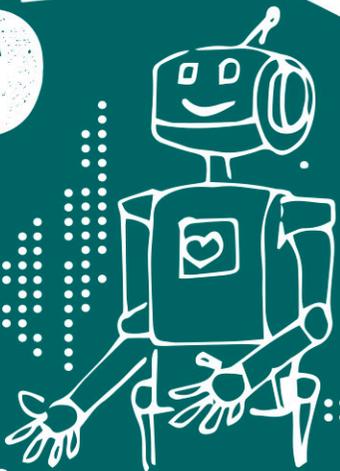
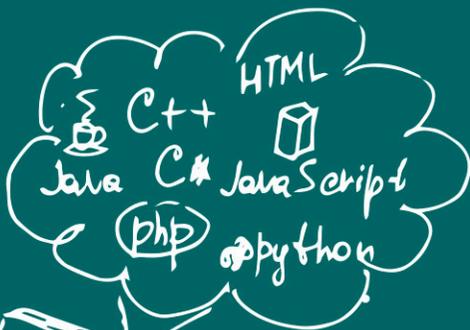


НАГЛЯДНЫЙ СПРАВОЧНИК ДЛЯ ПОДГОТОВКИ

К **ОГЭ**
И
ЕГЭ



Е.В. Тимофеева

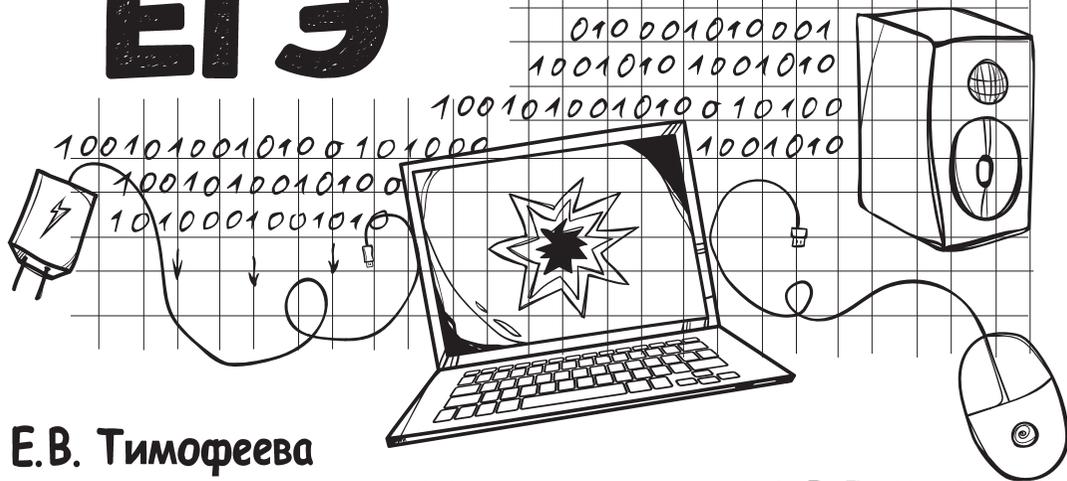
ИНФОРМАТИКА

- ✓ ВСЕ ТЕМЫ ШКОЛЬНОГО КУРСА
- ✓ ТРЕНИРОВОЧНЫЕ ЗАДАНИЯ
- ✓ ОТВЕТЫ И КОММЕНТАРИИ



НАГЛЯДНЫЙ СПРАВОЧНИК ДЛЯ ПОДГОТОВКИ

к **ОГЭ**
и
ЕГЭ



Е. В. Тимофеева

ИНФОРМАТИКА



МОСКВА
2023

УДК 373.5:004
ББК 32.81я721
Т41

Макет подготовлен при содействии ООО «Айдиономикс»

Тимофеева, Елена Викторовна.
Т41 Информатика / Е. В. Тимофеева. — Москва : Эксмо, 2023. —
304 с. : ил. — (Наглядный справочник для подготовки к ОГЭ и ЕГЭ).
ISBN 978-5-04-179033-2

Справочник содержит теоретические сведения за весь школьный курс информатики, а также практические задания с ответами и пояснениями. Весь материал изложен в наглядной и доступной форме, что способствует быстрому усвоению большого количества информации.

Издание окажет помощь старшеклассникам при подготовке к ОГЭ и ЕГЭ, урокам, различным формам текущего и промежуточного контроля.

УДК 373.5:004
ББК 32.81я721

ISBN 978-5-04-179033-2

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

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5	Представление звуковой информации	78
❶ ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ	6	Элементы алгебры логики	81
Информация и её кодирование	6	Логические высказывания	81
Виды информации	6	Истинность высказывания	81
Свойства информации	9	Логические операции	83
Единицы измерения количества информации	10	Приоритеты логических связей	87
Виды информационных процессов	16	Основные законы логики	90
Кодирование и декодирование информации	17	Предикаты и кванторы	98
Процесс передачи информации, источник и приёмник	27	☐ СРЕДСТВА ИКТ	101
Аналоговое и дискретное представление информации	28	Архитектура компьютера и компьютерных сетей	101
Искажение информации	31	Организация работы компьютеров	103
Скорость передачи информации	32	Аппаратное обеспечение	103
Система, её свойства и компоненты	37	Программное обеспечение	106
Понятие системы	37	Файловая система	112
Характеристики и свойства системы	38	Технологии создания и обработки текстовой информации	116
Информационная система и её компоненты	40	Программы для работы с текстовой информацией	116
Моделирование	42	Работа с текстовым документом	117
Классификация, цели и этапы моделирования	42	Шаблоны текстовых документов	120
Информационное моделирование	46	Проверка орфографии и грамматики	121
Математическое моделирование	48	Словари и тезаурусы	122
Компьютерное моделирование	50	Машинный перевод	122
Имитационное моделирование	52	Редактирование математических текстов. Графическое представление математических объектов	123
Системы счисления	53	Использование систем распознавания текстов	125
Позиционные системы счисления	54	Технология создания и обработки мультимедийной и графической информации	134
Двоичное представление информации	56	Форматы графических объектов	135
Сложение и умножение в различных системах счисления	67	Создание, ввод и обработка графических объектов	137
Представление информации в компьютере	70	Звуковые файлы	139
Представление числовой информации	70	Технология обработки информации в электронных таблицах	142
Представление текстовой информации	72	Табличный процессор	143
Представление графической информации	73		

Объекты табличного процессора Excel и их свойства	145
Технологии поиска и хранения информации.....	150
Составляющие банка данных.....	150
Типы моделей баз данных.....	151
Табличные базы данных.....	153
Использование инструментов поисковых систем (формирование запросов)	155
Телекоммуникационные технологии	160
Программное обеспечение средств.....	161
Инструменты создания информационных объектов для Интернета.....	171
 АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ	175
Алгоритмы и алгоритмизация.....	175
Историческая справка.....	175
Основные понятия.....	179
Свойства и способы представления алгоритмов	180
Виды алгоритмов	182
Выигрышная стратегия	196
Программирование	210
Классификация языков программирования.....	212
Основные понятия языков программирования.....	212
Типы данных.....	213
Структурированные типы данных.....	214
Графы.....	215
Массив	227
Списки.....	232
Основы языков программирования	234
Виды информации.....	234
Основные служебные слова.....	236
Разделители языка.....	238
Структура программы	239

Идентификаторы	240
Переменные и константы.....	242
Функции.....	245
Операторы и операции	248
Задачи на запись и анализ алгоритмов	257
Алгоритмы обработки массивов	263
Решение задач на действия над массивами.....	265
Исправление ошибок в программе.....	273
 ИНФОРМАЦИОННАЯ ДЕЯТЕЛЬНОСТЬ ЧЕЛОВЕКА.....	282
Этапы развития информационного общества	282
Профессиональная информационная деятельность.....	284
Применение технических средств и информационных ресурсов в профессиональной деятельности.....	285
Системное администрирование.....	286
Интернет и безопасность его использования	287
Информационные ресурсы.....	288
Национальные информационные ресурсы, их классификация	289
Ресурсосбережение	290
Рынок информационных ресурсов и услуг.....	291
Экономика информационной сферы.....	291
Информационная этика и этикет.....	293
Информационное право	295
Информационная безопасность	298
Искусственный интеллект и машинное обучение	300
Эксплуатация компьютерного рабочего места	303

ВВЕДЕНИЕ



Перед вами самый удобный справочник, который поможет школьнику систематизировать и закрепить знания по информатике за курс средней школы.

Пособие содержит основную и самую важную информацию из следующих разделов курса: «Информация и информационные процессы», «Средства информационных и коммуникационных технологий», «Алгоритмизация и программирование», «Информационная деятельность человека».

Материал книги представлен в виде таблиц, схем, рисунков, упорядочен и систематизирован, изложен доступным для усвоения языком. Это обеспечит максимальную сконцентрированность внимания, эффективное повторение и подготовку школьника по предмету.

Теоретический материал каждой темы сопровождается блоком практических заданий. Приведённые примеры с развернутыми разъяснениями позволят детально разобраться в темах школьного курса и отработать навыки выполнения различных заданий, составленных в том числе в соответствии с форматом требований ЕГЭ.

Справочник предназначен учащимся средней школы для самоподготовки к различным видам контроля, сдаче ОГЭ и ЕГЭ, а также может использоваться учителями информатики для работы на уроке.

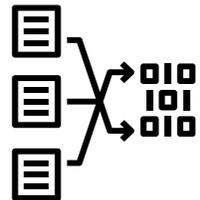
Желаем успехов!

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

Деление информации на виды проводится по нескольким критериям.

По способу восприятия

Визуальная

Информация, которую мы получаем с помощью зрения. Например, можем увидеть северное сияние, танец, 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

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 почувствовать, как ароматно пахнет домашняя еда, 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

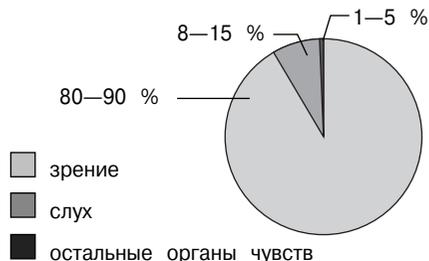
Информация, которую можно получить с помощью вестибулярного аппарата, отслеживающего наше положение в пространстве. Например, закрыв глаза, мы способны понять, куда идём: направо или налево. При падении мы чётко понимаем, что двигаемся вниз

Мышечная

Информация, которую мы получаем, используя мышцы. Например, можем определить, какой из двух предметов тяжелее. Благодаря мышечной информации развивается способность печатать на клавиатуре вслепую



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.



По назначению**Массовая**

Информация, передаваемая широким аудиториям, рассредоточенным во времени и пространстве, с помощью искусственных каналов. К такой информации относятся печатные, аудио-, 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**

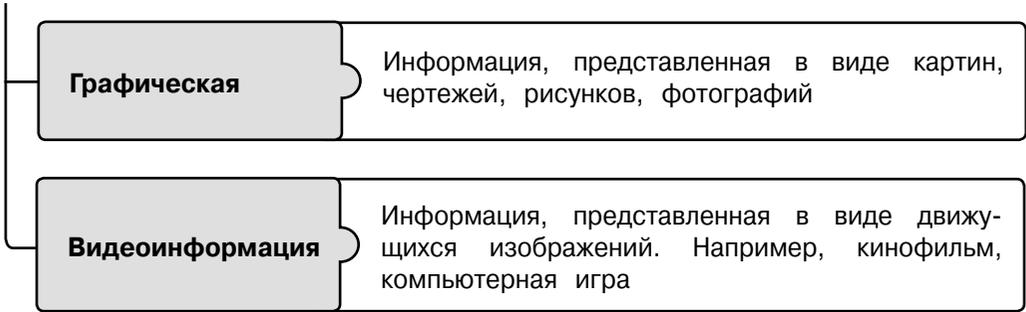
Информация, представленная в форме текстового сообщения, рукописного либо печатного. Например, газетные статьи, записи в блоге, книги, письма

Числовая

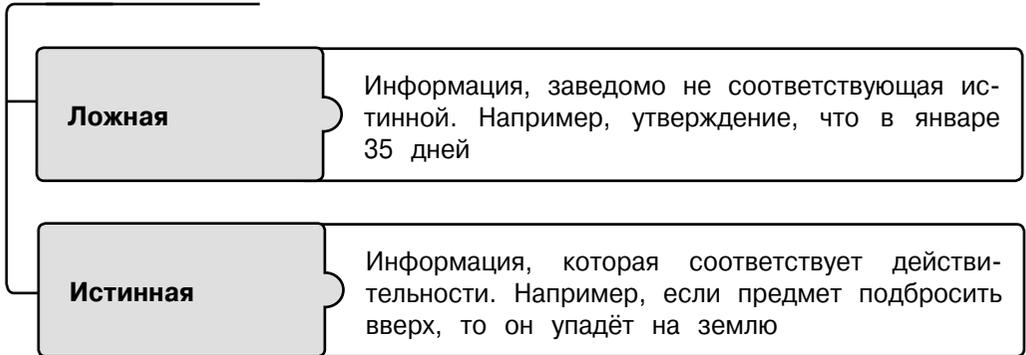
Информация, выраженная в виде специальных символов, чисел. Например, номер телефона, номер квартиры

Звуковая

Любая информация, которую можно услышать: музыка, речь человека, шум моря, звук колокола, шорох листьев



По истинности



СВОЙСТВА ИНФОРМАЦИИ

Свойства информации: объективность, актуальность, полнота, достоверность, полезность, понятность и дискретность.

