

ШКОЛЬНЫЙ КУРС В НАТЛЯДНЫХ ТАБЛИЦАХ

ИНФОРМАТИКА

Теория

Понятия

Примеры

ВВЕРХ
ПО ТУРНИРНОЙ
ТАБЛИЦЕ
ЗНАНИЙ!

5-11
КЛАССЫ



ШКОЛЬНЫЙ КУРС В НАГЛЯДНЫХ ТАБЛИЦАХ

Е. В. ТИМОФЕЕВА



**ВВЕРХ
ПО ТУРНИРНОЙ
ТАБЛИЦЕ
ЗНАНИЙ!**

ИНФОРМАТИКА: 5-11 КЛАССЫ



Москва
2023

УДК 373.5:004
ББК 32.81я721
Т41

Макет подготовлен при содействии ООО «Аудиономикс»

Тимофеева, Елена Викторовна.

Т41 Информатика : 5–11 классы / Е. В. Тимофеева. — Москва : Эксмо, 2023. — 224 с. — (Школьный курс в наглядных таблицах).

ISBN 978-5-04-189580-8

В пособии школьный курс информатики представлен в виде наглядных таблиц и схем, что заметно ускоряет и упрощает процесс усвоения материала. Использование таблиц в качестве опорного конспекта не даст школьникам запутаться в большом объеме информации и позволит без труда найти ответ на интересующий их вопрос.

Книга окажет помощь учащимся 5–11 классов при подготовке к контрольным и экзаменам ОГЭ и ЕГЭ, а также будет полезна учителям при организации работы на уроках.

**УДК 373.5:004
ББК 32.81я721**

ISBN 978-5-04-189580-8

© Тимофеева Е.В., 2023
© ООО «Аудиономикс», 2023
© Оформление. ООО «Издательство «Эксмо», 2023

СОДЕРЖАНИЕ

Введение	6	Позиционные системы счисления	37
ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ	7	Двоичное представление информации	40
Информация и её кодирование	7	Сложение и умножение в различных системах счисления	45
Виды информации	7	Представление информации в компьютере.....	48
Свойства информации.....	10	Представление числовой информации	49
Единицы измерения количества информации.....	11	Представление текстовой информации.....	50
Виды информационных процессов.....	12	Представление графической информации	51
Кодирование и декодирование информации	13	Представление звуковой информации	54
Процесс передачи информации, источник и приёмник.....	16	Элементы алгебры логики	56
Аналоговое и дискретное представление информации	17	Логические высказывания. Истинность высказывания	57
Искажение информации	19	Логические операции.....	59
Скорость передачи информации	20	Приоритеты логических связей.....	64
Система, её свойства и компоненты	21	Основные законы логики	64
Понятие системы.....	21	Предикаты и кванторы	66
Характеристики и свойства системы.....	22	СРЕДСТВА ИКТ	68
Информационная система и её компоненты	24	Архитектура компьютера и компьютерных сетей.....	68
Моделирование	26	Организация работы компьютеров и компьютерных систем	70
Информационное моделирование.....	30	Аппаратное обеспечение.....	70
Математическое моделирование.....	30	Программное обеспечение.....	75
Компьютерное моделирование	32	Файловая система	82
Имитационное моделирование	34	Технологии создания и обработки текстовой информации	85
Системы счисления.....	36	Программы для работы с текстовой информацией	85
Языковые системы	36		
Языки	36		
Виды систем счисления	37		

Работа с текстовым документом	87	Инструменты создания информационных объектов для Интернета	136
Этапы работы с текстовым документом	88	АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ	141
Шаблоны текстовых документов	89	Алгоритмы и алгоритмизация	141
Проверка орфографии и грамматики	90	Историческая справка	141
Словари и тезаурусы. Машинный перевод	91	Основные понятия	145
Редактирование математических текстов.		Свойства представления алгоритмов	146
Графическое представление математических объектов	92	Способы представления алгоритмов	147
Использование систем распознавания текстов	94	Виды алгоритмов	148
Технология создания и обработки мультимедийной и графической информации	108	Выигрышная стратегия	153
Форматы графических объектов	108	Программирование	154
Создание, ввод и обработка графических объектов	112	Основные понятия языков программирования	155
Звуковые файлы	114	Типы данных	157
Технология обработки информации в электронных таблицах	118	Структурированные типы данных	158
Табличный процессор	118	Графы	159
Объекты табличного процессора Excel и их свойства	120	Массив	166
Технологии поиска и хранения информации	125	Списки	173
Составляющие банка данных	125	Основы языков программирования	175
Типы моделей баз данных	127	Виды информации	175
Табличные базы данных	130	Основные служебные слова	177
Использование инструментов поисковых систем (формирование запросов)	131	Разделители языка	179
Телекоммуникационные технологии	132	Структура программы	180
Программное обеспечение средств телекоммуникационных технологий	133	Идентификаторы	181
		Переменные и константы	183
		Функции	187
		Операторы и операции	190
		Задачи на запись и анализ алгоритмов	196
		Алгоритмы обработки массивов	196

Исправление ошибок в программе	201	Информационные ресурсы	210
ИНФОРМАЦИОННАЯ ДЕЯТЕЛЬНОСТЬ ЧЕЛОВЕКА	203	Национальные информационные ресурсы	210
Этапы развития информационного общества	203	Ресурсосбережение	211
Профессиональная информационная деятельность	205	Рынок информационных ресурсов и услуг	212
Применение технических средств и информационных ресурсов в профессиональной деятельности	206	Экономика информационной сферы	213
Системное администрирование	207	Информационная этика и этикет	215
Интернет и безопасность его использования	209	Информационное право	216
		Информационная безопасность	219
		Компьютерные вирусы	220
		Искусственный интеллект и машинное обучение	221

ВВЕДЕНИЕ

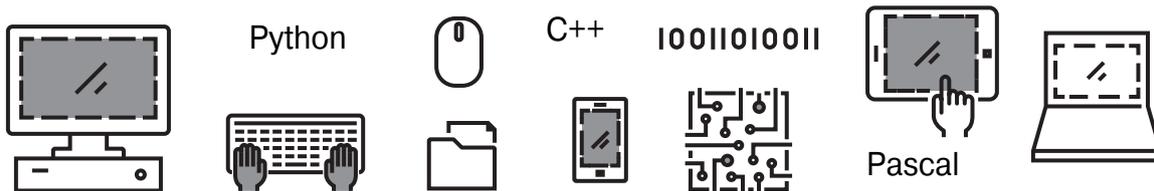
Данное пособие является помощником в изучении, систематизации и обобщении знаний по информатике за курс средней школы. Материал представлен в наглядной и удобной для восприятия форме — в виде таблиц, что существенно упрощает его запоминание.

Книга состоит из четырёх глав: «Информация и информационные процессы», «Средства ИКТ», «Алгоритмизация и программирование» и «Информационная деятельность человека».

Структура и содержание пособия позволяют ученику актуализировать, систематизировать и закрепить знания по информатике за курс основной школы.

Пособие предназначено для учащихся средней школы при самоподготовке к различным видам контроля, основному и единому государственному экзамену, а также для учителей информатики.

Желаем успехов на экзамене!



ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

ИНФОРМАЦИЯ И ЕЁ КОДИРОВАНИЕ



На протяжении всей жизни человек непрерывно получает и использует информацию — всё, что мы видим, слышим, осязаем, чувствуем. Источниками являются любые находящиеся в зоне восприятия предметы и приборы.



Информация — сведения об окружающем мире, которые снижают уровень неопределённости знаний о нём.

Виды информации

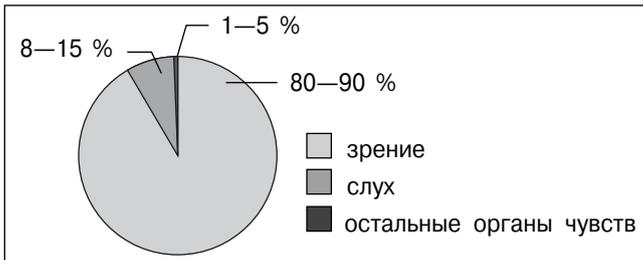
Вид	Характеристика
По способу восприятия	
Визуальная	Информация, которую мы получаем с помощью зрения. Например, можем увидеть некоторый предмет, определить его цвет, форму, положение в пространстве
Аудиальная	Информация, которую можно получить с помощью слуха. Например, послушать музыку, узнать, как журчит ручей или звенит колокольчик

Вид	Характеристика
Вкусовая	Информация, которую мы получаем с помощью вкусовых рецепторов, расположенных преимущественно на языке. Например, можем узнать, каков на вкус арбуз: сладкий, солёный, горький или кислый
Обонятельная	Информация, которую мы получаем с помощью носа. Например, можем определить пряный, терпкий, приятный или неприятный запах
Тактильная	Информация, которую мы получаем с помощью кожи. Например, дотрагиваясь до предмета, можно понять, горячий он или холодный, влажный или сухой
Вестибулярная	Информация, которую можно получить с помощью вестибулярного аппарата, отслеживающего наше положение в пространстве. Например, закрыв глаза, мы способны понять, куда идём: вправо или влево. При падении мы чётко понимаем, что двигаемся вниз
Мышечная	Информация, которую мы получаем, используя мышцы. Например, можем определить, какой из двух предметов тяжелее. Благодаря мышечной информации развивается способность печатать на клавиатуре вслепую
По назначению	
Массовая	Информация, передаваемая широким аудиториям, рассредоточенным во времени и пространстве, с помощью искусственных каналов. К такой информации относятся печатные, аудиовизуальные и иные сообщения и материалы. Например, новостные передачи, газеты
Специальная	Информация, которая может быть не понятна основной массе социума, но необходима и понятна в рамках узкой социальной группы. Например, технология производства бумаги

Секретная	Информация, не подлежащая разглашению. Например, сведения, составляющие государственную тайну, секретный рецепт шеф-повара
Личная	Набор сведений о какой-либо личности, определяющих социальное положение и типы социальных взаимодействий
По форме представления	
Текстовая	Информация, представленная в форме текстового сообщения: рукописного либо печатного. Например, газетные статьи, записи в блоге, книги, письма
Числовая	Информация, выраженная в виде специальных символов, чисел. Например, номер телефона, номер квартиры
Звуковая	Любая информация, которую можно услышать: музыка, речь человека, шум моря, звук колокола, шорох листьев
Графическая	Информация, представленная в виде картин, чертежей, рисунков, фотографий
Видеоинформация	Информация, представленная в виде движущихся изображений. Например, кинофильм, компьютерная игра
По истинности	
Ложная	Информация, заведомо не соответствующая истинной. Например, утверждение, что в январе 35 дней
Истинная	Информация, которая соответствует действительности. Например, в десятичной системе счисления $5 + 8 = 13$



Здоровый человек получает с помощью органов зрения (визуально) около 80—90 % информации, с использованием органов слуха (аудиально) — порядка 8—15 %, благодаря остальным органам чувств (обонянию, вкусу, осязанию) — только 1—5 %. При утрате одного из информационных каналов (зрения, слуха, вкуса, обоняния или осязания) усиливается информационная роль оставшихся.



Свойства информации

Свойства	Характеристика
Объективная информация	Не зависит от чьего-либо мнения, суждения
Актуальная информация	Важна и существенна для настоящего времени
Полная информация	Достаточна для понимания ситуации и принятия решения
Достоверная информация	Отражает истинное положение дел
Полезная информация	Оценивается по тем задачам, которые можно решить с её помощью
Понятная информация	Выражена на языке, доступном для получателя
Дискретная информация	Может быть разбита на элементарные фрагменты или части

Единицы измерения количества информации



Термин «**количество информации**» используют в устройствах цифровой обработки и передачи информации, например в цифровой вычислительной технике (компьютерах), для записи объёма запоминающих устройств, количества памяти, используемого программой. Наименьшей единицей информации является **бит** (англ. *binary digit (bit)* — «двоичная единица информации»).

Бит	Количество информации, необходимое для однозначного определения одного из двух равновероятных событий. Например, один бит информации получает человек, когда узнаёт, происходит какое-то событие или нет: выпал снег или нет, опаздывает работник или нет, правильно решено задание или нет и т. д.
Байт	Последовательность из 8 двоичных разрядов битов (наиболее популярная единица измерения информации)

Производные единицы измерения количества информации

Название	Условное обозначение	Соотношение с другими единицами	Название	Условное обозначение	Соотношение с другими единицами
Бит	Бит		Байт	Байт	8 бит
Килобит	Кбит	2^{10} бит = 1024 бит	Килобайт	Кбайт	2^{10} байт = 1024 байт
Мегабит	Мбит	2^{20} бит = 1024 Кбит	Мегабайт	Мбайт	2^{20} байт = 1024 Кбайт
Гигабит	Гбит	2^{30} бит = 1024 Мбит	Гигабайт	Гбайт	2^{30} байт = 1024 Мбайт
Терабит	Тбит	2^{40} бит = 1024 Гбит	Терабайт	Тбайт	2^{40} байт = 1024 Гбайт

Виды информационных процессов



Всё, что происходит с информацией, представляет собой **информационный процесс**. Выделяют три основных вида информационных процессов: передача, хранение и обработка информации. Они являются базовыми, а их выполнение порождает другие информационные процессы.

Основные информационные процессы

Передача	Хранение	Обработка информации
<p>Перемещение информации от источника к приёмнику по каналу передачи. Информация передаётся в форме сигналов (световых, звуковых, ультразвуковых, текстовых, электрических, графических и др.). Каналом передачи может быть воздух (сигнальные огни), электрические и оптоволоконные кабели (звук или видео), отдельные люди (новости или идеи), нервные клетки человека (импульсы) и т. д.</p>	<p>Информация хранится в памяти людей или же на каких-либо внешних носителях. На протяжении многих столетий основным носителем информации была бумага. В настоящее время также распространены электронные носители информации: облачные сервисы (удалённые серверы), внешние диски, флеш-карты и др.</p>	<p>Вся совокупность операций (сбор, защита, преобразование, в том числе кодирование и декодирование, считывание, уничтожение), осуществляемых при помощи человека, технических и программных средств, включая обмен по каналам передачи данных. В результате обработки информации можно получить новые знания из имеющихся</p>

Некоторые операции обработки информации

Операция	Характеристика
Сбор	Предполагает поиск и отбор необходимых данных из различных источников: работа с литературой, справочниками, проведение экспериментов, наблюдений, опросов, поиск в Интернете, а также в других информационных сетях и т. д. Например, чтобы написать реферат, необходимо найти информацию по данной теме. Для сбора информации используют различные измерительные устройства. Например, для определения температуры воздуха нужен термометр
Защита	Создание условий, при которых не допускается случайной потери, изменения, повреждения данных или несанкционированного доступа к ним. Способы защиты: создание резервных копий, хранение на защищённом носителе, предоставление прав доступа узкому кругу пользователей, шифрование сообщений и др.
Кодирование	Преобразование информации в символьную форму, удобную для хранения, передачи, обработки. Кодирование используется в технических средствах работы с информацией (компьютер)
Декодирование	Процесс восстановления изначальной формы представления информации, т. е. процесс, обратный кодированию, при котором закодированное сообщение переводится на язык, понятный получателю

Кодирование и декодирование информации



В процессах передачи, хранения и обработки информации происходит её кодирование.

Виды кодирования



Информационный код (кодовое слово или просто код) — символьная последовательность, которая несёт в себе конкретную информацию. Каждый код имеет определённую длину, т. е. состоит из конечного числа символов, это количество называется **длиной кода**.

Равномерное кодирование	Неравномерное кодирование
<p>Равномерное кодирование — вид кодирования, при котором все символы какого-либо алфавита кодируются кодами одинаковой длины.</p> <p>Чтобы понять, какая длина кода будет у одного символа при равномерном кодировании, понадобится формула Хартли, позволяющая определить количество информации. Эту закономерность Р. Хартли обнаружил в 1928 г.</p> <p>Формула Хартли:</p> $N = 2^i,$ <p>где N — количество равновероятных событий, i — количество информации, которую мы получим при наступлении одного из событий.</p> <p>Из этой формулы можно вывести i:</p> $i = \log_2 N.$ <p>Мощность алфавита — количество символов/знаков, из которых состоит рассматриваемый алфавит</p>	<p>Неравномерное кодирование — вид кодирования, при котором все элементы какого-либо множества кодируются кодами различной длины. Следует понимать общий принцип неравномерного кода, суть которого заключается в том, чтобы кодировать наиболее часто используемые элементы как можно меньшим количеством битов, поскольку ими приходится оперировать очень часто. Главное при таком кодировании — обеспечить возможность однозначного декодирования записанной с помощью этих кодов строки (поочерёдного, слева направо, выделения и распознавания из сплошной последовательности нулей и единиц кодов отдельных букв). Для этого коды символам необходимо назначать в соответствии с условиями Фано. С математической точки зрения условие звучит так: «Если код содержит слово A, то для любой непустой строки B слова AB не существует в коде». Существует также обратное условие Фано, которое с математической точки зрения можно сформулировать следующим образом: «Если код содержит слово B, то для любой непустой строки C слова CB не существует в коде»</p>

Неравномерный код может быть однозначно декодирован, если никакой из кодов не совпадает с началом (префиксом) какого-либо другого, более длинного кода.

A	B	C
10	11	001

D: 00
недопустимо:



C	001
D	00

Код D совпадает с началом кода C

A	B	C
10	11	00

D: 11
недопустимо:



B	11
D	11

Код D совпадает с кодом B

A	B	C
100	110	010

D: 00
допустимо:



Код D не совпадает ни с одним другим кодом и с началом никакого другого кода

Неравномерный код может быть однозначно декодирован, если никакой из кодов не совпадает с окончанием (постфиксом) какого-либо другого, более длинного кода.

A	B	C
10	11	001

D: 01
недопустимо:



C	001
D	01

Код D совпадает с окончанием кода C

A	B	C
10	11	00

D: 11
недопустимо:



B	11
D	11

Код D совпадает с кодом B

A	B	C
100	110	010

D: 01
допустимо:



Код D не совпадает ни с одним другим кодом и с окончанием никакого другого кода



Для однозначности декодирования последовательности кодов достаточно выполнения хотя бы одного из двух вышеуказанных условий Фано:

- при выполнении прямого условия Фано последовательность кодов однозначно декодируется с начала;
- при выполнении обратного условия Фано последовательность кодов однозначно декодируется с конца.

Выбрать, какое из двух правил Фано используется при решении конкретной задачи, можно, проанализировав коды в условии задачи (без учёта кода, проверяемого в вариантах ответа): если для исходных кодов выпол-

няется прямое правило Фано, то его и нужно использовать при решении, и наоборот.

Необходимо помнить, что правила Фано — это **достаточное, но не необходимое** условие однозначного декодирования: если не выполняется ни прямое, ни обратное правило Фано, конкретная двоичная последовательность может оказаться такой, что она декодируется однозначно (т. к. остальные возможные варианты до конца декодирования довести не удаётся). В подобном случае необходимо пытаться строить дерево декодирования в обоих направлениях.

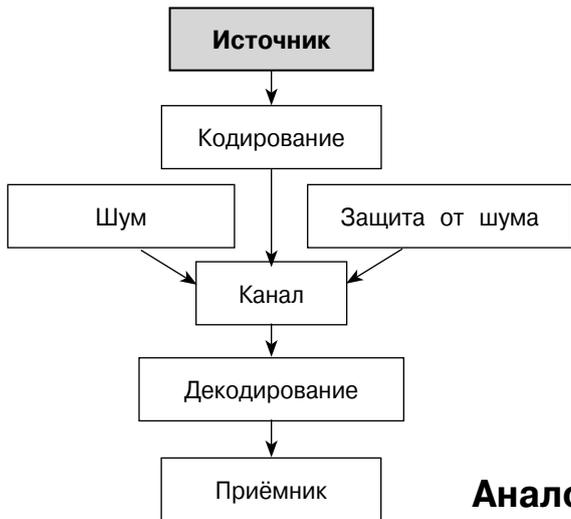
Процесс передачи информации, источник и приёмник



Передача информации — один из самых важных информационных процессов в современном мире. Без него невозможно создание общемирового информационного пространства, быстрое развитие науки и общение в социальных сетях.

Участники процесса передачи информации

Участник	Характеристика
Источник	Тот, кто отправляет информацию
Приёмник	Тот, кто принимает информацию, причём их может быть несколько
Канал связи	То, по чему передаётся информация

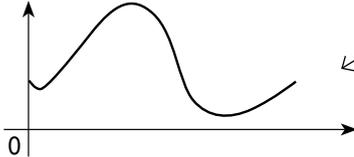
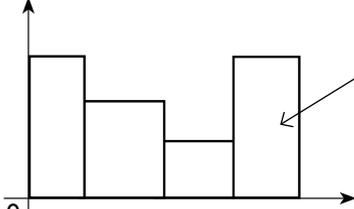


К. Шеннон, создатель теории информации и основоположник цифровой связи, разработал общую схему передачи информации. Процесс передачи информации/сообщения/сигнала идёт от источника, далее сигнал кодируется и по каналу связи передаётся (иногда с шумом и помехами). Перед тем как попасть к приёмнику, сообщение декодируется.

Аналоговое и дискретное представление информации



Всякая информация представляет собой набор знаков (в тексте — символы, в изображении — пиксели). Для того чтобы быть представленной в компьютере, информация кодируется определённым образом, т. е. превращается в последовательность электрических сигналов. У каждого цвета пикселя или текстового символа есть свой уникальный код. Это позволяет нам преобразовывать информацию из одной формы в другую (кодировать и декодировать).

Сигнал	Характеристика
Аналоговый (непрерывный)	<p>Может меняться в любой момент времени. Например, речь человека, пение птиц, кардиограмма.</p> <p>Непрерывные (аналоговые) информационные сигналы могут принимать любые значения из всех возможных в рамках заданного интервала, т. е. бесконечное множество значений.</p>  <p>Пример аналоговой функции</p> <p>В аналоговой форме сигнал описывается непрерывной функцией времени</p>
Дискретный (цифровой)	<p>Может меняться в определённый момент времени и принимать заранее определённые значения. Например, сигналы светофора, азбука Морзе, текст в книге.</p> <p>В дискретной (цифровой) форме представления информации величины могут принимать лишь отдельные, неделимые значения и не могут принимать значения, промежуточные между ними. Все значения дискретного сигнала можно пронумеровать целыми числами.</p>  <p>Пример дискретной функции</p>

В дискретной форме сигнал представляется совокупностью символов из некоторого набора, называемого алфавитом. Если каждому символу присвоить числовое значение, то сигнал будет иметь цифровую форму отображения информации. В цифровой технике используется два символа: 0 и 1. Увеличивая количество разрядов, можно повысить точность представления информационного объекта. Благодаря этому достоинству цифровая обработка занимает ведущие позиции в современных информационных технологиях



Аналоговый сигнал можно преобразовать в дискретный, и наоборот. Для этого используются аналого-цифровые преобразователи (АЦП).



Дискретизация — процесс превращения непрерывного сигнала в цифровой путём измерения числовых значений амплитуды сигнала через равные интервалы времени (шаг дискретизации).



Шаг дискретизации — промежуток времени, через который производятся замеры между сигналами.

Искажение информации



В процессе передачи информация может теряться, искажаться: искажение звука в телефоне, атмосферные помехи по радио, искажение или затемнение изображения в телевидении, ошибки при передаче в телеграфе. Существует наука, разрабатывающая способы защиты информации, — **криптология**.

Ошибки, возникающие при передаче информации

Часть правильной информации заменяется на неправильную

К передаваемой информации добавляются лишние, посторонние сообщения

Часть информации при передаче пропадает



Понятие «шум» применяется ко всему, что маскирует полезный сигнал, поэтому шумом может оказаться какой-нибудь другой сигнал (в таком случае он называется помехой), но обычно данный термин означает случайный шум физической (чаще всего тепловой) природы. Шум характеризуется своим частотным спектром, распределением амплитуды и источником (происхождением).

Скорость передачи информации

Канал связи — технические средства, позволяющие осуществлять передачу данных на расстоянии. Информация по каналам связи движется с определённой скоростью.

Скорость передачи информации	Пропускная способность канала
Количество информации, передаваемое в единицу времени. Скорость измеряется в битах в секунду (бит/с) и в кратных им величинах — Кбит/с, Мбит/с, а также в байтах в секунду (Б/с) и кратных им Кбайт/с, Гбайт/с	Максимальное количество информации, которое может быть получено по этому каналу в единицу времени. Пропускная способность канала измеряется в тех же единицах, что и скорость передачи информации



Скорость передачи информации в конкретной паре «источник — приёмник» может быть разная, но пропускная способность канала неизменна.



Объём переданной информации:

$$V = v \cdot t,$$

где V — объём передаваемого файла, v — скорость передачи файла, t — время передачи файла. Из формулы объёма можно вывести формулу скорости передачи информации: $v = \frac{V}{t}$.

СИСТЕМА, ЕЁ СВОЙСТВА И КОМПОНЕНТЫ

Понятие системы



Система — совокупность элементов, которые взаимодействуют друг с другом и образуют определённую целостность, единство.

Элементы системы	Главная цель системы
Школа	
Люди, оборудование, материалы, здания и др.	Создание базы знаний, обучение
Компьютер	
Электронные и электромеханические элементы, линии связи и др.	Обработка данных
Телекоммуникационная система	
Компьютеры, модемы, кабели, сетевое программное обеспечение и др.	Передача информации

Элементы системы	Главная цель системы
Информационная система	
Компьютеры, компьютерные сети, люди, информационное и программное обеспечение	Производство профессиональной информации

Характеристики и свойства системы

Архитектура системы	Совокупность существенных для пользователя свойств системы
Элемент системы	Часть системы, имеющая определённое функциональное назначение. Элементы, состоящие из простых взаимосвязанных элементов, часто называют подсистемами
Структура системы	Состав, порядок и принципы взаимодействия элементов системы, определяющие основные свойства системы. Если отдельные элементы системы разнесены по разным уровням и характеризуются внутренними связями, то принято говорить об иерархической структуре системы

Свойства системы (в том числе информационной)

Свойства	Характеристика
Целостность	Данное свойство означает, что все элементы системы функционируют как единое целое. Например, стрелки часов отдельно не могут показать время, а целая система «Часы» может
Иерархичность	Каждый компонент может рассматриваться как система, а исследуемая система представляется как элемент более широкой системы. Например, живая клетка многоклеточного организма — элемент системы, но при этом сама является также сложной системой

Множественность	В силу сложности каждой системы её изучение требует построения различных моделей. Например, человека можно рассматривать, с одной стороны, как совокупность частей тела (руки, ноги, голова), с другой — как совокупность нервной, мышечной, кровеносной систем, а с третьей — как совокупность химических элементов
Делимость	Система состоит из ряда подсистем или элементов, выделенных по определённым критериям и отвечающих конкретным целям и задачам. Например, предприятие по производству пирожных состоит из трёх отделов: маркетинга, бухгалтерии и производства. Каждый отдел выполняет определённую задачу
Сложность	Система обладает большим количеством элементов, компонентов и подсистем, прямых и обратных связей между ними. Например, человек — очень сложная, до сих пор не изученная полностью система
Структурность	Данное свойство означает возможность описания системы через построение её структуры по уровням и иерархиям. Например, в строительной фирме есть несколько подразделений, которые управляются другими подразделениями
Адаптивность	Система должна приспосабливаться к условиям конкретной предметной области. Например, раньше при лечении зубов применялись цементные пломбы, а в настоящее время — композитные световые. Система здравоохранения адаптируется под новые знания
Интегрируемость	Данное свойство означает возможность взаимодействия системы с вновь подключаемыми компонентами или подсистемами. Например, если в школе ввели дополнительный урок, то нужно перестроить расписание для него
Безопасность	Система должна обеспечить конфиденциальность и целостность информации. Например, любой сотовый оператор или банк должен обеспечивать конфиденциальность информации: паспортные данные, номер телефона

Информационная система и её компоненты

Информационная система (ИС)	
В широком смысле	В узком смысле
Любая система, предназначенная для обработки информации. Федеральный закон «Об информации, информационных технологиях и о защите информации» трактует данное понятие следующим образом: «Информационная система — совокупность содержащейся в базах данных информации и обеспечивающих её обработку информационных технологий и технических средств»	Совокупность аппаратных и программных средств, предназначенных для хранения, обработки и применения большого объёма информации. Современное понимание информационной системы предполагает использование компьютера в качестве основного технического средства переработки информации

Виды архитектуры



Под **архитектурой информационной системы** понимается совокупность основных её элементов и схема их взаимодействия. Существует два вида архитектуры: логическая и физическая.

Логическая архитектура	Физическая архитектура
Включает в себя комплекс методов, решений и алгоритмов, по которым работает система, а также обеспечивает функционирование и взаимодействие подсистем	Включает в себя элементы, обеспечивающие работу системы, т. е. является комплексом вычислительной техники, средств связи между элементами системы и обслуживающим персоналом

Компоненты информационной системы

База данных и её структура

Программы, необходимые для работы ИС, а также программа, позволяющая управлять базами данных

Необходимые технические средства

Пользователи



Компонент — простейшая неделимая часть системы. Принято считать, что компоненты информационной системы отвечают за хранение информации, её кодирование и т. д.

Классификация информационных систем

По целевому назначению	По степени автоматизации	По сфере применения
<ul style="list-style-type: none">• Управляющие.• Информационно-справочные.• Системы принятия решений.• Информационно-поисковые и т. п.	<ul style="list-style-type: none">• Ручные.• Автоматические.• Автоматизированные	<ul style="list-style-type: none">• В образовании.• В торговле.• В науке.• В здравоохранении и т. п.
По уровню управления	По степени распределения	По типу данных
<ul style="list-style-type: none">• Стратегические.• Функциональные.• Операционные	<ul style="list-style-type: none">• Локальные.• Распределённые	<ul style="list-style-type: none">• Фактографические.• Документальные

МОДЕЛИРОВАНИЕ



Модель — некоторый объект, созданный специально по образу другого объекта, предмета или явления и обладающий его существенными признаками.

Классификация моделей

По характеристике объекта моделирования	С учётом фактора времени
<ul style="list-style-type: none"> • Модели внешнего вида. • Модели структуры. • Модели поведения 	<ul style="list-style-type: none"> • Статические. • Динамические: дискретные и непрерывные
По области использования	По степени формализации
<ul style="list-style-type: none"> • Учебные. • Опытные. • Научно-технические. 	<ul style="list-style-type: none"> • Неформализованные. • Частично формализованные. • Нормализованные
По форме представления	
<ul style="list-style-type: none"> • Материальные. • Информационные: знаковые (компьютерные, некомпьютерные) и вербальные 	

Виды моделей

Модели	Характеристика
По форме представления	
Материальные	<p>Данные модели называются также предметными, поскольку всегда имеют реальное воплощение. Они отражают внешнее свойство и внутреннее устройство исходных объектов, суть процессов и явлений объекта-оригинала. Воплощают экспериментальный метод познания окружающей среды (глобус, скелет человека, макет Солнечной системы, химические опыты)</p>
Информационные	<p>Целенаправленно отобранная информация об объекте, которая отражает наиболее существенные для исследователя свойства этого объекта.</p> <ul style="list-style-type: none"> • Вербальные модели — модели, которые могут быть представлены в мысленной или разговорной форме (ход решения уравнения или задачи). • Знаковыми называются модели, выраженные специальными знаками любого формального языка: <ul style="list-style-type: none"> — компьютерными являются модели, реализованные с применением программных средств (3D-моделирование человека для компьютерной игры); — некомпьютерными называют модели, реализованные без использования программных средств (математические, специальные (ноты, химические элементы) модели)
С учётом фактора времени	
Статические	Модели, которые не меняются с течением времени (фотография)
Динамические	<p>Модели, которые способны меняться с течением времени:</p> <ul style="list-style-type: none"> • дискретные модели описывают прерывистые процессы (модель численности населения Российской Федерации); • непрерывные модели отражают непрерывные процессы (отображение закона Ньютона в виде формулы)

Модели	Характеристика
По области использования	
Учебные	Модели, которые используются в процессе обучения (тренажёр для полётов)
Опытные	Увеличенные или уменьшенные копии реально проектируемых объектов (модель корабля/автомобиля/здания)
Научно-технические	Модели, созданные для исследования процессов или явлений (прибор для получения грозового электрического разряда)
Игровые	Модели, созданные специально для изучения поведения людей в различных ситуациях (военные игры, экономическая игра «Монополия»)
Имитационные	Модели, отражающие логику и закономерности поведения моделируемых объектов во времени и пространстве (модель call-центра, линии производства на заводе, игра «Жизнь»)

Цели моделирования

Цель	Характеристика
Познание окружающего мира	Зачем человек создаёт модели? Чтобы ответить на этот вопрос, надо обратиться к далёкому прошлому. Несколько миллионов лет назад, на заре человечества первобытные люди изучали окружающую природу, чтобы научиться противостоять стихиям, пользоваться природными благами, выживать
Создание объектов с заданными свойствами	Решается задача типа «Как сделать, чтобы...». Накопив достаточно знаний, человек задаёт себе вопрос: «Нельзя ли создать объект с заданными свойствами и возможностями, чтобы противодействовать стихиям или ставить себе на службу природные явления?»

<p>Определение последствий воздействия на объект и принятие правильного решения</p>	<p>Речь идёт о решении задачи типа «Что будет, если...». Например, что будет, если увеличить плату за проезд в транспорте, или что произойдёт, если закопать ядерные отходы в какой-то местности</p>
<p>Эффективность управления объектом (или процессом)</p>	<p>Управление станет эффективным при условии, что будет соблюдено большинство критериев и задач. Решается задача типа «Как совместить несовместимое?». Например, как сделать питание в столовой качественным, чтобы при этом оно нравилось школьникам и было по карману родителям</p>

<p style="text-align: center;">Этапы моделирования</p>	
<p style="text-align: center;">1</p>	<p>Постановка цели моделирования</p>
<p style="text-align: center;">2</p>	<p>Анализ и выделение всех известных свойств моделирования</p>
<p style="text-align: center;">3</p>	<p>Анализ выделенных свойств с точки зрения цели моделирования и определение, какие из них следует считать существенными</p>
<p style="text-align: center;">4</p>	<p>Выбор формы представления модели</p>
<p style="text-align: center;">5</p>	<p>Формализация</p>
<p style="text-align: center;">6</p>	<p>Анализ полученной модели на непротиворечивость</p>
<p style="text-align: center;">7</p>	<p>Анализ адекватности полученной модели объекта и цели моделирования</p>

Информационное моделирование



Информационное моделирование — моделирование объектов и процессов в образной или знаковой форме.

Формы представления информационных моделей

Сигналы

Устная, словесная

Символьная

Табличная

Схемы, карты

Графики



В зависимости от поставленной цели один и тот же объект можно представить несколькими информационными моделями, отличающимися набором параметров и способом их представления.

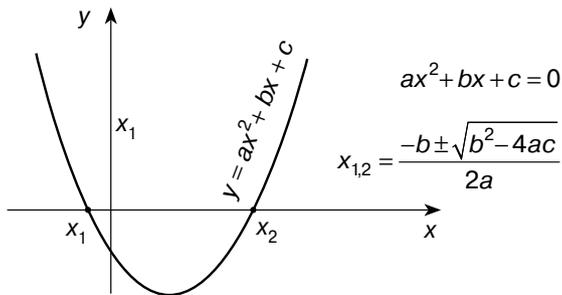
Рассмотрим пример информационной модели. Возьмём реальный объект — ученика. Для него мы можем создать модель — анкету данного учащегося, в которой будет представлено его описание, например цвет глаз, рост, возраст. В таком случае анкета является информационной моделью, по которой можно представить учащегося. Ещё один пример. Имеется класс из 25 учащихся, каждый из которых имеет свои персональные данные. Учителю достаточно взглянуть на список учащихся, чтобы представить этот класс. Список учащихся в журнале и сведения о них будут являться информационной моделью данного класса.

