, но не основным. Основным же является вопрос, как сделать данную операцию эффективно.

## **Истоки и предпосылки информатики**

Слово информатика происходит от французского слова *Informatique*, образованного в результате объединения терминов *Informaschon* (информация) и *Automatique* (автоматика), что выражает ее суть как науки об автоматической обработке информации. Кроме Франции термин информатика используется в ряде стран Восточной Европы. В то же время, в большинстве стран Западной Европы и США используется другой термин — *Computer Science* (наука о средствах вычислительной техники).

В качестве источников информатики обычно называют две науки — документалистику и кибернетику. Документалистика сформировалась в конце XIX века в связи с бурным развитием производственных отношений. Ее расцвет пришелся на 20-30-е годы XX века, а основным предметом стало изучение рациональных средств и методов повышения эффективности документооборота.

Основы близкой к информатике технической науки кибернетики были заложены трудами по математической логике американского математика Норберта Винера, опубликованными в 1948 году, а само название происходит от греческого слова (*kybemeticos* — искусный в управлении).

**Ампер Андре Мари**  
(22.01.1775 -  
10.06.1836) .

Открыл  
взаимодействие  
электрических токов  
и установил закон  
этого  
взаимодействия  
(закон Ампера),  
разработал теорию  
магнетизма (1820).



Впервые термин кибернетика ввел французский физик Андре Мари Ампер в первой половине XIX века. Он занимался разработкой единой системы классификации всех наук и обозначил этим термином гипотетическую науку об управлении, которой в то время не существовало, но которая, по его мнению, должна была существовать.

Сегодня предметом кибернетики являются принципы построения и функционирования систем автоматического управления, а основными задачами — методы моделирования процесса принятия решений техническими средствами, связь между психологией человека и математической логикой, связь между информационным процессом отдельного индивидуума и информационными процессами в обществе, разработка принципов и методов искусственного интеллекта.

На практике кибернетика во многих случаях опирается на те же программные и аппаратные средства вычислительной техники, что и информатика, а информатика, в свою очередь, заимствует у кибернетики математическую и логическую базу для развития этих средств.

## **Подведение итогов**

Все процессы в природе сопровождаются сигналами. Зарегистрированные сигналы образуют данные. Данные преобразуются,

транспортируются и потребляются с помощью методов. При взаимодействии данных и адекватных им методов образуется информация. Информация — это динамический объект, образующийся в ходе информационного процесса. Он отражает диалектическую связь между объективными данными и субъективными методами. Свойства информации зависят как от свойств данных, так и от свойств методов.

Данные различаются типами, что связано с различиями в физической природе сигналов, при регистрации которых образовались данные. В качестве средства хранения и транспортировки данных используются носители данных. Для удобства операций с данными их структурируют. Наиболее широко используются следующие структуры: линейная, табличная и иерархическая — они различаются методом адресации к данным. При сохранении данных образуются данные нового типа — адресные данные.

Вопросами систематизации приемов и методов создания, хранения, воспроизведения, обработки и передачи данных средствами вычислительной техники занимается техническая наука — информатика. С целью унификации приемов и методов работы с данными в вычислительной технике применяется универсальная система кодирования данных, называемая двоичным кодом. Элементарной единицей представления данных в двоичном коде является двоичный разряд (бит). Другой, более крупной единицей представления данных является байт.

Основной единицей хранения данных является файл. Файл представляет собой последовательность байтов, имеющую собственное имя. Совокупность файлов образует файловую структуру, которая, как правило, относится к иерархическому типу. Полный адрес файла в файловой структуре является уникальным и включает в себя собственное имя файла и путь доступа к нему.