▲ **Объективная информация.** Не зависит от чьего-либо мнения, суждения.

▲ **Актуальная информация.** Важна и существенна для настоящего времени.

▲ **Полная информация.** Достаточно 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. Характеризуется последовательными точными значениями некоторой величины.

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 информа-

ции»). **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, правильно решено задание или нет и т. д. **Байт** — последовательность из 8 двоичных разрядов битов (наиболее популярная единица измерения информации).

ТАБЛИЦА СТЕПЕНЕЙ ЧИСЛА 2

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
1	2	4	8	16	32	64	128	256	512	1024	2048

ПРОИЗВОДНЫЕ ЕДИНИЦЫ ИЗМЕРЕНИЯ КОЛИЧЕСТВА ИНФОРМАЦИИ

Название	Условное обозначение	Соотношение с другими единицами
Бит	Бит	
Килобит	Кбит	2^{10} бит = 1024 бит
Мегабит	Мбит	2^{20} бит = 1024 Кбит
Гигабит	Гбит	2^{30} бит = 1024 Мбит
Терабит	Тбит	2^{40} бит = 1024 Гбит
Петабит	Пбит	2^{50} бит = 1024 Пбит
Байт	Байт	8 бит
Килобайт	Кбайт	2^{10} байт = 1024 байт
Мегабайт	Мбайт	2^{20} байт = 1024 Кбайт
Гигабайт	Гбайт	2^{30} байт = 1024 Мбайт
Терабайт	Тбайт	2^{40} байт = 1024 Гбайт
Петабайт	Пбайт	2^{50} бит = 1024 Пбайт

Формула объёма информации:

$$I_{\text{об}} = k \cdot i,$$

где I — искомый объём, k — количество символов, i — количество бит, необходимое для хранения одного символа.

✓ Выполните перевод

- а) в биты: 5 Кбайт;
- б) в байты: 1 Мбайт;
- в) в килобайты: 4096 бит.

Решение:

- а) В биты: $5 \text{ Кбайт} = 5 \cdot 8 \text{ Кбит} = 40 \text{ Кбит}$
(т. к. в 1 байте 8 бит).
 $40 \text{ Кбит} = 40 \cdot 1024 = 40\,960 \text{ бит}$.
- б) В байты: $1 \text{ Мбайт} = 1024 \text{ Кбайт} =$
 $= 1024 \cdot 1024 \text{ байт} = 1\,048\,576 \text{ байт}$.
- в) В килобайты: $4096 \text{ бит} = 4096 : 1024 =$
 $= 4 \text{ Кбит}$ (1 Кбит = 1024 бит).
 $4 \text{ Кбит} = 4 : 8 = 0,5 \text{ Кбайт}$.

✓ Статья, набранная на компьютере, содержит 32 страницы, на каждой странице 40 строк, в каждой строке 48 символов. Определите размер статьи в кодировке КОИ-8, в которой каждый символ кодируется 8 битами.

Решение:

Найдём количество символов в статье: $32 \cdot 40 \cdot 48 = 32 \cdot 5 \cdot 8 \cdot 3 \cdot 16 =$
 $= 2^5 \cdot 5 \cdot 2^3 \cdot 3 \cdot 2^4 = 15 \cdot 2^{12}$ ($40 = 5 \cdot 8$;
 $48 = 3 \cdot 16$).
Один символ кодируется одним байтом, 2^{10} байт составляют 1 Кбайт, поэтому информационный объём статьи составляет $15 \cdot 2^{12} \text{ байт} = 15 \cdot 2^2 \times$
 $\times 2^{10} \text{ байт} = 15 \cdot 2^2 \text{ Кбайт} = 60 \text{ Кбайт}$.

Ответ: 60 Кбайт.

✓ В одной из кодировок Unicode каждый символ кодируется 16 битами. Коля написал текст (в нём нет лишних пробелов, символы переноса слов не учитываются):
«Москва, Санкт-Петербург, Новосибирск, Екатеринбург, Пенза, Казань, Челябинск, Омск, Самара, Ростов-на-Дону, Уфа, Красноярск, Воронеж, Пермь, Волгоград, Ульяновск, Барнаул, Иркутск, Липецк, Ярославль, Владивосток — города России».

Ученик вычеркнул из списка название одного города. Заодно он вычеркнул ставшие лишними запятые и пробелы — два пробела не должны идти подряд. При этом размер нового предложения в данной кодировке оказался на 14 байт меньше, чем размер исходного предложения. Среди городов, имеющих одинаковое количество букв, Коля вычёркивает первый по порядку.

Напишите в ответе вычеркнутое название города.

Решение:

Один символ кодируется 16 битами. Если перевести их в байты, то получится 2 байта. Следовательно, из текста удалили 7 символов. Заметим, что лишние запятая и пробел занимают 2 байта. Значит, название города, которое удалили из списка, должно состоять из 5 букв, поскольку $(7 - 2) : 1 = 5$ символов. Первое название города, которое состоит из 5 букв, — Пенза.

Ответ: Пенза.



Практические задания

1 Сколько килобайт информации содержит сообщение объёмом 2^{19} бит?

Решение:

1 байт = 8 бит = 2^3 бит.

1 Кбайт = 1024 байт = 2^{10} байт.

Отсюда 2^{19} бит = $2^{19} : 2^3 = 2^{16}$ байт = $2^{16} : 2^{10} = 2^6 = 64$ Кбайт.

Ответ: 64 Кбайт.

2 Информационный объём одного сообщения составляет 1 Кбайт, а другого — 384 бит. Сколько байт информации содержат эти два сообщения вместе? В ответе укажите одно число.

Решение:

Переведём обе величины в байты:

1 Кбайт = 1024 байт; 384 бит = 48 байт.

Выполним сложение: $1024 + 48 = 1072$ байт.

Ответ: 1072 байт.

3 В одной из кодировок Unicode каждый символ кодируется 16 битами. Ученик написал текст (в нём нет лишних пробелов):

«Уфа, Ейск, Жуков, Амурск, Воронеж, Воткинск, Соликамск — города России».

Ученик удалил из списка название одного города, а также лишние запятую и пробел — два пробела не должны идти подряд. При этом размер нового предложения в данной кодировке оказался на 16 байт меньше, чем размер исходного предложения. Напишите в ответе удалённое название города.

Решение:

По условию задачи каждый символ кодируется 16 битами, а после вычёркивания размер оказался на 16 байт меньше, значит, вычёркнутое слово вместе с одним пробелом и одной запятой составляет 16 байт.

Преобразуем байты в биты: 1 байт = 8 бит; 16 байт = $8 \cdot 16 = 128$ бит. Зная, что один символ кодируется 16 битами, определим количество символов: $128 : 16 = 8$ символов. Два из восьми символов — это запятая и пробел. Таким образом, на само слово, обозначающее город, остаётся $8 - 2 = 6$ символов. Это количество соответствует названию Амурск.

Ответ: Амурск.

- 4** Статья, набранная на компьютере, содержит 24 страницы, на каждой странице 40 строк, в каждой строке 128 символов. Определите информационный объём статьи в килобайтах в кодировке КОИ-8, в которой каждый символ кодируется 8 битами.

Решение:

Воспользуемся формулой объёма информации в сообщении: $I_{об} = k \cdot i$. По условию $i = 8$. Найдём значение k : $k = 24 \cdot 40 \cdot 128 = 8 \cdot 3 \cdot 8 \cdot 5 \cdot 128$. Представим числа 8 и 128 в виде степеней числа 2: $k = 15 \cdot 2^3 \cdot 2^3 \cdot 2^7 = 15 \cdot 2^{13}$.

$$I_{об} = 15 \cdot 2^{13} \cdot 8 = 15 \cdot 2^{16} \text{ бит.}$$

Переведём в килобайты: $I_{об} = 15 \cdot 2^{16} : 2^{13} = 15 \cdot 2^3 = 120$ Кбайт.

Ответ: 120 Кбайт.

- 5** Статья, набранная на компьютере, содержит 32 страницы, на каждой странице 64 строки, в каждой строке 100 символов. Информационный объём статьи составляет 200 Кбайт.

Определите, сколько бит памяти используется для кодирования каждого символа, если известно, что для представления каждого символа отводится одинаковый объём памяти.

Решение:

Переведём 200 Кбайт в биты: $200 \text{ Кбайт} = 200 \cdot 2^{13} \text{ бит}$. Из формулы объёма информации в сообщении $I_{об} = k \cdot i$ найдём i : $i = I_{об} : k$.

Вычислим k : $k = 32 \cdot 64 \cdot 100 = 2^5 \cdot 2^6 \cdot 4 \cdot 25 = 25 \cdot 2^{13}$.

Отсюда следует, что для кодирования каждого символа понадобится $k = 200 \cdot 2^{13} : 25 \cdot 2^{13} = 8$ бит.

Ответ: 8 бит.

- 6** В одной из кодировок Unicode каждый символ кодируется 16 битами. Определите размер приведённого ниже предложения в данной кодировке. Ответ выразите в байтах.
- «Октябрь уж наступил — уж роща отряхает последние листья с нагих своих ветвей».

Решение:

Способ 1. Посчитаем количество символов в предложении: $k = 76$, $i = 16$ (по условию).

Найдём $I_{об}$, используя формулу объёма информации в сообщении: $I_{об} = k \cdot i$.

$I_{об} = 76 \cdot 16 = 1216$ бит. В байтах: $I_{об} = 1216 : 8 = 152$ байта.

Способ 2. Посчитаем количество символов в предложении: $k = 76$. По условию $i = 16$ бит = 2 байта (для удобства сразу переведём в байты). Найдём $I_{об}$, используя формулу объёма информации в сообщении: $I_{об} = k \cdot i$. Получим: $I_{об} = 76 \cdot 2 = 152$ байта.

Ответ: 152 байта.

- 7** Текст рассказа набран на компьютере. Информационный объём получившегося файла составляет 15 Кбайт. Текст занимает 20 страниц, на каждой странице одинаковое количество строк, в каждой строке 32 символа. Все символы представлены в кодировке Unicode. В используемой версии Unicode каждый символ кодируется 2 байтами. Определите количество строк на каждой странице.

Решение:

Используя формулу объёма информации $I_{об} = k \cdot i$, найдём k : $k = I_{об} : i$.
 $k = 15 \cdot 2^{13} : 16 = 15 \cdot 2^{13} : 2^4 = 15 \cdot 10^9$.

Поскольку число k состоит из строк, страниц и символов, то количество строк найдём, поделив k на все известные данные: $15 \cdot 10^9 : 20 : 32 = 3 \cdot 4 = 12$ строк.

Ответ: 12 строк.