Математическое моделирование



Математическая модель — знаковая модель, сформулированная на языке математики и логики. Это система математических соотношений — формул, уравнений, неравенств, графиков и т. д., отображающих связи различных параметров объекта, системы объектов, процесса или явления.

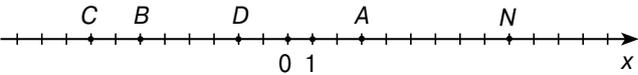
Характеристика	Пример
Над элементами математической модели можно выполнять определённые математические преобразования	В модели нахождения наименьшего числа выполняются операции сравнения, а в модели вычисления корня уравнения — различные арифметические операции
С помощью математических моделей описываются решения различных инженерных задач, многие физические процессы, технологические процессы	Движение планет, автомобиля, сварка, плавление металла и т. п.
Графики, таблицы, диаграммы позволяют отображать различные закономерности и зависимости реального мира	Модель развития эпидемии можно описать как с помощью формул, так и с помощью графика. Полёт снаряда, выпущенного из орудия, можно математически смоделировать с помощью известных формул движения, а затем построить график движения снаряда — баллистическую кривую, которая отображает реальный полёт. Математически изменяя параметры снаряда или характеристики движения, можно изучать, например, вопросы увеличения дальности или высоты полёта и т. п.



Формула и график описывают один и тот же объект — параболу



Как известно, не все математические задачи можно решить аналитически и получить решение в виде формул. Значительно больше задач, которые решаются приближённо, с заданной точностью, т. е. с использованием численных методов. Реализация приближённых расчётов на компьютерах позволяет повысить точность и скорость выполнения.

Реальная ситуация	Математическая модель
У Даши и Саши одинаковое количество конфет	$x = y$
У Кости на семь пятёрок больше, чем у Паши	$x + 7 = y$
Площадь квадрата со стороной a	$S = a^2$
Точка A расположена правее точки B на координатной прямой	

Компьютерное моделирование



При создании какого-либо нового объекта возникает необходимость протестировать его до запуска в массовое производство и до использования. Для этого применяют компьютерное моделирование, которое имитирует возможное поведение объекта. Данный вид моделирования нашёл широкое применение в таких отраслях, как строительство, медицина, транспорт, экология, метеорология и т. п.



Компьютерное моделирование — метод решения задачи с использованием компьютера.



Симулятор — комплекс программных и технических средств, которые имитируют управление каким-либо процессом, устройством.

Области применения

Научные исследования, образование	Мобильные устройства (КПК, смартфоны, коммуникаторы и др.)
Разработка бытовых и промышленных приборов	
Информационные и справочные службы	Компьютерные сети
Системы безопасности	Искусство и дизайн

Этапы компьютерного моделирования

1	Выделение условия задачи (цель и выбор модели)
2	Выделение основных понятий, концепция
3	Построение логики решения (логическая модель)
4	Построение математического решения (математическая модель)
5	Создание алгоритма
6	Реализация на языке программирования
7	Тестирование
8	Отладка — поиск ошибок и возврат к п. 7, если они обнаружены
9	Процесс исследования модели

Имитационное моделирование



Имитационное моделирование — распространённый способ исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему (модель должна отражать логику и закономерности поведения моделируемого объекта во времени и пространстве). Над устойчивой моделью проводится целенаправленное исследование структуры и функций реальной системы. При этом результаты испытаний (как одного, так и множества) будут определяться случайным характером процессов. По данным этих экспериментов можно получить достаточно устойчивую статистику. Имитационное моделирование — частный случай математического моделирования.



К имитационному моделированию прибегают, когда:

- дорого или практически невозможно экспериментировать на реальном объекте;
- невозможно построить аналитическую модель: в системе есть время, причинные связи, последствия, нелинейности, стохастические (случайные) переменные;
- необходимо симитировать поведение системы во времени.



Цель имитационного моделирования состоит в воспроизведении поведения исследуемой системы на основе результатов анализа наиболее существенных взаимосвязей между её элементами или, другими словами, в разработке симулятора исследуемой предметной области для проведения различных экспериментов.

Виды имитационного моделирования

Моделирование	Характеристика
Дискретно-событийное	Подход к моделированию, предлагающий абстрагироваться от непрерывной природы событий и рассматривать только основные события моделируемой системы, такие как «ожидание», «обработка заказа», «движение с грузом», «разгрузка» и др. Дискретно-событийное моделирование наиболее развито и имеет огромную сферу приложений — от логистики и систем массового обслуживания до транспортных и производственных систем. Этот вид моделирования наиболее подходит для моделирования производственных процессов. Основан Дж. Гордоном в 1960-х гг.
Системная динамика	Для исследуемой системы строятся графические диаграммы причинных связей и глобальных влияний одних параметров на другие во времени, а затем созданная на основе этих диаграмм модель имитируется на компьютере. Такой вид моделирования более всех других парадигм помогает понять суть происходящих причинно-следственных связей между объектами и явлениями. С помощью системной динамики строят модели бизнес-процессов, развития города, модели производства, динамики популяции, экологии и развития эпидемии. Метод основан Дж. Форрестером в 1950-х гг.
Агентное	Относительно новое (1990—2000-е гг.) направление в имитационном моделировании. Используется для исследования децентрализованных систем, динамика функционирования которых определяется не глобальными правилами и законами (как в других парадигмах моделирования), эти глобальные правила и законы являются результатом индивидуальной активности членов группы. Цель агентных моделей — получить представление об этих глобальных правилах, общем поведении системы, исходя из предположений об индивидуальном, частном поведении её отдельных активных объектов и взаимодействии этих объектов в системе. Агент — некая сущность, обладающая активностью, автономным поведением, может принимать решения в соответствии с некоторым набором правил, взаимодействовать с окружением, а также самостоятельно изменяться

СИСТЕМЫ СЧИСЛЕНИЯ



Для того чтобы представить информацию в определённом виде, необходима знаковая система.

Языковые системы



Знаковая система — система, состоящая из алфавита и правил составления слов и формул из этого алфавита. Примером знаковой системы является **язык**.



Основными составляющими языка являются:

- алфавит;
- грамматика (свод правил, по которым знаки превращаются в слова);
- синтаксис (слова формируются в предложения);
- семантика (интерпретация, смысл).

Языки

Естественные	Формальные
<p>Сложились исторически.</p> <p>✓ Русский, английский и т. п.</p>	<p>Созданы для определённой цели.</p> <p>✓ Языки логики, C++, Pascal</p>

Виды систем счисления



Система счисления — формальный язык, предназначенный для записи чисел.

Системы счисления	Позиционные	Значение каждой цифры зависит от её положения в числе (разряда): десятичная, двоичная системы счисления																			
	Непозиционные	<p>Значение цифры не зависит от её положения в числе (разряде):</p> <ul style="list-style-type: none">• унарная, древнерусская и др.;• римская система счисления (I = 1, V = 5, X = 10, L = 50, C = 100, D = 500, M = 1000). <table><tr><td>I = 1</td><td>VI = 6</td><td>XI = 11</td><td>XVI = 16</td></tr><tr><td>II = 2</td><td>VII = 7</td><td>XII = 12</td><td>XVII = 17</td></tr><tr><td>III = 3</td><td>VIII = 8</td><td>XIII = 13</td><td>XVIII = 18</td></tr><tr><td>IV = 4</td><td>IX = 9</td><td>XIV = 14</td><td>XIX = 19</td></tr><tr><td>V = 5</td><td>X = 10</td><td>XV = 15</td><td>XX = 20</td></tr></table>	I = 1	VI = 6	XI = 11	XVI = 16	II = 2	VII = 7	XII = 12	XVII = 17	III = 3	VIII = 8	XIII = 13	XVIII = 18	IV = 4	IX = 9	XIV = 14	XIX = 19	V = 5	X = 10	XV = 15
I = 1	VI = 6	XI = 11	XVI = 16																		
II = 2	VII = 7	XII = 12	XVII = 17																		
III = 3	VIII = 8	XIII = 13	XVIII = 18																		
IV = 4	IX = 9	XIV = 14	XIX = 19																		
V = 5	X = 10	XV = 15	XX = 20																		

Позиционные системы счисления



Позиционная система счисления определяется натуральным числом $k > 1$, называемым **основанием системы счисления**.



Место цифры в числе называется **разрядом**, а количество цифр в числе — его **разрядностью**.

Система счисления	Основание	Алфавит
Десятичная	10	0 1 2 3 4 5 6 7 8 9
Восьмеричная	8	0 1 2 3 4 5 6 7
Двоичная	2	0 1
Шестнадцатеричная	16	0 1 2 3 4 5 6 7 8 9 A B C D E F

Свойства записи чисел

1	Для записи чисел в позиционной системе счисления с основанием k требуется k знаков (алфавит системы состоит из k цифр или букв)
2	Основание системы счисления, записанное в ней, всегда имеет вид 10 (читается «один ноль»)
3	С помощью n разрядов в позиционной системе счисления с основанием k могут быть записаны k^n чисел (от 0 до $k^n - 1$)
4	Если основание системы k больше 10, то числа старше 10 при записи обозначают прописными буквами латинского алфавита: A, B... Z. При этом числу 10 соответствует знак A, числу 11 — знак B и т. д.
5	При записи чисел в позиционной системе, отличной от десятичной, принято писать основание справа внизу: 345 — число дано в пятеричной системе счисления

СВЁРНУТАЯ ФОРМА ЗАПИСИ ЧИСЛА

Обычная форма записи числа выполняется как последовательность цифр, стоящих на своих разрядах.

$$A_{10} = a_{n-1}a_{n-2}\dots a_0, a_{-1}\dots a_{-m}$$

РАЗВЕРНУТАЯ ФОРМА ЗАПИСИ ЧИСЛА

Выполняется в виде разрядных слагаемых, записанных с помощью степени соответствующего разряда и основания степени (основания счёта).

$$A_{10} = a_{n-1} \cdot 10^{n-1} + a_{n-2} \cdot 10^{n-2} + \dots + a_0 \cdot 10^0, \\ a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m}$$

Цифры шестнадцатеричной системы счисления

16-ричная система	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Соответствие в десятичной системе	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Алфавиты основных систем счисления

Основание	Название	Алфавит
2	Двоичная	0 1
8	Восьмеричная	0 1 2 3 4 5 6 7
10	Десятичная	0 1 2 3 4 5 6 7 8 9
16	Шестнадцатеричная	0 1 2 3 4 5 6 7 8 9 A B C D E F



Другие системы счисления записываются аналогично вышеприведённым системам с тем лишь исключением, что основание степени будет соответствовать основанию счёта.

Двоичное представление информации



В двоичной системе счисления используются всего две цифры: 0 и 1. В алфавите десятичной системы счисления есть знаки от 0 до 9. Когда счёт достигает 9, вводится новый разряд (десятки), а единицы обнуляются и счёт начинается снова. Когда десятки доходят до 9, появляется третий разряд — сотни. Двоичная система счисления аналогична десятичной, за исключением того, что в формировании числа участвуют всего два знака-цифры: 0 и 1. Как только разряд достигает своего предела (т. е. единицы), появляется новый разряд, а старый обнуляется.

Перевод чисел из двоичной системы счисления в десятичную

В десятичной системе счисления любое число можно представить в форме суммы единиц, десятков, сотен и т. д.

$$1476 = 1000 + 400 + 70 + 6$$

Можно пойти ещё дальше и разложить так:

$$1476 = 1 \cdot 10^3 + 4 \cdot 10^2 + 7 \cdot 10^1 + 6 \cdot 10^0.$$

Посмотрите на эту запись внимательно. Здесь 1, 4, 7 и 6 — это набор цифр, из которых состоит число 1476. Все эти цифры поочерёдно умножаются на 10, возведённое в ту или иную степень

Перевод десятичного числа в двоичное

Один из способов перевода десятичного числа в двоичное — деление на 2 и формирование двоичного числа из остатков. Допустим, необходимо получить из числа 77 его двоичную запись:

$$77 / 2 = 38 \text{ (остаток 1),}$$

$$38 / 2 = 19 \text{ (остаток 0),}$$

$$19 / 2 = 9 \text{ (остаток 1),}$$

$$9 / 2 = 4 \text{ (остаток 1),}$$

$$4 / 2 = 2 \text{ (остаток 0),}$$

$$2 / 2 = 1 \text{ (остаток 0),}$$

$$1 / 2 = 0 \text{ (остаток 1)}$$

Десять — это основание десятичной системы счисления. Степень, в которую возводится десятка, — разряд цифры за минусом единицы.

Аналогично можно разложить и любое двоичное число. Только основание здесь будет 2:

$$10001001_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0.$$

Если посчитать сумму составляющих, то в итоге мы получим десятичное число, соответствующее 10001001:

$$1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 128 + 0 + 0 + 0 + 8 + 0 + 0 + 1 = 137.$$

Значит, число 10001001 по основанию 2 равно числу 137 по основанию 10. Записать это можно так:

$$10001001_2 = 137_{10}$$

Собираем остатки вместе, начиная с конца: 1001101. Это и есть число 77 в двоичном представлении. Проверим:

$$1001101_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 64 + 0 + 0 + 8 + 4 + 0 + 1 = 77_{10}.$$

Другой вид записи решения таких заданий — деление уголком. Перевод 19_{10} в двоичную систему выполняется делением на 2.

$$\begin{array}{r}
 -19 \mid 2 \\
 -18 \mid 9 \quad 2 \\
 \quad 1 \quad 1 \quad 1 \quad 1 \\
 \quad \quad 8 \quad 4 \quad 2 \\
 \quad \quad \quad 4 \quad 2 \\
 \quad \quad \quad \quad 0 \quad 0 \\
 \quad \quad \quad \quad \quad 2 \quad 1 \\
 \quad \quad \quad \quad \quad \quad 0 \quad 0
 \end{array}$$

Записываем остатки от деления справа налево для ответа. Получим: $19_{10} = 10011_2$



Обратите внимание на то, что в десятичной системе 57 и 75 — разные числа, так же и в двоичной системе 11001 и 10011 — разные числа, поскольку обе системы являются позиционными.

Порядок перевода числа из k -ичной системы в десятичную

1	Расставить над каждой цифрой числа его разряд справа налево, начиная с 0
2	Записать число в развёрнутом виде, используя степени числа
3	Цифры и буквы перевести в десятичную систему
4	Выполнить сложение и получить ответ

Порядок перевода числа из десятичной системы в k -ичную

1	Разделить число на основание системы уголком
2	Выписать остаток и перевести его в нужную систему счисления (это касается букв)
3	Продолжить делить частное на основание системы счисления до тех пор, пока в частном не получится 0
4	Выписать остатки в обратном порядке (справа налево)

Метод триад и тетрад



Иногда приходится сталкиваться с задачей перевода числа из двоичной системы в восьмеричную или из шестнадцатеричной в двоичную. Для подобных случаев существует два способа решения. Первый способ заключается в переводе числа в десятичную систему, а потом в нужную нам. Второй способ заключается в применении **метода триад и тетрад**.

Таблица триад и тетрад

A8	Триада
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

A16	Тетрада
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

A16	Тетрада
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111



Для преобразования двоичного числа в число с основанием «степень двойки» (4, 8, 16 и т. п.) необходимо разбить двоичную последовательность справа налево на группы по количеству цифр, равному степени, и каждую группу заменить соответствующей цифрой новой системы счисления. И наоборот, для перевода в двоичную систему числа с основанием, равным степени двойки, необходимо каждую цифру заменить группой цифр в двоичной системе.

Перевод числа из двоичной системы в восьмеричную	Перевод числа из двоичной системы в шестнадцатеричную
<p>1. Разбить число на триады (тройки цифр), начиная с младшего разряда, дополнив в случае необходимости старшую триаду нулями.</p> <p>2. Заменить каждую триаду соответствующей восьмеричной цифрой (см. таблицу триад и тетрад).</p> <p>✓ Число 1001011_2 перевести в восьмеричную систему счисления.</p> $001\ 001\ 011_2 = 113_8$	<p>1. Разбить число на тетрады (четвёрки цифр), начиная с младшего разряда, дополнив в случае необходимости старшую тетраду нулями.</p> <p>2. Заменить каждую тетраду соответствующей шестнадцатеричной цифрой (см. таблицу триад и тетрад).</p> <p>✓ Число 1011100011_2 перевести в шестнадцатеричную систему счисления.</p> $0010\ 1110\ 0011_2 = 2E3_{16}$
Перевод восьмеричного числа в двоичное	Перевод шестнадцатеричного числа в двоичное
<p>Заменить каждую цифру эквивалентной ей двоичной триадой (см. таблицу триад и тетрад).</p> <p>✓ Число 531_8 перевести в двоичную систему счисления.</p> $531_8 = 101011001_2$	<p>Заменить каждую цифру эквивалентной ей двоичной тетрадой (см. таблицу триад и тетрад).</p> <p>✓ Число $EE8_{16}$ перевести в двоичную систему счисления.</p> $EE8_{16} = 111011101000_2$
Переход из восьмеричной системы счисления в шестнадцатеричную и обратно	
Необходим промежуточный перевод чисел в двоичную систему (см. таблицу триад и тетрад)	
<p>✓ Число FEA_{16} перевести в восьмеричную систему счисления.</p> $FEA_{16} = 111111101010_2 = 111\ 111\ 101\ 010_2 = 7752_8$	<p>✓ Число 6635_8 перевести в шестнадцатеричную систему счисления.</p> $6635_8 = 110110011101_2 = 1101\ 1001\ 1101_2 = D9D_{16}$

Сложение и умножение в различных системах счисления



Обычно мы используем десятичную систему счисления и в ней проводим такие математические операции, как сложение, вычитание, умножение, деление и др. Рассмотрим, как можно складывать и умножать в других системах счисления.

Двоичная система счисления



Правила сложения и умножения в двоичной системе счисления можно представить в виде схемы или таблицы.

Правила сложения	Правила умножения
$0 + 0 = 0$	$0 \cdot 0 = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$
$1 + 1 = 10$	$1 \cdot 1 = 1$

Таблица сложения

+	0	1
0	0	1
1	1	10

Таблица умножения

×	0	1
0	0	0
1	0	1



В данных примерах действия выполняются в двоичной системе счисления:

- ✓ $1000_2 + 11_2 = 1011_2$;
- ✓ $111_2 + 1_2 = 1000_2$.

Троичная система счисления



Рассмотрим правила сложения и умножения в троичной системе счисления, представив их аналогично двоичной системе.

Правила сложения	Правила умножения
$0 + 0 = 0$	$0 \cdot 0 = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$
$0 + 2 = 2$	$0 \cdot 2 = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$
$1 + 1 = 2$	$1 \cdot 1 = 1$
$1 + 2 = 10$	$1 \cdot 2 = 2$
$2 + 0 = 2$	$2 \cdot 1 = 2$
$2 + 1 = 10$	$2 \cdot 2 = 11$
$2 + 2 = 11$	

Таблица сложения

+	0	1	2
0	0	1	2
1	1	2	10
2	2	10	11

Таблица умножения

×	0	1	2
0	0	0	0
1	0	1	2
2	0	2	11



В данных примерах действия выполняются в троичной системе счисления:

$$\checkmark 1000_3 + 11_3 = 1011_3;$$

$$\checkmark 111_3 + 1_3 = 112_3.$$



Если сравнить данные примеры с приведёнными выше подобными примерами сложения в двоичной системе, можно заметить, что в первом примере ответ повторяется, а во втором примере нет перехода через разряд.

Восьмеричная система счисления



Аналогично можно написать правила сложения и умножения для других систем счисления. Рассмотрим восьмеричную систему счисления, в которой используется 8 символов: 0, 1, 2, 3, 4, 5, 6, 7. Правил сложения и умножения очень много, удобнее представить их с помощью таблицы.

Таблица сложения

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Таблица умножения

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61



В данных примерах действия выполняются в восьмеричной системе счисления:

$$\checkmark 1000_8 + 11_8 = 1011_8;$$

$$\checkmark 111_8 + 1_8 = 112_8.$$



Если сравнить данные примеры с приведёнными выше подобными примерами сложения в двоичной и троичной системах, можно заметить, что в обоих примерах ответы получились такими же, как и в троичной системе.

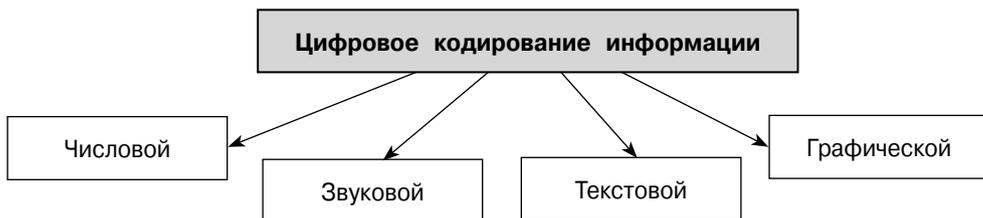
ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ



Рассматривая кодирование на абстрактном уровне, можно сказать, что вся информация представлена в виде 0 и 1, иными словами, используется двоичная система счисления.



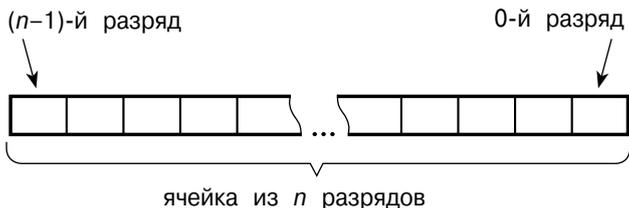
Ячейка, в которой хранится информация, называется **битом**. Представление информации с помощью последовательности битов называется **двоичным** или **цифровым кодированием**.



Представление числовой информации



При переводе из k -ичной системы в десятичную разряды нумеруются справа налево. Любое число, которое кодируется двоичным кодом, хранится в ячейке, состоящей из n разрядов. В каждом разряде содержится 0 или 1.



Для представления **целых чисел без знака** отводится 1, 2 байта информации, реже — 4 (при необходимости больше). При этом используются 8, 16 или 32 ячейки.

Порядок представления на компьютере целого числа без знака	
1	Перевести число в двоичную систему
2	Дополнить его нулями слева до заполнения ими нужного количества ячеек

Порядок представления на компьютере целого числа со знаком	
1	Перевести модуль числа в двоичную систему, получить прямой код
2	Создать обратный код, для этого заменить все 0 на 1 и все 1 на 0
3	К полученному коду прибавить 1



Под хранение **чисел со знаками** (отрицательных) также отводится 1, 2, 4 байта информации (при необходимости больше).

Представление текстовой информации



Для компьютерного представления текстовой информации все буквы кодируются числами, а текст представляется в виде набора цифр. Для того чтобы понять, какой символ какой букве соответствует, используется специальная **кодировочная таблица символов**.

Десятичный код	Двоичный код	Символ
32	100000	пробел
33	100001	!
34	100010	«
35	100011	#
36	100100	\$
37	100101	%
38	100110	&
40	101000	(
51	110011	3
52	110100	4
53	110101	5

Десятичный код	Двоичный код	Символ
54	110110	6
55	110111	7
56	111000	8
72	1001000	H
73	1001001	I
74	1001010	J
75	1001011	K
76	1001100	L
77	1001101	M
91	01011011	s
98	01100010	z



Основой компьютерных стандартов кодирования служит код для обмена информацией **ASCII**. В нём используется семибитовое кодирование: $2^7 = 128$ символов. В этой кодировке первые 32 символа — **управляющие**, остальные — **изображаемые**, т. е. представленные так, как они будут отражены на экране компьютера. Код числа и код изображаемого символа при переводе в двоичную систему не совпадают. Например, число 3 в двоичной системе имеет код 11, а изображаемый символ числа 3 кодируется как 110011.



Для русского языка используются кодировки CP-866, KOI-8, Windows-1251, в которых первые 128 символов совпадают с кодировкой ASCII, а остальные имеют отличия, т. е. одна и та же буква в этих кодировках имеет разные коды. Наиболее распространённой является **кодировка Unicode** — основной стандарт кодирования символов, включающий в себя знаки практически всех языков мира. Для кодирования одного символа информации используется 2 байта.

Представление графической информации



Дискретизация — преобразование графической информации из аналоговой формы в дискретную, т. е. разбиение непрерывного графического изображения на отдельные элементы.

Параметры перевода графической информации из аналоговой формы в дискретную

Цветовая модель	Математическая модель описания представления цветов в виде набора чисел (обычно из трёх или четырёх значений), называемых цветовыми компонентами или цветовыми координатами
Разрешение	Величина, определяющая количество точек на единицу площади
Глубина цвета	Количество битов, которое используется для кодирования цвета одного пикселя. От глубины цвета зависит количество воспроизводимых цветов и размер файла. Единицей измерения глубины цвета является бит/пиксель

Цветовые модели

RGB	CMYK	HSB
Red, Green, Blue — красный, зелёный, синий	Синий, Magenta, Yellow, Black — голубой, пурпурный, жёлтый, чёрный; для black в сокращение вошла последняя буква цвета, чтобы не возникло путаницы с обозначением синего Blue в RGB	Hue, Saturation, Brightness — тон, насыщенность, яркость
Применяется для вывода изображения на экраны мониторов и другие электронные устройства	Используется в полиграфии для стандартной печати	Является аналогом RGB, основана на её цветах, но отличается системой координат



Пиксель — наименьший логический двумерный элемент цифрового изображения. Другими словами, это минимальный и неделимый элемент (точка), из которого состоит изображение на экране монитора компьютера.



Формула зависимости количества отображаемых цветов от глубины цвета:

$$N = 2^i,$$

где N — количество цветов в палитре, i — глубина цвета в битах на один пиксель.



Формула расчёта информационного объёма растрового графического изображения:

$$V = \frac{K \cdot i}{k},$$

где V — информационный объём растрового графического изображения, K — количество пикселей в изображении, i — глубина цвета, k — коэффициент сжатия данных.



Информационный объём растрового графического изображения V измеряется в байтах, килобайтах, мегабайтах; количество пикселей (точек) в изображении K определяется разрешающей способностью носителя информации (экрана монитора, сканера, принтера); коэффициент сжатия данных k без сжатия равен 1.

Определение объёма файла размером $A \cdot B$ пикселей

1	Определить количество пикселей в изображении ($A \cdot B$)
2	Умножить количество пикселей на глубину цвета ($K \cdot i$)
3	При необходимости перевести количество битов в байты, Кбайты, Мбайты
4	Разделить результат на коэффициент сжатия при его наличии (указывается в условии задачи)

Зависимость количества кодируемых цветов от глубины цвета

Глубина цвета	Количество цветов	Кодируемые цвета	Объём файла размером 100×200 пикселей
1 бит/пиксель	2	Чёрно-белые	2,4 Кбайт
8 бит/пиксель	256	Цвета от белого до чёрного, проходящие через все градации серого. Аналогично для любого цвета, например все градации красного. Формат GIF	19,5 Кбайт
16 бит/пиксель и больше	65 536	Подходит для цветовой модели RGB	39 Кбайт

Представление звуковой информации



Звук — волновые колебания давления в упругой среде (воздухе, воде). Уровень звука измеряют в децибелах (дБ).

Параметры для кодирования звука

Частота звуковой дискретизации

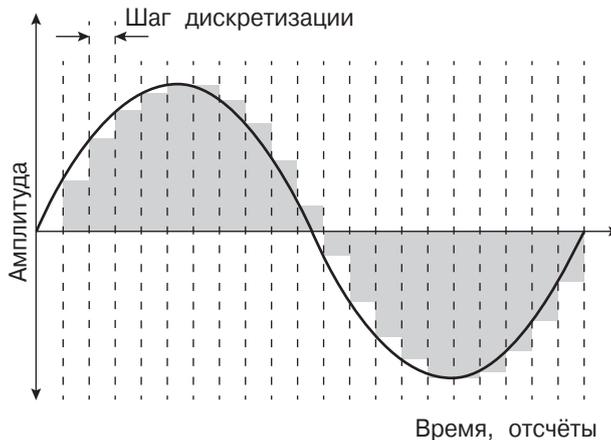
Количество измерений громкости звука за одну секунду. Количество измерений может лежать в диапазоне от 8 до 48 кГц

Глубина звука

Количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука



Герц (Гц или Hz) — единица измерения частоты периодических процессов (например, колебаний). 1 Гц обозначает одно исполнение такого процесса за одну секунду: 1 Гц = 1 колебание в секунду.



Аналого-цифровое преобразование звука. Непрерывная линия — аналоговый звук, прямоугольники — цифровой. Аналоговый звук разбивается на равные отрезки, и для каждого находится его значение

Качество оцифрованного звука

Качество звука	Глубина кодирования	Частота дискретизации
Радиотрансляция	8 бит	До 8 кГц
Звучание CD-диска	16 бит	До 48 кГц



Оцифровка звука — технология деления звука на небольшие временные шаги с последующей записью полученных значений в численном виде.

Формула расчёта объёма звукового файла	Пример
$V = f \cdot t \cdot l \cdot n,$ <p>где V — объём звукового файла, f — частота дискретизации (Гц), t — продолжительность звучания (с), l — глубина звука (бит), n — количество каналов трансляции (моно — 1, стерео — 2, квадро — 4)</p>	<p>Определите длительность звучания стереоаудиофайла, занимающего 1,3 Мбайт памяти при глубине звука 16 бит и частоте 48 кГц.</p> <p>Решение: $V = 600 \text{ Кбайт} = 600 \cdot 1024 \cdot 8 \text{ бит} = 4\,915\,200 \text{ бит}.$ $f = 48 \text{ кГц} = 48\,000 \text{ Гц}.$ $l = 16 \text{ бит}.$ Поскольку в условии говорится о стереофайле, значит, $n = 2$. Подставим имеющиеся значения в формулу $V = f \cdot t \cdot l \cdot n$. $4\,915\,200 = 48\,000 \cdot t \cdot 16 \cdot 2.$ $4\,915\,200 = 1\,536\,000 \cdot t.$ $t = 4\,915\,200 : 1\,536\,000 = 3,2 \text{ с}.$</p> <p>Ответ: 3,2 с</p>

ЭЛЕМЕНТЫ АЛГЕБРЫ ЛОГИКИ



Алгебра логики — один из основных разделов математической логики, в котором методы алгебры используются в логических преобразованиях. Она изучает логические операции над высказываниями с точки зрения их истинностного значения (истина, ложь).



Создателем алгебры логики является Дж. Буль (1815—1864), английский математик и логик, положивший в основу своего учения аналогию между алгеброй и логикой.



Действия, которые производятся над высказываниями, записываются в виде логических выражений.

Простое логическое выражение	Сложное логическое выражение
Состоит из одного высказывания и не содержит логических операций. В простом логическом выражении возможен только один из двух результатов — либо «истина», либо «ложь»	Содержит высказывания, объединённые логическими операциями. По аналогии с понятием функции в алгебре сложное логическое выражение содержит аргументы, которыми являются высказывания

Логические высказывания. Истинность высказывания



В повседневной жизни часто приходится слышать фразы-утверждения наподобие «Сейчас идёт дождь» или «Маша — девочка». Если в отношении повествовательного предложения (высказывания) можно однозначно утверждать, что его содержание истинно или ложно, то такое утверждение называется **логическим высказыванием**.



Не всякое предложение является логическим высказыванием, поэтому не всегда есть смысл говорить о его ложности или истинности. Например, высказывание «Саша — молодец» не является логическим высказыванием, потому что оценка «молодец» субъективна: для кого-то Саша молодец, а для кого-то — нет.

Высказывания	Характеристика
Простые (элементарные)	Обозначаются латинскими буквами (<i>A, B, C, D...</i>)
Сложные	Составленные из нескольких простых высказываний с помощью логических связок, таких как «не», «и», «или», «тогда и только тогда», «если... то...». Истинность или ложность получаемых таким образом сложных (составных) высказываний определяется значением входящих в них простых высказываний



Высказывание не может быть выражено побудительным или вопросительным предложением, оценка истинности или ложности которых невозможна.



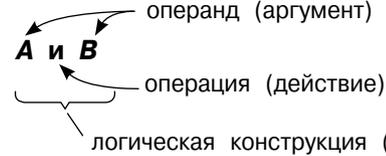
Обозначение результатов логических утверждений представлено в таблице.

1	Да (истинно)	True	Истина
0	Нет (ложно)	False	Ложь

В информатике для решения истинности конструкций используются обозначения 0 и 1, в программировании — True/False и 1/0.



Выражение « A и B » — логическая конструкция (функция), где A , B — операнды (аргументы), «и» — операция (действие).



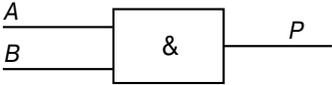
Операнд — аргумент операции, т. е. то, над чем выполняется операция.

Операция	Характеристика
Унарная (одноместная)	Операция, которая применяется к одному операнду. ✓ Изменение знака числа. К A применим операцию изменения знака, получим $(-A)$
Бинарная (двуместная)	Операция, которая выполняется над двумя операндами. Бинарную операцию принято обозначать знаком действия, который ставится между операндами. Такая форма записи называется инфиксной . ✓ Сложение $(A + B)$, вычитание $(A - B)$, умножение $(A \cdot B)$
Тернарная (трёхместная)	Операция, которая выполняется над тремя операндами. ✓ Среднее арифметическое трёх чисел, смешанное произведение векторов. Аналогом тернарной условной операции в математической логике является операция, которая записывается как $[a, b, c]$ и реализует алгоритм «если a , то b , иначе c ». ✓ Если завтра будет хорошая погода, то я пойду гулять, иначе останусь дома. Обычно тернарная условная операция ассоциируется с операцией $a ? b : c$, используемой в языках программирования

Логические операции



Таблицы истинности — таблицы с результатами каждой логической операции.

Логическое отрицание, или инверсия	Логическое умножение, или конъюнкция
<p>Унарная операция, в результате которой из данного высказывания (например, A) получается новое высказывание (не A), которое называется отрицанием исходного высказывания</p>	<p>Бинарная операция, соединяющая два высказывания и более с помощью связки «и» (например, «A и B»). Суть конъюнкции легко запомнить с помощью следующего выражения: «Конъюнкция (высказывание $A \& B$) истинна только тогда, когда оба высказывания A, B истинны»</p>
 <p>Схема операции логического отрицания (инвертор)</p>	
<p>Порядок работы схемы: на вход подаётся значение A, после прохождения прямоугольника с операцией \neg на выходе получается значение $\neg A$.</p> <p>✓ «Катя получила двойку», после применения отрицания получим: «Катя не получила двойку».</p> <p>Если на вход схемы отправить значение 0, то на выходе будет значение 1 ($\neg 0 = 1$). У данной схемы один вход и один выход, операция унарная</p>	<p>Порядок работы схемы: на вход подаются два значения — A, B, на выходе будет значение конъюнкции этих двух входящих.</p> <p>✓ На вход подали значения 0 и 1, выполнится $0 \& 1$ и на выходе получим 0</p>

Логическое отрицание, или инверсия

ОБОЗНАЧАЕТСЯ

$\neg A$, \bar{A} , не A , not A .

ЧИТАЕТСЯ

«Не A ».

- ✓ Высказывание A «10 простое число» — ложно.
- ✓ Высказывание $\neg A$ «10 не простое число» — истинно.