8 Пользователь создал сообщение из 256 символов в кодировке Unicode, в которой каждый символ кодируется 16 битами. После редактирования информационный объём сообщения составил 3120 бит.

Определите, сколько символов удалили из сообщения, если его кодировка не изменилась.

Решение:

Найдём информационный объём сообщения до редактирования:
 $256 \cdot 16 = 4096$ бит.

Определим, на сколько бит уменьшился информационный объём сообщения после редактирования: $4096 - 3120 = 976$ бит.

Тогда количество удалённых из сообщения символов будет равно $976 : 16 = 61$.

Ответ: 61 символ.

9 В одной из кодировок Unicode каждый из символов кодируется 2 байтами. Текст, набранный в этой кодировке, был перекодирован в 8-битную кодировку КОИ-8. При этом в памяти компьютера текст стал занимать на 2048 бит меньше. Из скольких символов состоит текст?

Решение:

Переведём байты в биты: 2 байта = $8 \cdot 2 = 16$ бит.

По формуле информационного объёма $I = k \cdot 16$ — текст до перекодировки.

Объём после перекодировки: $I - 2048 = k \cdot 8$.

Подставим I из первого выражения во второе и решим уравнение:

$$k \cdot 16 - 2048 = k \cdot 8;$$

$$k \cdot 16 = k \cdot 8 + 2048;$$

$$k \cdot 8 = 2048;$$

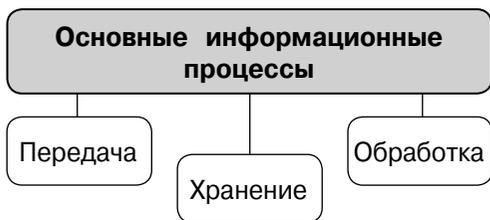
$$k = 2048 : 8;$$

$$k = 256.$$

Ответ: 256 символов.

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.



Передача — 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.

Хранение. 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.

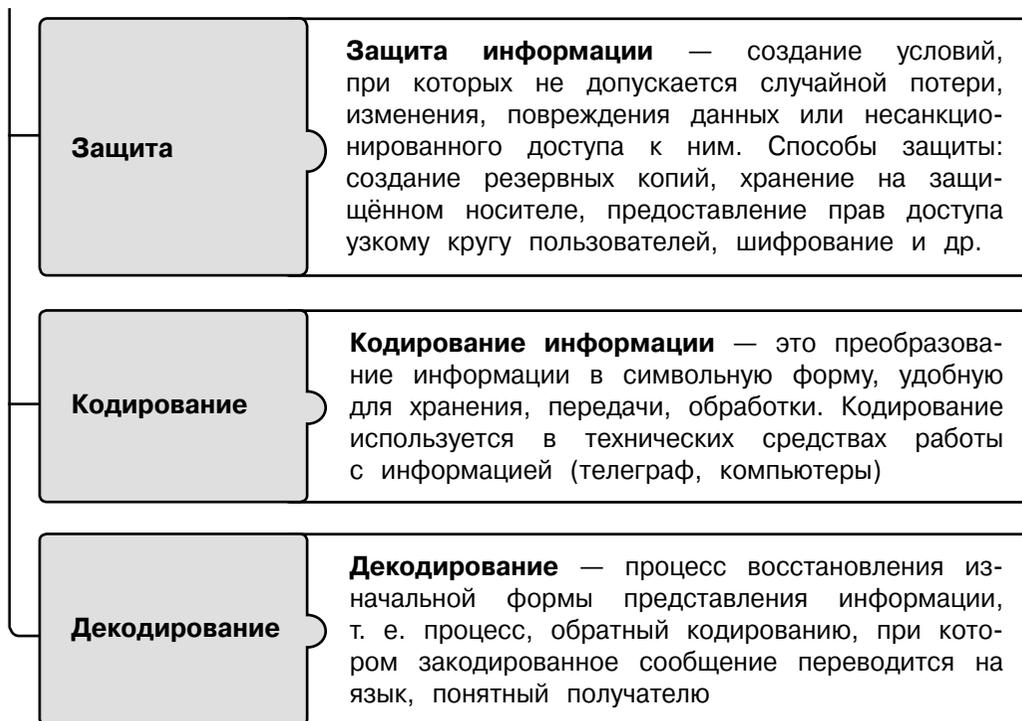
Обработка информации — вся совокупность операций (сбор, защита, преобразование, в том числе кодирование и декодирование, считывание, уничтожение), осуществляемых при помощи человека, технических и программных средств, включая обмен по каналам передачи данных. В результате обработки информации можно получить новые знания из имеющихся.

Рассмотрим подробнее некоторые из этих операций.

Некоторые операции обработки информации

Сбор

Сбор информации предполагает поиск и отбор необходимых данных из различных источников: работу с литературой, справочниками, проведение экспериментов, наблюдений, опросов, поиск в Интернете и т. д. Например, чтобы написать реферат, необходимо найти информацию по данной теме. Для сбора информации используют различные **измерительные устройства**. Для определения температуры воздуха на улице необходим термометр



✓ Какие информационные процессы можно назвать из приведённого текста?

«Школьник получил информацию в виде условия задачи, подумал, решил её в соответствии с определёнными правилами (например, правилами решения математических задач) и получил искомый результат».

Решение:

Школьник получил информацию — сбор и передача информации.

Школьник решил задачу — обработка информации.

Ответ: сбор, передача и обработка.

КОДИРОВАНИЕ И ДЕКОДИРОВАНИЕ ИНФОРМАЦИИ

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

онным кодом (кодовым словом или просто кодом) называется символьная последовательность, которая не-

сёт в себе конкретную информацию. Каждый код имеет определённую длину, т. е. состоит из конечного числа символов, это количество называется **длиной кода**. Например, последовательность цифр в номере телефона, букв в текстовом сообщении является информационным кодом.

■ Виды кодирования

Равномерное кодирование — вид кодирования, при котором все символы какого-либо алфавита кодируются кодами одинаковой длины. Чтобы понять, какая длина кода будет у одного символа при равномерном кодировании, нам понадобится формула Хартли, позволяющая определить количество информации.

Формула Хартли:

$$N = 2^i,$$

где N — количество равновероятных событий, i — количество информации, которую мы получим при наступлении одного из событий.

Эту закономерность Р. Хартли обнаружил в 1928 г. Из данной формулы можно вывести i :

$$i = \log_2 N.$$

Мощность алфавита — количество символов/знаков, из которых состоит рассматриваемый алфавит.

✓ Чтобы провести кодирование русскоязычного сообщения, нужно воспользоваться равномерным кодом. В первую очередь необходимо вспомнить количество букв в русском алфавите — 33. Затем следует воспользоваться формулой Хартли, чтобы определить количество бит, необходимых для кодирования одного символа.

Решение:

Посчитаем, сколько требуется бит информации для кодирования одного символа из русского алфавита: $2^i = 33$. Поскольку i — минимальное натуральное число, то $i = 6$.

Можно сделать вывод, что для кодирования сообщения нам требуется равномерный код длиной 6 бит.



Практические задания

10 Ученик загадал число от 1 до 16. Какое количество информации необходимо для угадывания числа из этого промежутка?

Решение:

N — количество чисел в промежутке 1–16. $N = 16$, подставим известные значения в формулу и найдём i :

$16 = 2^i$, $16 = 2^4$, отсюда $i = 4$.

Также i можно найти из выведенной ранее формулы: $i = \log_2 16 = 4$.

11 Объём сообщения — 7,5 Кбайт. Известно, что данное сообщение содержит 7680 символов. Какова мощность алфавита?

Решение:

1) I — информационный объём сообщения, K — количество символов в сообщении, i — вес одного символа: $I = K \cdot i$, откуда $i = \frac{I}{K}$.

2) Переведём 7,5 Кбайт в биты: 7,5 Кбайт = $7,5 \cdot 2^{13}$ бит.

3) $i = \frac{7,5 \cdot 2^{13}}{7680} = \frac{7,5 \cdot 2^{10} \cdot 2^3}{7,5 \cdot 2^{10}} = 8$ бит.

4) $M = 2^i$, где M — мощность алфавита, i — вес символа, т. е. то, чему равен один символ в битах. $M = 2^8 = 256$.

Ответ: 256.

12 Сколько слов длины 7, начинающихся с гласной буквы, можно составить из букв Т, О, Р? Каждая буква может входить в слово несколько раз. Слова необязательно должны быть осмысленными словами русского языка.

Решение:

Начальная мощность алфавита: $N = 3$ (буквы Т, О, Р).

На первом месте может стоять только буква О, т. к. по условию слово начинается с гласной, на остальных шести позициях — любые из трёх букв. Таким образом, можно составить равенство: $1 \cdot 3^6 = 729$ слов.

Ответ: 729 слов.



Для решения подобных задач можно использовать следующую формулу:

$$N = n_1 \cdot n_2 \cdot n_3 \cdot \dots \cdot n_L,$$

где n_1 — количество вариантов выбора первой буквы, n_2 — количество вариантов выбора второй буквы и т. д.

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 бит, 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, лежащими в основе теории кодирования.

Сформулировать прямое условие можно следующим образом: «Ни одно кодовое слово не может выступать в качестве начала любого другого кодового слова». 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 гласит: «Ни одно кодовое слово не может выступать в качестве окончания любого другого кодового слова».

С математической точки зрения обратное условие можно сформулировать следующим образом: «Если код содержит слово B , то для любой непустой строки C слова CB не существует в коде».

ПРЯМОЕ УСЛОВИЕ ФАНО

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 началом никакого другого кода

ОБРАТНОЕ УСЛОВИЕ ФАНО

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 окончанием никакого другого кода

Декодирование может быть однозначным и многозначным. Если при расшифровке возможно получить несколько различных вариантов исходного сообщения, то такое декодирование **неоднозначно**. При расшифровке с единственным возможным результатом сообщения имеет место **однозначное декодирование**.

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 выполняется 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.

✓ По каналу связи передаются сообщения, содержащие только семь букв: А, Б, И, К, Л, О, С. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны: А — 001, И — 01, С — 10. Какое наименьшее количество двоичных знаков потребуется для кодирования слова КОЛОС?

Решение:

Нам нужно закодировать три буквы: К, О, Л. Буква О встречается в слове 2 раза, поэтому её кодируем минимальным количеством знаков, например О — 000. К и Л не могут начинаться с 000, 10, 01, 001 по условию Фано, поэтому они кодируются как 110 и 111.

Можно рассмотреть вариант, где О — 11, тогда К и Л не могут начинаться с 11, 001, 10, 01. В таком случае это будет вариант 0001 и 00001.

Посчитаем количество знаков, необходимое для кодирования слова КОЛОС в каждом случае.

14 символов

110	000	111	000	10
К	О	Л	О	С

15 символов

0001	11	00001	11	10
К	О	Л	О	С

Минимальное количество — 14 символов.

Ответ: 14.



Иногда процесс декодирования проще выполнять с конца зашифрованного сообщения.

✓ Закодируем неравномерным кодом часть товаров, находящихся на товарном складе. Предположим, что на складе размещается около 3000 различных товаров, но наиболее ходовыми являются хлеб, соль, молоко и сахар.

Решение:

Данные четыре товара покупают огромными партиями, и очень утомительно вести записи в базе данных, постоянно вбивая названия этих продуктов. Можно применить следующее кодирование:

Хлеб — 00, Соль — 01, Молоко — 10, Сахар — 11.

Итого нам потребовалось 2 бита информации, чтобы закодировать в бинарном виде четыре наиболее ходовых товара.

Далее кодируем менее популярные, но всё-таки довольно востребованные товары, например муку и перец. Данные товары можно закодировать следующим образом:

Мука — 001, Перец — 010.

Значит, мы выделяем на их кодирование уже по 3 бита информации.

Необходимо уловить общий принцип: чем менее популярен товар, тем бóльшим количеством бит он будет закодирован.



Практические задания

- 13** Для кодирования растрового рисунка, напечатанного с использованием шести красок, применили неравномерный двоичный код. Для кодирования цветов используются следующие кодовые слова.