Таблица истинности операции инверсии

A	$\neg A$
0	1
1	0

Логическое умножение, или конъюнкция

ОБОЗНАЧАЕТСЯ

$A \wedge B$, $A \& B$, $A \cdot B$, $A \times B$, A и B , A and B .

ЧИТАЕТСЯ

« A и B ».

- ✓ «Этот треугольник равнобедренный и прямоугольный».

Данное высказывание может быть истинным только в том случае, если выполняются оба условия, в противном случае высказывание ложно.

Таблица истинности операции конъюнкции

A	B	$A \& B$
0	0	0
0	1	0
1	0	0
1	1	1

Логическое сложение, или дизъюнкция

Бинарная операция, соединяющая два высказывания и более с помощью связки «или» (например, « A или B »)



Схема операции дизъюнкции (дизъюнктор)

Порядок работы схемы: на вход подаются два значения — A , B , на выходе будет значение дизъюнкции этих двух входящих.

- ✓ На вход подали значения 0 и 1, выполнится $0 \vee 1$ и на выходе получим 1

ОБОЗНАЧАЕТСЯ

$A \vee B$, $A + B$, $A \mid B$, A или B , A or B .

ЧИТАЕТСЯ

« A или B ».

Высказывание будет истинным, если выполняются оба условия или хотя бы одно из условий

Дизъюнкция строго-разделительная, «исключающее ИЛИ»

Бинарная операция, соединяющая два высказывания с помощью связки «или», употреблённой в исключительном смысле. Суть операции легко запомнить с помощью следующего выражения: «Исключающее ИЛИ (высказывание $A \oplus B$) истинно только тогда, когда высказывания A и B имеют различные значения»

ОБОЗНАЧАЕТСЯ

$A \oplus B$, $A \vee\vee B$, $A \underline{\vee} B$, A xor B .

ЧИТАЕТСЯ

«Либо A , либо B ».

Высказывание истинно, если выполняется какое-то одно из условий

Таблица истинности операции дизъюнкции

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Таблица истинности операции «исключающее ИЛИ»

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Логическое следование, или импликация

Логическая операция, соединяющая два высказывания с помощью связки «если... то...»

ОБОЗНАЧАЕТСЯ

 $A \Rightarrow B, A \rightarrow B.$

ЧИТАЕТСЯ

«Если A , то B », « A влечёт B », «из A следует B », « A имплицирует B ».

Эквивалентность, или двойная импликация (равнозначность)

Бинарная операция, позволяющая из двух высказываний A и B получить новое высказывание $A = B$. Операция эквивалентности противоположна исключающему ИЛИ и имеет результат «истина» тогда и только тогда, когда значения переменных совпадают

ОБОЗНАЧАЕТСЯ

 $A \Leftrightarrow B, A \leftrightarrow B, A \equiv B.$

ЧИТАЕТСЯ

« A эквивалентно B », « A тогда и только тогда, когда B », « A то же самое, что B », « A равносильно B ».

Импликация — сокращённая запись для выражения $(\neg A \vee B)$. При решении задач можно уверенно заменять в выражении импликацию этой записью. Данная операция связывает два простых логических выражения, из которых первое является условием, а второе — следствием. Результат операции ложен только тогда, когда предпосылка есть истина, а следствие — ложь.

✓ «Если $3 \cdot 3 = 9$ (A), то Солнце — планета (B)», результат импликации $A \rightarrow B$ — ложь

Эквивалентность — сокращённая запись для выражения $(\neg A \wedge \neg B) \vee (A \wedge B)$. При решении задач можно уверенно заменять в выражении эквивалентность этой записью.

✓ «Треугольник будет прямоугольным тогда и только тогда, когда один из его углов равен 90° »

Таблица истинности операции импликации

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Таблица истинности операции эквивалентности

A	B	$A \leftrightarrow B$
0	0	1
1	1	1
1	0	0
0	1	0



Обозначения для логических связей (операций):

- отрицание (инверсия, логическое НЕ) обозначается \neg ($\neg A$);
- конъюнкция (логическое умножение, логическое И) обозначается \wedge (например, $A \wedge B$) либо $\&$ ($A \& B$);
- дизъюнкция (логическое сложение, логическое ИЛИ) обозначается \vee (например, $A \vee B$) либо $|$ ($A | B$);

- следование (импликация) обозначается \rightarrow ($A \rightarrow B$);
- тождество обозначается \equiv (например, $A \equiv B$); выражение $A \equiv B$ истинно тогда и только тогда, когда значения A и B совпадают (либо они оба истинны, либо они оба ложны);
- символ 1 используется для обозначения истины (истинного высказывания), символ 0 — для обозначения лжи (ложного высказывания).

Приоритеты логических связей



При работе с логическими конструкциями также важно сначала понять, какое действие выполнить первым. Для этого существуют определённые правила. Операции одинакового приоритета выполняются слева направо.

Правила приоритетов для логических связей

1	Выражение в скобках
2	Логическое НЕ
3	Логическое И
4	Логическое ИЛИ

Основные законы логики



Законы логики выражают необходимые условия для правильного последовательного мышления. Знание этих законов и порядка выполнения логических операций требуется для результативного упрощения логических выражений.

Название закона, формула	Содержание
Закон тождества: $A = A$	Любое высказывание тождественно самому себе
Закон непротиворечия: $A \& \neg A = 0$	Высказывание не может быть истинно и ложно одновременно
Закон исключённого третьего: $A \vee \neg A = 1$	В конкретный момент времени высказывание может иметь истинное или ложное значение, третье исключено

<p>Закон двойного отрицания: $\neg(\neg A) = A$</p>	<p>При двойном отрицании исходного суждения в итоге получится оно же</p>
<p>Свойства констант: $A \vee 0 = A$ $A \& 0 = 0$ $A \vee 1 = 1$ $A \& 1 = A$</p>	<p>Отрицание лжи есть истина. Отрицание истины есть ложь</p>
<p>Закон идемпотентности: $A \& A = A$ $A \vee A = A$</p>	<p>Закон, дающий возможность исключить повторяющиеся суждения</p>
<p>Закон коммутативности: $A \& B = B \& A$ $A \vee B = B \vee A$</p>	<p>Перемена мест высказываний не влияет на результат рассуждения</p>
<p>Закон ассоциативности: $A \vee (B \vee C) = (A \vee B) \vee C$ $A \& (B \& C) = (A \& B) \& C$</p>	<p>Логическое сложение и умножение ассоциативно, т. е. при наличии в выражении лишь конъюнкции или лишь дизъюнкции можно опускать скобки</p>
<p>Законы дистрибутивности: $A \vee (B \& C) = (A \vee B) \& (A \vee C)$ $A \& (B \vee C) = (A \& B) \vee (A \& C)$</p>	<p>Правила раскрытия скобок при конъюнкции и дизъюнкции</p>
<p>Закон поглощения: для конъюнкции: $A \vee (A \& B) = A$ для дизъюнкции: $A \& (A \vee B) = A$</p>	<p>По предыдущему закону можно раскрыть скобки и убрать лишнее A</p>
<p>Законы де Моргана: $\neg(A \& B) = \neg A \vee \neg B$ $\neg(A \vee B) = \neg A \& \neg B$</p>	<p>Отрицание суждения «A и B» эквивалентно суждению «не A или не B»</p>

Предикаты и кванторы



В алгебре высказываний для записи различных утверждений применяют логические знаки. Однако этих знаков недостаточно для выражения мысли типа «*Всякий элемент x из множества D обладает свойством $P(x)$* ». Введём новые логические знаки, обозначаемые \forall , \exists и $\exists!$.

Определение	Характеристика
Предикат	<p>Утверждение, содержащее переменные и принимающее значение 1 или 0 (истинно или ложно) в зависимости от значений переменных.</p> <p>✓ Утверждение «x делится на 9 на множестве целых положительных чисел» при $x = 9, 18, 27$ — истинно, при $x = 8, 16, 33$ — ложно.</p> <p>Предикатом в программировании является функция, которая принимает один аргумент и более и возвращает значения булева типа (от англ. <i>Boolean</i>, или <i>logical data type</i>)</p>
Множество истинности предиката	<p>Множество, на котором предикат принимает только истинные значения.</p> <p>✓ В примере «x делится на 9 на множестве целых положительных чисел» множеством истинности предиката является множество чисел $r = 9n$, где $n \in N$ (N — множество натуральных чисел: 1, 2, 3...)</p>
Невыполнимый предикат	<p>Предикат, принимающий значение «ложь» при всех допустимых значениях переменной. Так, выражение $x^2 + 3 = 0$ при всех действительных значениях x не принимает значения «истина», уравнение не имеет действительных корней</p>
Следствие предиката	<p>Предикат A является следствием предиката B, если для любых значений переменных, при которых B — истина, A тоже истина. Например, если число a де-</p>

	лится на 9 (B), то число a делится на 3 (A). Обратное утверждение не всегда будет являться истинным
Равносильные предикаты	Предикаты A и B называются равносильными, если из A следует B и из B следует A . Например, если число a делится на 3 (A), то число $a + 3$ делится на 3 (B). Обратное утверждение является истинным для любых значений a
Кванторы	Логические операторы, применение которых к предикатам превращает их в ложные или истинные высказывания; логические операции, которые ограничивают область истинности предиката и создают высказывание

Часто упоминаемые кванторы

Кванторы	Характеристика
Всеобщности	Обозначение: \forall . Читается: «для любого...», «для каждого...», «для всех...» или «каждый...», «любой...», «все...»
Существования	Обозначение: \exists . Читается: «существует...» или «найдётся...»
Существования и единственности	Обозначение: $\exists!$. Выражение «существует и единственный...»



В математической логике приписывание квантора к формуле называется **связыванием** или **квантификацией**.



Пусть предикат « x кратно 7».

С помощью квантора всеобщности можно записать следующие ложные высказывания:

- 1) любое натуральное число делится на 7;
- 2) каждое натуральное число делится на 7;
- 3) все натуральные числа делятся на 7.

Квантор будет иметь вид: $(\forall x \in N) P(x)$.

СРЕДСТВА ИКТ

АРХИТЕКТУРА КОМПЬЮТЕРА И КОМПЬЮТЕРНЫХ СЕТЕЙ



В современном мире одним из основных средств информационных и коммуникационных технологий (ИКТ) является компьютер. Архитектура компьютера описывает его организацию и принципы функционирования структурных элементов.



Архитектура компьютера — структура вычислительной машины, определяющая проведение обработки информации и включающая методы преобразования информации в данные и принципы взаимодействия технических средств и программного обеспечения.



Человек создал компьютер для упрощения своей работы. В процессе обработки информация проходит несколько уровней, поэтому то, что отражается на экране компьютера или телефона, внутри выглядит совсем иначе. Э. Таненбаум выделяет 6 уровней изменения информации: от понятной компьютеру до понятной человеку.

Уровень 5

Уровень языка прикладных программистов

Уровень 4

Трансляция (компилятор)

Уровень ассемблера

Уровень 3

Трансляция (ассемблер)

Уровень операционной системы

Уровень 2

Частичная интерпретация (операционная система)

Уровень архитектуры набора команд

Уровень 1

Интерпретация (микропрограмма) или непосредственное выполнение

Уровень микроархитектуры

Уровень 0

Аппаратное обеспечение

Цифровой логический уровень

Уровни изменения информации

Уровень	Характеристика
0	<p>Цифровой логический уровень — аппаратное обеспечение машины, включающее ряд логических элементов — вентиляей.</p> <p>Вентиль — часть электронной логической схемы, выполняющая элементарную логическую функцию (НЕ, И, ИЛИ). Несколько вентиляей формируют одну неделимую ячейку памяти (бит), в которой может храниться 0 или 1. Биты памяти, объединённые по 8, 16, 32, 64, образуют регистры — быстродействующие ячейки памяти различной длины, доступ к которым организован быстрее, чем к простым ячейкам</p>
1	<p>Микроархитектурный уровень, интерпретация (микропрограмма). На этом уровне находятся наборы, состоящие обычно из 8 или 32 регистров, которые формируют локальную память и схему, называемую АЛУ (арифметико-логическое устройство). АЛУ исполняет простые арифметические операции</p>
2	<p>Уровень архитектуры системы команд. На этом уровне набор машинных команд описывается и исполняется микропрограммой-интерпретатором или аппаратным обеспечением. Правильный выбор архитектуры системы команд важен для реализации вычислений наиболее эффективным образом (т. е. за минимальное время)</p>
3	<p>Уровень операционной системы. Это гибридный уровень: одна часть команд интерпретируется операционной системой, другая — микропрограммой</p>
4	<p>Уровень ассемблера представляет собой символическую форму одного из языков более низкого уровня</p>
5	<p>Уровень языка прикладных программистов. Прикладная программа или приложение — программа, предназначенная для выполнения определённых задач и рассчитанная на непосредственное взаимодействие с пользователем. Языки, разработанные для прикладных программистов, называются языками высокого уровня. Наиболее известные среди них — С, С++, Java, Perl, Python и PHP</p>

Организация работы компьютеров и компьютерных систем



Компьютер — электронно-вычислительное устройство или система, способная выполнять заданную, чётко определённую, изменяемую последовательность операций.

Компьютер = аппаратная часть + программное обеспечение

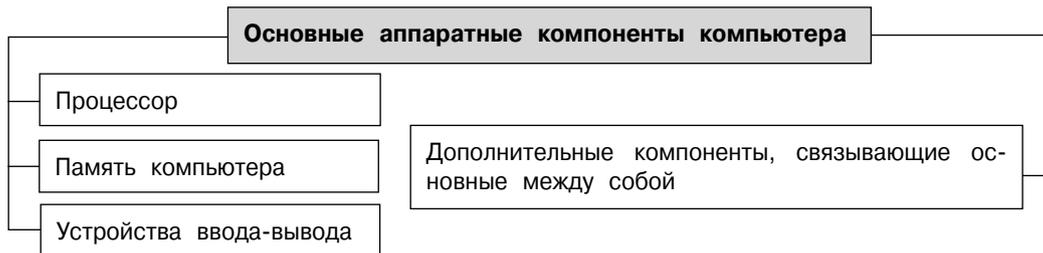


Аппаратное и программное обеспечение логически эквивалентны. Любая операция, исполняемая программным обеспечением, может быть реализована аппаратным обеспечением, и, наоборот, любая команда, исполняемая аппаратным обеспечением, может быть смоделирована программно.

Аппаратное обеспечение



Аппаратное обеспечение компьютера, или *hardware* (англ. *hard* — «твёрдый», *ware* — «изделие»), — набор устройств, из которых он состоит.





Корпус системного блока



Жёсткий диск



Процессор



Видеокарта



Кулер



Оперативная память



Материнская плата



Монитор



Шлейф



Блок питания

Аппаратные компоненты компьютера



Функции по объединению всех компонентов компьютера осуществляет **материнская (системная) плата**, на которой размещаются центральный процессор, основная память, системная шина, различные контроллеры и другие устройства.

Процессор



Процессор — центральная часть компьютера, которая выполняет заданные программой преобразования информации и осуществляет управление всем вычислительным процессом.

Процессор	Характеристика
Мобильный	Применяется в современных телефонах, планшетах
Десктопный	Используется в компьютерах и ноутбуках



Эти два типа процессоров основаны на разных архитектурах, они взаимодействуют с операционной системой и всем софтом с помощью разного набора команд.



Микропроцессор — программно управляемое электронное цифровое устройство, обрабатывающее информацию в цифровом виде и выполняющее арифметические и логические операции. Микропроцессор совместно с микропроцессорным комплектом (чипсетом) также управляет работой основных узлов и блоков компьютера.

Устройство управления (УУ)

Основной координатор всех действий компьютера, управляющий последовательностью выполнения команд и движением (потоками) данных в компьютере

Арифметико-логическое устройство (АЛУ)

Устройство, выполняющее все вычислительные операции

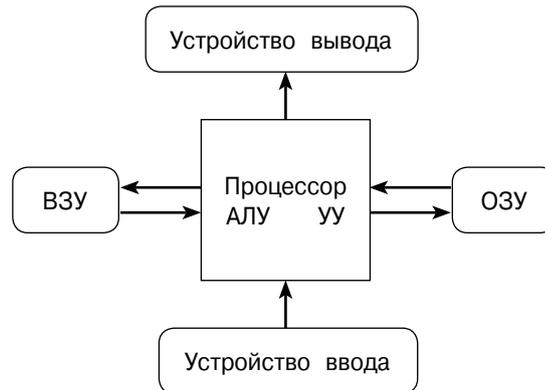


Схема Дж. фон Неймана. Стрелки отражают движение информации

Память компьютера



Память	Характеристика
Оперативная (ОЗУ — оперативное запоминающее устройство)	Представляет собой набор микросхем, предназначенных для временного хранения данных, когда компьютер включён. В этой памяти хранятся исполняемые программы и данные
Постоянная (ПЗУ — постоянное запоминающее устройство)	Микросхема, предназначенная для длительного хранения данных, в том числе когда компьютер выключен. К ПЗУ можно обращаться только при чтении данных и программ, но записывать в неё нельзя. ПЗУ хранит вспомогательные программы, которые используются многократно в процессе решения любых задач. Этот набор программ образует базовую систему ввода-вывода (BIOS — <i>Basic Input/Output System</i>), он получает управление при выключении и сбросе (reset) системной платы
Внешняя (ВЗУ — внешнее запоминающее устройство)	Состоит из накопителей на жёстких магнитных дисках (HDD), дисководов чтения и записи оптических дисков (CD и DVD), а также карты памяти и флеш-накопителей
Кеш-память	Сверхбыстрая память, используемая процессором для временного хранения наиболее часто используемых данных

Системная шина, контроллеры и другие устройства

Устройство	Функция
Системная шина	Передаёт данные между функциональными блоками компьютера
Контроллеры	Обеспечивают обмен информацией с внешними устройствами
Чипсет (набор микросхем, обеспечивающих согласованную работу всех аппаратных средств компьютера и ряда других подсистем)	Обеспечивает удобство и функциональность конкретной материнской платы (электропитания, мониторинга физических и электрических параметров)
Видеокарта (видеоадаптер, графическая карта)	Преобразует изображение, находящееся в памяти компьютера, в видеосигнал для монитора
Блок питания	Обеспечивает питание всех компонентов, подключённых к материнской плате, включая её саму

Устройства ввода-вывода



Периферийные устройства — устройства, предназначенные для ввода или вывода информации: принтеры, клавиатуры, мыши, сканеры и т. д. Подсоединение их к компьютеру производится через специальные интерфейсы — порты ввода/вывода. По способу передачи информации порты ввода/вывода могут быть **последовательными** и **параллельными** (несколько битов информации передаётся одновременно).

Устройства ввода информации	
Клавиатура	Сканер
Манипуляторы	Звуковая карта, микрофон
Графические планшеты	Игровые манипуляторы
Устройства вывода информации	
Монитор	Акустические колонки, наушники
Принтер	Модем

Программное обеспечение



Программное обеспечение (ПО), или *software* (англ. *soft* — «мягкий», *ware* — «изделие»), — набор программ, управляющих действиями компьютера при его использовании для решения задач.

Структура программного обеспечения компьютера

Системное ПО	Базовое ПО	Операционные системы (ОС)
		Оболочки ОС
		Сетевые ОС
	Сервисное ПО	Драйверы специфических и специальных устройств
		Программы диагностики работоспособности компьютера
		Антивирусные программы
		Программы обслуживания дисков
		Программы архивирования данных
Прикладное ПО	Приложения общего назначения	
	Приложения специального назначения	
Инструментальное ПО	Языки программирования	



Программное обеспечение — неотъемлемая часть компьютерной системы. Оно является логическим продолжением технических средств. Сфера применения конкретного компьютера определяется созданным для него системным ПО.

Системное ПО



Системное ПО — программы, которые предназначены для управления работой компьютера и вычислительной сети, их диагностики и профилактики, выполнения различных вспомогательных технологических процессов.

Базовое ПО	Минимальный набор программных средств, обеспечивающих работу компьютера. В него входят операционная система, операционные оболочки (обычно текстовые и графические), сетевая операционная система
Сервисное ПО	Включает программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя. К ним относятся драйверы специфических и специальных устройств, которые не поставляются в составе ОС

Виды системного ПО

Базовое ПО	Операционная система	Предназначена для управления выполнением пользовательских программ, планирования вычислительных ресурсов ЭВМ и управления ими
	Операционные оболочки	Специальные программы, предназначенные для облегчения общения пользователя с командами операционной системы. Операционные оболочки имеют текстовый и графический варианты интерфейса
	Сетевые операционные системы	Комплекс программ, обеспечивающих обработку, передачу и хранение данных в сети

Сервисное ПО	Драйверы	Компьютерные программы, с помощью которых другие программы получают доступ к аппаратному обеспечению устройства
		Программы диагностики работоспособности компьютера
	Антивирусные программы	Программы, выполняющие задачи обнаружения компьютерных вирусов, а также нежелательных вредоносных программ, восстановления заражённых (модифицированных) такими программами файлов и предотвращения заражения файлов или ОС вредоносным кодом
	Программы обслуживания дисков	Программы, способные тестировать производительность, проверять кластеры, объединять разделы или восстанавливать винчестер
	Программы архивирования данных	Программы, предназначенные для сжатия без потерь одного и более файлов в единый файл-архив или в серию архивов для удобства переноса и (или) хранения данных. Обратное действие — распаковка архива — выполняется с помощью этого же архиватора
	Программы обслуживания сети	Коммуникационные программы, организующие обмен информацией между компьютерами

Прикладное ПО



Прикладное ПО — программы, предназначенные для реализации конкретных задач по обработке данных, которые пользователь решает в ходе своей деятельности.

Виды прикладного ПО

Приложения	Характеристики
Общего назначения	К данному типу относится комплект офисных приложений MS Office или LibreOffice (для macOS). Эти комплекты предназначены для работы с текстом (текстовые редакторы), таблицами (электронные таблицы), пакетами для создания презентаций, СУБД (базы данных)
Специального назначения	В число таких приложений входят браузеры — средства просмотра веб-страниц, графические редакторы, бухгалтерские системы типа 1С, веб-редакторы

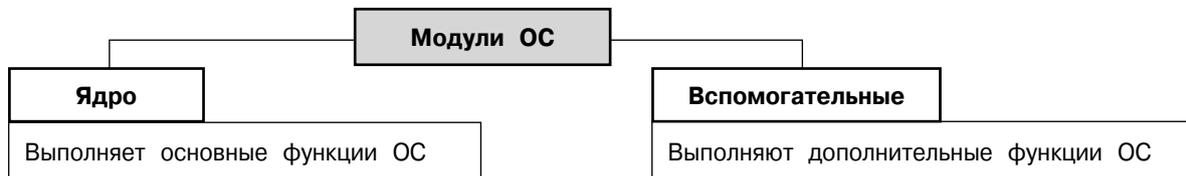


Инструментальное ПО — совокупность программ, обеспечивающих разработку новых компьютерных программ на языке программирования (Pascal, Basic, C, C++, Java, LOGO, LISP и др.).

Операционные системы



Операционная система (ОС) — программный комплекс, предоставляющий пользователю среду для выполнения прикладных программ и управления ими, а прикладным программам — средства доступа и управления аппаратными ресурсами.





Ядро — центральная часть операционной системы, управляющая выполнением процессов, ресурсами вычислительной системы и предоставляющая процессам координированный доступ к этим ресурсам. Основными ресурсами являются **процессорное время, память и устройства ввода-вывода**. Доступ к файловой системе и сетевое взаимодействие также могут быть реализованы на уровне ядра.

Структура ОС	Характеристика и функции
Программный модуль	Управляет файловой системой: процесс работы компьютера сводится к обмену файлами между устройствами. Файлы ОС хранятся во внешней, долговременной памяти (на жёстком, гибком или лазерном диске). Но программы могут выполняться, только если они находятся в компьютере, поэтому файлы нужно загрузить в оперативную память (ОП)
Командный процессор	Часть операционной системы, обрабатывающая команды, вводимые с терминала или из командного файла, и запускающая задачи для их выполнения
Драйверы	Специальное программное обеспечение, необходимое для взаимодействия операционной системы и подключаемого устройства, причём каждому устройству соответствует свой драйвер. Без драйверов ОС не будет работать, ведь именно они осуществляют взаимодействие между всем аппаратным обеспечением компьютера и ОС. При включении компьютера производится загрузка драйверов в оперативную память. Пользователь имеет возможность вручную установить или переустановить драйверы
Графический интерфейс	Программные модули графического интерфейса-программы позволяют осуществлять взаимодействие человека с компьютером в форме диалога
Справочная система	Обеспечивает получение информации о функционировании ОС в целом и о работе её отдельных модулей

Структура ОС	Характеристика и функции
Сервисные программы (утилиты)	Сервисные программы для обслуживания дисков (проверка, сжатие, дефрагментация и т. д.), выполнения операций с файлами (архивирование, копирование и т. д.) и работы в компьютерных сетях



Команда — описание элементарной операции, которую должен выполнить компьютер или формальный исполнитель. У каждой команды есть свой двоичный код.



Графический интерфейс пользователя (ГИП, ГПИ) — разновидность пользовательского интерфейса, в котором элементы интерфейса (меню, кнопки, значки, списки и т. п.), представленные пользователю на дисплее, исполнены в виде графических изображений.

Классификация ОС

По числу одновременно обслуживаемых пользователей	
Однопользовательские — одновременно поддерживается не более одного сеанса пользователя	Многопользовательские — одновременно поддерживается множество сеансов пользователей
По числу одновременно выполняемых процессов	
Однозадачные — не более одного работающего процесса	Многозадачные — множество работающих процессов
По назначению	
Универсальные — рассчитаны на решение любых задач пользователя	Специализированные — работают с фиксированным набором программ

По способу загрузки

Загружаемые ОС (большинство)

Системы, находящиеся в памяти вычислительной системы постоянно

По типу компьютеров

Мобильные (ОС Android и iOS)

Десктопные системы (ОС семейства Windows и системы класса UNIX — Linux и macOS)

Загрузка ОС



Системный диск — жёсткий, гибкий или лазерный диск, на котором находятся файлы операционной системы и с которого производится её загрузка.

POST (Power-On Self-Test)

Самотестирование после включения

BIOS (Basic Input/Output System)

Базовая система ввода-вывода

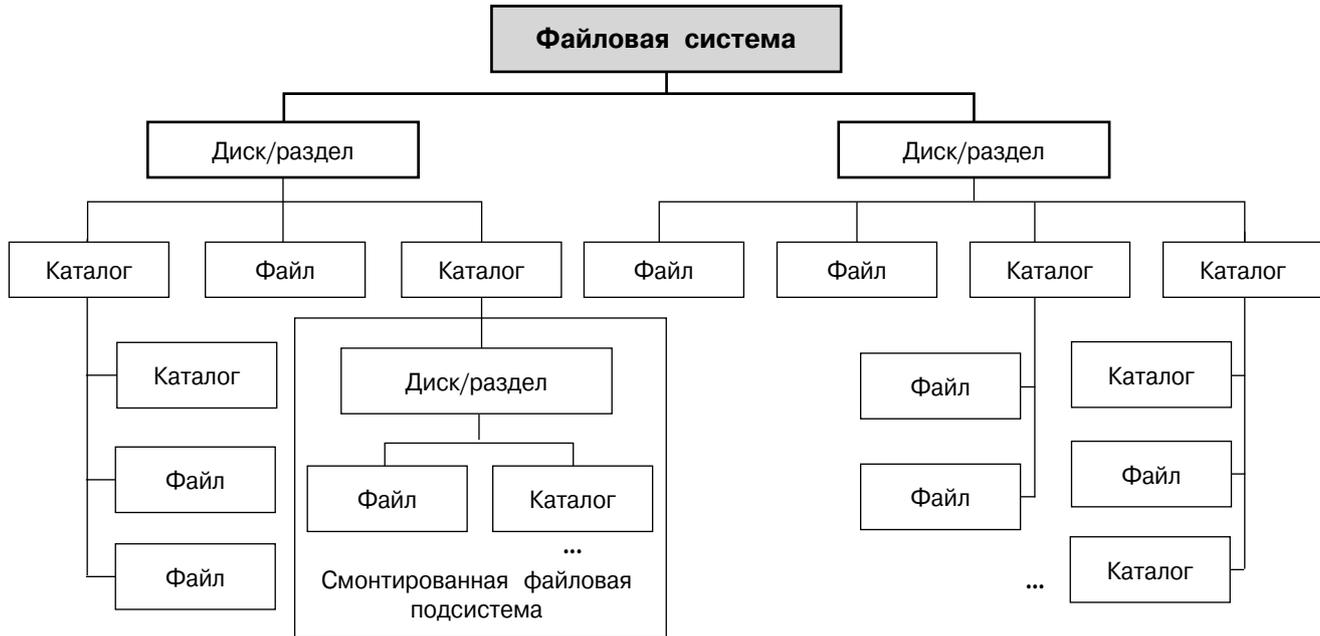
Этапы процесса загрузки ОС

1	Включение
2	Самотестирование компьютера (POST)
3	Программа BIOS ищет программу-загрузчика ОС на системном диске
4	Программа-загрузчик ОС загружается в оперативную память
5	Программа-загрузчик находит файлы ОС на системном диске и загружает их в оперативную память
6	Поиск логического раздела жёсткого диска, загрузка ядра ОС
7	Передача управления командному процессору

Файловая система



Файловая система — система, которая определяет и контролирует способ названия и хранения данных на носителе/накопителе информации: жёстком диске или SSD, флеш-накопителе и др.



Определение	Характеристика
Каталоги	Системные файлы, предназначенные для поддержки структуры файловой системы. Если на диске хранится много файлов, то для упрощения и удобства поиска их объединяют в папки по тематике. Одни папки могут находиться внутри других, тогда их называют вложенными каталогами или папками
Файл	Программа или данные, имеющие имя и хранящиеся в долговременной (внешней) памяти



Существуют универсальные файловые системы, например FAT32, NTFS и exFAT, которые видят все ОС: Windows, Linux, macOS, IOS, Android и др. Их читают практически все фотоаппараты, видеокамеры и иное современное оборудование.

Типы файлов

Тип	Расширение
Исполняемые файлы	.exe
Текстовые файлы	.txt, .doc, .docx, .rtf
Файлы баз данных	.dbf, .mdb
Графические файлы	.bmp, .tif, .cdr, .jpg, .png, .ico
Звуковые файлы	.mod, .wav, .wma, .mp3
Видеофайлы	.avi, .mpg, .mp4
Форматы Internet	.html, .asp, .asf, .webp
Системные файлы	.sys, .bat



Действия с файлами:

- создание;
- удаление;
- открытие;
- закрытие;
- поиск;
- переименование;
- чтение;
- запись;
- добавление данных;
- установление атрибутов.



Все операционные системы связывают с файлом другую информацию, например дату и время последней модификации и его размер. Такие дополнительные сведения называются **атрибутами файла**.

Атрибут	Значение
Защита	Кто и каким образом имеет право получить доступ к файлу
Пароль	Набор символов для получения доступа к файлу
Создатель	Идентификатор создателя файла
Владелец	Текущий владелец
Флаг «только для чтения»	0 — для чтения и записи, 1 — только для чтения
Флаги блокировки	0 — незаблокированный, 1 — заблокированный
Длина записи	Количество байтов в записи
Позиция ключа	Смещение ключа внутри каждой записи
Длина ключа	Количество байтов в поле ключа
Время создания	Дата и время создания файла
Время последнего доступа	Дата и время последнего доступа к файлу
Время внесения последних изменений	Дата и время внесения в файл последних изменений
Текущий размер	Количество байтов в файле
Максимальный размер	Количество байтов, до которого файл может увеличиваться

ТЕХНОЛОГИИ СОЗДАНИЯ И ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ



Написание текстовых сообщений, редактирование рефератов, курсовых и дипломных работ, создание исследовательских проектов, удаление старых текстовых файлов, составление электронных писем, кодов для программ — вся эта деятельность относится к **обработке текстовой информации**.

Программы для работы с текстовой информацией

Виды программ	Описание
Текстовые редакторы	Позволяют создавать, редактировать, сохранять текстовую информацию. Функция форматирования вида текста отсутствует. ✓ Стандартный «Блокнот» от Windows, Notepad++, KeyPad+, Мой Офис Текст
Редакторы кодов программ	Позволяют упрощать процесс создания кодов для программ. Подобные программы выдают подсказки функций и операторов, подсвечивают необходимые слова или строки, выполняют другие подобные задачи. ✓ Visual Studio Code, Sublime Text, Atom, Notepad++
Текстовые процессоры	Позволяют создавать текстовые документы с последующей возможностью вставки текста и другой информации, а также форматирования текста. Эти программы отображают создаваемый документ на экране в том виде, в каком

Виды программ	Описание
	<p>он будет выводиться на печать, предоставляют широкие возможности для редактирования. Они позволяют изменять размер, вид и цвет шрифта, отступы абзацев, параметры страниц, вставлять в текст таблицы, формулы, графические изображения и звуковые объекты.</p> <p>✓ Microsoft Word, LibreOffice Writer, WPS Office Writer</p>
Настольные издательские системы (НИС)	<p>Расширенные программы, предназначенные для профессиональной издательской деятельности, позволяющие осуществлять компьютерную вёрстку книг и периодических изданий — газет и журналов.</p> <p>✓ QuarkXPress, Adobe InDesign, Scribus, Microsoft Publisher, Pages</p>



Компьютерная вёрстка — создание макетов произвольного размера, которые впоследствии можно распечатывать в типографии. НИС создают макеты не только книг или газет, но и любой печатной продукции: рекламных объявлений, подарочной упаковки и т. п.

Что позволяют делать НИС

Компоновать (верстать) текст

Использовать всевозможные шрифты и полиграфические изображения

Осуществлять редактирование на уровне лучших текстовых процессоров

Обрабатывать графические изображения

Обеспечивать вывод документов высокого качества и др.

Работа с текстовым документом



Текст является одной из важнейших форм представления информации об окружающем мире. **Текстовый документ** — информация, представленная на бумажном, электронном или ином материальном носителе в текстовой форме.

Основные элементы	Характеристика
Символ	Минимальная единица текстовой информации
Слово	Произвольная последовательность букв и цифр, ограниченная с двух сторон служебными символами
Строка	Произвольная последовательность символов между левой и правой границами абзаца
Предложение	Произвольная последовательность слов, завершающаяся точкой
Абзац	Часть текста, которая завершается специальным символом конца абзаца, при этом допускаются пустые абзацы
Страница	Страницу составляют строки и абзацы, таблицы и внедрённые в документ объекты
Документ	Наиболее крупная единица, в которой все составляющие текст абзацы определённым образом структурированы, снабжены при необходимости заголовками, выстроена иерархия структурных разделов

Этапы работы с текстовым документом

Открытие		Сохранение		Создание	
1	Найти файл на рабочем столе или в папке	1	Нажать <i>Файл</i> => <i>Сохранить как</i>	1	Открыть текстовый редактор (процессор)
2	В зависимости от настроек кликнуть на нём один или два раза или вызвать контекстное меню правой кнопкой мыши и выбрать <i>Открыть</i>	2	Назвать документ любым именем и сохранить	2	Нажать в строке меню вкладку <i>Файл</i> => <i>Создать (Новый)</i>
		3	Закреть документ или продолжить редактирование	3	Внести нужный текст, необходимые дополнения, отформатировать текст



Кроме создания, открытия, закрытия, сохранения, печати файла, необходимо уметь осуществлять редактирование и форматирование. При **редактировании** документа изменяется его содержание, при **форматировании** — внешний вид.

Редактирование	Форматирование
<p>Редактирование предполагает внесение изменений в текст документа. К редактированию можно отнести следующие операции:</p> <ul style="list-style-type: none"> • набор документа; • копирование; • перемещение частей текста; • вырезание частей текста; • исправление и т. п. 	<p>Процесс изменения характеристик документа или его частей. К форматированию относятся операции:</p> <ul style="list-style-type: none"> • изменение параметров абзацев; • изменение параметров шрифта; • изменение параметров заголовков; • преобразование текстовой информации в таблицы; • изменение колонтитулов, нумерация страниц

Шрифт

Параметры шрифта	Характеристика
Тип шрифта (гарнитура)	Может быть Times, Courier и т. д.
Размер шрифта (кегель)	Задаётся в пунктах: 10 пт, 14 пт, 16 пт и т. д.
Начертание	Обычный, полужирный, курсив, полужирный курсив
Тип подчёркивания	Одинарное, двойное, волнистое и т. д.
Цвет	Цвет символов, цвет фона символов
Эффекты	Верхний и нижний индекс, зачёркивание, тень и т. д.



Для поиска отдельных слов используются **поисковые средства текстового редактора**. При этом нужно воспользоваться строкой поиска или вызвать её с помощью набора клавиш. В macOS вызвать строку поиска можно с помощью сочетания клавиш `cmd + F`, в Windows — `Ctrl + F`.

Шаблоны текстовых документов



При работе на компьютере часто возникает необходимость использовать повторяющиеся тексты, предполагающие определённую форму изложения, например трудовой договор, заявление на отпуск и т. п. Для того чтобы не перепечатывать весь документ заново, при его составлении можно воспользоваться готовым образцом или шаблоном.



Шаблоны — документы, содержащие рекомендуемые разделы, обязательный текст или специальную эмблему, а также элементы управления содержимым или стили форматирования. Можно сказать, что шаблон — это модель для создания нового документа.

Этапы создания нового документа

1	Открыть новый документ
2	Внести нужный текст, необходимые дополнения, отформатировать текст
3	Нажать <i>Файл => Сохранить как => Шаблон</i>
4	Назвать документ любым именем
5	Закреть документ



Если открыть шаблон, моментально создаётся его копия, которая имеет такое же расширение, как и текстовый документ: .dotx, .dot, .dotm.

Проверка орфографии и грамматики

Вид проверки	Функция
Автоматическая проверка	Обеспечивает выявление ошибок в тексте при его наборе. После того как слово или предложение набрано, программа подчёркивает слова и выражения с грамматическими и орфографическими ошибками линиями различных цветов. С помощью контекстного меню можно выбрать варианты исправления ошибки, добавить слово в словарь пользователя или указать пропуск всех таких элементов

Проверка правописания вручную	Проводится после набора некоторого объёма текста и отображает для каждой ошибки диалоговое окно. Такая проверка удобна, если нужно убедиться в отсутствии ошибок в некоторой части документа. Для выполнения этой задачи используется диалоговое окно <i>Правописание (Правописание и грамматика)</i> и т. п.
Автозамена	Служит для автоматического исправления ошибок или опечаток, которые наиболее часто возникают при наборе. В словарь автозамены заносятся ошибочные и соответствующие правильным варианты. При ошибочном наборе слова из этого словаря текстовый редактор автоматически исправит его написание



Словарь автозамены пополняется и редактируется. Пользователь может самостоятельно настроить программу по своим предпочтениям. Например, при наборе слова «комп» автозамена выдаст «компьютер».

Словари и тезаурусы. Машинный перевод



Тезаурус — словарь, дающий описание лексики данного языка во всём её объёме и полноте через соотношение с другими лексическими единицами (синонимы, антонимы, паронимы). Этот тип словарей используется в машинном переводе.



Машинный перевод — процесс перевода текста с одного языка на другой с помощью специальной компьютерной программы.

Области применения машинного перевода

Перевод письма

Чтение иностранных сайтов

Перевод инструкции для техники

Чтение профессиональной документации

Решение задач профессионального перевода



Некоторые переводчики можно использовать в браузере, для работы с другими надо скачать файл или приложение. Программами машинного перевода являются PROMT, Pragma, ProLing Office. Часто используемый переводчик Google Translate насчитывает в своём арсенале более 100 языков. Его можно использовать онлайн.



Существует множество программ — словарных оболочек, которые организуют поиск слов в **электронных словарях**. Такие словари могут быть электронными версиями печатных изданий либо же разработанными специально для использования с конкретной словарной оболочкой, как, например, ABBYY Lingvo.

Редактирование математических текстов.

Графическое представление математических объектов



Технические тексты часто содержат математические формулы. Текстовый процессор позволяет вводить в текст формулы с помощью **библиотеки математических символов**.



Для упорядочения представления данных используются **таблицы**, состоящие из ячеек, которые формируют строки и столбцы. Ячейки могут содержать различную информацию: текст, рисунки, формулы. Таблицы позволяют сортировать данные, выполнять различные вычисления, обрабатывать и обобщать большие массивы числовых данных, упорядочивать по заданным критериям, создавать макеты страниц.

Редактирование

Вставка формулы		Добавление таблицы	
1	Нажать вкладку <i>Вставка</i> => <i>Формула</i> / <i>Вставить новую формулу</i> (в зависимости от редактора и его возможностей)	1	Нажать вкладку <i>Вставка</i> => <i>Таблица</i>
2	Вставить нужную формулу и значения для неё	2	Выбрать нужное количество строк и колонок
Добавление простых рисунков		Добавление диаграммы	
1	Нажать вкладку <i>Вставка</i> => <i>Рисунок</i> , <i>Фигура</i>	1	Выделить необходимые для диаграммы ячейки с данными
2	Выбрать рисунок из имеющихся на ПК или из Интернета	2	Нажать в строке меню вкладку <i>Вставка</i> => <i>Диаграмма</i>
3	При необходимости выбрать из предложенного меню параметры цвета, заливки, изменения границ объекта и т. п.	3	Выбрать из предложенных вариантов нужную диаграмму
		4	В созданной диаграмме изменить название и при необходимости отформатировать оси



Диаграммы используются для более явного сравнения или оценки объектов. Чтобы отформатировать ось, необходимо кликнуть на ней в области диаграммы и в появившемся окне настроить нужные параметры.

Использование систем распознавания текстов



Сканер (англ. *scanner*) — устройство, которое создаёт цифровое изображение сканируемого объекта. С его помощью можно получить изображение страницы текста в графическом файле. Однако полноценно работать с таким текстом-картинкой невозможно. Его можно читать, распечатывать, но нельзя редактировать и форматировать. Для того чтобы перевести графическое изображение в текст, необходимо произвести распознавание текста. Наиболее распространённым инструментом для распознавания текста является ABBY FineReader. Можно использовать также Adobe Acrobat DC, FreeOCR, gImageReader, ABBY Screenshot Reader, Free Online OCR.

Этапы проведения сканирования

Этапы проведения сканирования	
1	Сканирование документа и выбор области окончательного сканирования
2	Сегментация и распознавание текста. Прежде чем получить готовый текст, необходимо разбить фрагменты документа на блоки (текст, рисунок, таблица и т. д.) для корректного распознавания (преобразования в текстовый документ)
3	Проверка орфографии и передача текстового документа в нужное приложение для дальнейшей работы или сохранение в файл

Виды сканеров

Вид	Характеристика
Планшетные	Наиболее распространённые, обеспечивают максимальное удобство для пользователя — высокое качество и приемлемую скорость сканирования. Такой сканер представляет собой планшет, внутри которого под прозрачным стеклом расположен механизм сканирования

Барабанные	Барабанные сканеры применяются в полиграфии, имеют большое разрешение (около 10 000 точек на дюйм). Оригинал располагается на внутренней или внешней стенке прозрачного цилиндра (барабана)
Ручные	В данных сканерах отсутствует двигатель, следовательно, объект приходится сканировать вручную. Единственным его плюсом является дешевизна и мобильность, при этом он имеет массу недостатков — низкое разрешение, малая скорость работы, узкая полоса сканирования, возможны перекосы изображения, поскольку пользователю сложно перемещать сканер с постоянной скоростью
Сканеры штрихкода	Небольшие, компактные модели для сканирования штрихкодов товара в магазинах



Оптимальным разрешением для обычных текстов является 300 dpi (dots per inch — логическая единица измерения, т. е. количество точек на дюйм), для текстов, набранных мелким шрифтом (размером 9 пт и менее), — 400—600 dpi.

ABBYY FineReader



FineReader — омнифонтовая (распознающая символы разных размеров и начертаний) система оптического распознавания текстов. Данная программа позволяет распознавать тексты, набранные практически любыми шрифтами, без предварительного обучения. Особенностью программы FineReader является высокая точность распознавания и малая чувствительность к дефектам печати.

Распознавание текста и изображений

В процессе распознавания происходит обработка текста. Обнаруженные изображения программа выделяет красным цветом и в дальнейшем их не обрабатывает, а передаёт в том виде, как они получились при сканировании

Выделение ошибок и проверка текста

Все предполагаемые ошибки выделяются цветом. Затем программа производит проверку текста на наличие орфографических ошибок и все некорректные слова подчёркивает красными линиями

Если соперник не отступает, хозяин территории принимает более энергичные меры, танцуя вниз головой, бешено кусает ртом песок, сявно желая показать: «Если не уйдешь, я и тебя могу так отделать!» А если и это не устршило агрессора, тогда танцор разворачивается к нему широкой стороной тела и оттопыривает две большие брюшные иглы. Это угроза высшей степени, и она граничит с отчаянием. К ней прибегает колюшка и в критических случаях, обороняясь от жуки или окуня.

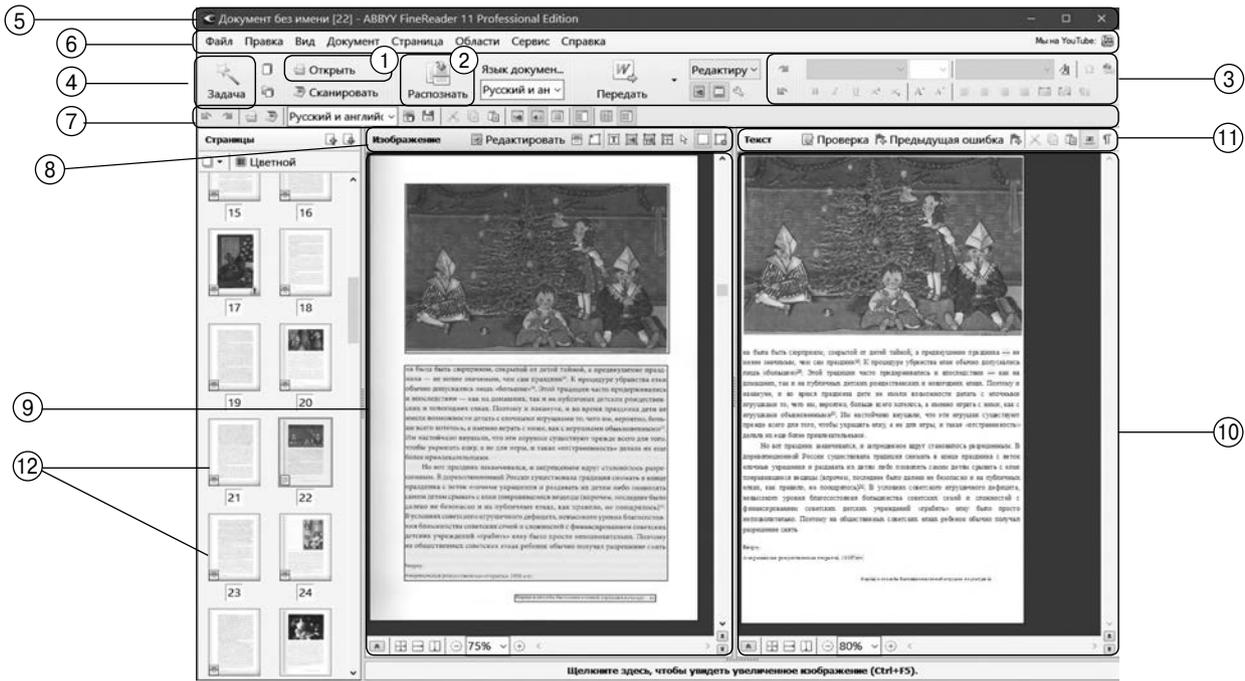
Вид текста после проверки орфографии

Редактирование полученного документа

Все изображения, относящиеся к данному документу, могут быть скопированы, вырезаны, вставлены или повернуты с помощью соответствующей панели редактирования

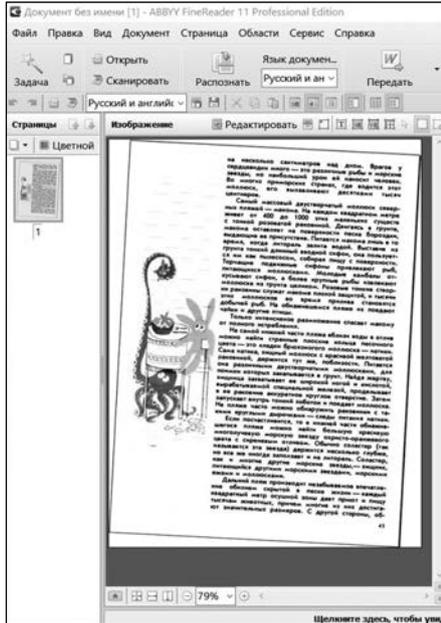
Если изображение во время сканирования было немного повернуто, этот недостаток устраняется с помощью функции *Исправить перекося*, которая активизируется автоматически при преданализе изображения

Функция *Фотокоррекция* позволяет исправить искажение строк, устранить размытие картинки, уменьшить шум

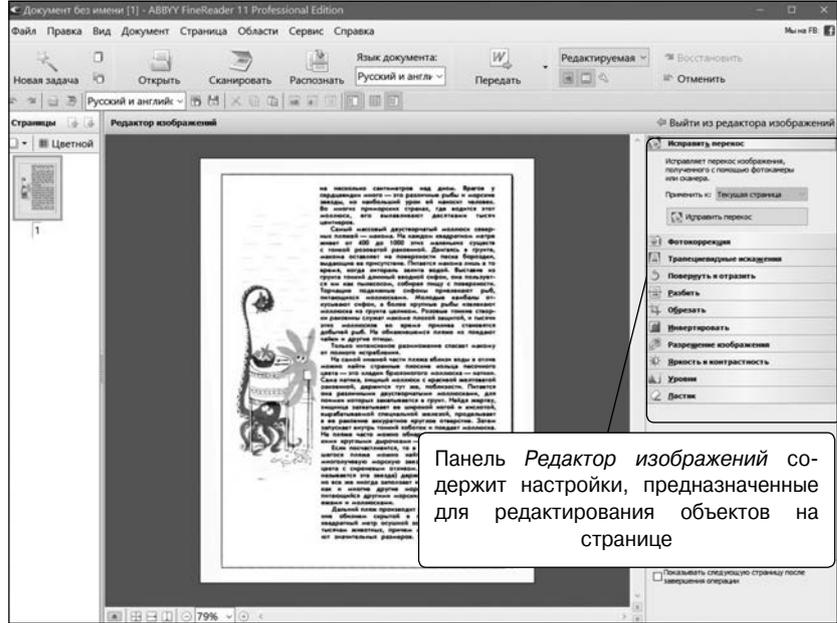


Интерфейс ABBY FineReader 11 Professional Edition:

- 1 — кнопка быстрого доступа к добавлению страниц;
- 2 — кнопка быстрого доступа к распознаванию;
- 3 — функциональные кнопки для редактирования документа;
- 4 — кнопка быстрого доступа к типовым сценариям;
- 5 — строка заголовка;
- 6 — строка меню;
- 7 — панель быстрого доступа;
- 8 — панель инструментов окна *Изображение*;
- 9 — окно *Изображение*;
- 10 — окно *Текст*;
- 11 — панель инструментов окна *Текст*;
- 12 — миниатюры страниц



Вид до активизации кнопки *Исправить перекося*



Вид после исправления перекося

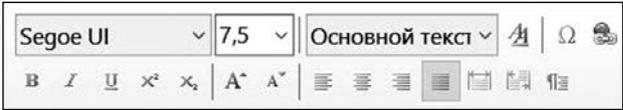
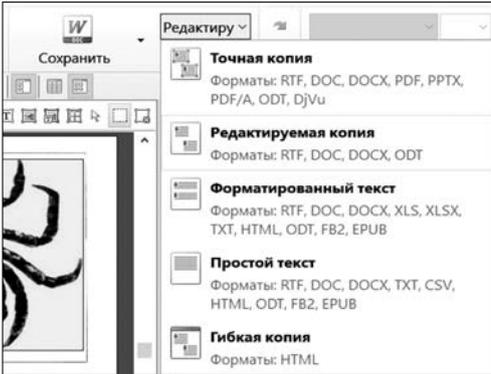


Программа также допускает устранение трапецевидных искажений, позволяет отражать и поворачивать картинку, автоматически разбивать изображения на части, удалять лишнее путём обрезки краёв, инвертировать цвет изображения, менять разрешение снимка и даже стирать часть изображения.



Иногда при сканировании штриховых рисунков, узоров, формул они ошибочно распознаются как текст. Для решения этой проблемы нужно использовать кнопку *Картинка* и отметить все формулы как изображения. После редактирования необходимо повторить анализ скана.

Редактирование ошибок в тексте

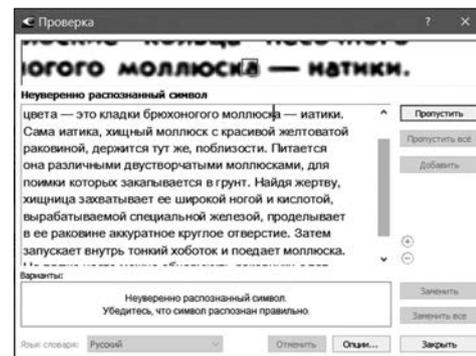
Характеристика	Наглядный пример
Редактирование всего текста осуществляется с помощью соответствующей панели	 <p>Панель редактирования текста</p>
Сначала необходимо выбрать режим редактирования (простой текст, точная копия, редактируемая копия, форматированный текст)	 <p>Выбор режима редактирования</p>

Пользователь может изменить шрифт текста, его размер, включить такие эффекты, как курсив, подчёркивание, выделение жирным шрифтом. Существуют возможности при необходимости выровнять текст (четыре варианта), изменить направление письма (справа налево), включить режим редактирования гиперссылок



Панель *Текст* позволяет выявлять и исправлять ошибки в распознанном тексте

Для редактирования ошибок в тексте используется кнопка *Проверка орфографии*. В случае необходимости можно воспользоваться функцией расстановки переносов (в ABBYY FineReader 11 эта функция не вошла)



Пример работы в режиме проверки орфографии



Структура документа изменяется в зависимости от применяемого режима работы. Пользователь может включить режим *Без картинок* и *Без колонтитулов*, что упростит работу с текстом.

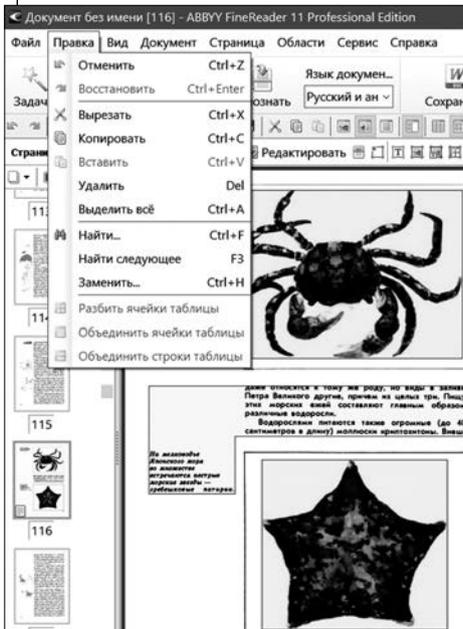
Работа с функционалом строки меню

Закладка	Функция
<i>Файл</i> (1, с. 102)	Позволяет производить действия с самим файлом: сохранять, печатать, передавать и т. д.
<i>Правка</i> (2, с. 102)	Помогает вставлять элементы, удалять, копировать, находить символы
<i>Вид</i> (3, с. 102)	Отвечает за внешний вид и расположение окон <i>Изображение</i> , <i>Текст</i> и <i>Крупный план</i> , позволяет выбрать и настроить панели инструментов
<i>Документ</i> (4, с. 103)	Позволяет запустить анализ и распознавание документа, перейти к предыдущей или следующей странице, а также к странице с указанным номером
<i>Страница</i> (5, с. 103)	Даёт возможность распознавать, анализировать страницу, редактировать отдельные области
<i>Сервис</i> (6, с. 103)	Позволяет использовать словари, редактор языков и проверку
<i>Области</i> (7, с. 104)	Помогает изменять тип области с изображения на таблицу, текст, картинку или штрихкод
<i>Справка</i> (8, с. 104)	Открывает справочные материалы

①



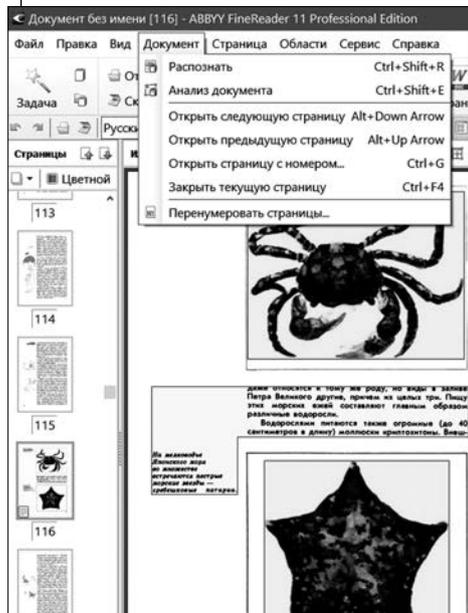
②



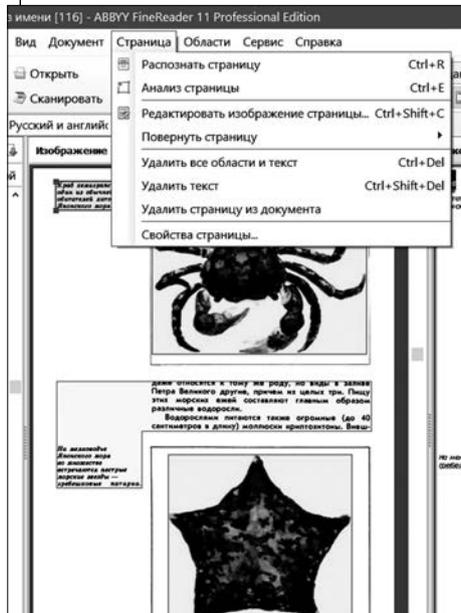
③



4

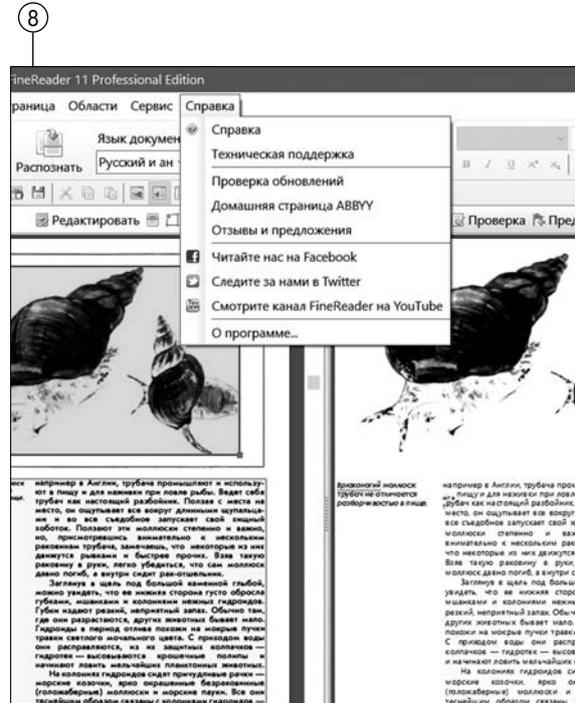
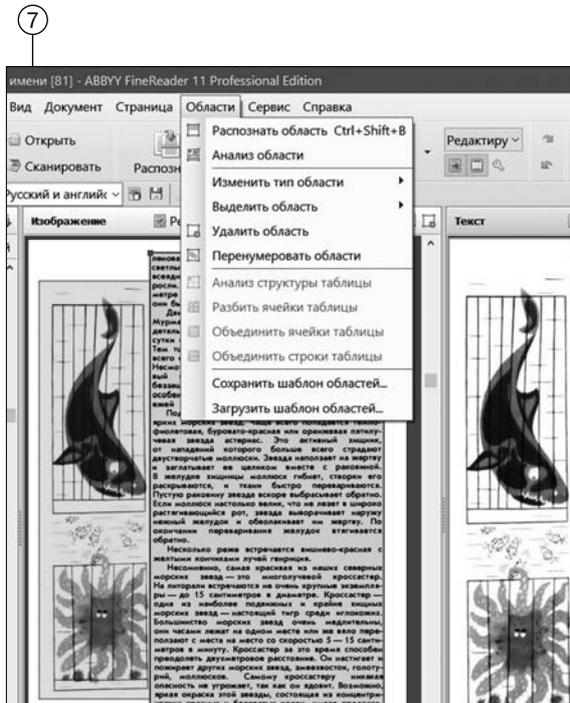


5



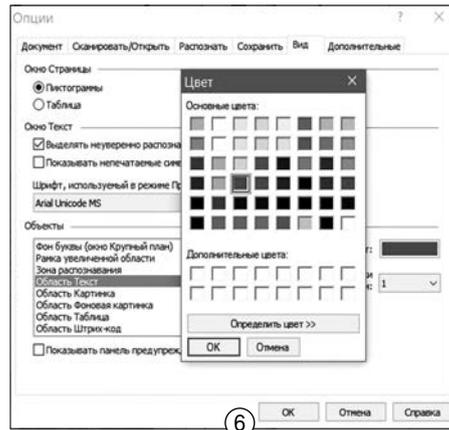
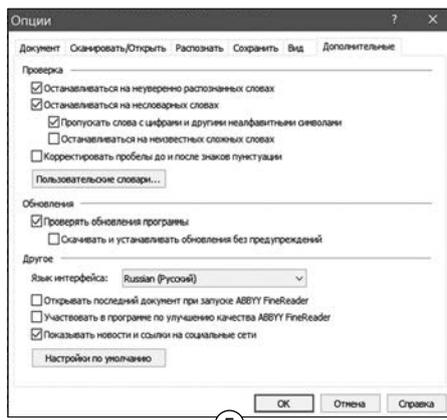
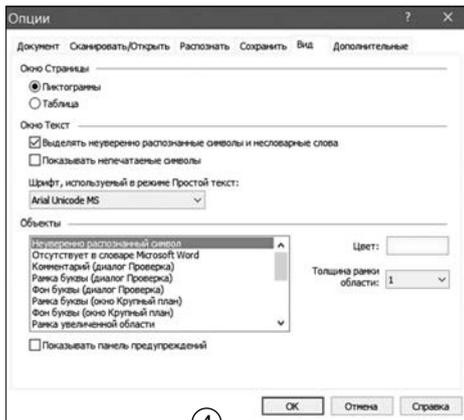
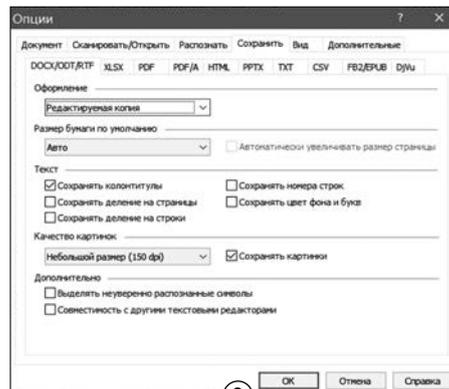
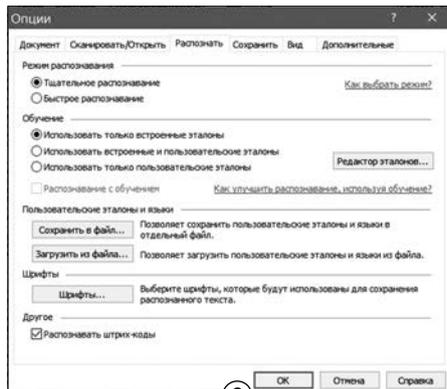
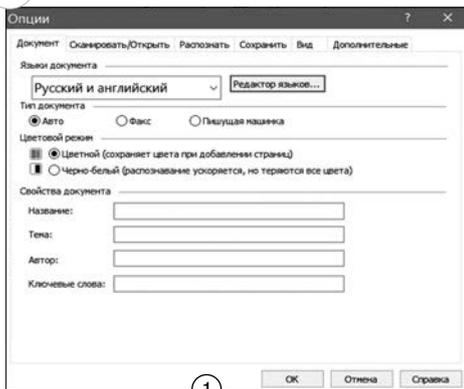
6





Раздел *Опции* в подменю *Сервис*

Вкладка	Функции
<i>Документ</i> (1, с. 106)	Применяется, чтобы определить язык распознаваемого документа, тип документа, цветность, задать атрибуты конечного файла (указать название, тему и автора документа, ввести ключевые слова)
<i>Распознать</i> (2, с. 106)	Определяет режим распознавания (быстрый или тщательный), а также позволяет применять пользовательский эталон для обучения программы
<i>Сохранить</i> (3, с. 106)	Позволяет выбрать режим сохранения в файл типа *.rtf, *.doc, *.docx, *.xls, *.html, *.pptx, *.csv, *.txt или *.pdf. Здесь же пользователь выбирает оформление документа (форматируемый текст, точная копия, простой текст) и размер бумаги. Допускается сохранение текста вместе с колонтитулами и делением на страницы и строки. Для каждого формата имеются небольшие отличия в настройках
<i>Вид</i> (4, 5, с. 106)	Позволяет представлять страницы в виде таблиц или пиктограмм. Здесь находятся настройки, согласно которым программа может выделять неуверенно распознанные символы и несловарные слова и задавать цвет выделения для каждого объекта на странице
<i>Дополнительные</i> (6, с. 106)	Содержит список шрифтов, которые используются для отображения распознанного текста. Здесь же можно настроить проверку орфографии. Программа может обрабатывать неуверенно распознанные слова или слова, не входящие в программные словари, также возможна автоматическая обработка сложных слов, цифр и других неалфавитных символов. Дополнительно программа может корректировать пробелы между словами. На этой же вкладке можно подключить пользовательские словари, указав папку, в которой они хранятся



Обработка документа

Характеристика	Пример
Работа с функционалом панели быстрого доступа	
<p>Панель быстрого доступа содержит набор самых востребованных команд, что позволяет отменить или вернуть совершённое действие, установить язык распознавания, открыть изображение для сканирования, сканировать, распознать или сохранить документ. Выбранную область можно вырезать, скопировать, вставить. Здесь же расположены кнопки настройки интерфейса программы</p>	 <p style="text-align: center;">Панель быстрого доступа</p>
Работа в ручном и автоматическом режиме расшифровки	
<p>Помимо редактирования формата отсканированной страницы, пользователь может самостоятельно выделять области с текстом, картинками и таблицы, а затем распознавать обработанную страницу. В некоторых условиях ручной режим определения типа блока может значительно повысить качество обрабатываемого документа</p>	<p>Для этого следует выделить нужную часть отсканированной страницы и выбрать требуемый тип блока на этой панели. После ручной обработки необходимого объёма материала нужно запустить распознавание. Программа допускает совместное использование автоматического и ручного определения типов блоков</p>
Сохранение файла	
<p>Обработанный документ может быть сохранён в формате Word, Excel или Acrobat Reader</p>	 <p style="text-align: center;">Необходимый формат выбирается на панели сохранения</p>

ТЕХНОЛОГИЯ СОЗДАНИЯ И ОБРАБОТКИ МУЛЬТИМЕДИЙНОЙ И ГРАФИЧЕСКОЙ ИНФОРМАЦИИ



Мультимедийная информация — содержание или данные, которые представляются одновременно в разных формах: звук, анимированная компьютерная графика, видеоряд.



Мультимедиа — комплекс аппаратных и программных средств, позволяющих пользователю работать в диалоговом режиме с разнородными данными (графикой, текстом, звуком, видео и анимацией), организованными в виде единой информационной среды, создавая новые объекты.

Форматы графических объектов

Определение	Характеристика
Компьютерная графика	Технология создания и обработки графических изображений с помощью аппаратных и программных средств компьютера. Компьютерная графика бывает двухмерной (2D; от англ. <i>two dimensions</i> — «два измерения») и трёхмерной (3D; от англ. <i>three dimensions</i> — «три измерения»)
Графический объект	Любая видимая глазом форма, которую сознание человека выделяет среди прочих форм как самостоятельную сущность

Цифровой графический объект	Любой нарисованный объект, который можно редактировать и форматировать с помощью панели инструментов рисования встроенным редактором графических объектов
Графический редактор	Программа, предназначенная для создания, редактирования и просмотра графических изображений. В современных компьютерах существует два принципиально различных способа хранения изображений: растровый и векторный . Соответственно, и графические редакторы можно разделить на две категории: растровые и векторные



Графическая информация — информация, представленная в виде изображений. Изображения могут быть неподвижными (рисунки, фотографии, слайды) и подвижными (анимация, видео). Графическую информацию, как и текстовую, можно создавать и редактировать на компьютере.

Категории графических редакторов

Критерии сравнения	Растровая графика	Векторная графика
Элементарный объект	Пиксель	Линия
Представление объектов реального мира	Возможно	Невозможно
Качество при редактировании	Возникают искажения при редактировании	Без потери качества при редактировании
Размер файла	Большой	Небольшой

Критерии сравнения	Растровая графика	Векторная графика
Фотографическое качество	Да	Нет
Распечатка на принтере	Без искажений	Возможны искажения
Форматы	.bmp, .jpeg, .gif, .png	.svg, .eps, .cgm, .wmf
Группировка и разгруппировка	Нет	Да
Обеспечение градации цветов и полутонов	Да	Нет
Примеры графических редакторов	Corel Photo-Paint, Adobe Photoshop, PictureMan Painter, PhotoFinish, Paint	CorelDRAW, Adobe Illustrator, MS Draw

Форматы растровых изображений

Формат	Характеристика
BMP	Формат, который позволяет сохранить изображения точно, без сжатия. Изображение состоит из массы точек, каждая из которых содержит свой цвет. Файлы формата BMP имеют очень большой размер
JPEG	Наиболее широко используемый формат, применяемый в основном для хранения фотографических изображений. В работе с форматом JPEG используются сложные алгоритмы для сжатия изображений, при котором возможна потеря данных и ухудшение их качества
GIF	Данный формат объединяет несколько изображений, которые выводятся на экран в определённой последовательности и в итоге образуют анимированный элемент. Не имеет звукового сопровождения, прост в создании и поддерживается абсолютно всеми браузерами. Формат GIF позволяет передавать не более 256 цветов одновременно

TIFF	Сохранение в формате TIFF позволяет максимально точно передать подготовленный проект. При этом не происходит потерь, в то же время размер изображения достаточно сжимается. Данный формат используется при сканировании, распознавании текста, в полиграфии, широко поддерживается графическими приложениями
PNG	Формат содержит 8-битную цветовую палитру, использует сжатие без потерь и может служить заменой формата изображения GIF. Однако, в отличие от файлов формата GIF, файлы PNG не поддерживают анимацию

Форматы векторных изображений

Формат	Характеристика
EPS	Данный формат удобно использовать для печати больших объёмов графической информации. Обеспечивает сохранение качества различных векторных изображений при их редактировании
CGM	Международный стандарт для хранения и обмена графическими данными (векторной и растровой 2D-графики и текста). Формат широко используется в конструкторских, авиационных и других технических областях
SVG	Формат предусматривает работу с двумерной векторной графикой. Он подходит как для создания веб-иллюстраций, так и для редактирования файлов, поскольку даже в случае масштабирования графика, сохранённая в данном формате, отличается высоким качеством. Файлы формата SVG могут содержать не только изображения и текст, но и анимацию
PDF	Формат предназначен для оптимизации хранения и отображения полиграфической продукции и текстовых документов. Основное преимущество данного формата — его универсальность: он позволяет сохранить шрифты, изображения и макет исходного документа независимо от того, на какой платформе он создан

Создание, ввод и обработка графических объектов



Получение графических объектов с помощью копирования готовых изображений с цифровой камеры, устройств внешней памяти или скачивания из Интернета является наиболее доступным и распространённым способом.