Цвет	Кодовое слово
Фиолетовый	11110
Чёрный	10
Белый	0
Зелёный	11111
Красный	1110
Голубой	?

Укажите кратчайшее кодовое слово для кодирования голубого цвета, при котором код будет удовлетворять условию Фано. Если таких кодов несколько, укажите код с наименьшим числовым значением.

Решение:

Подберём кодовое слово для голубого цвета. Поскольку нам нужно наименьшее кодовое слово, начнём с однозначных чисел: 0, 1 использовать нельзя, т. к. 0 занято, а 1 не подходит, поскольку является началом других кодовых слов. Заметим, что кодовые слова 11 и 10 использовать нельзя, т. к. 10 уже занято, а 11 является началом других кодовых слов. Попробуем трёхзначные кодовые слова. 100 не подходит, т. к. его можно получить из чёрного и белого, 101 нельзя использовать, т. к. его можно получить из чёрного и фиолетового. Поэтому можно использовать только кодовое слово 110, оно будет минимальным.

Ответ: 110.

- 14** От разведчика было получено следующее сообщение.

1101001110010100

В этом сообщении зашифрован пароль — последовательность букв русского алфавита.

В пароле использовались только буквы С, Б, К, Л, О, Я; каждая буква кодировалась двоичным словом по следующей таблице.

С	Б	К	Л	О	Я
01	100	101	111	00	110

Расшифруйте сообщение. Запишите в ответе пароль.

Решение:

Изучив таблицу, видим, что каждая буква зашифрована двоичным словом из двух или трёх цифр. Выполним разбиение зашифрованного сообщения. Первые две цифры — 11, но 11 нет в кодовых словах, значит, первая буква кодируется двоичным кодом, которое состоит из трёх цифр. 110 — буква Я. Рассуждая таким же образом, получим разбиение сообщения:

110 100 111 00 101 00, что соответствует слову ЯБЛОКО.

Ответ: яблоко.

15 Разведчик передал в штаб следующую радиограмму.

. . . _ _ _ . _ _ . . _ _ . . .

В этой радиограмме содержится последовательность букв, в которой встречаются только О, Б, В, С, Р. Каждая буква закодирована с помощью аналога азбуки Морзе. Разделителей между кодами букв нет.

Запишите в ответе переданную последовательность букв.

Нужный фрагмент приведён ниже.

О	Б	В	С	Р
. _ _	_ _ _ .	.	_ .	_ . _ _

Решение:

Изучив таблицу, видим, что каждая буква зашифрована одним, двумя, тремя или четырьмя символами. Выполним разбиение зашифрованного сообщения. Первая буква определяется только двумя символами «-·».

Продолжим разбивать на символы, исходя из условия таблицы, получим следующий результат.

· · · · ·
 _ _ _ _ _
 С О Р В О В С В

Ответ: СОРВОВСВ.

- 16** По каналу связи передаются сообщения, содержащие только семь букв: А, Б, Г, И, М, Р, Я. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны: А — 010, Б — 011, Г — 100. Какое наименьшее количество двоичных знаков потребуется для кодирования слова МАГИЯ?

Решение:

Следующая буква должна кодироваться как 11, поскольку 10 мы взять не можем. 100 взять не можем из-за Г, значит, следующая буква должна быть закодирована кодом 101. Следующая буква должна кодироваться как 000, поскольку 00 взять не можем, иначе не хватит кодовых слов для оставшейся буквы, которые удовлетворяют условию Фано. Значит, последняя буква будет кодироваться как 001. Тогда наименьшее количество двоичных знаков, которое потребуется для кодирования слова МАГИЯ, равно $2 + 3 + 3 + 3 + 3 = 14$.

Ответ: 14.

- 17** Передаются сообщения, содержащие только следующие буквы: Р, Е, К, А. Для передачи используется двоичный код, удовлетворяющий условию Фано. Для букв А, Р, Е используются такие кодовые слова: А — 111, Р — 0, Е — 100.

Укажите наименьшее числовое и кратчайшее кодовое слово для буквы К.

Решение:

Рассмотрим возможные коды в порядке возрастания длины и числового значения: 0 — нельзя, это буква Р; 1 — нельзя, буквы Е и К

начинаются с 1; 01 — нельзя из-за P; 10 — нельзя из-за E; 11 — нельзя из-за A; 000 и 001 — нельзя из-за P; 101 — можно использовать.

Ответ: 101.

- 18** Для пяти букв латинского алфавита заданы их двоичные коды (для некоторых букв — из двух бит, для некоторых — из трёх). Эти коды представлены в таблице.

a	b	c	d	e
100	110	011	01	10

Какой набор букв закодирован двоичной строкой 1000110110110? Все буквы в последовательности разные.

Решение:

Мы видим, что условие Фано и обратное условие Фано не выполняются, значит, код можно декодировать неоднозначно. Будем пробовать разные варианты, отбрасывая те, в которых получаются повторяющиеся буквы.

Первая буква определяется однозначно, её код 100: a.

Вторая буква может быть 011 (c) или 01 (d).

Пусть вторая буква — 011 (c), тогда следующая буква — 01 (d), потом — 10 (e) и 110 (b) — 100 011 01 10 110.

Если вторая буква 01 (d), то получаем такое кодирование: 100 01 10 110 110. Однако это решение нам не подходит, т. к. 110 повторяется 2 раза, что противоречит условию задачи.

Следовательно, первое решение удовлетворяет условию, окончательный ответ: acdeb.

Ответ: acdeb.

- 19** Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную

двоичную последовательность. Вот этот код: А — 1; Б — 0100; В — 000; Г — 011; Д — 0101. Требуется сократить для одной из букв длину кодового слова так, чтобы код по-прежнему можно было декодировать однозначно. Коды остальных букв меняться не должны. Каким из указанных способов это можно сделать?

- 1) Для буквы Г — 11.
- 2) Для буквы В — 00.
- 3) Для буквы Г — 01.
- 4) Это невозможно.

Решение:

Для однозначного декодирования получившееся в результате сокращения кодовое слово не должно быть началом никакого другого. Первый вариант ответа не подходит, поскольку код буквы А является началом кода буквы Г. Второй вариант ответа подходит. Третий вариант ответа не подходит, поскольку в таком случае код буквы Г является началом кода буквы Д. Значит, правильный ответ указан под номером 2.

Ответ: 2.

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.

Клод Шеннон, создатель теории информации и основоположник цифровой связи, разработал общую схему передачи информации.

Процесс передачи информации/сообщения/сигнала идёт от источника,

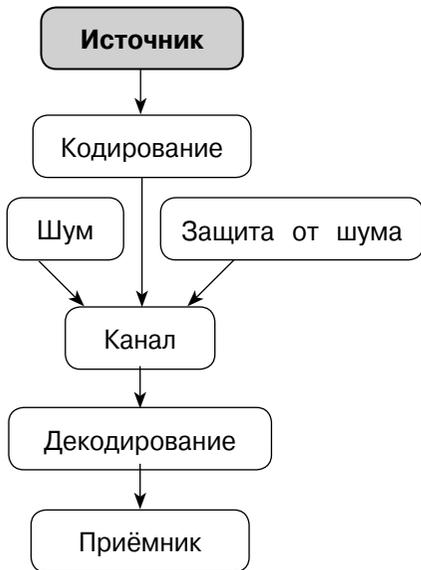
далее сигнал кодируется и по каналу связи передаётся (иногда с шумом и помехами). Перед тем как попасть к приёмнику, сообщение декодируется.

Участники процесса передачи информации:

▲ **источник** (тот, кто отправляет информацию);

▲ **приёмник** (тот, кто принимает информацию, причём их может быть несколько);

▲ **канал связи** (то, по чему передаётся информация).



Практическое задание

20

Каких участников процесса передачи информации можно назвать из приведённого ниже текста?
«Саша получил письмо по электронной почте».

Решение:

Саша получил письмо, значит, Саша — приёмник.

Получил по электронной почте, т. е. Интернет / электронная сеть — канал.

Ответ: приёмник, канал.

АНАЛОГОВОЕ И ДИСКРЕТНОЕ ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

В жизни мы постоянно слышим или видим какие-нибудь сигналы: сигналы светофора, сирену полицейской машины или скорой помощи. Сигнал — способ передачи 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 ин-

формацию из одной формы в другую (кодировать и декодировать).

Формы представления информации: аналоговая и дискретная.

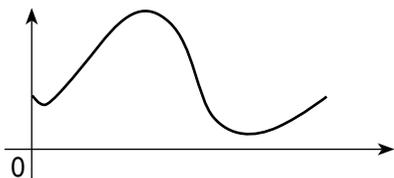
▲ **Аналоговый (непрерывный) сигнал.** Может меняться в любой момент времени.

✓ Речь человека, пение птиц, кардиограмма.

▲ **Дискретный (цифровой) сигнал.** Может меняться в определённый момент времени и принимать заранее определённые значения.

✓ Сигналы светофора, азбука Морзе, текст в книге.

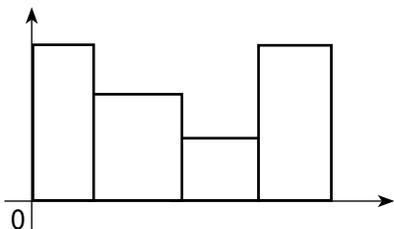
Непрерывные (аналоговые) информационные сигналы могут принимать любые значения из всех возможных в рамках заданного интервала, т. е. бесконечное множество значений.



Пример аналоговой функции

В аналоговой форме сигнал описывается непрерывной функцией времени.

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.



Пример дискретной функции

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, поэтому в данном учебном пособии ей уделено основное внимание.

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.



Практические задания

- 21** Определите, какие сигналы являются аналоговыми, а какие — дискретными: скорость автомобиля, картинка на мониторе, температура воздуха на улице.

Решение:

Скорость автомобиля, температура воздуха на улице — аналоговые сигналы, т. к. они могут изменяться в любой момент времени. Изменение происходит плавно, без резких скачков. Картинка на мониторе — дискретный сигнал. Яркость луча изменяется скачком: есть луч (яркая белая или цветная точка), нет луча (чёрная точка).

Ответ: скорость и температура — аналоговые сигналы; картинка на мониторе — дискретный.

22

При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 17 символов и содержащий только символы из 8-символьного набора: А, В, С, D, Е, F, G, H. В базе данных для хранения сведений о каждом пользователе отведено одинаковое минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт, одно и то же для всех пользователей.

Для хранения сведений о 25 пользователях потребовалось 320 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число — количество байт.

Решение:

Для кодирования 8-символьного алфавита требуется 3 бита (т. к. $2^3 = 8$). Для хранения 17 символов требуется $3 \cdot 17 = 51$ бит.

Переведём в байты: $51 : 8 = 6,375$. Таким образом, минимальное количество байт, вмещающее в себя 51 бит, — 7 (округляем 6,375 до большего целого числа).

Найдём, сколько байт понадобится на одного пользователя: $350 : 25 = 14$ байт. Из них 7 байт отводится на пароль. Значит, остальные 7 байт выделены для хранения дополнительных сведений.

Ответ: 7.

23 При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 11 символов и содержащий только символы А, Б, В, Г, Д, Е. Каждый такой пароль в системе записывается минимально возможным и одинаковым целым количеством байт (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством бит).

Определите объём памяти, отводимый системой для записи 70 паролей. В ответе запишите только целое число — количество байт.

Решение:

С помощью k бит можно закодировать 2^k различных символов. Для кодирования 6-символьного алфавита требуется 3 бита, т. к. $2^2 < 6 < 2^3$. Для хранения 11 символов потребуется $11 \cdot 3 = 33$ бита. Переведём в байты: $33 : 8 = 4,125$. Значит, минимальное количество байт, вмещающее в себя 33 бита, — 5 (округляем 4,125 до большего целого числа).

Тогда 70 паролей занимают $5 \cdot 70 = 350$ байт.

Ответ: 350.

ИСКАЖЕНИЕ ИНФОРМАЦИИ

Очень часто при передаче информации возникают помехи, и тогда информация от источника к приёмнику поступает в искажённом виде.