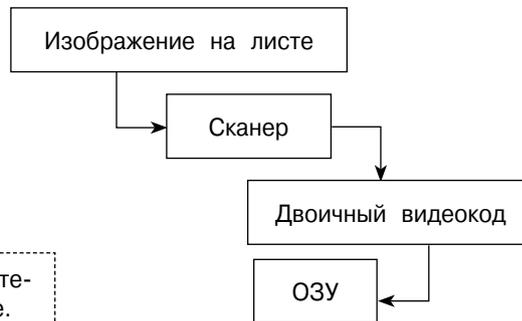
Способы получения цифровых графических объектов	Основные способы обработки изображения
Копирование готовых изображений с цифровой фотокамеры, из Интернета	Изменение масштаба изображения
Создание новых изображений с помощью программного обеспечения	Изменение размера с помощью обрезки
Ввод графических изображений с помощью сканера	Коррекция изображения: <ul style="list-style-type: none"> • изменение контраста, яркости, насыщенности; • задание параметров резкости или размытия



Для создания графического файла с помощью программного обеспечения нужно запустить любой имеющийся графический редактор и нажать в строке меню *Файл => Создать* (или *Новый*). Затем выбрать в появившемся окне параметры изображения. С этой целью можно использовать такие графические редакторы, как Microsoft Paint, GIMP, Adobe Photoshop, OpenOffice.org, CorelDRAW и др.



Ввод графических изображений, существующих на бумажных носителях, с помощью сканера происходит в указанном справа порядке.





Основная функция сканеров — получение изображений различных бумажных носителей (книг, журналов, фотографий, открыток, рисунков, слайдов и т. д.) для последующей обработки, хранения и распространения в цифровом формате.

Этапы работы с файлом в графическом редакторе

1	Открыть графический редактор
2	Найти вкладку <i>Файл</i> => <i>Открыть</i> . В появившемся окне выбрать нужный файл
3	Произвести необходимые изменения в файле
4	Сохранить в требуемом формате



Графическая информация обрабатывается с помощью монитора, видеокарты, планшета, камеры и других устройств. При необходимости графические объекты можно вывести на бумагу или другой носитель (плёнку, ткань и т. д.).



Информационные технологии обработки графической информации включают специфические модели представления информации данного вида, особые методы ввода, формирования и вывода изображений, свои аппаратные и программные средства.

Звуковые файлы



Звук — продольная волна сжатия/расширения, распространяющаяся в воздушной, водной или другой упругой среде. Звуковые волны представляют собой механические колебания, взаимодействующие с органами слуха.



Звук характеризуется **интенсивностью**, которую человек воспринимает как громкость, и **частотой**, воспринимаемой как тон. Диапазон частот звука, воспринимаемых человеком, очень велик (от 20 до 20 000 Гц). Интенсивность (мощность) звука измеряется в децибелах (дБ). Например, уровень шума движущегося поезда равен 90 дБ, работающего перфоратора — 100 дБ, шелеста листьев — 10 дБ. Разность между максимальной и минимальной интенсивностью звука называется **динамическим диапазоном**, для человеческого уха она составляет 140 дБ.

Звуки	Естественные	Речь, шум моря, свист ветра, звук скрипки
	Воспроизводимые электроакустическими устройствами	Радиопередача, звук из усилителя



В идеале звуки, воспроизводимые электроакустическими устройствами, должны в точности повторять естественные звуки, но на практике их точное соответствие часто невозможно или трудноосуществимо.

Преобразование аналогового звукового сигнала в цифровой

1	Аналоговый звуковой сигнал подаётся на аналоговый фильтр, который устраняет помехи и шумы
2	Из аналогового сигнала с помощью схемы выборки/хранения выделяются отсчёты
3	Отсчёты поступают в аналого-цифровой преобразователь (АЦП), который преобразует мгновенное значение каждого отсчёта в цифровой код
4	Полученная последовательность битов цифрового кода является звуковым сигналом в цифровой форме

Форматы звуковых файлов

Без потери качества (lossless-формат)	С потерей качества (lossy-формат)
К lossless-форматам относятся WAV, APE, FLAC. В них сжатие не применяется совсем или применяется так называемое обратимое сжатие с помощью механизма архивации	В lossy-форматах MP3, OGG, WMA и др., кроме архивации, применяется психоакустическое кодирование (например, маскировка более слабого сигнала более сильным), учитывающее физиологические свойства человеческого слуха. При этом ради достижения большего сжатия из сигнала выбрасывается часть информации, и его полное восстановление становится невозможным

Запись и воспроизведение звука



Для преобразования звуковых колебаний в электрические используются микрофоны. Чтобы микрофон можно было подключить к компьютеру, он должен быть оборудован **звуковой картой**. В современных устройствах звуковые карты интегрированы в материнские платы, однако лучшее качество обеспечивают дискретные и внешние карты.

Компьютерная запись звука		Создание нового файла в звуковом редакторе	
1	Подключить источник сигнала к порту звуковой карты компьютера	1	Открыть звуковой редактор
2	Настроить звуковую карту — выбрать источник и уровень сигнала. Для этого может быть использована стандартная программа <i>Регулятор громкости</i> , входящая в состав ОС Windows	2	Выбрать команду <i>File => New (Файл => Создать/Новый)</i>
3	При выборе команды <i>Параметры => Свойства</i> откроется окно <i>Свойства</i> , в котором нужно выбрать переключатель <i>Запись</i> , указать нужный вход (Microphone или Line-in) и установить необходимый уровень записи	3	Установить частоту дискретизации и глубину кодирования звукового сигнала, а также режим записи (стерео или моно). Для музыкального сигнала лучше использовать параметры компакт-диска (44 100 Гц, 16 бит, 2 канала); если требования к качеству звука выше, можно поднять частоту дискретизации до 48 000 Гц
4	Далее можно использовать любой звуковой редактор	4	Для прослушивания звукового файла подключить к линейному выходу звуковой карты колонки или наушники



Задача звуковой карты — производить с определённой частотой измерения уровня звукового сигнала (преобразованного в электрические колебания) и результаты измерений записывать в память компьютера. Этот процесс называют **оцифровкой звука**.

Программы, работающие с аудиофайлами

Программа	Функция	Примеры
Аудиоконвертеры	Перевод аудиофайлов из одного формата в другой	FormatFactory, Free Audio Converter, Mobile Media Converter
Аудиоплагины	Расширение возможностей проигрывателей и редакторов	Vocoder, Stereo Tool, WhiteCap
Микшеры, синтезаторы	Самостоятельное создание музыки, а также редактирование, микширование и запись аудиофайлов	Adobe Audition CC, Virtual DJ Studio, PianoFX STUDIO
Программы для записи звука	Запись аудиопотока как с микрофона, так и со звуковой карты компьютера	NanoStudio, Free Audio Recorder, Echo, Pro Tools
Аудиоредакторы	Редактирование аудиофайлов любых форматов	Audacity, MP3Gain, Sibelius, MAGIX Music Maker, Adobe Audition CC
Аудиоплееры и аудиопроигрыватели	Воспроизведение музыкальных файлов	Winamp, iTunes, Winyl Player, AIMP
Аудиокодеки	Воспроизведение и редактирование звуковых файлов	Windows Essentials Codec Pack, FLAC, LameXP, Lame MP3 Encoder

ТЕХНОЛОГИЯ ОБРАБОТКИ ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ



Технология обработки информации в электронных таблицах — процесс, использующий методы обработки информации в таблицах для расчётов, решения логических задач, исследования информационных моделей и др.

Табличный процессор



Табличный процессор — прикладная программа, предназначенная для обработки данных, представленных в таблицах.

Предназначение	Функции	Современные табличные процессоры
Автоматизация расчётов	Ввод данных в ячейки электронных таблиц, их редактирование и форматирование	Excel (англ. <i>excel</i> — «преобладать, превосходить») входит в пакет программ Microsoft Office, один из самых популярных в настоящее время
Моделирование различных ситуаций	Построение диаграмм и графиков на основе данных, которые содержатся в ячейках электронных таблиц	Calc (англ. <i>calculator</i> — «вычислитель, калькулятор») входит в пакет программ StarOffice

Отсеивание изменений получаемых результатов	Поиск данных в соответствии с указанными условиями	GNumeric (англ. <i>GNU</i> — проект по созданию программного обеспечения свободного распространения, <i>numeric</i> — «числовой») находится в свободном распространении
Выбор наиболее подходящего варианта из множества решений задачи	Вычисление новых результатов при изменении данных или формул для их обработки	
Удобная реализация для проведения бухгалтерских и статистических расчётов	Обработка числовых данных по формулам, которые могут содержать арифметические выражения, математические, логические, статистические и другие функции	Numbers (англ. <i>numbers</i> — «числа») разработан Apple Inc. как часть пакета iWork
Выполнение расчётов прочно-сти конструкции	Отбор данных, которые удовлетворяют определённым условиям (сортировка данных)	
Ведение учёта товаров в магазине	Печать данных ЭТ, диаграмм и графиков	

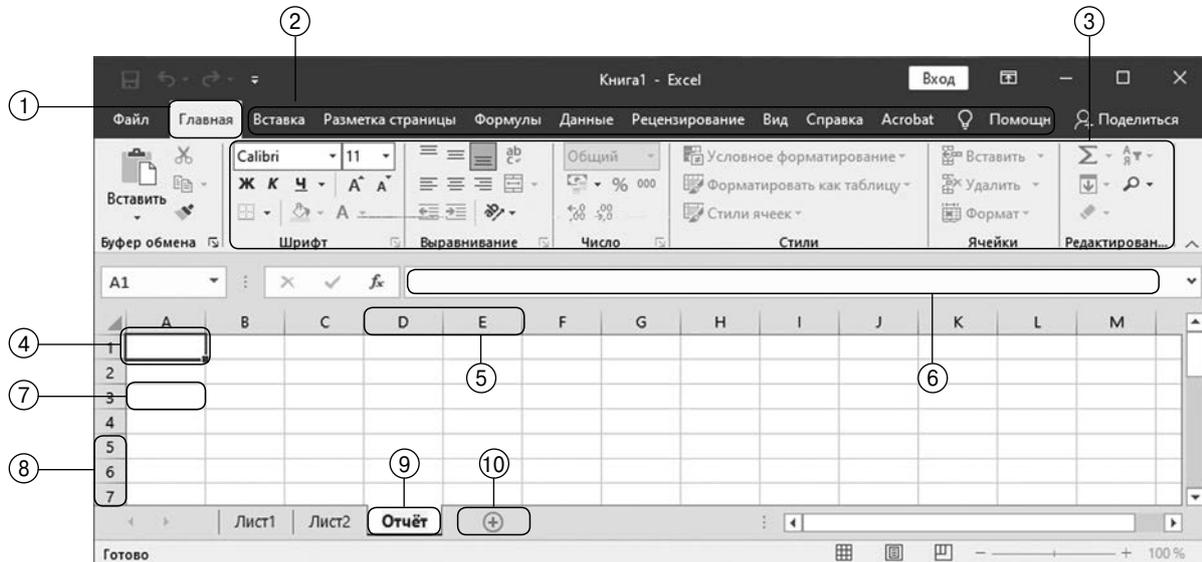


Таблицы, в которых представлены данные для обработки табличным процессором, называются **электронными таблицами** (ЭТ).



В табличных процессорах возможно создание гипертекстовых ссылок с целью перехода в другие файлы, хранящиеся на компьютере, в локальной сети или сети Интернет. Данные и диаграммы можно сохранять как отдельную веб-страницу.

Объекты табличного процессора Excel и их свойства



Интерфейс Microsoft Excel:

- 1 — активная вкладка главного меню; 2 — вкладки главного меню; 3 — лента кнопок;
 4 — активная ячейка; 5 — имена столбцов; 6 — строка для формул; 7 — ячейка; 8 — имена строк;
 9 — имя листа; 10 — добавление листа

Основные объекты

Объект	Характеристика
Электронная книга	Документ приложения Excel, который состоит из листов (worksheet), которые представляют собой большие таблицы ячеек с числами и текстовой информацией
Лист	Каждый лист книги имеет имя, которое указывается на его ярлычке. Если на листах размещаются электронные таблицы, то такие листы по умолчанию имеют имена Лист1, Лист2 и т. д. Эти листы можно переименовать, кликнув на листе дважды. Если листы предназначены для размещения только диаграмм, они по умолчанию имеют имена Диаграмма1, Диаграмма2 и т. д.
Электронная таблица	Специальная модель структурирования, предоставления и обработки числовой или текстовой информации, тесно связанная с текстовыми документами и с базами данных
Строка и столбец	Строки таблиц нумеруются цифрами, а столбцы — буквами латинского алфавита. В таблице содержится 1 048 576 строк (номера от 1 до 1 048 576), 16 384 столбца (A, B, C...Z, AA, AB...ZZ, AAA, AAB...XFD). Перемножив количество строк на количество столбцов, получим количество ячеек — 17 179 869 184
Ячейка	Адрес ячейки задаётся номерами столбца и строки, на пересечении которых она находится, например A5, C8, D16. Всегда одна из ячеек электронной таблицы текущая, она выделяется табличным курсором в виде чёрной рамки. Номер строки и номер столбца текущей ячейки выделяются другим цветом фона. Выделение двух ячеек и более в электронной таблице называется диапазоном ячеек . В диапазон могут входить как смежные, так и несмежные ячейки. Адрес связного диапазона задаётся адресами двух ячеек, расположенных в противоположных углах диапазона. При этом адреса ячеек разделены двоеточием

Ячейка или диапазон	Использование
Ячейка на пересечении столбца В и строки 12	B12
Диапазон ячеек: столбец А, строки 12—22	A12:A22
Диапазон ячеек: строка 15, столбцы В—Е	B15:E15
Все ячейки в строке 5	5:5
Все ячейки в строках 2—10	2:10
Все ячейки в столбце В	B:B
Все ячейки в столбцах с Н по Р	H:P
Диапазон ячеек: столбцы А—Х, строки 10—20	A10:X20



Для выделения целого столбца следует щёлкнуть левой кнопкой мыши по его названию (латинской букве), для выделения строки — по названию строки (числу). Строка и столбец также являются диапазонами ячеек. Адресом диапазона ячеек, в который входят все ячейки десятой строки, является 10:10, а адресом диапазона ячеек, в который входят все ячейки столбца В, — В:В. Чтобы указать адрес несвязного диапазона ячеек, нужно указать адреса его связанных частей, разделив их точкой с запятой.

Работа с файлами в MS Excel



Как и в любом другом процессоре, в MS Excel можно выполнять различные операции с файлами: создавать, сохранять, форматировать, переименовывать и пр.

Создание электронной книги	
1	Найти на рабочем столе ярлык программы или найти программу с помощью меню <i>Пуск</i>
2	Один или два раза (в зависимости от настроек) кликнуть по значку программы <i>Excel</i>
3	На панели инструментов найти кнопку меню <i>Файл => Создать</i>

Сохранение файла	
1	На панели инструментов найти кнопку меню <i>Файл => Сохранить</i>
2	В появившемся окне указать имя файла, тип файла, выбрать диск и папку, в которой будет храниться рабочая книга, и нажать кнопку <i>Сохранить</i>
3	При сохранении файла под другим именем использовать команду <i>Файл => Сохранить как</i>
4	Для форматирования документа открыть файл и произвести изменения, затем сохранить их



В каждой ячейке можно изменить размер, тип и цвет шрифта, величину ячеек, заливку, границы, формат, выравнивание текста.

Формулы в MS Excel



Работа с формулами относится к важнейшим функциям процессора Excel и обеспечивает почти бесконечное поле его применения. Формула предписывает программе Excel порядок действий с числами, значениями в ячейке или группе ячеек. Частями формул в Excel могут выступать знаки математических и логических операций, ссылки на другие ячейки, функции, числа или текст.

Этапы введения формулы	
1	Поставить курсор в нужную ячейку
2	Нажать знак «=»
3	Ввести нужную формулу
4	Нажать <i>Enter</i> . В ячейке появится результат вычислений

Некоторые из основных операций программы MS Excel

Оператор	Операция	Пример
+ (плюс)	Сложение	B4+7
- (минус)	Вычитание	A9-100
* (звёздочка)	Умножение	A3*2
/ (наклонная черта)	Деление	A7/A8
^ (циркумфлекс)	Степень	6^2
СРЗНАЧ (функция)	Среднее значение выбранных элементов	СРЗНАЧ(A2:A5)
СУММ (функция)	Сумма выбранных элементов	СУММ(A2:A10)
ДНИ (функция)	Находит количество дней между двумя датами	ДНИ(B5;B23) или ДНИ("15.03.11","1.02.11")

ТЕХНОЛОГИИ ПОИСКА И ХРАНЕНИЯ ИНФОРМАЦИИ



В современном мире компьютеры являются основными помощниками человека в поиске, хранении, использовании текстовой, графической, видео- и аудиоинформации. Это стало возможным благодаря хранилищу, которое называется банком данных.

Составляющие банка данных



Банк данных — разновидность информационной системы, предназначенная для хранения, обработки и применения информации, представленной в одной или нескольких базах данных.

Составляющая	Характеристика
База данных	Ядро банка данных, структура, позволяющая хранить информацию об объектах в строго упорядоченном виде
Системы управления базами данных	Создание баз данных, а также операции поиска и сортировки данных при решении различных задач выполняются специальными программами — системами управления базами данных (СУБД). Необходимо различать собственно базы данных, которые являются упорядоченными наборами данных, и системы управления базами данных — приложения, управляющие хранением и обработкой данных

Составляющая	Характеристика
Приложения	Программы, разработанные для решения конкретной прикладной задачи. Приложения создаются в среде СУБД или с помощью внешних средств (языков программирования)
Компьютерная система	Вычислительная система (технические средства и операционная система)
Словарь данных	Содержит всю основную информацию: <ul style="list-style-type: none">• связи между файлами;• модель представления данных;• информация о пользователях (о разграничении доступа, о парольной защите). Иногда роль словаря данных берёт на себя СУБД
Администратор банка данных	Обеспечивает выполнение следующих функций: <ul style="list-style-type: none">• проектирование банка данных;• эффективная работа с базами данных;• безопасность;• продвижение банка данных;• руководство техническим персоналом
Обслуживающий персонал	К обслуживающему персоналу относятся все, кто подчиняется администратору: <ul style="list-style-type: none">• сетевой администратор;• программисты;• статисты;• технический персонал

Типы моделей баз данных

Преимущества

Недостатки

Реляционная модель

Реляционную модель представления данных часто называют табличной, информация размещается в ячейках.

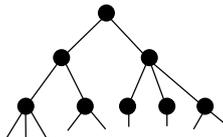
✓ MS Access, FoxPro, Paradox, dBase

- Простота организации.
- Удобство.
- Наглядность.
- Высокая скорость обработки.
- Простота реализации

- Строгость записи.
- Наличие разных ограничений.
- Сложность представления иерархических и сетевых данных.
- Возможность избыточных данных

Иерархическая модель

Иерархическая модель является одной из самых старых. В ней связи между объектами можно представить в виде упорядоченного графа базы данных. Характеризуется определённым порядком подачи данных — одни элементы считаются главными, другие им подчиняются.



✓ IMS, PC/Focus

Преимущества

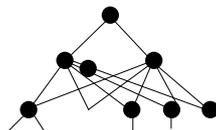
- Наглядность структуры.
- Эффективное использование памяти.
- Небольшое время обработки данных

Недостатки

Сложность хранения информации с большим количеством связей

Сетевая модель

Связи между объектами можно представить в виде произвольного графа. Эта модель, по сравнению с иерархической, более гибкая, поскольку в ней присутствуют не только вертикальные, но и горизонтальные связи.



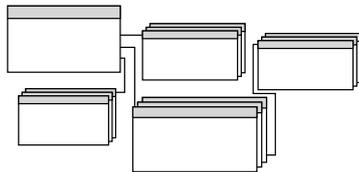
✓ IDS, IDS-2, DMS-1100

- Учёт разнообразия связей между объектами.
- Небольшие затраты памяти

- Сложность самой схемы.
- Сложность контроля целостности и непротиворечивости.
- Сложность обработки

Объектно-ориентированная модель

Модель, в которой взаимодействия между объектами описываются на языке механизмов объектно-ориентированного программирования.



✓ MongoDB, Jasmin, Orion

Простота представления объектов со сложными связями

- Большая сложность понятий.
- Невысокая скорость обработки

Многомерная модель

Данная модель возникла параллельно с реляционной, поскольку они похожи по своей структуре. Её можно представить как трёхмерную модель в виде куба.

	Май	Июнь	Июль
Рюмин	115 000	130 000	140 000
Орлов	120 000	145 000	160 000

✓ InterSystems Caché, OpenQM, Rocket U2

- Компактное размещение на диске.
- Небольшое время обработки информации

- Сложность реализации.
- Ограниченный круг пользователей (в основном программисты)

Табличные базы данных



Самой простой базой данных является **реляционная (табличная)**. Вся информация в ней находится в ячейках таблиц, главным достоинством является простота использования. Именно такую базу данных изучают на уроках информатики и к ней относят MS Access.

Свойства двумерных таблиц

В каждой ячейке содержится один элемент данных

В каждый столбец можно вписать элемент одного типа

Каждое имя столбца уникально

Не имеется одинаковых строк

Не существует определённого порядка строк и столбцов

Свойства поля	Описание
Размер	Выражается в знаках (или символах)
Имя	Должно быть уникальным для каждого поля
Подпись	Отображается в заголовке столбца
Формат	Устанавливает формат данных



Таблица — это данные определённого типа. В одной таблице могут быть данные только из какой-то конкретной области. Все эти данные расположены в полях (столбцах) и записях (строках).



Поле — место для основных элементов в базе данных. От выбора типа поля в настройках зависит тип элементов, который будет в нём расположен. В поле задаётся информация об операциях, которые можно производить над размещёнными в нём элементами.

Типы данных, находящихся в поле

Тип	Характеристика
Текстовый	В поле может размещаться текст размером до 255 символов
Числовой	В качестве данных могут быть числа
Логический	Данными могут выступать два значения: да (1), нет (0)
Дата и время	В качестве данных могут быть часы, минуты, секунды или день, месяц, год
Денежный	Записываются числа в форме денежных знаков
Поле объектов OLE	В таких полях могут находиться медиафайлы
Счётчик	Для данных, которые могут изменять свои значения в результате наращивания
Поле MEMO	В таком поле может быть текст размером до 65 535 символов

Использование инструментов поисковых систем (формирование запросов)

Определение	Характеристика
Фильтр	Отбор информации, необходимой пользователю. Можно задать сложные условия отбора
Сортировка	Процесс упорядочения информации по некоторому признаку. Различают возрастающую и убывающую сортировку. Если база данных имеет числовую информацию, в качестве сортировки используют ранжирование, в случае работы с текстовой информацией пользуются сортировкой по алфавиту. В MS Access системы управления базами данных осуществляют сортировку в пределах одного поля. Если используется новая сортировка, то предыдущие её результаты теряются. Если сортировка вложенная, то выполнить её можно с помощью запроса

Определение	Характеристика
Поиск	Для поиска данных используются специальные символы. Последовательность допустимых символов, применяемых при поиске файлов по имени, называется маской файла . Фильтр по знакам можно использовать для всей таблицы либо для части данных, отобранных ранее по некоторым критериям
Запрос	Масштабная функция, которая может заключать в себе как сортировку, так и фильтрацию. Данная функция позволяет производить отбор данных в нескольких полях и таблицах. Можно создавать библиотеку запросов, которая сохраняется для будущего использования. Существует специальный язык запросов SQL

ТЕЛЕКОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ



На разных этапах развития общества применялись новые методы, средства и технологии передачи информации в телекоммуникационных системах. Телекоммуникационные сети (радио-, телефонные, телевизионные и вычислительные) открывают всё более широкие возможности для передачи данных, голоса и изображения.



Телекоммуникационные технологии — совокупность алгоритмов, методов и средств передачи информации.



Можно выделить следующие виды телекоммуникационных технологий:

- телефонная связь;
- телеграфная связь;
- радиосвязь;
- спутниковая связь;
- компьютерные сети.

Использование средств телекоммуникационных технологий

Все сферы коммерческой и производственной деятельности

Осуществление контроля работы предприятий

Ведение отчётности в учреждениях

Дистанционное обучение

Личные цели и др.

Программное обеспечение средств телекоммуникационных технологий



Компьютерная сеть — система, обеспечивающая обмен данными между устройствами — компьютерами, серверами, маршрутизаторами и другим оборудованием или программным обеспечением.

Локальная сеть (Local Area Network, или LAN)

Замкнутая система, способная объединить компьютеры, находящиеся на небольшом расстоянии друг от друга. Локальная сеть отдельной организации (юридического лица) независимо от занимаемой ею территории называется **корпоративной сетью**, её используют для ограниченного доступа к определённой информации

Глобальная сеть (Wide Area Network, или WAN)

Охватывает большие географические регионы и состоит из множества локальных сетей. Серверы глобальной сети способны передавать и использовать информацию на небольших расстояниях за короткое время. Интернет — глобальная компьютерная сеть. Каждый компьютер в сети имеет адреса трёх уровней

Уровни компьютерных адресов в сети

Уровни		Характеристика
1	MAC-адрес	Назначается производителем оборудования и является уникальным. Связь между IP-адресами и доменными именами осуществляет международная база данных, основанная на использовании DNS-серверов (Domain Name System, или DNS, — доменная система имён)
2	IP-адрес (IPv6, IPv4)	IPv4 состоит из четырёх десятичных чисел со значением от 0 до 255, разделённых точками, а IPv6 — из 32 шестнадцатиричных чисел: восемь секций по четыре цифры в каждой, разделённых двоеточиями. ✓ IPv4: 198.168.70.36, IPv6: 1234:5828:34ab:abfd:1844:5A78:90ab:cebf
3	Адрес домена	Состоит из имён и точек. Такой адрес указывает на название страны и организации, которая поставляет сеть. Домен может быть совсем простым и состоять из доменов первого и второго уровня, а может включать три или четыре уровня. Иногда домены первого уровня называют доменами верхнего уровня



Домены верхнего или первого уровня подразделяются на два вида:

- национальные домены (каждой стране выделено двухбуквенное обозначение):
✓ «RU» — Россия, «RF» — Российская Федерация, «UA» — Украина, «DE» — Германия;
- общие домены:
✓ «COM» — коммерческие организации, «NET» — предприятия, связанные с развитием сетей и телекоммуникационных технологий, «ORG» — некоммерческие организации.



Сеть Интернет состоит из компьютерных узлов и каналов связи. К узлам связи подключаются персональные компьютеры, а услуги абонентам предоставляют провайдеры. Все объекты в сети соединены каналами связи. В качестве каналов связи могут использоваться радиоволны, оптоволокно, электрические, а также телефонные кабели.

Характеристики каналов связи

Пропускная способность, т. е. скорость обмена данными

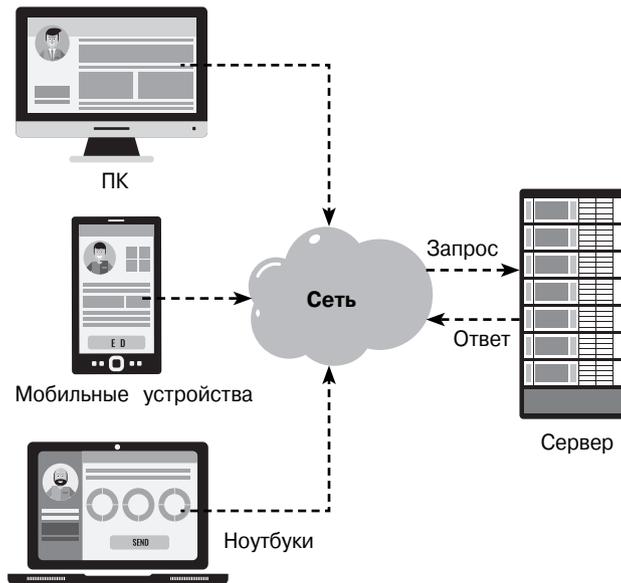
Надёжность — способность канала связи передавать информацию без искажений

Стоимость

Возможность усовершенствования



Сеть функционирует благодаря **сетевым операционным системам**, отвечающим за приём и передачу информации по сети, и **сетевым приложениям**, к которым можно отнести браузер.



Пример клиент-серверной архитектуры

Определение	Характеристика
Линия связи	Приспособление, используемое для объединения в сеть отдельных компьютеров
Сервер	Компьютер, ресурсами которого пользуются другие компьютеры в сети
Рабочая станция	Компьютер, который пользуется сетью
Браузер	Компонента, необходимая для отображения страницы в Интернете

Инструменты создания информационных объектов для Интернета

Инструмент	Характеристика
Информационный объект	Объект предметной области в информационной системе, определяющий её структуру, атрибуты, целостность данных и поведение
Простые информационные объекты	Звук, изображение, текст, число
Комплексные информационные объекты	Элемент, база данных, таблица, гипертекст, гипермедиа



Информационные объекты отображаются на веб-страницах.

Определение	Характеристика
Веб-страница	Документ, созданный в формате html, который может включать в себя видео, фото, текст, графику, гиперссылки
Гиперссылка (ссылка)	Текст, фотография, иконка или другой элемент, при нажатии на который открывается другая веб-страница
Веб-сайт	Несколько страниц с общими интерфейсом, оформлением, гиперссылками, а также расположением на одном веб-сервере



Веб-разработка — динамичный процесс создания сайтов, который позволяет вносить на них изменения.



Чтобы изменить, добавить информацию на сайте, используют специальные инструменты, относящиеся к системам управления содержимым сайта. Изменение настроек в таких системах приводит к автоматическому изменению HTML-кода.



Для разработки качественного сайта необходимы услуги веб-дизайнера, программиста и администратора. Также сайт можно разработать самостоятельно, имея минимальные навыки. Существует много конструкторов и программ, на базе которых создаются сайты: Adobe Dreamweaver, Tilda Publishing.

Язык HTML



HTML (англ. *HyperText Markup Language*) — язык разметки гипертекста, предназначенный для создания веб-страниц.

Структура HTML-документа

Компонент	Характеристика
Тег (tag)	<p>Элементы языка HTML, позволяющие формировать содержимое веб-сайтов. Существуют, например, теги создания абзаца или вставки изображения. Конструкция любого HTML-кода управляется тегами. В тегах должны находиться ключевые запросы. Тег вставляется в угловые скобки: <...>. Теги бывают одиночными и парными.</p> <p>Парный начинается с открывающего тега и заканчивается закрывающим тегом. Символ «/» в парных тегах означает его окончание.</p> <ul style="list-style-type: none">✓ Тег <p>...</p> — парный: <p> означает, что абзац начался, </p> — абзац закончился.✓ Тег — одиночный. У него нет парного ему закрывающего тега
Атрибут (аргумент)	<p>Изменяет тег. Состоит из имени, знака равенства и значения атрибута.</p> <ul style="list-style-type: none">✓ Выравнивание абзаца, изменение цвета шрифта и т. п.

Компонент	Характеристика
Значение	<p>Значения присваиваются атрибутам и определяют вносимые изменения. Значения могут быть текстовыми, а также числовыми, как, например, ширина и высота изображения.</p> <p>✓ <code><p style="font-family:verdana; color:red;"></code></p> <p>Тег <code><p></code> означает начало абзаца, атрибут <code>style</code> присваивает значение тексту: красный цвет и шрифт Verdana</p>



HTML-документ — текстовый документ, который может быть создан как в обычном текстовом редакторе (Блокнот), так и в специализированном (Notepad++, Visual Studio Code и т. п.).



HTML-документ имеет расширение `.html`. На сайте обязательно должна быть страница `index.html` — главная страница сайта. Справа приведена основная структура документа HTML, а также примеры тегов и их атрибутов.

Структура документа HTML

```

<!DOCTYPE html>
<html>
  <head>
    <Мета-теги>
    <title>
      Заголовок документа
    </title>
  </head>
  <body>
    Основная часть документа
  <Функции скриптов>
  </body>
</html>

```

Значение основных тегов	Примеры тегов	Атрибуты тегов
<ul style="list-style-type: none"> • Элемент <code><!DOCTYPE></code> предназначен для указания типа текущего документа. • <code><html></code> показывает начало и конец кода. • <code><head></code> описывает то, что находится в документе, и как это следует выводить. • <code><title></code> показывает начало и конец основного заголовка. • <code><body></code> содержит весь контент, отображаемый на веб-странице в браузере 	<ul style="list-style-type: none"> • От <code><h1></h1></code> до <code><h6></h6></code> — шесть уровней заголовков. • <code><p></p></code> — абзац. • <code>
</code> — переход на другую строку. • <code><hr></code> — горизонтальная линия. • <code></code> и его атрибуты <code>face</code>, <code>size</code>, <code>color</code> — гарнитура, размер, цвет шрифта. • <code></code> — нумерованный список. • <code></code> — элемент списка. • <code></code> — маркированный список, атрибут <code>TYPE</code> задаёт вид маркера: <code>disc</code> — диск, <code>square</code> — квадрат, <code>circle</code> — окружность. • <code></code> — вставка изображения. • <code><video> </video></code> — вставка видео. • <code><link></code> — установка связи текущего документа с внешним ресурсом 	<ul style="list-style-type: none"> • <code>src</code> — исходный URL-адрес файла изображения. • <code>align</code> — способ выравнивания изображения. • <code>border</code> — видимая рамка вокруг изображения. • <code>width</code> — ширина изображения. • <code>height</code> — высота изображения. • <code>loop</code> — число повторений видео. • <code>start</code> — условие запуска видео. • <code>colspan</code> — количество столбцов, занимаемое ячейкой. • <code>rowspan</code> — количество рядов, занимаемое ячейкой. • <code>bgcolor</code> — цвет фона ячейки. • <code>background</code> — фоновый рисунок в ячейке



Практически в каждой странице хорошо разработанного профессионального веб-сайта используются каскадные таблицы стилей (CSS), которые управляют форматированием элементов в браузере.

Стили в документе

Стиль	Характеристика
Внутренние	<p>Определяются атрибутом <code>style</code> конкретных тегов. Он действует только на элементы, определённые этими тегами.</p> <ul style="list-style-type: none"> ✓ <code><p style=»color:green»>Абзац</code> с текстом зелёного цвета<code></p></code>
Глобальные	<p>CSS размещаются в теге <code><style>...</style></code>, расположенном, в свою очередь, в <code><head>...</head></code>.</p> <ul style="list-style-type: none"> ✓ <code><style type=»text/css»></code> <code> p {color:#808080;}</code> <code></style></code> <p>Результатом будет серый цвет текста во всех абзацах</p>
Внешние	<p>Определяются в отдельном файле с расширением <code>css</code>. Внешние стили позволяют всем страницам сайта выглядеть единообразно. Для подключения требуемой таблицы стилей используется элемент <code><link></code> в блоке <code><head></code> документа HTML5.</p> <ul style="list-style-type: none"> ✓ <code><link href=»style.css»</code> <code> rel=»stylesheet»></code>



Для создания веб-страниц нужен:

- браузер (программа, пригодная для просмотра HTML-файлов);
- редактор текстовых файлов, поддерживающий язык выбранной вами кодировки (Notepad++ или программа «Блокнот»);
- специальные программы, позволяющие создавать веб-страницы из сайтов даже тем людям, которые не знают о языке HTML, путём визуального редактирования на экране дисплея.

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

АЛГОРИТМЫ И АЛГОРИТМИЗАЦИЯ

Историческая справка

Дата	Событие
Более 2000 лет назад	Открыт алгоритм Евклида (нахождение наибольшего общего делителя двух целых чисел)
Начало IX в.	Возник термин «алгоритм» от имени великого персидского учёного Мухаммада аль-Хорезми, описавшего четыре арифметических действия. При переводе его трудов на латинский язык имя учёного было записано как <i>Algorithmi</i> . Постепенно значение слова расширялось, и математики стали применять его не только к вычислительным, но и к другим операциям
Около 1360 г.	Н. Орем использовал слово «алгоритм» в математическом трактате «Вычисление пропорций» (<i>Algorismus proportionum</i>)
1684 г.	Г. Лейбниц в сочинении впервые использовал слово «алгоритм» (<i>Algorithmo</i>) в более широком смысле: как систематический способ решения проблем дифференциального исчисления
XVIII в.	Понятие алгоритма использовал выдающийся математик XVIII в. Л. Эйлер, его работа называлась «Использование нового алгоритма для решения проблемы Пелля»
Начало 1930-х гг.	Теория алгоритмов

Машина Тьюринга



Машина Тьюринга — модель универсального исполнителя алгоритмов обработки символьных последовательностей, состоящая из центра управления и ряда ячеек. Управляющее устройство машины имеет возможность двигаться по ряду ячеек и считывать все значения, которые им присвоены.

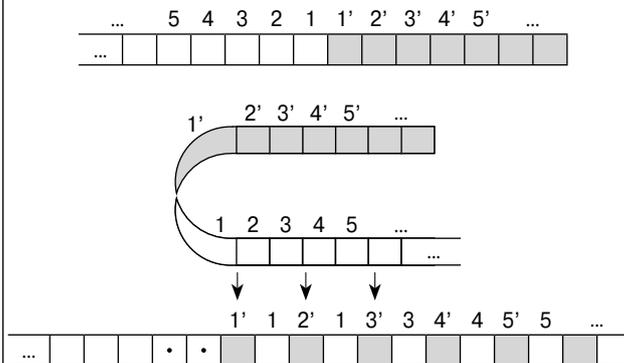
Принцип действия

Алгоритм позволяет задавать значение ячейки, в которой находится управляющее устройство. Также можно задать некоторое действие устройства в данной ячейке. Для задания значений ячейкам в машине Тьюринга используют ограниченный алфавит. Для перехода от одного значения к другому применяют команду:

$$q_1 a_1 \rightarrow q_2 a_2 D.$$

Слева от команды написано состояние или значение данной ячейки в рассматриваемый момент времени, справа — значение, которое будет иметь данная ячейка после изменения свойств. D — передвижение управляющего устройства. Если на месте данной буквы стоит L , это означает, что управляющее устройство необходимо передвинуть влево, если R — передвинуть вправо, буква N в команде оставляет устройство на месте

Пример

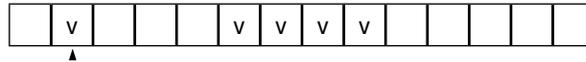


Схематичное изображение машины Тьюринга, работающей на полубесконечной ленте

Машина Поста



Машина Поста в своей реализации проще, чем машина Тьюринга: она имеет бесконечный ряд клеток (ячеек), в каждой из которых стоит либо отсутствует метка или какой-нибудь знак, соответствующий принимаемым значениям 1 или 0. Имеется каретка, которая может передвигаться вправо или влево. Алгоритм данной машины состоит из нескольких строк и имеет небольшое количество команд.



Модель машины Поста

Команды	Действия машины Поста
$n \leftarrow m$	Сдвиг каретки на шаг влево и переход к выполнению команды с номером m
$n \rightarrow m$	Сдвиг каретки на шаг вправо и переход к выполнению команды с номером m
$n \vee m$	Запись метки в текущую пустую клетку и переход к выполнению команды с номером m
$n \dagger m$	Стирание метки в текущей клетке и переход к выполнению команды с номером m
$n !$	Остановка выполнения программы
$n ? m, k$	Переход в зависимости от содержимого текущей клетки: если текущая клетка пустая, то следующей будет выполняться команда с номером m , если непустая — команда с номером k

Алгоритм Маркова



Нормальный алгоритм Маркова позволяет изменять слова заменой их частей. Данный алгоритм имеет множество значений, замена производится подстановкой.



Изначально имеется непустой конечный алфавит и последовательность знаков алфавита — слова, с помощью которых описываются алгоритм и данные. Есть пустые слова, для удобства их можно обозначить μ . Существует алгоритм, который переводит одно слово в другое, и есть непустая строка символов, к которой нужно применить алгоритм.

Принцип действия алгоритма

1	В упорядоченной последовательности подстановок найти самую первую подстановку, левое слово которой входит в строку данных
2	В строке данных найти самое первое (левое) вхождение левого слова найденной подстановки
3	Это вхождение в строке данных заменить на правое слово найденной подстановки (преобразование данных)



Шаг работы алгоритма прекращается в следующих случаях:

- если возникнет ситуация, когда шаг не сможет быть выполнен из-за того, что ни одна подстановка не подходит (левое слово любой подстановки уже не входит в строку данных), — правило остановки;
- если будет установлено, что процесс подстановок не может прекратиться.

Основные понятия

Понятие	Характеристика
Алгоритм	Точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату
Алгоритмизация	Метод описания систем или процессов путём составления алгоритмов их функционирования
Исполнитель	Любое устройство или живое существо, которое способно точно и однозначно выполнять заданные команды (действия, определённые алгоритмом)
Система команд исполнителя	Совокупность команд, которые он способен выполнить. Другими словами, алгоритм представляет собой запись в той или иной форме последовательности команд для исполнителя
Команда	Описание элементарной операции, которую должен выполнить исполнитель
Блок-схема	Графический способ представления алгоритма, в котором шаги изображаются в виде блоков различной формы, соединённых между собой стрелками

Основные элементы блок-схемы

	Блок начала или конца программы		Блок ввода или вывода информации
	Логический блок с одним входом и двумя выходами: проверка заданного условия		Блок организации циклического процесса
	Блок обработки информации		Направление процесса

Свойства представления алгоритмов

Свойство	Характеристика
Дискретность	Алгоритм должен представлять процесс решения задачи как упорядоченное выполнение простых действий (шагов, этапов), т. е. должен быть разбит на последовательность отдельных шагов, каждый из которых называется командой. При этом для выполнения каждого действия алгоритма требуется время
Понятность	Чтобы исполнитель мог выполнить все команды согласно алгоритму, он должен их понять и выполнить каждую, поэтому любой алгоритм должен составляться для конкретного исполнителя с учётом его возможностей. У каждого исполнителя имеется перечень команд, которые он способен выполнить (система команд исполнителя). Понятными исполнителю будут являться только те команды, которые попадают в этот список
Результативность	Алгоритм должен приводить к решению задачи или сообщению, что задача решений не имеет, за конечное число шагов. Результат выполнения алгоритма должен быть обязательно получен, т. е. правильный алгоритм не может обрываться безрезультатно из-за какого-либо непреодолимого препятствия в ходе выполнения. При наличии ошибок возможны нарушения результативности
Корректность	Решение задачи, которая задаётся алгоритмом, должно быть правильным (корректным) для любых допустимых исходных данных
Детерминированность	Каждая команда должна быть чёткой, однозначной, исключать произвольное толкование и не оставлять места для двусмысленности. По сути, выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний. Результат не может зависеть от какой-либо дополнительной информации извне алгоритма. Сколько бы раз ни запускали программу, для одних и тех же исходных данных всегда будет получаться один и тот же результат. При наличии ошибок в алгоритме это свойство может иногда нарушаться

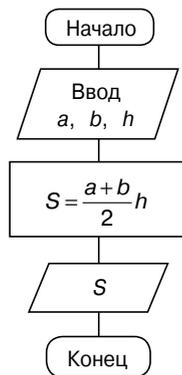
Конечность	Каждое отдельное действие, как и весь алгоритм, должно иметь возможность реального исполнения, поэтому алгоритм имеет предел, т. е. конечен
Массовость	Алгоритм разрабатывается в общем виде так, чтобы его можно было применять для класса задач, различающихся только исходными данными. При этом исходные данные выбираются из некоторой области, которая называется областью применимости алгоритма

Способы представления алгоритмов

Словесно-формульный способ	Графический способ описания алгоритма	Операторный способ
Представление алгоритма в виде текстовых записей и (или) формулы	Представление алгоритма в виде графической схемы, рисунка. В программировании для написания определённой программы часто используют алгоритмы, представленные блок-схемой, и после применяют операторный способ	Описание алгоритма на языке программирования (алгоритмическом языке)
<p>✓ Требуется решить линейное уравнение $ax + b = 0$, $a \neq 0$ (без этого условия уравнение не будет линейным). Математической моделью решения этой задачи является формула $x = \frac{-b}{a}$</p>	<p>✓ Пример графического способа записи алгоритма для вычисления площади S трапеции, где a, b — основание трапеции, h — высота трапеции</p>	<p>✓ Пример описания на алгоритмическом языке.</p> <p>алг наибольшее число нач цел a, b, K вывод "Введите два целых числа", nc ввод a, b</p>

✓ На основании этой формулы запишем алгоритм.

- 1) Задать значения a , b . $a \neq 0$.
- 2) Перенести b в правую часть с противоположным знаком.
- 3) Разделить обе части уравнения на a .
- 4) Записать результат: «Корень уравнения $x = c$ »



$K := b$
 если $a > b$ то
 $K := a$
 все
 вывод "Наибольшее число", K
 кон

Данный алгоритм на экран выводит наибольшее из двух вводимых целых чисел

Виды алгоритмов



Большинство наших действий можно записать в виде алгоритма. Эти действия выполняются за несколько неповторяющихся шагов, иногда приходится делать выбор между двумя вариантами, в других случаях для получения результата приходится повторять одни и те же шаги. Исходя из способа достижения результата, выделяют три основных вида алгоритмов: **линейный** (шаги выполняются последовательно друг за другом), **разветвляющийся** (порядок выполнения шагов изменяется в зависимости от условия) и **циклический** (определённая последовательность шагов повторяется несколько раз).

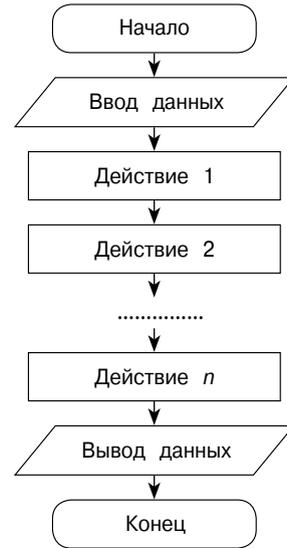
Линейный алгоритм



Наиболее простым в информатике считается **линейный алгоритм**. Он образуется командами, выполняющимися однократно и именно в той последовательности, в которой они записаны. Алгоритм с линейной структурой можно записать как в текстовой, так и в графической форме.

Действия, необходимые для организации линейного алгоритма

1	Определить тип/класс исходных, промежуточных данных и конечного результата, выбрать имена переменных и конечного результата
2	Определить, какие математические формулы связывают результат и исходные данные
3	Записать окончательный алгоритм. Он должен включать в себя ввод данных, вычисления, вывод результатов



Блок-схема для линейного алгоритма

Разветвляющийся алгоритм



Алгоритм ветвления, или **разветвляющийся алгоритм**, используется для реализации задач, связанных с условиями «если... то...». Логика такого алгоритма можно описать следующим образом:

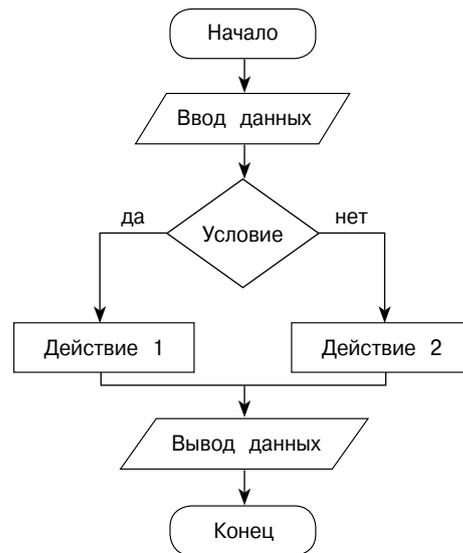
ЕСЛИ <условие истинно> ТО <действие 1> ИНАЧЕ <действие 2>.

Действия, необходимые для организации алгоритма ветвления

1	Определить тип/класс исходных, промежуточных данных и конечного результата, выбрать имена переменных и конечного результата
2	Установить, какие математические формулы связывают результат и исходные данные
3	Определить результаты, если выполнится или не выполнится условие
4	Записать окончательный алгоритм. Он должен включать в себя ввод данных, вычисления, условия, вывод результатов



Разветвляющийся процесс, включающий в себя две ветви, называется **простым**, более двух ветвей — **сложным**. Сложный разветвляющийся процесс можно представить с помощью простых разветвляющихся процессов.



Блок-схема для разветвляющегося алгоритма

Циклический алгоритм



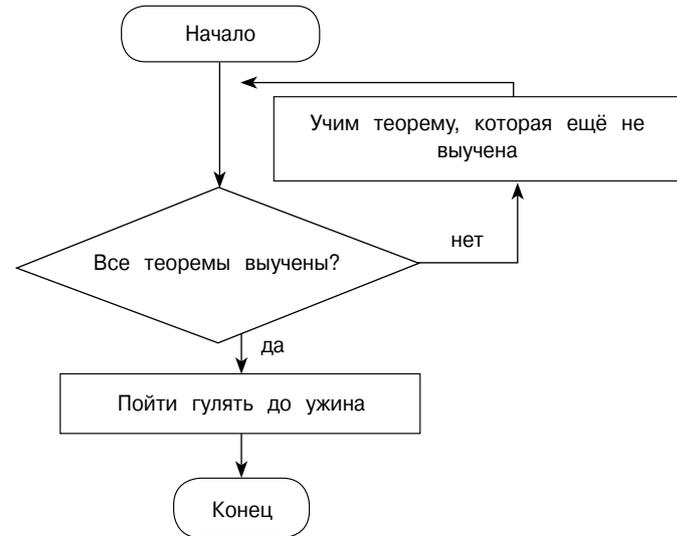
Циклический алгоритм — описание действий, которые должны повторяться указанное число раз или до тех пор, пока не выполнено заданное условие.



В представленном ниже порядке действий, необходимых для организации циклического алгоритма, действия 5—7 повторяются многократно до тех пор, пока цикл не будет закончен, т. е. до определённого условия. Переменная, которая изменяется в цикле, называется **параметром цикла**. Действия, повторяемые многократно, называются **телом цикла**. Обычно тело цикла выполняется определённое количество раз, т. е. диапазон изменения параметра известен заранее.

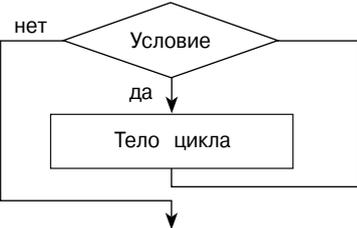
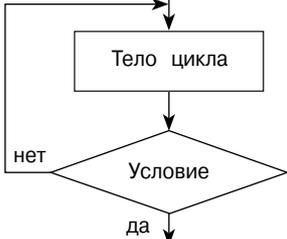
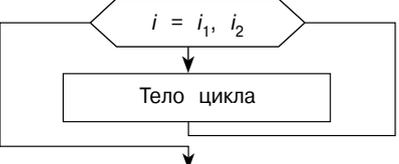
Действия, необходимые для организации циклического алгоритма

1	Определить тип/класс исходных, промежуточных данных и конечного результата, выбрать имена переменных и конечного результата
2	Определить, какие математические формулы связывают результат и исходные данные
3	Перед циклом задать начальное значение переменной, которая будет изменяться в цикле
4	Произвести необходимые действия, записать математические формулы
5	Изменить переменную на величину шага
6	Проверить условие окончания (или повторения) цикла
7	Перейти к п. 5, если цикл не закончен



Блок-схема для циклического алгоритма

Классификация циклических алгоритмов

Цикл с предусловием (цикл ПОКА)	Цикл с постусловием (цикл ДО)	Цикл с параметром (цикл ДЛЯ)
<p>Цикл, который выполняется, пока истинно некоторое условие, указанное до его начала. Это условие проверяется до выполнения тела цикла, поэтому, если условие ложно, тело цикла может быть не выполнено ни разу</p>	<p>Цикл, в котором условие проверяется после выполнения тела цикла. Отсюда следует, что тело цикла всегда выполняется хотя бы один раз. Этот цикл предписывает повторять команды тела цикла до тех пор, пока не выполнится условие в его заголовке (т. е. пока оно остаётся ложным)</p>	<p>Имеет также название «цикл со счётчиком» — цикл, в котором некоторая переменная изменяет своё значение от заданного начального до конечного с некоторым шагом, и для каждого значения этой переменной тело цикла выполняется один раз</p>
<p>Блок-схема цикла с предусловием</p> 	<p>Блок-схема цикла с постусловием</p> 	<p>Блок-схема цикла с параметром</p> 



Алгоритмическая конструкция, в которой внутри одного (глобального) цикла расположен другой (локальный) цикл, называется **алгоритмом со структурой вложенных циклов**. Вложенный цикл по отношению к циклу, в тело которого он вложен, именуется **внутренним циклом**, а цикл, в теле которого существует вложенный цикл, является **внешним** по отношению к вложенному.

Выигрышная стратегия



К задачам, которые нуждаются в применении алгоритма, относятся игры. Победа в них достигается за счёт нахождения выигрышного алгоритма (выигрышной стратегии). Игры очень результативны, поскольку даже самая маленькая и простая игра может иметь прикладное значение.

Постановка задачи в игре

1	Начальные условия игры
2	Правила игры для каждого из игроков
3	Указание условий выигрыша



Методы нахождения выигрышной стратегии:

- перебор вариантов;
- использование дерева решений;
- инварианты;
- симметрия;
- чётность и нечётность;
- анализ с конца.

Порядок решения задач на определение выигрышной стратегии

1	Проанализировать алгоритм игры, данный в задаче
2	Построить дерево партий
3	Найти и описать выигрышную стратегию
4	Описать выигрышную стратегию и указать возможное количество шагов, которое потребуется для выигрыша. Простого указания на того, кто имеет выигрышную стратегию, недостаточно
5	Привести доказательства, что указанная стратегия приводит к выигрышу. Просто назвать стратегию выигрышной недостаточно

ПРОГРАММИРОВАНИЕ



Программирование — теоретическая и практическая деятельность, связанная с созданием программ.



Программирование включает создание сайтов (веб-программирование), решение прикладных задач, администрирование баз данных, создание игр и макросов для них, программирование приложений для смартфонов, планшетов и других устройств.

Основные этапы создания программы

1	Анализ поставленной задачи
2	Проектирование — разработка комплекса алгоритмов
3	Кодирование и компиляция — написание исходного текста программы и преобразование его в исполнимый код с помощью компилятора
4	Тестирование и отладка — выявление и устранение ошибок в программе
5	Испытание и сдача программы
6	Создание документации
7	Сопровождение, внесение изменений, оптимизация

Классификация языков программирования

По степени зависимости от аппаратных средств	По принципам программирования	По ориентации на класс задач
Языки низкого уровня. ✓ Assembler (автокод)	Процедурные. ✓ Assembler, Fortran, Basic, Pascal, Ada	Универсальные. ✓ Algol, PL/1, Simula, Basic, Pascal
Языки высокого уровня. ✓ Fortran, Pascal, Ada, Basic	Непроцедурные, включающие объектно-ориентированные. ✓ Prolog, Langin, C++, Visual Basic, Java	Специализированные. ✓ Fortran (инженерные расчёты), Cobol (коммерческие задачи), Refal, Lisp (символьная обработка), Modula, Ada (программирование в реальном времени)

Основные понятия языков программирования



Языки программирования, как и естественные языки, имеют алфавит, словарь, грамматику, синтаксис и семантику. Взаимодействие синтаксических и семантических правил определяет основные понятия языка, такие как операторы, идентификаторы, константы, переменные, функции, процедуры и т. д.

Понятие	Характеристика
Алфавит	Разрешённый к использованию набор символов, с помощью которого могут быть образованы слова и величины данного языка
Синтаксис	Система правил, определяющих допустимые конструкции языка программирования из букв алфавита
Семантика	Система правил однозначного толкования каждой языковой конструкции, позволяющих производить процесс обработки данных
Словарь	Все используемые слова языка программирования
Грамматика	Свод правил и законов языка



Машинный язык — язык программирования, элементами которого являются команды компьютера, характеризующиеся:

- количеством операндов в команде;
 - назначением информации, задаваемой в операндах;
 - набором операций, которые может выполнить компьютер, и др.
- Конструкции машинного языка интерпретируются аппаратурой.

Понятие	Характеристика
Система программирования	Программная система, предназначенная для разработки программ на конкретном языке программирования. Система программирования предоставляет пользователю специальные средства разработки программ: транслятор, (специальный) редактор текстов программ, библиотеки стандартных подпрограмм, программную документацию, отладчик и др.

Алгоритмический язык	Искусственный язык, предназначенный для записи алгоритмов, позволяющий представить алгоритм в виде текста, составленного по определённым правилам с использованием специальных служебных слов. Количество служебных слов ограничено, и каждое из них имеет точно определённый смысл, назначение и способ применения. При записи алгоритма служебные слова выделяют полужирным шрифтом или подчёркиванием. Алгоритмический язык не привязан к архитектуре компьютера и не содержит деталей, которые связаны с устройством машины
Псевдокод	Компактный, неформальный язык описания алгоритмов, использующий ключевые слова императивных языков программирования, но опускающий несущественные для понимания алгоритма подробности и специфический синтаксис. Псевдокод занимает промежуточное положение между естественным языком и языками программирования. Пример псевдокода — учебный алгоритмический язык

Типы данных



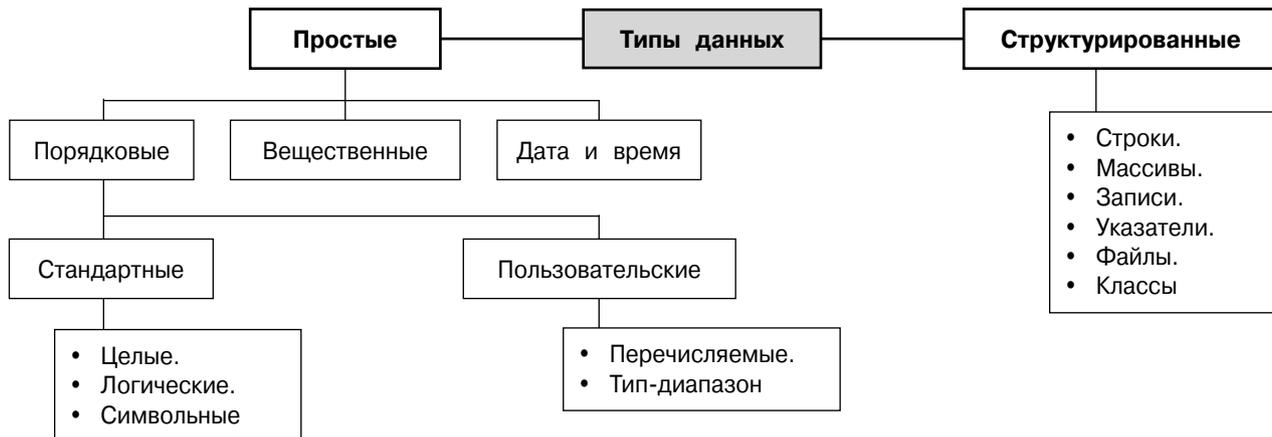
Тип данных — множество значений, которые могут принимать данные, и множество операций, которые применимы к этим значениям.

- ✓ Данные разных типов: 56 — целое число, 26.07.2020 — дата.



Тип данных определяет:

- внутреннюю форму представления данных в ЭВМ;
- возможные значения переменных, констант, функций, выражений, принадлежащих к определённому типу;
- операции и функции, которые могут выполняться над величинами, принадлежащими к определённому типу.



Структурированные типы данных



Структура данных — способ организации хранения данных и доступа к ним, предназначенный для выполнения определённого набора операций над этими данными.

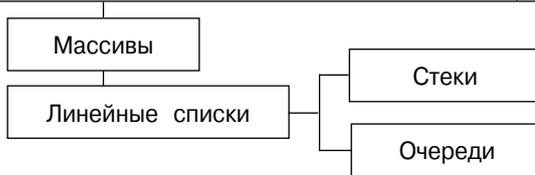


Узел (элемент структуры данных) — единица хранения данных, несущая в себе ссылки на связанные с ней элементы (узлы). Каждый узел содержит две составляющие: фактические данные, которые в нём хранятся (это могут быть данные любого типа), и указатель (или ссылку) на очередной узел в последовательности.

Структуры данных

Линейные

Элементы образуют последовательность или линейный список, обход узлов будет также линейен



Нелинейные

Обход узлов нелинейный и данные непоследовательны

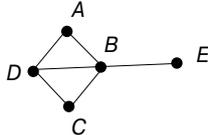
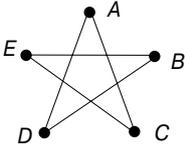
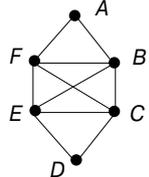


Графы



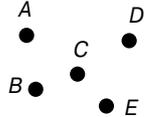
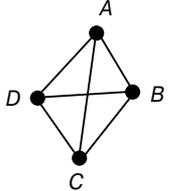
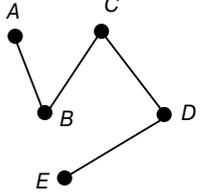
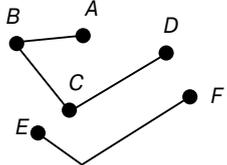
Граф — непустое множество точек и множество (возможно, и пустое) отрезков кривой, оба конца которой принадлежат заданному множеству точек.

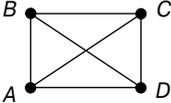
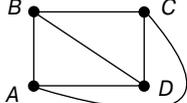
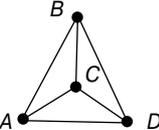
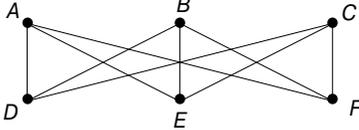
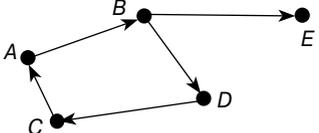
Понятие	Характеристика
Ребро	Отрезки графа. Если при отсутствии некоторого ребра граф делится на несколько компонент, то это ребро называется мостом
Вершина	Точки графа. Вершина графа, которая имеет только одно ребро, является висячей
Маршрут	Вершины, последовательно соединённые рёбрами. Маршрут должен иметь только одно направление. Маршрут, в котором рёбра не пересекаются, называют путём . Если путь замкнут, он является циклом

Определение	Пример
<p>Путь, который проходит через каждую вершину, причём единожды, называют гамильтоновым путём</p>	
<p>Если маршрут проходит по всем рёбрам, его называют эйлеровым маршрутом (путём). При этом никакое ребро маршрута не должно встречаться более одного раза. Вершина, от которой проложен маршрут, называется началом пути, вершина в конце маршрута — конец пути</p>	
<p>Эйлеровым циклом в графе называется цикл, содержащий все рёбра графа (т. е. это замкнутый эйлеров путь). Если все вершины цикла разные, такой цикл называется элементарным (или простым) циклом. Если же цикл включает в себя все рёбра графа по одному разу, такой цикл называется эйлеровой линией</p>	

Виды графов и их свойства

Вид	Характеристика
По количеству рёбер и вершин	
Нулевой	Состоит из одной точки <div style="text-align: center; margin-top: 10px;">  </div>

<p>Пустой</p>	<p>Нет ни одного ребра</p>	
<p>Полный</p>	<p>Каждые две различные вершины его соединены одним и только одним ребром</p>	
<p>По количеству компонент связности</p>		
<p>Связный</p>	<p>Граф, у которого одна компонента связности</p>	
<p>Несвязный</p>	<p>Граф, у которого больше одной компоненты связности</p>	

Вид	Характеристика
По расположению на плоскости	
Планарный	Граф можно изобразить на плоскости таким образом, что никакие два ребра, за исключением выходящих из общей вершины, не имеют общих точек (т. е. не пересекаются)
Плоский	<p>Граф, который можно уложить в одной плоскости. У всякого планарного графа имеется плоское представление. На рисунке <i>а</i> изображён граф, в котором имеется пересечение рёбер. Если ребро <i>AC</i> перебросить через вершину <i>B</i> (рис. <i>б</i>) и перерисовать граф в другом виде, получим граф <i>ABCD</i> (рис. <i>в</i>), в котором нет пересечения рёбер. Таким образом, можно утверждать, что граф, изображённый на рисунке <i>а</i>, планарный, а его плоское представление — плоский граф на рисунке <i>в</i>. В случае отсутствия возможности подобного преобразования граф не будет являться планарным.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>а)</p>  </div> <div style="text-align: center;"> <p>б)</p>  </div> <div style="text-align: center;"> <p>в)</p>  </div> </div>
Непланарный	<p>Граф, который не имеет плоского представления. Никакое перемещение рёбер не позволит превратить его в граф без их пересечения</p> <div style="text-align: center;">  </div>
По направленности	
Ориентированный	<p>Граф, имеющий рёбра с заданной направленностью</p> <div style="text-align: center;">  </div>

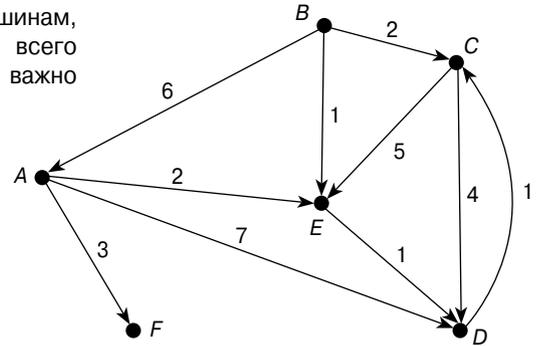
Неориентированный

Граф, ни одному ребру которого не присвоено направление

Другие виды графов

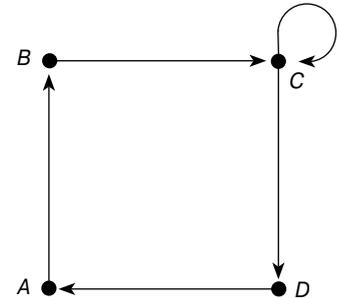
Взвешенный

Граф, некоторым элементам которого (вершинам, рёбрам или дугам) сопоставлены числа. Чаще всего числа сопоставляют рёбрам или дугам. Это важно при составлении алгоритмов



Псевдограф

Граф, в котором допускается наличие петель и существование более одного ребра между двумя вершинами



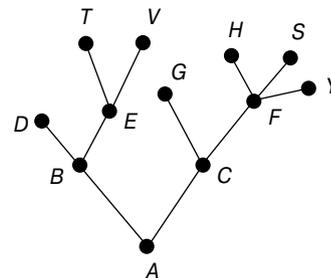
Свойства связных и несвязных графов

1	Если в связном графе все степени вершин чётные, то существует эйлеров путь, причём начало пути совпадает с его концом
2	Если в связном графе степень двух вершин нечётная, то существует эйлеров путь, причём начало и конец пути не совпадают
3	Если в графе более двух вершин с нечётной степенью, то эйлерова пути не существует

Деревья



Деревом называется любой связный граф, не имеющий циклов. Это частный случай графа, наиболее широко применяемый в программировании. Принято считать деревом также всякий граф, состоящий из одной (изолированной) вершины. Существуют и ориентированные деревья, аналогичные ориентированным графам. Вершина дерева, имеющая степень единицу, называется **висячей вершиной**.



Граф-дерево

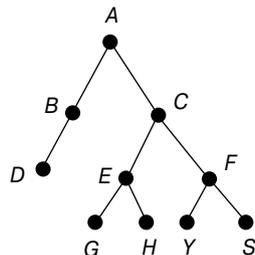
Свойства графа-дерева	Типовые операции над элементами дерева
Для каждой пары вершин дерева существует единственный путь, соединяющий их	Добавление элемента в дерево
	Удаление элемента из дерева
Всякое ребро в дереве является мостом. Действительно, после удаления любого ребра дерево распадается на два	Обход дерева
	Поиск элемента в дереве



Дерево с n вершинами имеет $n - 1$ ребро.



Бинарное дерево — корневое дерево, каждая вершина которого имеет не более двух потомков. Данный граф иногда называют двоичным деревом, поскольку из каждой вершины выходит два ребра. Бинарные, или двоичные, деревья обычно применяются для реализации множеств и массивов, а также различных алгоритмов вычислительной геометрии.



Бинарное дерево

Способы представления графа

Способ	Характеристика
Матрица смежности	Представляет собой квадратную матрицу размером $n \times n$ (n — количество вершин в графе), заполненную единицами и нулями по следующему правилу: если в графе имеется ребро, соединяющее вершины A и B , то $S_m[A, B] = 1$, в противном случае $S_m[A, B] = 0$. Матрицу можно представлять в виде таблицы, где в каждой ячейке есть значение 0 или 1
Матрица инцидентности графа	Количество строк в этой матрице соответствует числу вершин, а количество столбцов — числу рёбер
Список рёбер	Этот способ задания графов наиболее удобен для внешнего представления входных данных. Каждая строка входного файла содержит информацию об одном ребре (дуге). ✓ <номер_начальной_вершины> <номер_конечной_вершины> [<вес_ребра>]
Списки смежности	Для каждой вершины будет указан список всех смежных с ней вершин (для ор-графа — список вершин, являющихся концами исходящих дуг). ✓ <номер_начальной_вершины>: <номера_смежных_вершин>

Массив



Массив — упорядоченный набор данных, предназначенный для хранения данных одного типа, идентифицируемых с помощью одного или нескольких индексов.



Массивы используются для обработки большого количества одно-типных данных, например последовательностей (одномерные массивы) и таблиц (двухмерные массивы). Массив характеризуется:

- размерностью;
- базовым типом элементов;
- типом индекса (может быть только порядковым типом);
- множеством значений для индекса.