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. Эти помехи (шумы) создают проблемы. Существует на-

ука, 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 обозначает

случайный шум физической (чаще всего тепловой) природы. Шум характеризуется своим частотным спектром, распределением амплитуды и источником, т. е. происхождением.



Практическое задание

24 По каналу связи с помощью равномерного двоичного кода передаются сообщения, содержащие только четыре буквы: П, Р, С, Т. Каждой букве соответствует своё кодовое слово, при этом для набора кодовых слов выполнено следующее свойство: любые два слова из набора отличаются не менее чем в трёх позициях. Это свойство важно для расшифровки сообщений при наличии помех. Для кодирования букв П, Р, С используются 5-битовые кодовые слова: П — 01111, Р — 00001, С — 11000. 5-битовый код буквы Т начинается с 1 и оканчивается 0. Определите кодовое слово для буквы Т.

Решение:

- 1) Для решения задачи расположим данные четыре кода букв друг под другом и будем подбирать код для Т по условию «любые два слова из набора отличаются не менее чем в трёх позициях».
- 2) 01111 — П, 00001 — Р, 11000 — С, 10110 — Т.

Ответ: 10110.

СКОРОСТЬ ПЕРЕДАЧИ ИНФОРМАЦИИ

Обмен информацией или её передача производится по каналам связи. **Канал связи** — технические средства, позволяющие осуществлять передачу данных на расстоянии. Например, каналами связи являют-

ся органы чувств человека, техническими информационными каналами являются телефон, компьютерные сети. Информация по каналам связи движется с определённой скоростью.

Скорость передачи информации — количество информации, передаваемое в единицу времени.

При увеличении скорости передачи информации v в определённое количество раз время передачи t уменьшится в такое же количество раз (при неизменном размере файла). И наоборот, при уменьшении скорости передачи информации в определённое количество раз время передачи увеличится в такое же количество раз (при неизменном размере файла).

При увеличении скорости передачи информации v в определённое количество раз количество передаваемой информации I может быть увеличено в такое же количество раз (при неизменном времени передачи). И наоборот, при уменьшении скорости передачи информации v в определённое количество раз количество информации I , которое может быть передано, уменьшится в такое же количество раз (при неизменном времени).

Скорость измеряется в битах в секунду (бит/с) и в кратных им ве-

личинах — Кбит/с, Мбит/с, а также в байтах в секунду (Б/с) и кратных им величинах — Кбайт/с, Гбайт/с.

Пропускная способность канала — 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:

$$I = v \cdot t,$$

где I — объём передаваемого файла, v — скорость передачи файла, t — время передачи файла.

Из формулы объёма можно вывести формулу скорости передачи инфор-

мации: $v = \frac{I}{t}$.



Практические задания

25 Скорость передачи информации через соединение равна 256 000 бит/с. Передача файла через это соединение заняла 2 мин. Определите размер файла в килобайтах.

Решение:

1) Чтобы определить размер файла, нужно найти его объём по формуле, т. е. умножить время передачи на скорость передачи:

$$l = t \cdot v = 120 \text{ с} \cdot 256 \text{ 000 бит/с} = 30 \text{ 720 000 бит.}$$

2) Переведём биты в килобайты:

$$30 \text{ 720 000 бит} : 8 = 3 \text{ 840 000 байт} : 1024 = 3750 \text{ Кбайт.}$$

Ответ: 3750 Кбайт.

26 Файл размером 128 Кбайт передаётся через некоторое соединение со скоростью 2048 бит/с. Определите размер файла (в Кбайтах), который можно передать за то же время через другое соединение со скоростью 128 бит/с.

В ответе укажите одно число — размер файла в килобайтах. Единицы измерения писать не нужно.

Решение:

Способ 1 (математический).

128 Кбайт — 2048 бит/с

x Кбайт — 128 бит/с

Решая пропорцию, найдём x : $2048x = 128 \cdot 128$, отсюда $x = 8$ Кбайт.

Способ 2 (с применением формул).

Переведём скорость в байт/с: 2048 бит/с = 256 байт/с.

128 Кбайт = $128 \cdot 2^{10}$ байт.

Используем формулу скорости передачи информации, найдём t_1 .

$$t_1 = l : v = 128 \cdot 2^{10} : 256 = 2^9 \text{ с.}$$

Зная, что время одинаковое, вычислим объём.

$$l = vt = 128 \cdot 2^9 \text{ бит} = 2^7 \cdot 2^9 \text{ бит} = 2^{16} : 2^{13} \text{ Кбайт} = 2^3 \text{ Кбайт} = 8 \text{ Кбайт.}$$

Ответ: 8.

27 Файл размером 1200 Кбайт передаётся через некоторое соединение в течение 1 мин. Определите размер файла (в Кбайтах), который можно передать через это соединение за 48 с.

В ответе укажите одно число — размер файла в килобайтах. Единицы измерения писать не нужно.

Решение:

Способ 1 (математический).

$$1200 \text{ Кбайт} — 1 \text{ мин} = 60 \text{ с}$$

$$x \text{ Кбайт} — 48 \text{ с}$$

Решая пропорцию, найдём x : $60x = 1200 \cdot 48$, отсюда $x = 960$ Кбайт.

Способ 2 (с применением формул).

Используем формулу скорости передачи информации, найдём v_1 .

$$v_1 = l : t = 1200 : 60 = 20 \text{ Кбайт/с.}$$

Зная, что скорость одинаковая, вычислим объём.

$$l = vt = 20 \cdot 48 = 960 \text{ Кбайт.}$$

Ответ: 960.

28 Файл размером 15 Кбайт передаётся через некоторое соединение за 60 с. Определите, за сколько секунд можно передать этот же файл через соединение, скорость которого больше на 1024 бит/с.

В ответе укажите одно число — количество секунд. Единицы измерения писать не нужно.

Решение:

Сделаем краткую запись задачи:

$$l_1 = 15 \text{ Кбайт}$$

$$t_1 = 60 \text{ с}$$

$$v_2 = v_1 + 1024 \text{ бит/с}$$

$$t_2 = ?$$

Используем формулу скорости передачи информации, найдём v_1 .

$$v_1 = l_1 : t_1 = 15 \cdot 2^{13} : 60 = \frac{1}{4} \cdot 2^{13} = 2^{11} \text{ бит/с} = 2048 \text{ бит/с.}$$

Зная, что объём одинаковый, а $v_2 = v_1 + 1024$ бит/с, найдём время.

$$l_1 = l_2 = 15 \text{ Кбайт, } v_2 = 3072 \text{ бит/с} = 3 \cdot 2^{10}, t_2 = 15 \cdot 2^{13} : 3 \cdot 2^{10} = 40 \text{ с.}$$

Ответ: 40.

- 29** Каково время (в минутах) передачи полного объёма данных по каналу связи, если известно, что передано 1200 Мбайт данных, причём треть времени передача шла со скоростью 60 Мбит/с, а остальное время — со скоростью 90 Мбит/с?

Решение:

1) Обозначим время передачи t . Тогда $t_1 = \frac{1}{3}t$, $t_2 = \frac{2}{3}t$.

2) Обозначим Q — количество данных, q — скорость передачи. Переведём Q в Мбиты: 1200 Мбайт = 9600 Мбит.

3) Составим уравнение по формуле $qt = Q$:

$$\frac{1}{3}t \cdot 60 + \frac{2}{3}t \cdot 90 = 9600; \quad t(20 + 60) = 9600.$$

4) Отсюда находим время: $t = 9600 : 80 = 120 \text{ с} = 2 \text{ мин}$.

Ответ: 2 мин.

- 30** У Васи есть доступ к сети Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации 2^{17} бит/с. У Пети нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по низкоскоростному телефонному каналу со средней скоростью 2^{15} бит/с. Петя договорился с Васей, что тот будет скачивать для него данные объёмом 4 Мбайт по высокоскоростному каналу и ретранслировать их ему по низкоскоростному каналу. Компьютер Васи может начать ретрансляцию данных не раньше чем им будут получены первые 512 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Петей? В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Решение:

Нужно определить, сколько времени будет передаваться файл объёмом 4 Мбайт по каналу со скоростью передачи данных 2^{15} бит/с; к этому

времени нужно добавить задержку файла у Васи (пока он не получит 512 Кбайт данных по каналу со скоростью 2^{17} бит/с).

1) Переведём все данные в биты: $4 \text{ Мбайт} = 4 \cdot 2^{20} \text{ байт} = 2^{25} \text{ бит}$,
 $512 \text{ Кбайт} = 512 \cdot 1024 \text{ байт} = 512 \cdot 1024 \cdot 8 \text{ бит} = 2^9 \cdot 2^{10} \cdot 2^3 = 2^{22} \text{ бит}$.

2) Время задержки: $t_0 = 2^{22} \text{ бит} : 2^{17} \text{ бит/с} = 2^5 \text{ с}$.

3) Время скачивания данных Петей: $t_1 = 2^{25} \text{ бит} : 2^{15} \text{ бит/с} = 2^{10} \text{ с}$.

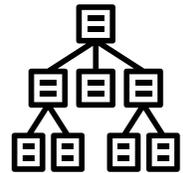
4) Полное время: $t = t_0 + t_1 = 2^{10} \text{ с} + 2^5 \text{ с} = (1024 + 32) \text{ с} = 1056 \text{ с}$.

Ответ: 1056.



СИСТЕМА, ЕЁ СВОЙСТВА И КОМПОНЕНТЫ

Понятие «система» относится к основным как в науке, так и в обычной жизни. Каждая система обладает определённой структурой и предполагает взаимодействие её компонентов друг с другом и с окружающей средой.



ПОНЯТИЕ СИСТЕМЫ

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

Данное понятие применяется по отношению к самым различным предметам, явлениям и процессам. Примерами систем являются коллектив людей, живой организм, тех-

ническое устройство, организация здравоохранения и т. д. Они состоят из множества частей, каждая из которых работает во взаимодействии с другими для создания целого.

Системы отличаются между собой как по составу, так и по целям. Рассмотрим несколько примеров систем подробнее.

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
Информационная система	
Компьютеры, компьютерные сети, люди, информационное и программное обеспечение	Производство профессиональной информации

ХАРАКТЕРИСТИКИ И СВОЙСТВА СИСТЕМЫ

Каждая система обладает определёнными характеристиками: функция (цель, назначение) системы, состав системы (элементы), структура 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)

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 из отделов 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

Понятие информационной системы можно трактовать как в широком, так и в узком смысле.

В широком смысле **информационная система (ИС)** — это любая система, предназначенная для обработки информации. Закон РФ «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».

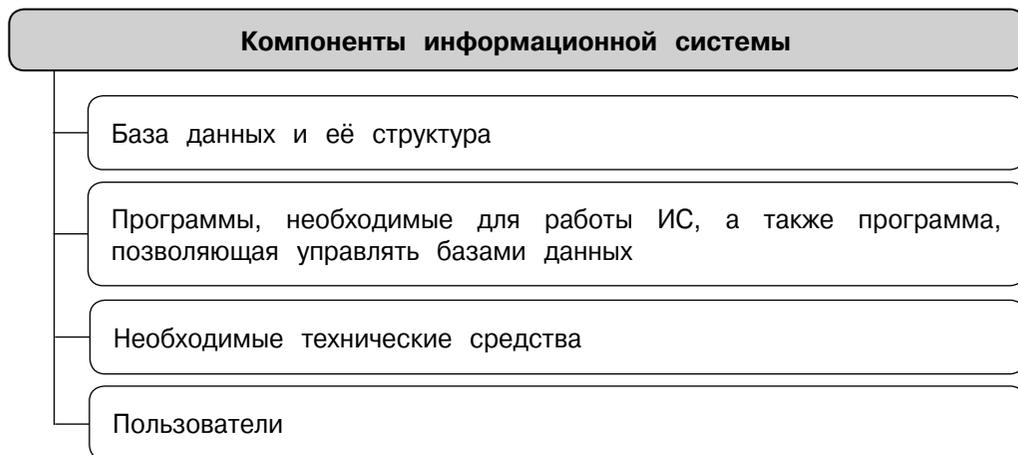
В узком смысле **информационная система** — совокупность аппаратных и программных средств, предназначенных для хранения, обработки и применения большого объёма информации. Современное понимание информационной системы предполагает использование компьютера в качестве основного технического средства переработки информации.