Понятие	Характеристика
Элемент массива	Отдельная переменная, входящая в массив
Индекс элемента массива	Номер элемента в этом массиве
Размерность массива	Количество элементов, которое содержит массив

Массив	Характеристика
Векторный (одномерный)	Массив, состоящий из некоторого ряда элементов. ✓ [1, 2, 5, 7, 8] — векторный массив, состоящий из пяти элементов

Матричный (многомерный)

Массив включает несколько рядов.

$$\begin{bmatrix} 3 & 5 & 3 \\ 4 & -1 & 2 \\ 2 & -3 & 1 \\ 10 & 2 & 4 \end{bmatrix}$$

✓ Массив является многомерным (в данном случае — двухмерным) размерности 3×4 и включает 12 элементов



Массивы могут состоять не только из чисел, но и из строк.

✓ Массив [“пример”, “использования”, “строк”] состоит из трёх строк, для каждой из которых выделено определённое количество информации.

Способы заполнения массива данными

Непосредственное присваивание значений элементам

Заполнение массива произвольными элементами, случайными числами

Ввод значений элементов с клавиатуры или чтение из файла



Количество элементов в массиве задаётся изначально и обозначается латинской буквой, например n . Доступ к элементам производится по числовому индексу, записывается как `mydogs[i]`, где `mydogs` — имя массива, `[i]` — числовой индекс элемента. Так, для обращения к символу под номером 5 запишем: `mydogs[5]`. В многомерном массиве также используются латинские буквы, например j и k . В этом случае доступ к массиву осуществляется по записи вида `inf[5,6]`, где `inf` — имя массива; числа обозначают, что запрошенный элемент находится на пересечении строки 5 и столбца 6. Доступ к массиву произвольный, поэтому можно обращаться к элементам под индексами 0, 33, 2020, и независимо от того, какой из элементов мы запросим, он сразу появится.

Действия над массивами и их элементами



Для того чтобы выбрать конкретный элемент массива, надо после идентификатора массива указать его индекс или индексы, поэтому элементы массива часто называют индексированными переменными. Элемент одномерного массива имеет один индекс, двумерного — два и т. д. Над элементами массива могут производиться те же действия, что и над простыми переменными того же типа.

Действие	Характеристика
Проверка на равенство	$A=B$. Результатом является значение «истина», если каждый элемент массива A равен соответствующему элементу массива B
Проверка на неравенство	$A <> B$. Результатом является значение «истина», если значение хотя бы одного элемента массива A не равно значению соответствующего элемента массива B
Присваивание	$A:=B$. Все значения элементов массива B присваиваются элементам массива A . Значения элементов массива B остаются без изменения
Инициализация массива	Инициализация — это присваивание каждому элементу массива какого-либо значения
Ввод и вывод элементов массива	Массив выводится на экран компьютера
Копирование элементов	Для выполнения копирования из одного массива в другой необходимо, чтобы массивы, участвующие в данной операции, были описаны одинаково

Поиск в массиве элементов	Пусть нам задано некоторое условие, по этому условию нам нужно найти элементы в массиве
Поиск минимального (максимального) элемента	Осуществляется отбор минимального или максимального элемента среди всех элементов массива
Вычисление суммы (произведения) элементов массива	В результате получаем одно число
Перестановка элементов массива	Осуществляется с использованием вспомогательной переменной того же типа, что и базовый тип массивов
Суммирование элементов матриц	Суммирование элементов матриц (двухмерного массива) по строкам или столбцам. В результате получаем одномерный массив
Поиск минимального (максимального) элемента строки (столбца)	В результате выводим минимальный или максимальный элемент по строке или столбцу
Сортировка массивов	Изменение массива в соответствии с заданным условием

Сортировка массива



Сортировкой называется процесс упорядочивания набора данных одного типа по возрастанию или убыванию значения какого-либо признака. **Алгоритм сортировки** — система последовательных операций для упорядочивания элементов в списке.

Параметры оценки	Характеристика
Время сортировки	Основной параметр, характеризующий быстродействие алгоритма
Память	Ряд алгоритмов требует выделения дополнительной памяти под временное хранение данных. При оценке используемой памяти не будет учитываться место, которое занимает исходный массив, и не зависящие от входной последовательности затраты, например на хранение кода программы
Устойчивость	Устойчивая сортировка не меняет взаимного расположения равных элементов. Такое свойство может быть очень полезным, если элементы состоят из нескольких полей, а сортировка происходит по одному из них
Естественность поведения	Определяется эффективностью метода при обработке уже отсортированных или частично отсортированных данных. Алгоритм ведёт себя естественно, если учитывает эту характеристику входной последовательности, и работает лучше



Сортировка массива может происходить по возрастанию или убыванию. Сортировка строк часто ранжируется по алфавиту. Различают обменную сортировку (пузырьком), сортировку выбором, сортировку вставками, сортировку слиянием.

Сортировка обмена (пузырьком)	Сортировка выбором
Идея метода: сортировка массива от элемента $a[0]$ к элементу $a[i]$ с попарным изменением местами двух элементов массива при определённых условиях	Идея метода: создавать отсортированную последовательность путём присоединения к ней одного элемента за другим в правильном порядке. Строим готовую последовательность, начиная с левого конца массива

Первый проход

1) Проходим снизу вверх (слева направо) по массиву.

2) Просматриваем пары соседних элементов.

3) Меняем местами элементы, если они не удовлетворяют условию сортировки.

После первого прохода по массиву вверх оказывается самый лёгкий элемент — отсюда аналогия с пузырьком.

Второй и последующие проходы

1) Следующий проход делаем до второго сверху элемента, таким образом, второй по величине элемент поднимается на правильную позицию.

2) Делаем проходы по всё уменьшающейся нижней части массива до тех пор, пока в ней не останется только один элемент.

На этом сортировка заканчивается, т. к. последовательность упорядочена по возрастанию

Алгоритм состоит из n последовательных шагов, начиная от нулевого и заканчивая $(n - 1)$ -м.

1) На первом шаге просмотрим весь данный массив и найдём наименьший элемент — k .

2) Меняем первый элемент и k -й (наименьший). Таким образом, на первом шаге получим, что на месте нулевого элемента $a[0]$ стоит наименьший.

3) На втором шаге находим наименьший элемент среди оставшихся $a[1]...a[n - 1]$ и меняем его с $a[1]$.

4) На i -м шаге выбираем наименьший из элементов $a[i]...a[n - 1]$ и меняем его местами с $a[i]$.

5) Продолжаем шаги до тех пор, пока не поменяем последние элементы.

Последовательность шагов при $n = 5$ изображена на рисунке ниже.

4	9	7	6	2	3
---	---	---	---	---	---

исходная последовательность

<u>2</u>	9	7	6	4	3
----------	---	---	---	---	---

шаг 1: 2 \longleftrightarrow 4

<u>2</u>	<u>3</u>	7	6	4	9
----------	----------	---	---	---	---

шаг 2: 3 \longleftrightarrow 9

<u>2</u>	<u>3</u>	<u>4</u>	6	7	9
----------	----------	----------	---	---	---

шаг 3: 4 \longleftrightarrow 7

<u>2</u>	<u>3</u>	<u>4</u>	<u>6</u>	7	9
----------	----------	----------	----------	---	---

шаг 4: 6 \longleftrightarrow 6

<u>2</u>	<u>3</u>	<u>4</u>	<u>6</u>	<u>7</u>	9
----------	----------	----------	----------	----------	---

шаг 5: 7 \longleftrightarrow 7

Вне зависимости от номера текущего шага i последовательность $a[0]...a[i]$ (выделена на рисунке курсивом) является упорядоченной. Таким образом, на $(n - 1)$ -м шаге вся последовательность оказывается отсортированной

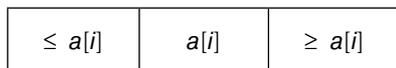
Быстрая сортировка (вставками)

Идея метода: разделение массива на подмассивы и сортировка этих подмассивов. Этот вид сортировки является наиболее применяемым и эффективным

1) Выбираем из массива некоторый опорный элемент $a[i]$.

2) Запускаем процедуру разделения массива, которая перемещает все ключи, меньшие либо равные $a[i]$, влево от него, а все ключи, большие либо равные $a[i]$, — вправо.

Теперь массив состоит из двух подмножеств, причём левое меньше правого либо равно ему.

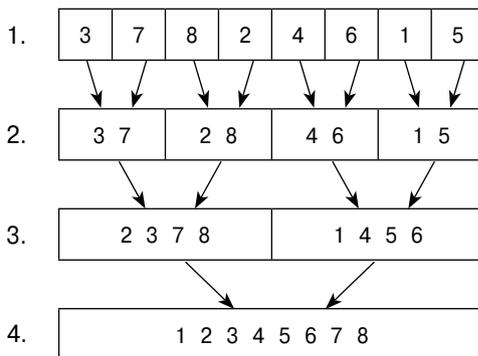


3) Если в подмассиве более двух элементов, рекурсивно запускаем для него ту же процедуру. Производим эту операцию для обоих подмассивов.

4) Продолжаем до тех пор, пока не получим полностью отсортированный массив

Сортировка слиянием

Идея метода: вместо деления по опорному элементу, как в принципе сортировки вставками, массив просто делится пополам



1. Массив разделён до последовательностей длины один
2. Слияние до упорядоченных пар
3. Слияние пар в упорядоченные четвёрки
4. Слияние четвёрок в общий массив

Списки



Список — последовательность связанных элементов, для которых определены операции добавления элементов в произвольное место списка и удаления любого из них. Список задаётся указателем на начало списка.



Каждый элемент списка содержит **ключ**, который идентифицирует его. Ключ может быть целым числом либо строкой и является частью поля данных. В качестве ключа в процессе работы со списком могут выступать разные части поля данных. Ключи разных элементов списка могут совпадать.

Операции, выполняемые над списками

1	Начальное формирование списка (создание первого элемента)
2	Добавление элемента в конец списка
3	Чтение элемента с заданным ключом
4	Вставка элемента в заданное место списка (до или после элемента с заданным ключом)
5	Удаление элемента с заданным ключом
6	Упорядочивание списка по ключу
7	Копирование списка
8	Поиск заданного элемента
9	Объединение в один несколько списков
10	Разбиение одного списка на несколько отдельных списков



Существует несколько разновидностей списков, для которых определены дополнительные правила работы с начальным и конечным элементами. По способу связи элементов различают линейные и циклические списки.

Понятие	Характеристика
Кольцо	Циклический список, имеющий дополнительную связь между последним и первым элементами (первый узел является следующим для последнего)
Стек	<p>Разновидность линейного списка, в котором все включения и исключения узлов (и, как правило, всякий доступ) производятся только на одном его конце. Этот концевой элемент называется вершиной стека. Иногда стек называют списком LIFO (<i>Last In — First Out</i> — «последним вошёл, первым вышел»).</p> <p>Каждый элемент структуры стека содержит указатель на следующий. К этому виду списка неприменима операция обхода элементов. Стек предполагает вставку и удаление элементов, поэтому он является динамической структурой. Механизм действия стека напоминает железнодорожный тупик, из которого первым может выехать только поезд, въехавший в него последним</p>
Очередь	Разновидность линейного списка, в котором все включения элементов производятся на одном его конце, а удаление узлов (и обычно всякий доступ) — на другом. Эти узлы называются началом и концом очереди. Иногда очередь называют списком FIFO (<i>First In — First Out</i> — «первым вошёл, первым вышел»). Механизм действия такого списка действительно похож на бытовую очередь

ОСНОВЫ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ



Программирование в современном мире постепенно становится базовым навыком, ведь информационные технологии развиваются с огромной скоростью. Программирование — фундаментальный навык, представляющий собой совокупность умений мыслить абстрактно, критически и разделять задачу на составные части. Развитие этого навыка невозможно без знания языков программирования.

Виды информации



Выполнение заданий по программированию на экзамене по информатике допускается на следующих языках программирования: C++, Java, C#, Pascal, Python, школьный алгоритмический язык.



Минимальная единица языка, имеющая самостоятельный смысл, называется **лексемой**. Для написания лексемы применяется алфавит. Каждый язык программирования использует свой алфавит.



Алгоритмический язык	Pascal
<ul style="list-style-type: none"> • Прописные и строчные буквы русского алфавита. • 26 латинских строчных и 26 латинских прописных букв. • Цифры 0—9. • Специальные символы: знаки арифметических операций, знаки препинания, скобки и т. п. • Неотображаемые символы: пробел, табуляция, переход на новую строку 	<ul style="list-style-type: none"> • Прописные и строчные буквы латинского алфавита (A...Z, a...z). • Цифры 0—9. • Специальные символы: знаки арифметических операций, знаки препинания, скобки и т. п. • Неотображаемые символы: пробел, табуляция, переход на новую строку. • Комбинации специальных символов, которые нельзя разделять пробелами, если они используются как знаки операций: «:=», «.», «<>», «<=», «>=», «{»
Python	C++
<ul style="list-style-type: none"> • Прописные и строчные буквы латинского алфавита (A...Z, a...z). • Цифры 0—9. • Знаки арифметических операций, знаки препинания, скобки и т. п. • Неотображаемые символы: пробел, табуляция, переход на новую строку 	<ul style="list-style-type: none"> • Прописные и строчные буквы латинского алфавита (A...Z, a...z). • Цифры 0—9. • Специальные символы: знаки арифметических операций, знаки препинания, скобки и т. п. • Неотображаемые символы: пробел, табуляция, переход на новую строку



Во всех названных языках в комментариях, строках и символьных константах могут использоваться также другие литеры (например, буквы русского алфавита).



В Python и C++ имеются различия между прописными и строчными буквами алфавита, например `chislo`, `CHISLO`, `Chislo` — разные имена.

Основные служебные слова



Служебные слова — это идентификаторы, зарезервированные в языке для специального использования: обозначения начала и конца алгоритма, применения ветвлений и т. д. Они являются ключевыми словами языка, которые используются для организации некоторого процесса, дающего программе возможность корректно работать, либо обозначают действия, которые определены заранее.



Программа всегда расценивает служебные слова как призыв к действию, поэтому нельзя использовать их в имени программы, переменной или константы. Каждый язык использует свой набор служебных слов.

Алгоритмический язык	Pascal	C++	Python
алг — алгоритм	and — логическое И	auto — автоматический	False — ложь
сим — символьный тип	mod — остаток от деления	break — прервать	True — правда
лит — литерный тип	array — массив	case — вариант	and — логическое И
цел — целый тип	not — логическое НЕ	char — символьный тип	break — выход из цикла
вещ — вещественный тип	begin — начало блока	const — константа	class — пользовательский тип, состоящий из методов и атрибутов
лог — логический тип	or — логическое ИЛИ	continue — продолжить	default — прочие
длин — длина	case — вариант	default — прочие	do — цикл
рез — результат	of — из	do — цикл	double — двойной
нач — начало	const — константа	double — двойной	else — иначе
кон — конец	object — объект	define — определить	else — если
таб — таблица, массив	div — деление нацело	else — определить	def — определение функции
нц — начало цикла	procedure — процедура	elif — иначе если	for — цикл for
кц — конец цикла	go to — переход на...	float — плавающий тип	if — если

Алгоритмический язык	Pascal	C++	Python
<p>дано надо для да нет и ввод вывод всё или если то иначе до пока</p>	<p>program — программа do — выполнять repeat — повторять string — строка else — иначе then — то end — конец блока file — файл type — тип for — для until — до function — функция if — если var — переменная interrupt — прерывание while — пока with — с</p>	<p>for — для goto — переход if — если int — целочисленный тип include — включать long — длинный return — возвращать short — короткий not — нет static — статический struct — структура switch — выбор typedef — определение типа union — совмещение void — пустой while — пока</p>	<p>continue — переход на следующую итерацию цикла elif — в противном случае, если except — перехватить исключение from — импорт нескольких функций из модуля global — позволяет сделать значение переменной, присвоенное ей внутри функции, доступным и за пределами этой функции import — импорт модуля in — проверка на вхождение is — наличие ссылки двух объектов на одно и то же место в памяти not — логическое НЕ or — логическое ИЛИ return — вернуть результат try — выполнить инструкции, перехватывая исключения while — цикл</p>

Разделители языка



Разделители — специальные символы, которые используются в служебных целях от организации текста программы до определения указаний компилятору языка.

Алгоритмический язык	Pascal
<p>Пробел — разделяет слова.</p> <p>Табуляция — позволяет оформить структуру программы.</p> <p>, (запятая) — разделение величин в строке, перечисление элементов.</p> <p>: (двоеточие) — используется в спецсимволах (например, оператор присваивания).</p> <p>() (скобки) — используются для ввода параметров команды.</p> <p>Перевод строки (enter) — переводит на новую строку</p>	<p>Пробел — разделяет слова.</p> <p>. (точка) — обозначение конца программы.</p> <p>; (точка с запятой) — разделяет описание и операторы.</p> <p>Табуляция — позволяет оформить структуру программы.</p> <p>, (запятая) — разделение величин в строке, перечисление элементов.</p> <p>: (двоеточие) — используется в спецсимволах (например, оператор присваивания).</p> <p>() (скобки) — используются для ввода параметров команды.</p> <p>Перевод строки (enter) — переводит на новую строку</p>
C++	
<p>Пробел — разделяет слова.</p> <p>; (точка с запятой) — признак конца инструкции (команды).</p> <p>: (двоеточие) — используется для указания базового класса при наследовании классов, задаёт список инициализации для конструктора.</p> <p>, (запятая) — разделение величин в строке, перечисление элементов.</p> <p>() (скобки) — используются для ввода параметров команды.</p> <p>Табуляция — отступы в блоках.</p> <p>'\n' — символ перехода на новую строку</p>	

Python

Пробел — разделение слов или отступы в блоках.

Табуляция — отступы в блоках.

: (двоеточие) — вход в блок из операторов.

, (запятая) — разделение величин в строке, перечисление элементов.

() (скобки) — используются для ввода параметров команды.

;(точка с запятой) — позволяет вводить несколько инструкций на одной строке.

sep=' ' — разделитель между аргументами функции print (). В пробел между кавычками можно поставить любой разделитель: тире, пробел, запятую

Структура программы



Структура программы — искусственно выделенные программистом взаимодействующие части программы.

Алгоритмический язык	Pascal
<p>алг <название алгоритма> (аргумент и результат)</p> <p>дано условия применимости алгоритма</p> <p>надо цель выполнения алгоритма</p> <p>нач описание промежуточных величин</p> <p>последовательность команд (тело программы)</p> <p>кон</p> <p>дано и надо используем при необходимости для понимания того, что нужно сделать в задаче. После знака можно писать комментарии, и они не будут отображаться</p>	<pre> Program <название программы>; const ...; (константы) var...; (переменные) (процедуры и функции) begin ... (основная программа — тело программы) end. {} или (* *) — комментарии </pre>

C++	Python
<pre>#include <iostream> // подключение за- головочного файла с помощью директивы include using namespace std; // использование пространства имён std int main() // функция main, внутри ско- бок идёт набор инструкций { тело программы return 0; } после двойной наклонной строки // можно писать любые однострочные комментарии /* многострочный комментарий */</pre>	<p>Любая Python-программа состоит из последовательности лексем (допустимых символов), записанных в определённом порядке и по определённым правилам.</p> <p>Этот язык не содержит операторных скобок (<code>begin...end</code> в Pascal или <code>{...}</code> в C++), вместо этого блоки выделяются отступами — пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием.</p> <p>Количество пробелов в отступах произвольно, однако по договорённости равняется четырём пробелам. При этом отступ всего блока должен быть одинаковым.</p> <p>Несколько строк кода с одинаковым отступом формируют отдельный блок кода. Благодаря такой системе значительно повышается читаемость кода и прививается привычка писать понятно и структурированно.</p> <p># — комментарий в программе идёт до конца строки</p>

Идентификаторы



Посредством **идентификаторов** обозначают имена переменных, констант, процедур и функций. Идентификатор представляет собой последовательность букв, цифр и символов подчёркивания. Выбирая идентификатор для имени, следует учитывать два условия. Во-первых, имя должно быть содержательным, т. е. отражать назначение идентификатора, что де-

лает программу более читабельной. Во-вторых, каждый язык программирования накладывает на имена ряд ограничений. Правильно подобранные идентификаторы упрощают программистам и аналитикам понимание порядка работы программы, а также помогают исправить или расширить исходный код, чтобы использовать его для текущей или новой задачи.

Алгоритмический язык

- +** Можно использовать:
- латинские и русские буквы, заглавные и строчные буквы не различаются;
 - цифры, но не в начале имени (имя не может начинаться с цифры);
 - символ нижнего подчёркивания `_`.
- Нельзя использовать:
- пробелы, скобки, знаки `+`, `-`, `?` и др.;
 - служебные слова

C++

- +** Можно использовать:
- латинские буквы;
 - цифры;
 - символы тире `—` и нижнее подчёркивание `_`;
 - первый символом должна быть буква.
- Нельзя использовать:
- символы `!`, `@`, `&`, `#`, пробел и др.;
 - служебные слова;
 - имена длиной больше 255 символов.

В C++ часто используется служебное слово `define` для задания символических имён констант. Так, строка `#define PI 3.1415` (где `#define` — служебное слово, `PI` — имя константы, `3.1415` — её значение) задаёт символическое имя `PI` для константы `3.1415`. После этого имя `PI` можно использовать вместо числового значения

Pascal

- +** Можно использовать:
- латинские буквы, заглавные и строчные буквы не различаются;
 - цифры, но не в начале имени.
- Нельзя использовать:
- русские буквы, пробелы, скобки, знаки `+`, `-`, `?` и др.;
 - служебные слова

Python

- +** Можно использовать:
- латинские буквы;
 - цифры, но не в начале имени.
- Слова могут быть разделены подчёркиванием для улучшения читаемости.
- Нельзя использовать:
- русские буквы, пробелы, скобки, знаки `+`, `-`, `?` и др.;
 - служебные слова;
 - латинские буквы `l` (строчная «эль»), `I` (прописная «и»), `O` (прописная «о») в качестве имён переменных, состоящих из одного символа

Переменные и константы



Переменные — величины, имеющие имя, тип и значение. Значение переменной может меняться во время работы программы.

Константы — данные, не изменяющиеся в процессе решения задачи.



Все программы работают с данными. Данные — это значения, которые используются в работе программы: строки, числа, ссылки, символы и т. п. Все они хранятся в оперативной памяти. Для каждого значения выделяется отдельная ячейка, в которой может храниться только один вид данных. На языке программирования даётся команда компьютеру: создать константу или переменную, после чего он выделяет соответствующую ячейку памяти.

Атрибут	Характеристика
Имя	Имя любой переменной уникально и не меняется в процессе выполнения программы
Тип	Определяет множество допустимых значений переменной и множество применимых к ней операций, объём занимаемой памяти и способ представления в памяти
Значение	Характеристика, которая может меняться многократно в ходе исполнения алгоритма

Основные типы переменных и констант

Тип	Характеристика
Числовой	Переменные и константы записываются в виде конкретного числа, они бывают двух типов: целые и вещественные
Целый	Положительные и отрицательные целые числа. ✓ -5897, +39, 20
Вещественный	Числа, имеющие целую и дробную части. Существует две формы записи вещественных констант: основная и экспоненциальная. К основной форме относятся записи вида: 1.02, -7.149. Экспоненциальная форма записи: $mE \pm p$, где m — мантисса (число в основной форме), E — основание 10, p — порядок числа. Например: $2.8E - 05$, $-6.9E + 07$. Чтобы перейти от экспоненциальной формы к основной, нужно m умножить на 10 в степени p (порядок). ✓ $2.4E^{-6} = 2.4 \cdot 10^{-6} = 0.0000024$, $-2.9E + 07 = 2.9 \cdot 10^7 = 29000000$
Строковый	Набор любых символов длиной, не превышающей 255 символов. Строковая константа в языках программирования заключается в кавычки (двойные или одинарные в зависимости от языка). ✓ «Решений нет», «котики», «cat_2»
Символьный	Один символ. Заключается в кавычки
Дата/время	Тип записи даты или времени. ✓ 12.12.2020
Логический	Предполагает вывод значения «истина» или «ложь» в разных интерпретациях (0 или 1, true или false)

Тип переменных и констант	Примеры
Алгоритмический язык	
<p>цел — целый вещ — вещественный сим — символьный, 1 символ лит — литерный (строковый) лог — логический</p>	<p>✓ цел i, mn, summ</p> <p>В программу внесли 3 переменные целочисленного типа с именами i, mn, summ.</p> <p>✓ n2:=34</p> <p>n2 присвоено значение 34</p>
Pascal	
<p>integer — целые (-32768...32767) longint — большое целое real — вещественный char — символьный, 1 символ string — строковый boolean — логический</p>	<p>В программу внесены несколько переменных. Первая строка: объявление целочисленных переменных a и b; вторая строка: объявление вещественной переменной H; третья строка: объявление строковых переменных s1, s2.</p> <p>✓ var a, b: integer; H: real; s1, s2: string</p> <p>Переменным присвоены их значения соответствующего типа: переменной a2 — значение целого типа, переменной r — значение строкового типа, переменной n — значение логического типа.</p> <p>✓ a2=45; r='Даша'; n=True;</p>

Тип переменных и констант	Примеры
C++	
<p>char — символьный short — целые (-32768...32767) int — целое число long — длинное целое число float — вещественное число bool — логическое значение</p>	<p>✓ <code>const int N=15;</code> Объявление целочисленной константы N со значением 15.</p> <p>✓ <code>const char simb='f';</code> Константа символьного типа, в котором хранится символ f. Записывается в одинарных кавычках.</p> <p>✓ <code>int x, y;</code> Переменные x и y целочисленного типа.</p> <p>✓ <code>char str[10];</code> Пример объявления строки, где str — имя переменной, 10 — размер массива, 9 символов и нуль-терминатор. Строки в C++ представляются как массивы элементов типа char, заканчивающиеся нуль-терминатором. \0 — символ нуль-терминатора. Символьные строки состоят из набора символьных констант, заключённых в двойные кавычки. При объявлении строкового массива необходимо учитывать наличие в конце строки нуль-терминатора и отводить под него дополнительный байт</p>
Python	
<p>int — целый float — вещественный str — строковый bool — логический</p>	<p>Не нужно указывать тип переменной во время присвоения значения. Python — язык с динамической типизацией, поэтому на этапе присвоения интерпретатор «понимает», что переменная hello имеет тип str (<i>string</i> — «строка»). При этом далее можно присвоить этой переменной значение 123, и её тип автоматически сменится на int (<i>integer</i> — «число»)</p>

```
✓ a, b, c = 1, 'hello', 12
```

Можно в одной строке присваивать разным переменным разные значения. Данная выше запись равнозначна следующей:

```
a = 1
```

```
b = 'hello'
```

```
c = 12
```



Если число заключается в кавычки, например «123», получается строковая константа. В таком случае с ней нельзя производить математические операции.

Функции



Функция — самостоятельная единица программы, которая спроектирована для реализации конкретной подзадачи.



Функция является подпрограммой, которая содержится в основной программе или создаётся отдельно (в библиотеке). Каждая функция выполняет в программе определённые действия. При программировании часто возникает необходимость вычислить значение функции (например, логарифм числа, корень квадратный и т. п.). Для обращения к функции (вычисления значения) необходимо указать её имя и в круглых скобках записать аргумент.

Название оператора	Форма записи функции	
	Алгоритмический язык	Pascal
Модуль числа	<code>abs(x)</code>	<code>abs(x)</code>
Корень квадратный	<code>sqr(x)</code>	<code>sqrt(x)</code>
Деление нацело	<code>div(x, y)</code>	<code>x div y</code>
Остаток от деления нацело	<code>mod(x, y)</code>	<code>x mod y</code>
Синус	<code>sin(x)</code>	<code>sin(x)</code>
Косинус	<code>cos(x)</code>	<code>cos(x)</code>
Тангенс	<code>tg(x)</code>	<code>tan(x)</code>
Натуральный логарифм	<code>log(x)</code>	<code>ln(x)</code>
Показательная функция	<code>exp(x)</code>	<code>exp(x)</code>
Случайное число	<code>rnd(x)</code>	<code>random(N)</code> — генерирует случайное число 0—N

Название оператора	Форма записи функции	
	C++	
Модуль числа	<code>abs(x)</code> — при подключении библиотеки <code>cmath</code> <code>fabs(x)</code> Для подключения библиотеки: <code>#include <cmath></code>	

Корень квадратный	<code>sqrt(x)</code>
Деление нацело	<code>/</code> При делении целых чисел остаток отбрасывается
Остаток от деления нацело	<code>x%y</code>
Синус	<code>sin(x)</code>
Косинус	<code>cos(x)</code>
Тангенс	<code>tan(x)</code>
Натуральный логарифм	<code>log(x)</code>
Показательная функция	<code>exp(x)</code>
Случайное число	<code>random(z)</code> — генерирует случайное число от 0 до $z-1$

Название оператора	Форма записи функции
	Python
Модуль числа	<code>abs(x)</code>
Корень квадратный	<code>sqrt(x)</code>
Деление нацело	<code>x//y</code>
Остаток от деления нацело	<code>x%y</code>
Синус	<code>sin(x)</code>

Название оператора	Форма записи функции
	Python
Косинус	<code>cos(x)</code>
Тангенс	<code>tan(x)</code>
Натуральный логарифм	<code>log(x)</code>
Показательная функция	<code>exp(x)</code>
Случайное число	<code>randint(a,b)</code> — генерирует случайное число от a до b ; <code>uniform(a,b)</code> — вещественное число от a до b

Операторы и операции

Определение	Характеристика
Операнд	В простейшем случае является константой или идентификатором. В общем случае каждый операнд выражения также представляет собой выражение, имеющее некоторое значение
Операции	Определяют действия, выполняемые над операндами, возвращают некоторое значение
Оператор	Некоторая конструкция, присущая данному конкретному языку, изменяющая состояние памяти компьютера, но ничего не возвращающая



Важно различать понятия «оператор» и «операция»: операция возвращает значение, а оператор — нет.

✓ $a + c$ — операция сложения двух операндов a и c , между которыми стоит оператор сложения «+».

Типы простейших операций

Арифметические операции: сложение, вычитание, деление и др.

Логические операции: логическое И, логическое ИЛИ, логическое НЕ и др.

Операции отношения: меньше, больше, меньше или равно, больше или равно, равно, не равно

Операция конкатенации (соединения) символьных значений друг с другом

Присваивание переменной значения

Индексация массива

Переход по ссылке на объект

Ввод и вывод



Во всех изучаемых языках программирования арифметические операции записываются одинаково.

Сложение	$A + B$
Вычитание	$A - B$
Умножение	$A * B$
Деление	A / B



Важно помнить, что при делении целого числа на целое можно получить дробное число. Это следует учитывать при выборе типа переменной, которую мы получаем. Например, нужно поделить целочисленную переменную A на целочисленную переменную B . При выполнении деления $10 : 3$ в ответе не получится целое число. Некоторые языки выдадут ошибку, некоторые возьмут от числа целую часть и выведут в ответ. Этот ответ будет неправильным, поэтому необходимо изменить тип переменной на вещественный или другой, подходящий к данным в задаче.

Операторы присваивания и ввода/вывода

Название оператора	Форма записи	
	Алгоритмический язык	Pascal
Присваивание	<code>:=</code>	<code>:=</code>
Ввод	<code>ВВОД</code>	<code>ReadLn</code> <code>Read</code>
Вывод	<code>ВЫВОД</code>	<code>WriteLn</code> <code>Write</code>



В ходе использования оператора присваивания при вводе нового значения предыдущее значение стирается, поэтому, чтобы не потерять вводимое число, его нужно сохранить в другую ячейку памяти для дальнейшего использования.

Название оператора	Форма записи	
	C++	Python
Присваивание	<code>=</code>	<code>=</code>
Ввод	<code>cin >></code>	<code>input</code>
Вывод	<code>cout <<</code>	<code>print</code>



Запись `A:=3` (или `A=3` в зависимости от языка) означает «переменной `A` присвоить значение, равное 3».

Операторы условия



В зависимости от ситуации процесс может пойти по одной или другой ветви, а может быть усложнён несколькими условиями. Чтобы написать в программе определённые условные конструкции, необходимо использовать английские слова: `if` — «если», `then` — «то», `else` — «иначе».

Алгоритмический язык	Pascal
<p>Сокращённый вариант: если условие то действия всё</p> <p>Полный вариант: если условие то операторы иначе операторы всё</p>	<p>Сокращённый вариант: if условие then оператор;</p> <p>Полный вариант: if условие then оператор else оператор;</p>
C++	Python
<p>Сокращённый вариант: if (условие) оператор;</p> <p>Полный вариант: if (условие){ блок операций; } else { блок операций; }</p>	<p>Полный вариант: if условие 1: операция elif условие 2: операции else: условие 3</p> <p>При необходимости можно использовать конструкцию elif.</p> <p>Замена полного варианта на сокращённый: if X: A = Y else: A = Z</p> <p>МОЖНО заменить на A = Y if X else Z</p>

Операторы отношения



Для записи условий часто используются операторы отношения, например при сравнении двух чисел.

Оператор	Форма записи			
	Алгоритмический язык	Pascal	C++	Python
больше	>	>	>	>
меньше	<	<	<	<
больше или равно	>=	>=	>=	>=
меньше или равно	<=	<=	<=	<=
равенство	=	=	==	==
не равно	<>	<>	!=	!=

Операторы цикла



При решении задач часто приходится многократно повторять одно и то же действие в связи с вводом различных значений параметров. Такой вычислительный процесс называется **циклическим**, а многократно повторяющиеся участки этого процесса — **циклами**.

Цикл с предусловием	Цикл с постусловием	Цикл с параметром
Алгоритмический язык		
нц пока условие действия кц	нц действия до условия кц	нц для параметр от а до b шаг h действия кц
Pascal		
while выражение do begin Внутренний оператор; end;	repeat оператор; until логическое выражение;	for управляющая переменная:=исходное значение to конечное значение do оператор
C++		
while (Условие) { блок операций; }	do { блок операций; } while (Условие);	for (Инициализация; Условие; Модификация) { блок операций; }
Python		
while условие: операторы else: операторы или без else: while условие: операторы	В этом языке программирования циклы с постусловием отсутствуют, но решение можно записать в таком виде: while True: if not условие: break	for <переменная> in <список> : операторы 1...N

ЗАДАЧИ НА ЗАПИСЬ И АНАЛИЗ АЛГОРИТМОВ



Изучив основные элементы языков программирования, можно приступить к решению задач.

Алгоритмы обработки массивов



При работе с массивами необходимо помнить следующее:

- массив — это структура данных, представляющая собой совокупность элементов одного типа;
- при объявлении массива указывается количество элементов массива;
- доступ к элементу массива осуществляется путём указания индекса (номера) элемента массива.



Каждому элементу массива присваивается его порядковый номер, который называется **индексом**. Количество индексов определяет размерность массива. Массивы бывают одномерными и многомерными.



Типовые алгоритмы обработки массивов:

- ввод и вывод значений;
- подсчёт минимума или максимума;
- суммирование элементов массива;
- поиск заданного элемента;
- замена заданного элемента;
- формирование нового массива из других заданных;
- сортировка и др.

Инициализация массива

Алгоритмический язык	Pascal
<p>Объявление массивов <u>Массив целых чисел:</u></p> <div style="margin-left: 20px;"> <p>Тип элемента ↓</p> <p>Имя ↓</p> <p>Начальный индекс ↓</p> <p>Конечный индекс ↓</p> <p>целтаб A[1:5]</p> </div> <p>цел N=5</p> <p><u>Размер через константу:</u> целтаб A [1:N]</p> <p><u>Ввод с клавиатуры:</u> нц для I от 1 до N вывод " A[" , i , "]" = " ввод A [i]</p> <p>кц</p> <p><u>Обработка каждого элемента:</u> нц для I от 1 до N A[i]:= A+1</p> <p>кц</p> <p><u>Вывод на экран:</u> вывод " Массив A " , нс</p> <p>нц для I от 1 до N вывод A [i] , " "</p> <p>кц</p>	<p>Объявление массива:</p> <div style="margin-left: 20px;"> <p>Имя ↓</p> <p>Начальный индекс ↓</p> <p>Конечный индекс ↓</p> <p>Тип элемента ↓</p> <p>var A : array [1..5] of integer;</p> </div> <p><u>Через константу:</u> const N=5; var A : array [1..N] of integer;</p> <p><u>Ввод с клавиатуры:</u> for i:=1 to N do begin write ('a[' , i , ']=') ; read (a[i]); end;</p> <p><u>Поэлементные операции:</u> for i:=1 to N do a[i]:=a[i]+1;</p> <p><u>Вывод на экран:</u> writeln ('Массив A :'); for i:=1 to N do write (a[i] :4);</p>

Массивы других типов:

вещтаб X[1:14], Y[1:12] | вещественный

X[1]:=4.32

симтаб S[1:10] | символьный

S[1]:= " d "

логтаб L[1:5] | логический

L[1]:=нет | да или нет

C++Объявление массива:

	Тип элемента		Имя		Размер массива
	↓		↓		↓
int	A	[5];			

```
const int N=5;
int A[N];
```

Ввод с клавиатуры:

```
for (int i=0; i<N; ++i){
int x
cin>>x;
A[i] = x;
}
```

Поэлементные операции:

```
for(int i=0; i<N; ++i) A[i]=A[i]*2;
```

Вывод на экран:

```
for (int i=0; i<N; ++i){
count<<A[i]<< " " ;
```

Python**Ввод массива с клавиатуры**Объявление массива:

```
N=10 # размер массива
A=[0]*N # выделили память для массива
```

Ввод по одному элементу в строке:

```
for i in range (N):
A[i]=int(input() )
```

Другой способ:

```
A = [int (input () )
for i in range (N)]
```

Вывод массива на экранКак список:

```
print(A)
```

В строчку через пробел:

```
for i in range (N:
print (A[i], end= " ")
```

Вариант:

```
print (1, 2, 3, 4, 5)
print (*A)
```

Решение задач на действия над массивами



Основным действием, производимым над массивом, является поиск, который лежит в основе множества других возможных манипуляций с массивами. В зависимости от упорядоченности массива поиск выполняется различными способами. Если он не упорядочен, то для поиска определённого элемента необходимо просмотреть каждое значение, имеющееся в массиве. Такой вид поиска называется **линейным**. Если же массив упорядочен, то используют метод половинного деления, или **бинарный**. Перед поиском для более быстрого его завершения производится сортировка — действие, которое приводит к изменению положения элементов в заданном массиве согласно поставленным условиям.

Сортировка обмена	Сортировка прямым включением (вставкой)	Сортировка выбором
Алгоритмический язык		
Данный способ сортировки предусматривает сравнение элемента с соседними и перемену их мест в случае необходимости. Подобные перемещения элементов массива относительно друг друга производятся до тех пор, пока массив не будет упорядочен. Элементы в массиве передвигаются на положенное место подобно пузырькам, которые поднимаются на высоту согласно собственному размеру, поэтому данный метод часто называют пузырьковым	В данном случае все элементы делятся на два массива: один — уже отсортированный, второй — произвольный. Из неотсортированного массива берётся элемент и вставляется на определённое место в отсортированный, т. е. программа ищет позицию необходимого элемента и вставляет его. Следует помнить, что это приводит к сдвигу всех остальных элементов	Данный способ заключается в поиске самого большого (малого) элемента, после чего он перемещается в начало массива. Подобные перемещения происходят до тех пор, пока сортировка не будет выполнена полностью

Сортировка обмена	Сортировка прямым включением (вставкой)	Сортировка выбором
<p>✓ алг Обменная_сортировка (арг цел n, арг рез вещь таб A[1:n])</p> <p>дано A - массив размерности n</p> <p>надо упорядочить массив по возрастанию</p> <p>нач</p> <p>цел i, j</p> <p>вещ Tmp</p> <p>нц для i от 2 до n</p> <p>нц для j от n до 1</p> <p>если A[j]<A[j-1]</p> <p>то Tmp :=A[j]; A[j] :=A[j-1]; A[j-1] :=Tmp</p> <p>все</p> <p>кц</p> <p>кц</p> <p>кон</p>	<p>✓ алг Сортировка_вставкой</p> <p>арг цел n, арг рез вещь таб A[1:n])</p> <p>дано A - массив размерности n</p> <p>надо упорядочить массив по возрастанию</p> <p>нач</p> <p>цел i, j</p> <p>вещ Tmp</p> <p>нц для i от 2 до n</p> <p>Tmp :=A[j]; j :=i-1;</p> <p>нц пока j>= 1 и A[j]>Tmp</p> <p>A[j+1] :=A[j]</p> <p>j :=i-1</p> <p>кц</p> <p>A[j+1] :=Tmp</p> <p>кц</p> <p>кон</p>	<p>✓ алг Сортировка_выбором (арг цел n, арг рез вещь таб A[1:n])</p> <p>дано A - массив размерности n</p> <p>надо упорядочить массив по возрастанию</p> <p>нач</p> <p>цел i, k</p> <p>вещ Min</p> <p>нц для i от 1 до n-1</p> <p>Min :=A[j]; k :=i</p> <p>нц для j от i+1 до n</p> <p>если Min > A[j]</p> <p>то Min :=A[j]; k :=j</p> <p>все</p> <p>кц</p> <p>A[k] :=A[i]; A[i] :=Min</p> <p>кц</p> <p>кон</p>

Исправление ошибок в программе



При создании программы часто возникают ошибки в написании служебного слова, постановке знаков и т. п. Существует определённый порядок выявления таких недостатков.

Характерные ошибки

Ошибка	Характеристика
Неправильная постановка задачи	Верное решение неверно сформулированной задачи
Неверный алгоритм	Выбор алгоритма, приводящего к неточному, неэффективному решению задачи
Ошибки анализа	Неполный учёт ситуаций, которые могут возникнуть, логические ошибки
Семантические ошибки	Неправильный порядок выполнения операций
Синтаксические ошибки	Нарушение правил, определяемых языком программирования
Ошибки при выполнении операций	Слишком большое число (переполнение), деление на ноль, извлечение квадратного корня из отрицательного числа и т. п.
Ошибки в данных	Неправильное определение возможного диапазона изменения данных
Ошибки ввода/вывода	Неверное считывание входных данных, задание форматов, отсутствие некоторых данных
Опечатки	Неверное указание близких по написанию символов, например путаницу могут вызвать цифра 1 и буква l, цифра 0 и прописная буква O, прописная латинская буква l (И) и строчная l (эль)



Существует три аспекта проверки программы:

- правильность;
- эффективность реализации;
- вычислительная сложность.

Поиск ошибок в программе

Порядок поиска ошибок		
1	Просмотр	Текст программы просматривается на предмет обнаружения ошибок и расхождений с алгоритмом. Для этого следует просмотреть организацию всех циклов, условия в условных операторах, аргументы в обращениях к подпрограммам, синтаксис, правильность написания служебных слов и т. д.
2	Проверка	На этом этапе необходимо восстановить вычислительный процесс по тексту программы. Чтобы выявить, в каких строках программы допущена ошибка, надо определить задачу, которую должна решать программа
3	Прокрутка	Для выполнения этой операции нужно задать исходные данные и произвести над ними необходимые вычисления. Прокрутка — трудоёмкий процесс, поэтому её следует применять только для контроля логически сложных участков программы
4	Отладка	Процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере
5	Тестирование	Испытание, проверка правильности работы программы в целом или её составных частей

ИНФОРМАЦИОННАЯ ДЕЯТЕЛЬНОСТЬ ЧЕЛОВЕКА

ЭТАПЫ РАЗВИТИЯ ИНФОРМАЦИОННОГО ОБЩЕСТВА

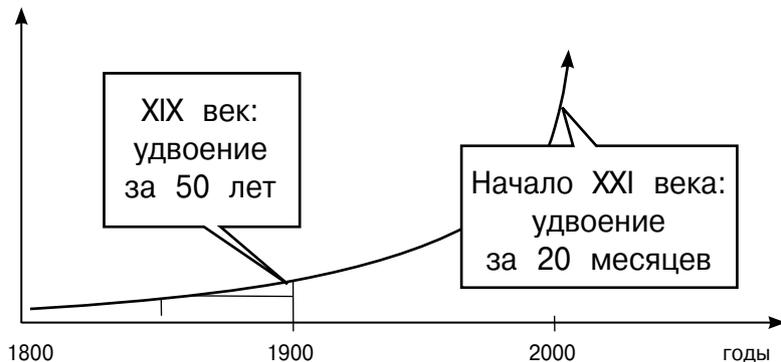


В развитии человечества можно выделить четыре этапа, названные информационными революциями, которые внесли большие изменения в развитие информационной деятельности человека.

Этапы развития информационной деятельности человека	
1	Изобретение письменности (середина IV тыс. до н. э.)
2	Изобретение книгопечатания (середина XV в.)
3	Изобретение электричества (XIX в.)
4	Изобретение микропроцессорной технологии и персональных компьютеров (1970-е гг.)



Появление и развитие новых знаний увеличивается в геометрической прогрессии. Если первое удвоение общего количества знаний на планете произошло за период от рубежа нашей эры до 1750 г., то второе удвоение случилось уже за 150 лет, к началу XX столетия, третье — за 50 лет — к 1950 г. В дальнейшем объёмы знаний удваивались ещё более стремительными темпами: до 1970 г. — на протяжении 10 лет, после 1970 г. — каждые 5 лет, а с 1991 г. — ежегодно. Данный процесс можно наглядно представить схемой, на которой стрелка обозначает удвоение количества знаний.



XIX век:
удвоение
за 50 лет

Начало XXI века:
удвоение
за 20 месяцев



К современным техническим средствам работы с информацией относятся не только компьютеры, но и другие устройства, обеспечивающие её передачу, дальнейшую обработку и хранение:

- сетевое оборудование: модемы, кабели, сетевые адаптеры;
- аналого-цифровые и цифро-аналоговые преобразователи;
- цифровые фото- и видеокамеры, цифровые диктофоны;
- записывающие устройства (CD-RW, DVD-RW, флеш-карты и др.);
- полиграфическое оборудование;
- цифровые музыкальные студии;

- медицинское оборудование для УЗИ и томографии;
- ТВ-тюнеры для подачи телевизионного сигнала в компьютер;
- плоттеры, сканеры, принтеры;
- мультимедийные проекторы;
- флеш-память, используемая также в плеерах и фотоаппаратах;
- мобильные телефоны, планшеты, смартфоны;
- электронные книги;
- мощные вычислительные системы для решения сложных научно-технических и оборонных задач, обработки огромных баз данных, работы телекоммуникационных сетей (Интернет).

ПРОФЕССИОНАЛЬНАЯ ИНФОРМАЦИОННАЯ ДЕЯТЕЛЬНОСТЬ

Понятие	Характеристика
Информационная деятельность человека	Любая деятельность, связанная с процессами получения, преобразования, накопления и передачи информации
Профессиональная деятельность	Вид трудовой деятельности человека, владеющего комплексом специальных теоретических знаний и практических навыков, который является основным источником дохода
Цель профессиональной деятельности	Предполагаемый результат, обеспечивающий развитие личности и её жизнедеятельность

Цели информационной деятельности человека

Удовлетворение спроса на информацию и потребности в ней

Реализация информационного потенциала человека, предприятия или страны

Налаживание сотрудничества с зарубежными информационными системами

Задачи, решаемые в процессе информационной деятельности

Введение в оборот всех информационных ресурсов страны, повышение качества информационных услуг

Пробуждение интереса к информации, повышение культуры её потребления у всех категорий населения

Создание материальной базы для технического оснащения научно-информационной деятельности, обеспечение защиты информационных работников, прав собственности на информационную продукцию и производителей

Ускорение процесса формирования системы баз и банков данных

Применение технических средств и информационных ресурсов в профессиональной деятельности

Профессии	Технические средства	Информационные ресурсы
Средства массовой информации		
Журналисты	Телевидение, радио, телекоммуникации, компьютеры, компьютерные сети	Интернет, электронная почта, библиотеки, архивы
Сфера связи: почта, телеграф, телефония		
Почтовые служащие, инженеры	Телеграф, телефон, компьютерные сети	Базы данных
Наука		
Учёные	Телекоммуникации, компьютеры, компьютерные сети	Библиотеки, архивы, базы данных, экспертные системы, Интернет
Техника		
Инженеры	Телекоммуникации, компьютеры, компьютерные сети	Системы автоматизированного проектирования (САПР), библиотеки, патенты, базы данных, экспертные системы, Интернет
Медицина		
Медицинские работники	Информационные системы, компьютеры, компьютерные сети	Базы данных, экспертные системы, архивы, Интернет
Финансы и экономика		
Бухгалтеры, экономисты	Информационные системы, телекоммуникации, компьютеры, компьютерные сети	Базы данных, экспертные системы, архивы, Интернет

Управление

Менеджеры	Информационные системы, телекоммуникации, компьютеры, компьютерные сети	Базы данных, экспертные системы
Образование		
Преподаватели	Информационные системы, телекоммуникации, компьютеры, компьютерные сети	Библиотеки, Интернет
Искусство		
Писатели, художники, музыканты, дизайнеры	Компьютеры, устройства ввода и вывода информации, системы мультимедиа, аудио- и видеосистемы, телекоммуникации, компьютерные сети	Галереи, библиотеки, музеи, Интернет

Системное администрирование



С появлением компьютеров и расширением их применения возникают новые профессии. Одна из них — администратор сети — приобретает всё большую популярность, т. к. системное администрирование становится ключевой областью в постоянно растущем числе организаций. Представитель данной профессии должен разбираться в своей системе, знать её пользователей и род их занятий.



Администратор сети (системный администратор) — специалист, в круг обязанностей которого входит выполнение следующих основных функций: инсталляция операционной системы и прикладных программ, описание информационной среды, настройка операционной среды, настройка сетевой печати, мониторинг сети и управление сетевыми ресурсами, архивирование и восстановление данных сети, обеспечение безопасности.

Функция	Характеристика
Инсталляция	Процесс установки программного обеспечения на компьютер, во время которого администратор отвечает на ряд вопросов в диалоговом режиме
Описание информационной среды	<ul style="list-style-type: none"> • Создание каталогов пользователей. • Назначение прав доступа по отношению к каталогам и файлам. • Назначение атрибутов каталогам и файлам
Настройка операционной среды	<ul style="list-style-type: none"> • Инсталляция конфигурационных файлов рабочих станций. • Разработка системных и пользовательских процедур подключения к сети. • Управление рабочими станциями
Настройка сетевой печати	<p>Описание:</p> <ul style="list-style-type: none"> • объектов печати (очередей, принтеров, серверов печати); • форм печати (размера бумаги); • конфигураций заданий на печать
Мониторинг сети	<p>Оценка характеристик производительности и надёжности сети. С помощью специальных программных пакетов можно реализовать:</p> <ul style="list-style-type: none"> • мониторинг файловых серверов и сегментов сети; • управление файловым сервером, концентраторами, рабочими станциями

Архивирование и восстановление данных сети	Предполагает выполнение ряда условий. Для архивации большого объёма данных компьютер должен иметь определённые технические характеристики и высокоскоростное (100 Мбит/с и выше) подключение к локальной сети, иначе процесс архивации может затянуться надолго
Обеспечение безопасности	Одна из основных проблем в современном мире. Системный администратор должен знать основы информационной безопасности, которая заключается в защите информации, находящейся на персональных компьютерах и серверах локальной сети предприятия, от несанкционированного доступа, умышленного искажения и повреждения. Администратор также отвечает за антивирусную защиту локальной сети и отдельных компьютеров

Интернет и безопасность его использования



Недостатки бесконтрольного использования Интернета в организациях:

- отвлечение сотрудников от рабочего процесса при использовании Интернета в личных целях;
- проникновение вредоносных программ в сеть предприятия;
- перерасход средств при интернет-активности пользователей, не связанной с трудовой деятельностью.

Учёт потребления интернет-ресурсов	Управление доступом в Интернет	Защита сети
Знание сотрудников о том, что каждое их действие в Глобальной сети контролируется, уменьшит потребление трафика и предотвратит нецелевое расходование средств учреждения	В данном случае указывается, когда, к каким ресурсам и кто из сотрудников может иметь доступ. Это позволяет сократить потери рабочего времени и повысить безопасность	Следует установить защиту от утечки информации, атак хакеров и сетевых вирусных программ

ИНФОРМАЦИОННЫЕ РЕСУРСЫ



В Федеральном законе «Об информации, информатизации и защите информации» прописано: «**Информационные ресурсы** — отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах)». Это определение даёт юридическое основание для охраны информационных ресурсов, но несколько сужает его (как и другие юридические формулы). В широком смысле к информационным ресурсам относят все научно-технические знания, произведения литературы и искусства, множество иной информации общественно-государственной значимости, зафиксированной в любой форме, на любом носителе.

Национальные информационные ресурсы



Национальные информационные ресурсы — это информационные ресурсы, созданные и широко используемые в конкретном государстве.

Классификация национальных информационных ресурсов

Ресурс	Характеристика
Библиотечные ресурсы	Библиотеки обладают огромным количеством информационных ресурсов. Преобладают бумажные формы их представления, но всё больше библиотечных ресурсов (книги, диссертации, различные научные работы) в последние годы переводится в электронный вид

Архивные ресурсы	Архивы хранят материалы, связанные с историей и культурой страны. Объёмы архивных материалов огромны и обычно накапливаются быстрее, чем их удастся обработать
Научно-технические ресурсы	Во всех развитых странах существуют специализированные системы научно-технической информации. Они включают многочисленные специальные издания, патентные службы и т. д.
Правовая информация	Своды законов, кодексы, нормативные акты, другие виды правовой информации — основа функционирования любого государства
Отраслевая информация	Свои отраслевые информационные ресурсы имеются у любой социальной, научной, промышленной, аграрной и иной сферы общества. Значительны по объёму информационные ресурсы оборонной сферы, системы образования и т. д.



Также к национальным информационным ресурсам можно отнести:

- информацию государственных (властных) структур;
- финансовую и экономическую информацию;
- информацию о природных ресурсах;
- информацию предприятий и учреждений.

Ресурсосбережение



Ресурсосбережение — основная результирующая часть научно-технического прогресса (НТП), представляющая собой эколого-социально-экономический эффект, полученный за счёт рационализации потребления ресурсов.

Задача	Характеристика
Экономическая	Определение эффективных форм организации производства, постоянный учёт наличия, движения и расходования ресурсов, управление затратами, внедрение прогрессивных стимулов экономии ресурсов, политики ценообразования и сбыта
Техническая	Научно обоснованный выбор ресурсоэкономичных технических средств на стадиях производства и эксплуатации с оптимальными показателями долговечности, безотказности, ремонтпригодности и сохраняемости
Технологическая	Разработка безотходных и малооперационных технологий, обеспечивающих при минимальном потреблении ресурсов формирование требуемых качественных характеристик производимой продукции
Экологическая	Установление гармоничного взаимодействия агропромышленного производства с окружающей средой на основе восстановления почвенного плодородия, энерго-ресурсов, водного баланса и минеральных ресурсов

Рынок информационных ресурсов и услуг



Наряду с мировым рынком информационных ресурсов и услуг во многих странах сформировался также национальный рынок. Как и на всяком рынке, там есть поставщики (продавцы) и потребители (покупатели). В качестве товара могут выступать:

- информация бытового характера о доступе к материальным товарам и услугам, их стоимости;
- информация научно-технического характера (патенты, авторские свидетельства, научные статьи и т. д.);
- информационные технологии, компьютерные программы;
- базы данных, информационные системы и многое другое.

ЭКОНОМИКА ИНФОРМАЦИОННОЙ СФЕРЫ



Информационная сфера — постоянно расширяющаяся область человеческой деятельности, связанная с производством новых информационных продуктов, услуг и технологий.

Информационная сфера

Научно-информационные центры

Информационные сети

Библиотеки и архивы

Дистанционное обучение

Средства массовой информации (СМИ)

Индустрия создания, хранения, обработки и распространения информации во всех областях человеческой деятельности

Структура экономической информационной системы

Информационное обеспечение

Техническое обеспечение

Лингвистическое обеспечение

Организационное обеспечение

Правовое обеспечение



Экономическая информационная система — система, функционирование которой заключается в сборе, хранении, обработке и распространении информации о деятельности экономического объекта реального мира.



Информационная экономика изучает закономерности использования информационных факторов как компонентов экономической системы.

Стадии развития информационной экономики

Стадия 1	Проникновение информационных технологий в производство
Стадия 2	Массовое внедрение информационных технологий и преобладание стандартизированных систем
Стадия 3	Значительное превышение производительности в сфере производства информации и информационных технологий над другими отраслями
Стадия 4	Переход к преобладающему производству информации и знаний



Основные задачи информационной экономики решаются на макро- и микроуровнях.

Макроуровень	Микроуровень
Происходит выделение наиболее перспективной отрасли для развития её информационной сферы, что определяется готовностью отрасли к новым переменам и интенсивному развитию	На местах происходит создание алгоритмов, которые позволят получить средства и возможности из микроуровня на развитие тех или иных информационных ресурсов, позволяющие создавать новые рабочие места



Информационный бизнес — направленная на получение дохода деятельность в сфере создания и коммерческого распространения информационных продуктов, технологий и услуг, т. е. бизнес в информационной сфере.

ИНФОРМАЦИОННАЯ ЭТИКА И ЭТИКЕТ



Информационная этика — дисциплина, исследующая моральные проблемы, которые возникают в связи с развитием и применением информационных технологий.



Проблематика информационной этики тесно связана с такими областями исследования, как компьютерная, виртуальная, сетевая и коммуникативная этика, интернет-этика, киберэтика, философия информации.

Проблемы, анализируемые информационной этикой

Связанные с разработкой морального кодекса для IT-профессионалов и обычных пользователей, чья работа связана с компьютерной техникой

Проблемы защиты прав собственности, авторских прав, права на личную жизнь и свободу слова применительно к области информационных технологий

Связанные с компьютерными преступлениями (преимущественно правовая сфера)



Понятие **«сетевой этикет»** появилось с возникновением первой сетевой услуги — электронной почты. Важно соблюдать правила сетевого этикета, чтобы не попасть в неприятное положение или не нарушить правовые нормы, а также чтобы не создать проблемы партнёрам по общению в Интернете. Сетевой этикет определяет базовые правила поведения в сети. Он регулирует:

- правила обмена сообщениями по электронной почте;
- стилистику сетевой коммуникации при коллективных обсуждениях (например, в чатах);
- общие правила написания публикуемых текстов в сети и пр.

Правила этикета для электронной почты	Правила этикета для общения в чате, форуме, телеконференции
Всегда заполняйте поле «Тема» для своего письма	Будьте тактичными и корректными в своих высказываниях
Цитируйте отрывки письма, если это необходимо для ответа	Выбирайте себе псевдоним (или ник), не оскорбляющий других участников чата
Используйте прописные буквы только по необходимости	Обращаясь к кому-либо, пишите его ник в начале вашей фразы
Не вставляйте в электронное письмо файлы большого объёма	Не используйте много прописных букв и восклицательных знаков
Используйте подпись, в которой содержатся данные об авторе письма	Не злоупотребляйте смайликами
Используйте смайлики только в неофициальной корреспонденции	Уходя из чата, не забудьте попрощаться с собеседниками

ИНФОРМАЦИОННОЕ ПРАВО



Информационное право — совокупность охраняемых государством правовых норм, возникающих в сфере производства, преобразования и потребления информации.



Общеправовые принципы информационного права базируются на Конституции РФ, специальные принципы основываются на федеральных законах и других нормативных актах. Предмет регулирования информационного права образуют общественные отношения:

- связанные с осуществлением права на производство, распространение, передачу, поиск и получение информации;
- возникающие, изменяющиеся и прекращающиеся в связи с применением информационных технологий;
- обусловленные созданием, эксплуатацией и развитием информационных систем;
- связанные с созданием условий для эффективного применения информационно-телекоммуникационных сетей.

Цели защиты информации

Предотвращение хищения, утечки, искажения, утраты и подделки информации

Предотвращение несанкционированных действий по уничтожению, модификации, копированию и блокированию информации

Реализация права на государственную тайну и конфиденциальную информацию

Понятие	Характеристика
Защита информации	Деятельность, которая направлена на предотвращение утечки защищаемых данных, непреднамеренных и несанкционированных воздействий на защищаемые данные
Средства защиты информации	Делятся на организационные, юридические и программно-технические
Организационные средства защиты информации	Совокупность организационно-технических (обеспечение компьютерными помещениями, настройка кабельной системы и др.) и организационно-правовых средств (повышение квалификации персонала, контролируемые каналы распространения информации, разделение прав доступа, уничтожение ненужных копий документов и др.)
Юридические средства защиты информации	Совокупность законов и правовых актов, регулирующих информационную деятельность



Программно-технические средства защиты информации объединяют программы, которые помогают контролировать, хранить и защищать информацию и доступ к ней, а также технические виды устройств, которые предотвращают несанкционированное проникновение и утечку данных. Эти средства обеспечивают:

- защиту от компьютерных вирусов;
- шифрование данных;
- резервное копирование данных;
- ограничение доступа к устройствам и файловой системе.

Юридические средства защиты информации

Закон	Пояснение
«О правовой охране программ для ЭВМ и баз данных»	Регламентирует юридические вопросы, связанные с авторскими правами на программные продукты и базы данных
«Об авторском праве и смежных правах»	Разделяет следующие права на информационные ресурсы: <ul style="list-style-type: none"> • право распоряжения состоит в том, что только субъект — владелец информации имеет право определять, кому эта информация может быть предоставлена; • право владения должно обеспечивать субъекту — владельцу информации хранение информации в неизменном виде (никто, кроме него, не может её изменять); • право пользования предоставляет субъекту — владельцу информации право её использования только в своих интересах

«Об информации, информатизации и защите информации»

Позволяет защищать информационные ресурсы (личные и общественные) от искажения, порчи, уничтожения. Данным законом определено, что информационные ресурсы, являясь объектом отношений физических, юридических лиц и государства, подлежат обязательному учёту и защите, как всякое материальное имущество собственника. При этом собственнику предоставляется право самостоятельно в пределах своей компетенции устанавливать режим защиты информационных ресурсов и доступа к ним

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ



Информационная безопасность — совокупность мер по защите информационной среды общества и человека. Информационная безопасность — механизм, который обеспечивает:

- конфиденциальность — доступ к информации только авторизованных пользователей;
- целостность — достоверность и полноту информации и методов её обработки;
- доступность информации для авторизованных пользователей.



Информационная безопасность предприятия — состояние защищённости информационных ресурсов и экономических интересов предприятия в информационной сфере.

Угрозы безопасности в информационной сфере

Угрозы	Характеристика
Естественные	Катаклизмы, не зависящие от человека: пожары, ураганы, наводнения, удары молнии и т. д.
Искусственные	<ul style="list-style-type: none"> • Непреднамеренные (совершаются людьми по неосторожности или незнанию). • Преднамеренные (хакерские атаки, в том числе компьютерные вирусы, противоправные действия конкурентов, месть сотрудников и пр.)
Внутренние	Источники угрозы находятся внутри системы
Внешние	Источники угрозы находятся за пределами системы



Виды воздействия на защищаемые данные: удаление, частичное редактирование, модификация, изменение кодировки, блокировка доступа, невозможность загрузки, бесконтрольное распространение.

Компьютерные вирусы



Компьютерные вирусы — специально написанные программы, способные самопроизвольно присоединяться к другим программам, создавать свои копии и внедрять их в файлы, системные области компьютера и в вычислительные сети с целью нарушения работы программ, порчи файлов и каталогов, создания всевозможных помех в работе компьютера.

Классификация компьютерных вирусов

Виды	Меры профилактики воздействия вирусных программ
По среде обитания: сетевые, файловые, загрузочные и файлово-загрузочные	Резервное копирование (создание копий файлов и системных областей жёстких дисков)
По способу заражения: резидентные и нерезидентные	Исключение использования случайных и неизвестных программ
По степени воздействия: неопасные, опасные и очень опасные	Ограничение доступа к информации, в частности защита носителя информации (диска, флеш-карты) во время копирования файлов
По особенностям алгоритмов: паразитические, репликаторы, невидимки, мутанты, троянские, макровирусы	Использование специальных антивирусных программ



Антивирусная программа — специализированная программа для обнаружения компьютерных вирусов, нежелательных вредоносных программ и восстановления заражённых (модифицированных) такими программами файлов, а также для профилактики — предотвращения заражения (модификации) файлов или операционной системы вредоносным кодом.

Искусственный интеллект и машинное обучение



Искусственный интеллект (ИИ), машинное обучение (МО) (Machine Learning, ML), искусственные нейронные сети (ИНС) — термины, используемые для описания технологий, способных решить множество задач из реального мира.

Искусственный интеллект	Машинное обучение	Искусственные нейронные сети
Способность компьютера обучаться, принимать решения и выполнять действия, свойственные человеческому интеллекту	Технология искусственного интеллекта, которая предоставляет вычислительным системам возможность автоматически учиться и совершенствоваться на основе опыта без явного программирования	Математические модели, созданные по аналогии с биологическими нейронными сетями. ИНС способны моделировать и обрабатывать нелинейные отношения между входными и выходными сигналами



С точки зрения структурной организации машинное обучение является разделом искусственного интеллекта, нейросети — одним из видов машинного обучения.



Цель машинного обучения — предсказать результат по входным данным. Чем разнообразнее и обширнее будут входные данные, тем проще машине найти закономерности и тем точнее будет результат.

Параметры, необходимые для обучения машины

Параметры	Характеристика
Данные	Направление сбора данных определяется выполняемой задачей. Для обучения информатике понадобятся все книги по информатике; если стоит задача научиться определять спам в письмах или сообщениях, нужно множество таких писем и сообщений. Большое количество данных способствует улучшению результата. Данные могут собираться вручную или автоматически
Характеристики	К характеристикам можно отнести данные пользователя: его пол, возраст и др. Чем точнее и полнее заданные характеристики, тем быстрее будет проходить обучение системы
Алгоритм	Алгоритм определяет метод решения задачи, от которого будет зависеть точность, скорость работы и размер готовой модели

Практическое применение искусственного интеллекта

Сфера	Применение
Банковская	Возможность оплаты без карты с помощью системы распознавания лиц, анализ валютного рынка
Автомобилестроение	Создание автомобилей с беспилотным управлением
Безопасность	Внедрение систем распознавания лиц с целью предотвращения преступлений и поиска преступников
Медицина	Применение телемедицины, опросников, фитнес-браслетов для сбора данных и консультирования пациентов
Промышленность	Использование систем ИИ в сборке деталей, бухгалтерских расчётах, применение программ, заменяющих работу консультантов
Сельское хозяйство	Создание машин для распознавания вредителей и болезней растений, сбора урожая, уничтожения сорняков, применение беспилотных летательных аппаратов для видеосъёмки и орошения полей
Бытовая сфера	Применение автоматизированных систем управления «умный дом», способных решать определённые повседневные задачи без участия человека

Справочное издание
анықтамалық баспа
Для среднего и старшего школьного возраста
Орта және ажа возраста үшін
ШКОЛЬНЫЙ КУРС В НАГЛЯДНЫХ ТАБЛИЦАХ

Тимофеева Елена Викторовна

ИНФОРМАТИКА

5—11 КЛАССЫ
(орыс тілінде)

Ответственный редактор *Т. Судакова*
Редактор *П. Умитбаева*
Художественный редактор *Н. Кривошта*

Во внутреннем оформлении использованы иллюстрации:
Andriy Lipkan, davooda, karpenko_ilia, Pranch, stockakia, VasutinSergey,
VikiVector / Shutterstock.com
Используется по лицензии от Shutterstock.com

Соответствует техническому регламенту ТР ТС 007/2011
КО ТР 007/2011 техникалық регламентіне сәйкес келеді
Страна происхождения: Российская Федерация
Шығарылған елі: Ресей Федерациясы

ООО «Издательство «Эксмо»
125308, Россия, город Москва, улица Зорге, дом 1, строение 1, этаж 20, каб. 2013.
Тел.: 8 (495) 411-68-86.
Home page: www.eksmo.ru E-mail: info@eksmo.ru
Фабрикал: «ЭКСМО» АҚС Баспасы.
125308, Ресей, қала Мәскеу, Зорге көшесі, 1 үй, 1 ғимарат, 20 қабат, офис: 2013 к.
Тел.: 8 (495) 411-68-86.
Home page: www.eksmo.ru E-mail: info@eksmo.ru
Тауар белгісі: «Эксмо»
Интернет-магазин: www.book24.kz



ISBN 978-5-04-189580-8



Хочешь стать
автором «Эксмо»?



Издательство «Эксмо» — универсальное
издательство №1 в России, является
одним из лидеров книжного рынка Европы.

eksmo.ru [eksmo](https://www.instagram.com/eksmo)

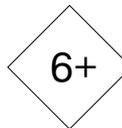


eksmo.ru

Официальный
интернет-магазин
издательства «Эксмо»

Интернет-магазин: www.book24.kz
Интернет-магазин: www.book24.kz
Импортер в Республику Казахстан ТОО «РДЦ-Алматы».
Қазақстан Республикасындағы импорттаушы «РДЦ-Алматы» ЖШС.
Дистрибутор и представитель по прямому поручению на продукцию,
в Республике Казахстан: ТОО «РДЦ-Алматы»
Қазақстан Республикасында дистрибутор және өнім бойынша арна-талалғанды
қабилдаушының өкілі «РДЦ-Алматы» ЖШС.
Алматы қ., Дембровский көш., 3-а, литер Б, офис 1.
Тел.: 8 (727) 251-59-90/31/32; E-mail: RDC-Almaty@eksmo.kz
Феникс қарамағының меркезі шетелдегі.
Сертификация туралы ақпарат сайты: www.eksmo.ru/certification

Сведения о подтверждении соответствия издания согласно законодательству РФ
о техническом регулировании можно получить на сайте Издательства «Эксмо»
www.eksmo.ru/certification
Өндiрген мемлекет: Ресей. Сертификация қарастырылған
Дата изготовления / Подписано в печать 16.08.2023.
Формат 60x90^{1/16}. Печать офсетная. Бумага типографская.
Усл. печ. л. 14,0. Тираж экз. Заказ



ШКОЛЬНЫЙ КУРС

В НАГЛЯДНЫХ ТАБЛИЦАХ

ИНФОРМАТИКА



ЧТО
в книге?

Сведения по основным разделам
школьного курса информатики:

- * ИНФОРМАЦИЯ и ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ
- * СРЕДСТВА ИКТ
- * АЛГОРИТМИЗАЦИЯ и ПРОГРАММИРОВАНИЕ
- * ИНФОРМАЦИОННАЯ ДЕЯТЕЛЬНОСТЬ ЧЕЛОВЕКА

В ЧЁМ
особенность?

Теоретические сведения представлены
в виде **НАГЛЯДНЫХ ТАБЛИЦ**, которые помогают
быстро и эффективно запомнить материал

Для ЧЕГО нужна
эта книга?

- * Для повторения материала перед контрольными, ОГЭ и ЕГЭ
- * Для заполнения пробелов в знаниях
- * Чтобы иметь краткий конспект по теме

Для КОГО?

Для учеников средней и старшей школы

ISBN 978-5-04-189580-8



9 785041 895808 >

www.yk.com/eksmo_kids