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

Логическая архитектура включает комплекс методов, решений и алгоритмов, по которым работает система, обеспечивает функционирование и взаимодействие подсистем на логическом уровне на протяжении всего жизненного цикла.

Физическая архитектура включает в себя элементы, обеспечивающие работу системы, т. е. является комплексом вычислительной техники, средств связи между элементами системы и обслуживающим персоналом.

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



■ Классификация информационных систем

Существует множество классификаций информационных систем. Выделим несколько из них.

По степени автоматизации:

- ▲ ручные;
- ▲ автоматические;
- ▲ автоматизированные.

По целевому назначению:

- ▲ управляющие;
- ▲ информационно-справочные;
- ▲ системы принятия решений;
- ▲ информационно-поисковые и т. п.

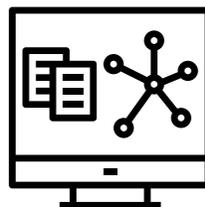
По сфере применения:

- ▲ в образовании;
- ▲ в торговле;
- ▲ в юриспруденции;
- ▲ в науке;
- ▲ в здравоохранении и т. п.



МОДЕЛИРОВАНИЕ

Практически во всех науках построение и использование моделей — часто применяемый инструмент для познания и исследования. Реальные процессы и объекты очень многогранные и сложные, поэтому наилучшим способом изучения является построение модели. Она отображает только какую-то часть реальности и потому значительно проще, чем сама реальность.



КЛАССИФИКАЦИЯ, ЦЕЛИ И ЭТАПЫ МОДЕЛИРОВАНИЯ

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

■ Классификация моделей

По форме представления:

- ▲ материальные;
- ▲ информационные:
 - знаковые:
 - компьютерные;
 - некомпьютерные;
 - вербальные.

С учётом фактора времени:

- ▲ статические;
- ▲ динамические;

- дискретные;
- непрерывные.

По области использования:

- ▲ учебные;
- ▲ опытные;
- ▲ научно-технические;
- ▲ игровые;
- ▲ имитационные.

По характеристике объекта моделирования:

- ▲ модели внешнего вида:
 - ✓ манекен;
- ▲ модели структуры:
 - ✓ анатомический атлас человека;
- ▲ модели поведения:
 - ✓ велотренажёр.

По степени формализации:

- ▲ неформализованные:

- ✓ описание внешности героя литературного произведения;
- ▲ частично формализованные:
 - ✓ кулинарный рецепт;
 - ▲ формализованные:
 - ✓ компьютерная программа.

Рассмотрим подробнее некоторые из представленных видов моделей.

■ По форме представления

Материальные модели

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.

Информационные модели

Целенаправленно отобранная информация об объекте, которая отражает наиболее существенные для исследователя свойства этого объекта.

- ✓ Математическая формула, блок-схема, географическая карта, ноты, химические формулы.

Информационные модели бывают знаковыми и вербальными.

▲ **Вербальные модели** — такие модели, которые могут быть представлены в мысленной или разговорной форме.

- ✓ Ход решения уравнения или задачи.

▲ **Знаковыми** называются модели, выраженные специальными знаками любого формального языка.

- ✓ Схема метро, чертёж шкафа.

Знаковые модели, в свою очередь, делятся на компьютерные и некомпьютерные.

• **Компьютерными** являются модели, реализованные с применением программных средств.

- ✓ Движение объекта в определённой среде, 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

Модели, которые не меняются с течением времени.

- ✓ Фотография.

Динамические модели

Модели, которые способны меняться с течением времени.

- ✓ График изменения температуры.

Динамические модели делятся на **дискретные** и **непрерывные**.

▲ **Дискретные модели** описывают прерывистые процессы.

- ✓ Модель численности населения.

- ▲ **Непрерывные модели** отражают непрерывные процессы.
- ✓ Отображение закона Ньютона в виде формулы.

■ По области использования

Учебные модели

- Модели, которые используются в процессе обучения.
- ✓ Тренажёр для полётов.

Опытные модели

- Увеличенные или уменьшенные копии реально проектируемых объектов.
- ✓ Модель корабля/автомобиля/здания.

Научно-технические модели

- Модели, созданные для исследования процессов или явлений.
- ✓ Прибор для получения грозового электрического разряда.

Игровые модели

- Модели, созданные специально для изучения поведения людей в различных ситуациях.
- ✓ Военные игры, экономическая игра «Монополия».

Имитационные модели

- Модели, отражающие логику и закономерности поведения моделируемых объектов во времени и пространстве.
- ✓ Модель call-центра, линии производства на заводе.



- ▲ Для одного объекта один субъект может построить несколько моделей, если он решает разные задачи, приводящие к разным целям моделирования.

- ▲ Для одного объекта разные субъекты могут построить разные модели, даже если задача моделирования у них одна. Выбор вида модели и её построение зависят от знаний, опыта, предпочтений, личных интересов субъекта.

- ▲ Разные объекты могут иметь одинаковые по виду модели, даже если их строили совершенно разные субъекты, исходя при этом из разных целей моделирования.

■ Цели моделирования

Познание окружающего мира

Зачем человек создаёт модели и какова их роль в постижении законов окружающего мира? 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, просто 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?»

Определение последствий воздействия на объект и принятие правильного решения

Речь идёт о решении задачи типа «Что будет, если...». Например, что будет, если увеличить плату за проезд в транспорте и одновременно снизить налоговую нагрузку на отдельные категории работников, или что произойдёт, если закопать ядерные отходы в какой-то местности.

Эффективность управления объектом (или процессом)

Управление станет эффективным при условии, если будет соблюдено большинство критериев и задач. Решается задача типа «Как совместить несовместимое?». Например, как сделать питание в столовой качественным, чтобы при этом оно нравилось ребятам и было по карману родителям.



ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ

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

Формы представления информационных моделей:

- ▲ в виде сигналов;
- ▲ устная, словесная;
- ▲ символная (числа, текст, символы, формулы);
- ▲ 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, у каждого из которых есть 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.



Практические задания

31 Встретились три подруги — Белова, Краснова и Чернова. На одной из них было чёрное платье, на другой — красное, на третьей — белое. Девушка в белом платье сказала Черновой: «Нам троим надо поменяться платьями, а то цвета наших платьев не соответствуют нашим фамилиям». В платье какого цвета была одета каждая из подруг?

Решение:

Задача легко решается, если представить её условие в виде таблицы. В таблице расставим прочерки, опираясь на условия задания. Очевидно, что у Беловой не белое, у Черновой не чёрное и у Красновой не красное платье. Также девушка в белом не Чернова, т. к. она не могла говорить самой себе.

Фамилия/Платье	Белое	Чёрное	Красное
Белова	–		
Чернова	–	–	
Краснова			–

Дальше из таблицы видим, что у Черновой может быть только красное платье. Значит, у остальных красного платья быть не может.

Фамилия/Платье	Белое	Чёрное	Красное
Белова	–		–
Чернова	–	–	+
Краснова			–

У Беловой не красное платье, значит, оно чёрное. Получается, у Красновой — белое. Заполняем окончательный вариант таблицы.

Фамилия/Платье	Белое	Чёрное	Красное
Белова	–	+	–
Чернова	–	–	+
Краснова	+	–	–

Ответ: у Беловой чёрное платье, у Черновой — красное, у Красновой — белое.

- 32** Катя сказала: «У нас в понедельник 5 уроков (математика, русский язык, литература, физкультура и ОБЖ), а во вторник — 4 урока (математика, история, музыка и литература)». Представьте это высказывание в виде информационной модели.

Ответ:

Представим высказывание в виде таблицы.

Понедельник	Вторник
Математика	Математика
Русский язык	История
Литература	Музыка
Физкультура	Литература
ОБЖ	

В процессе изучения информатики мы чаще всего встречаемся со знаковыми моделями. Рассмотрим их подробнее, в частности компью-

терные и математические модели (наиболее интересный для нас вид некомпьютерных моделей) и способы их представления.

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.

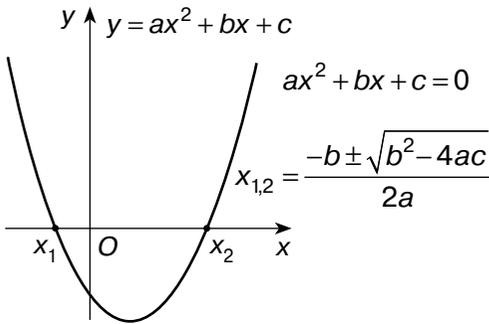
Над элементами математической модели можно выполнять определённые математические преобразования. Например, в модели нахождения наименьшего числа выполняются операции сравнения, а в модели вычисления корня урав-

нения — различные арифметические операции.

С помощью математических моделей описываются решения различных инженерных задач, многие физические процессы (движение планет, автомобиля и т. п.), технологические процессы (сварка, плавление металла и т. п.).

Графики, таблицы, диаграммы позволяют отображать различные закономерности и зависимости реального мира. Например, 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.

В настоящее время расчёты для большинства математических моделей производят на компьютерах, используя специальные прикладные программные комплексы, которые позволяют:

- ▲ в несколько раз сократить время проведения исследований;
- ▲ уменьшить количество участников эксперимента;
- ▲ повысить точность и достоверность эксперимента, а следовательно, увеличить контроль;
- ▲ за счёт средств графической визуализации, например анимации, получить реальную картинку;
- ▲ повысить качество и информативность эксперимента за счёт увеличения числа контролируемых параметров и более точной обработки данных; например, на экране компьютера возможно сформировать целую систему приборов, которые будут отслеживать изменение параметров объекта.

Приведём несколько примеров математических моделей.

Реальная ситуация	Математическая модель
У Даши и Саши одинаковое количество конфет	$x = y$
У Кости на семь пятёркок 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

Компьютерное моделирование — метод решения задачи с использованием компьютера. При создании какого-либо нового объекта возникает необходимость протестировать его до запуска в массовое производство и использование. 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.

При исследовании с целью прогноза погоды на ближайшие дни создаётся компьютерная модель, которая на основе имеющихся данных помогает установить параметры, ожидаемые завтра или через неделю.

В настоящее время существует множество аппаратов, которые способны воспроизвести головной мозг или коленный сустав человека, показать его внутренние проблемы, например аппарат МРТ. Все данные передаются на компьютер, врач изучает снимки и делает соответствующее заключение.

Учитель может использовать на уроке готовые компьютерные модели для демонстрации изучаемого явления, например круговорота воды в природе, движения планет, взаимодействия молекул или атомов. В Интернете можно найти различные виртуальные лаборатории для

исследования реальных процессов или явлений. Очень удобно проводить опыты в таких виртуальных лабораториях, особенно если в реальности они представляют опасность для жизни.

Ещё одним примером компьютерной модели является симулятор. **Симулятор** — это комплекс программных и технических средств, которые имитируют управление каким-либо процессом, устройством. В качестве примеров можно привести компьютерные игры, симуляторы вождения или полёта.

■ Области применения и этапы компьютерного моделирования

Одной из наиболее интересных областей компьютерного моделирования является моделирование процессов мышления, процессов распознавания образов, речи. В наши дни уже активно используются системы распознавания лиц в метро, управления некоторыми видами техники с помощью голоса.

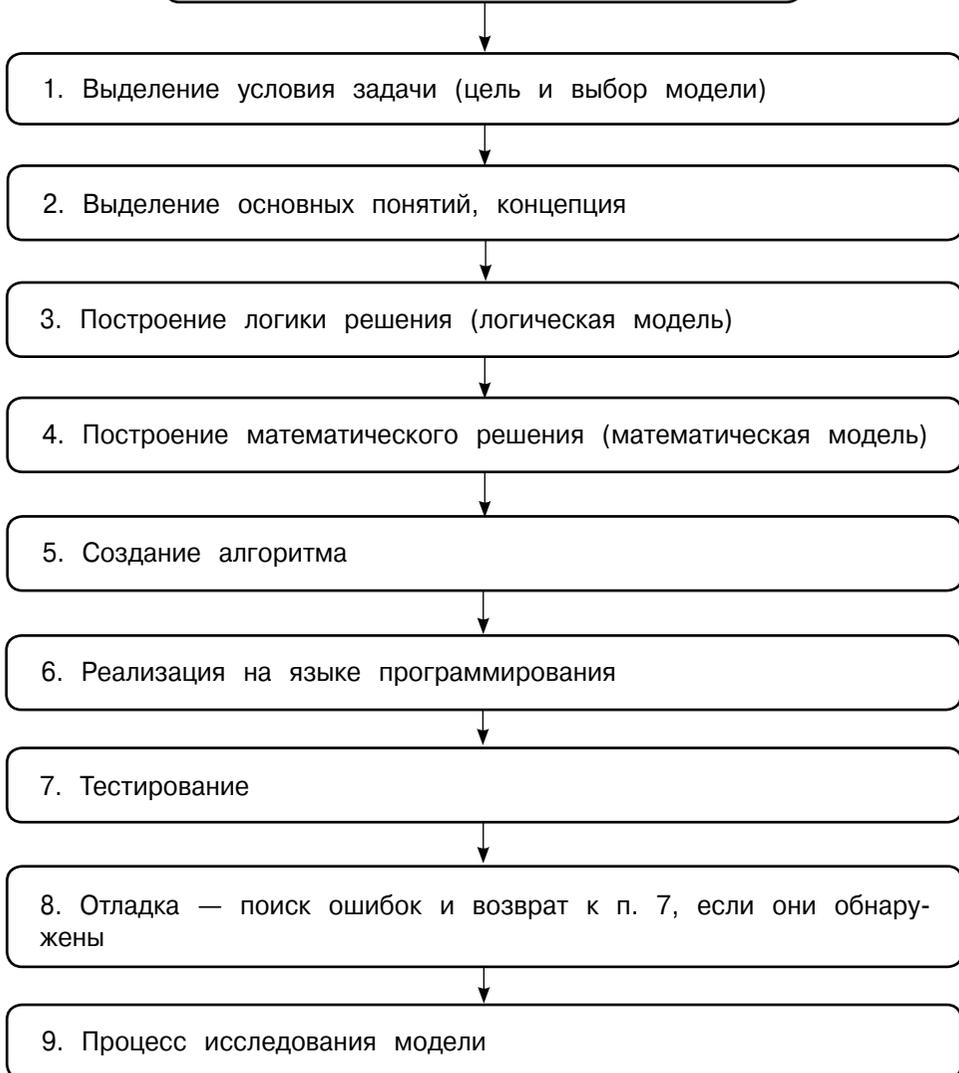
Области применения компьютерного моделирования:

- ▲ научные исследования, образование;
- ▲ разработка бытовых и промышленных приборов;
- ▲ системы безопасности;

▲ информационные и справочные службы;
 ▲ компьютерные сети;

▲ мобильные устройства (КПК, смартфоны, коммуникаторы и др.);
 ▲ искусство и дизайн.

Этапы компьютерного моделирования



ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

Имитационное моделирование — метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему (модель описывает процессы так, как они происходили бы в действительности). С моделью проводятся эксперименты с целью получения информации об этой системе. Такую модель можно «проиграть» во времени как для одного испытания, так и для заданного множества. При этом результаты будут определяться случайным характером процессов. По этим данным можно получить достаточно устойчивую статистику. Экспериментирование с моделью называют имитацией (постижение сути явления без проведения экспериментов на реальном объекте). 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 поведе-

ния 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

Дискретно-событийное моделирование

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 Джеффри Гордоном в 1960-х гг.

Системная динамика

Парадигма моделирования, в которой 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 Джейм Форрестером в 1950-х гг.

Агентное моделирование

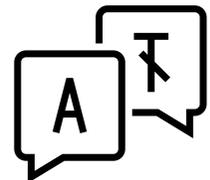
Относительно новое (1990—2000-е гг.) направление в имитационном моделировании. Используется для исследования децентрализованных систем, динамика функционирования которых определяется не

глобальными правилами и законами (как в других парадигмах моделирования), а, наоборот, 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

У разных народов в процессе развития сформировались свои изображения алфавита и чисел. Потребность в создании начертания чисел и букв появилась у человека ещё в древние времена, когда было необходимо записать и запомнить большое количество информации.



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

Знаковая система — система, состоящая из алфавита и правил составления слов и формул из этого алфавита.

■ Языки

Примером знаковой системы является **язык**. Основными составляющими языка служат:

- ▶ алфавит;
- ▶ грамматика (свод правил, согласных которым знаки превращаются в слова);
- ▶ синтаксис (слова формируются в предложения);
- ▶ семантика (смысловое значение единиц языка).

Естественные языки

К естественным относятся языки, которые сложились исторически.

- ✓ Русский, английский и т. п.

Формальные языки

Формальными называются языки, созданные для определённой цели.

- ✓ Языки логики, C++, Pascal.

■ Виды систем счисления

Система счисления — формальный язык, предназначенный для записи чисел.

Позиционные системы

В таких системах значение каждой цифры зависит от её положения в числе (разряда).

- ✓ Десятичная, двоичная системы счисления.

Непозиционные системы

В непозиционных системах значение цифры не зависит от её положения в числе (разряде).

- ✓ Унарная, древнерусская и др.
- ✓ Римская система счисления.

I = 1	C = 100
V = 5	D = 500
X = 10	M = 1000
L = 50	

Так образуются другие числа в римской системе счисления:

I = 1	XI = 11
II = 2	XII = 12
III = 3	XIII = 13
IV = 4	XIV = 14
V = 5	XV = 15
VI = 6	XVI = 16
VII = 7	XVII = 17
VIII = 8	XVIII = 18
IX = 9	XIX = 19
X = 10	XX = 20



Пример формирования чисел:

- ▶ в позиционной системе: 23 и 32 — разные числа;
- ▶ в непозиционной системе: CM = MC = 1100.

ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ

Позиционная система счисления определяется натуральным чис-

лом $k > 1$, называемым **основанием системы счисления**.

Систему с основанием 10 называют **десятичной**, с основанием 2 — **двоичной**, в общем случае с основанием k — **k -ичной**.

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

■ Свойства записи чисел

Для записи чисел в позиционной системе счисления с основанием k требуется k знаков (алфавит системы состоит из k цифр или букв).

Основание системы счисления, записанное в ней, всегда имеет вид 10 (читается «один ноль»).

С помощью n разрядов в позиционной системе счисления с основанием k могут быть записаны k^n чисел (от 0 до $k^n - 1$).

Если основание системы k больше 10, то числа старше 10 при записи обозначают прописными буквами латинского алфавита: А, В...Z. При этом числу 10 соответствует знак А, числу 11 — знак В и т. д.

При записи чисел в позиционной системе, отличной от десятичной, принято писать основание справа внизу: 34_5 — число дано в пятеричной системе счисления.

! При записи числа в позиционной системе счисления все цифры должны быть меньше основания:

в числе, данном в пятеричной системе счисления, могут присутствовать цифры от 0 до 4, цифр 7 или 5 быть не может.

Развёрнутая форма записи числа выполняется в виде разрядных слагаемых, записанных с помощью степени соответствующего разряда и основания степени (основания счёта).

Свёрнутая запись числа:

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

Развёрнутая запись числа:

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

✓ Десятичная система

Свёрнутая запись числа: 95438_{10} .

Пронумеруем разряды, начиная с младшего.

4	3	2	1	0
9	5	4	3	8

Развёрнутая запись числа: $95438_{10} = 9 \cdot 10^4 + 5 \cdot 10^3 + 4 \cdot 10^2 + 3 \cdot 10^1 + 8 \cdot 10^0$.

ЦИФРЫ ШЕСТНАДЦАТЕРИЧНОЙ СИСТЕМЫ СЧИСЛЕНИЯ

16-ричная система	0	1	2	3	4	5	6	7	8	9
Соответствие в десятичной системе	0	1	2	3	4	5	6	7	8	9

16-ричная система	A	B	C	D	E	F
Соответствие в десятичной системе	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

✓ Двоичная система

Свёрнутая запись числа: 1011010_2 .

Пронумеруем разряды, начиная с младшего.

6	5	4	3	2	1	0
1	0	1	1	0	1	0

Развёрнутая запись числа: $1011010_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$.

Другие системы счисления записываются аналогично вышеприведённым системам с тем лишь исключением, что основание степени будет соответствовать основанию счёта. Так, девятеричная система будет иметь следующий алфавит: 0 1 2 3 4 5 6 7 8.

✓ Запишем алфавит для двенадцатеричной системы счисления: 0 1 2 3 4 5 6 7 8 9 A B.

ДВОИЧНОЕ ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

В двоичной системе счисления используются всего две цифры: 0 и 1.

Чтобы научиться понимать числа в двоичной системе счисления, сна-

чала рассмотрим, как формируются числа в привычной для нас десятичной системе.

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. После 19 разряд десятков увеличивается на 1, а единицы снова обнуляются и т. д. Когда десятки доходят до 9, появляется третий разряд — сотни.

```

0 1 2... 9
      10 11 12... 19
            20 21... 99
                  100 101 102
    
```

Двоичная система счисления аналогична десятичной, за исключением того, что в формировании числа участвуют всего лишь 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

Нетрудно заметить, что в двоичной системе счисления длины чисел с увеличением значения растут быстрыми темпами. Как определить, что значит такая запись: 10001001? Непривычный к такой форме записи чисел человеческий мозг обычно не может понять, сколько это. Неплохо бы уметь переводить двоичные числа в десятичные.

В десятичной системе счисления любое число можно представить

в виде 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 поочередно умножаются на 10, возведённое в ту или иную степень. 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:

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

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:

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

✓ Переведите 75_{10} в двоичную систему.

Решение:

$$\begin{array}{r}
 75 \left| \begin{array}{l} 2 \\ -74 \\ \hline 1 \end{array} \right. \begin{array}{l} 2 \\ -37 \\ \hline 1 \end{array} \begin{array}{l} 2 \\ -18 \\ \hline 0 \end{array} \begin{array}{l} 2 \\ -9 \\ \hline 1 \end{array} \begin{array}{l} 2 \\ -4 \\ \hline 0 \end{array} \begin{array}{l} 2 \\ -2 \\ \hline 1 \end{array} \begin{array}{l} 2 \\ -0 \\ \hline 0 \end{array}
 \end{array}$$

Ответ записываем справа налево. Получим: $75_{10} = 1001011_2$.

Ответ: $75_{10} = 1001011_2$.

✓ Число 1001001001_2 переведите в десятичную систему счисления.

Решение:

Представим это число в виде развёрнутой записи, для этого пронумеруем разряды справа налево, начиная с нуля.

9876543210

$$\begin{aligned}
 1001001001_2 &= 1 \cdot 2^9 + 0 \cdot 2^8 + 0 \cdot 2^7 + \\
 &+ 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + \\
 &+ 0 \cdot 2^1 + 1 \cdot 2^0 = 512 + 0 + 0 + 64 + 0 + \\
 &+ 0 + 8 + 0 + 0 + 1 = 585_{10}.
 \end{aligned}$$

Ответ: 585_{10} .



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

✓ Число 468_{10} переведите в двоичную систему счисления.

Решение:

Целая часть числа находится делением на основание новой системы счисления.

$$\begin{array}{r}
 468 \left| \begin{array}{l} 2 \\ -468 \\ \hline 0 \end{array} \right. \begin{array}{l} 2 \\ -234 \\ \hline 0 \end{array} \begin{array}{l} 2 \\ -117 \\ \hline 1 \end{array} \begin{array}{l} 2 \\ -58 \\ \hline 0 \end{array} \begin{array}{l} 2 \\ -29 \\ \hline 1 \end{array} \begin{array}{l} 2 \\ -14 \\ \hline 0 \end{array} \begin{array}{l} 2 \\ -7 \\ \hline 1 \end{array} \begin{array}{l} 2 \\ -3 \\ \hline 1 \end{array} \begin{array}{l} 2 \\ -2 \\ \hline 0 \end{array} \begin{array}{l} 2 \\ -1 \\ \hline 1 \end{array}
 \end{array}$$

Ответ: $468_{10} = 111010100_2$.

✓ Число 100010111011_2 переведите в десятичную систему счисления.

Решение:

Представим это число в виде развёрнутой записи, пронумеруем разряды справа налево, начиная с нуля.

11109876543210

$$\begin{aligned}
 100010111011_2 &= 1 \cdot 2^{11} + 0 \cdot 2^{10} + \\
 &+ 0 \cdot 2^9 + 0 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + \\
 &+ 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + \\
 &+ 1 \cdot 2^1 + 1 \cdot 2^0 = 2048 + 0 + 0 + \\
 &+ 0 + 128 + 0 + 32 + 16 + 8 + 0 + \\
 &+ 2 + 1 = 2235_{10}.
 \end{aligned}$$

Ответ: 2235_{10} .

■ Перевод чисел из k -ичной системы в десятичную

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

✓ расставляем числа разрядов

$$\begin{array}{l}
 \overbrace{3210}^{\text{расставляем числа разрядов}} \\
 \underbrace{3212_4}_{\text{записываем в развёрнутом виде}} = 3 \cdot 4^3 + 2 \cdot 4^2 + 1 \cdot 4^1 + 2 \cdot 4^0 = \\
 = 192 + 32 + 4 + 2 = \underbrace{230}_{\text{ответ}}_{10} \\
 \underbrace{\hspace{10em}}_{\text{выполняем сложение}}
 \end{array}$$

✓ Переведите число 152_6 в десятичную систему счисления.

Решение:

$$\begin{aligned}
 152_6 &= 1 \cdot 6^2 + 5 \cdot 6^1 + 2 \cdot 6^0 = 1 \cdot 36 + \\
 &+ 5 \cdot 6 + 2 \cdot 1 = 36 + 30 + 2 = 68_{10}.
 \end{aligned}$$

Ответ: $152_6 = 68_{10}$.

✓ расставляем числа разрядов

$$\begin{array}{l}
 \overbrace{210}^{\text{расставляем числа разрядов}} \\
 \underbrace{AC5_{16}}_{\text{записываем в развёрнутом виде}} = A \cdot 16^2 + C \cdot 16^1 + 5 \cdot 16^0 = \underbrace{10 \cdot 16^2 + 12 \cdot 16^1 + 5 \cdot 16^0}_{\text{переводим буквы в десятичную систему}} = \underbrace{2560 + 192 + 5}_{\text{выполняем сложение}} = \underbrace{2757}_{\text{ответ}}_{10}
 \end{array}$$

■ Перевод чисел из десятичной системы в k -ичную

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)

✓ Переведём 345_{10} в троичную систему, для этого будем делить столбиком на 3.

Решение:

$$\begin{array}{r}
 345 \mid 3 \\
 \underline{-345} \quad 0 \\
 115 \quad 3 \\
 \underline{-114} \quad 1 \\
 38 \quad 3 \\
 \underline{-36} \quad 2 \\
 12 \quad 3 \\
 \underline{-12} \quad 0 \\
 4 \quad 3 \\
 \underline{-3} \quad 1 \\
 1
 \end{array}$$

Записываем полученное число справа налево: 110210_3 .

Ответ: $345_{10} = 110210_3$.

✓ Переведём 349_{10} в шестнадцатеричную систему.

Целая часть числа находится делением на основание новой системы — на 16.

Решение:

$$\begin{array}{r}
 349 \mid 16 \\
 \underline{-336} \quad 13 \\
 21 \quad 16 \\
 \underline{-16} \quad 5 \\
 5 \quad 1
 \end{array}$$

Записываем полученное число справа налево: $15D_{16}$.

Ответ: $349_{10} = 15D_{16}$.



Практическое задание

33 Все четырёхбуквенные слова, составленные из букв М, Р, Т, У, записаны в алфавитном порядке и пронумерованы, начиная с 1. Ниже приведено начало списка:

1. ММММ
2. МММР
3. МММТ
4. МММУ
5. ММРМ

...

Запишите слово, которое стоит на 216-м месте от начала списка.

Решение:

Заменяем буквы М, Р, Т, У на 0, 1, 2, 3 соответственно.

Выпишем начало списка, заменив буквы на цифры:

1. 0000
2. 0001
3. 0002
4. 0003
5. 0010

...

Полученная запись представляет собой числа, записанные в четверичной системе счисления в порядке возрастания. Тогда на 216-м месте будет стоять число 215 (т. к. первое число — 0). Переведём число 215 в четверичную систему:

$$215 / 4 = 53 (3),$$

$$53 / 4 = 13 (1),$$

$$13 / 4 = 3 (1),$$

$$3 / 4 = 0 (3).$$

В четверичной системе 215 запишется как 3113. Произведём обратную замену и получим слово УРРУ.

Ответ: УРРУ.

■ Метод триад и тетрад

Иногда приходится сталкиваться с задачей перевести число 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 состоит в применении метода триад и тетрад.

ТАБЛИЦА ТРИАД И ТЕТРАД

A8	Триада	A16	Тетрада
0	000	0	0000
1	001	1	0001
2	010	2	0010
3	011	3	0011
4	100	4	0100
5	101	5	0101
6	110	6	0110
7	111	7	0111
		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.

✓ Переведите число $С3A6_{16}$ в двоичную систему счисления.

Решение:

Заменим каждую цифру числа группой из четырёх цифр (т. к. $16 = 2^4$), дополнив при необходимости в начале группу нулями:

$$С3A6_{16} = 1100\ 0011\ 1010\ 0110_2.$$

Перевод числа из двоичной системы в восьмеричную

1. Разбить число на триады (тройки цифр), начиная с младшего разряда, дополнив в случае необходимости старшую триаду нулями.
2. Заменить каждую триаду соответствующей восьмеричной цифрой (см. таблицу триад и тетрад).

✓ Число 1001011_2 перевести в восьмеричную систему счисления.

$$001\ 001\ 011_2 = 113_8$$

Перевод числа из двоичной системы в шестнадцатеричную

1. Разбить число на тетрады (четвёрки цифр), начиная с младшего

разряда, дополнив в случае необходимости старшую тетраду нулями.

2. Заменить каждую тетраду соответствующей восьмеричной цифрой (см. таблицу триад и тетрад).

✓ Число 1011100011_2 перевести в шестнадцатеричную систему счисления.

$$0010\ 1110\ 0011_2 = 2E3_{16}$$

Перевод восьмеричного числа в двоичное

Заменить каждую цифру эквивалентной ей двоичной триадой (см. таблицу триад и тетрад).

✓ Число 531_8 перевести в двоичную систему счисления.

$$531_8 = 101011001_2$$

Перевод шестнадцатеричного числа в двоичное

Заменить каждую цифру эквивалентной ей двоичной тетрадой (см. таблицу триад и тетрад).

✓ Число $EE8_{16}$ перевести в двоичную систему счисления.

$$EE8_{16} = 111011101000_2$$

Переход из восьмеричной системы счисления

в шестнадцатеричную и обратно
Необходим промежуточный перевод чисел в двоичную систему (см. таблицу триад и тетрад).

✓ Число FEA_{16} перевести в восьмеричную систему счисления.

$$\begin{aligned} EFEA_{16} &= 111111101010_2 = \\ &= 111\ 111\ 101\ 010_2 = 7752_8 \end{aligned}$$

✓ Число 6635_8 перевести в шестнадцатеричную систему счисления.

$$\begin{aligned} 6635_8 &= 110110011101_2 = \\ &= 1101\ 1001\ 1101_2 = D9D_{16} \end{aligned}$$

При возведении степени в степень показатели умножаются.



Практические задания

34 На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

- 1) Строится двоичная запись числа N .
- 2) К этой записи дописываются справа ещё два разряда по следующему правилу:
 - а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа справа (например, запись 11100 преобразуется в запись 111001);

б) над этой записью производятся те же действия: справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R . Укажите минимальное число R , которое превышает 43 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

Решение:

1) Переведём число 44 в двоичную систему счисления: $44_{10} = 101100_2$, отсюда найдём $N = 1011$ (убираем два крайних разряда).

2) Пропустим через алгоритм полученное значение N :

а) 10111;

б) 101110 — R .

3) Переведём 101110 в десятичную систему счисления: $101110_2 = 46_{10}$, что удовлетворяет условию «минимальное число R , которое превышает 43 и может являться результатом работы алгоритма», и является ответом.

Ответ: 46.

35 Сколько единиц содержится в двоичной записи значения выражения:

$$4^3 \cdot 2^8 : 8?$$

Решение:

1) Чтобы найти результат выражения, нужно привести все числа к виду «2 в степени»: $2^6 \cdot 2^8 : 2^3 = 2^{6+8-3} = 2^{11}$.

2) Считаем результат в десятичной системе, а потом переводим в двоичную: $2^{11} = 2048_{10} = 10000000000_2$.

Ответ: 1.

36 Сколько единиц содержится в двоичной записи значения выражения:

$$4^{411} + 2^{411} - 511?$$

Решение:

Рассмотрим число -511 и преобразуем его в $-512 + 1 = -2^9 + 1$.

Значит, выражение $4^{411} + 2^{411} - 511$ равно $2^{822} + 2^{411} - 2^9 + 2^0$.

Возьмём все числа с положительным знаком, получим число в двоичной записи: $1000\dots0001000\dots0001$, где единицы стоят на 1-м, 412-м и 823-м местах.

Теперь из этого числа вычтем 2^9 — в двоичной записи это 1 и 9 нулей.

$$\begin{array}{r} 1000\dots000\overset{\cdot}{1}000\dots 0000000001 \\ - 1000000000 \\ \hline 1000\dots000011\dots111000000001 \end{array}$$

Нам нужно занять из 412-й позиции единицу, чтобы вычесть 2^9 , тогда получится единица с 411-й по 10-ю позицию. Всего таких позиций 402. Также присутствует единица на 1-м и 823-м месте. Общее количество единиц в двоичной записи значения выражения: $402 + 1 + 1 = 404$.

Ответ: 404.

- 37** Среди приведённых ниже трёх чисел, записанных в различных системах счисления, найдите максимальное и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно.

$$57_{16}, 1010110_2, 125_8$$

Решение:

1) Чтобы сравнить числа, нужно перевести каждое в десятичную систему: $57_{16} = 87_{10}$; $1010110_2 = 86_{10}$; $125_8 = 85_{10}$.

2) Выберем наибольшее из чисел: $57_{16} = 87_{10}$.

Ответ: 87.

- 38** Среди приведённых ниже трёх чисел, записанных в десятичной системе счисления, найдите число, в двоичной записи

которого наименьшее количество единиц. В ответе запишите количество единиц в двоичной записи этого числа.

$$100_{10}, 90_{10}, 75_{10}$$

Решение:

Переведём все числа в двоичную систему счисления:

$$100_{10} = 1100100_2, \text{ количество единиц — 3;}$$

$$90_{10} = 1011010_2, \text{ количество единиц — 4;}$$

$$75_{10} = 1001011_2, \text{ количество единиц — 4.}$$

Таким образом, числом с наименьшим количеством единиц является 100_{10} , и количество единиц в нём равно 3.

Ответ: 3.

СЛОЖЕНИЕ И УМНОЖЕНИЕ В РАЗЛИЧНЫХ СИСТЕМАХ СЧИСЛЕНИЯ

Обычно мы используем десятичную систему счисления и в ней производим 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

$$0 + 0 = 0 \quad 0 + 1 = 1$$

$$1 + 0 = 1 \quad 1 + 1 = 10$$

ТАБЛИЦА СЛОЖЕНИЯ

+	0	1
0	0	1
1	1	10

Правила умножения

$$0 \cdot 0 = 0 \quad 0 \cdot 1 = 0$$

$$1 \cdot 0 = 0 \quad 1 \cdot 1 = 1$$

ТАБЛИЦА УМНОЖЕНИЯ

×	0	1
0	0	0
1	0	1

В данных примерах действия выполняются в двоичной системе счисления:

$$\checkmark 1000_2 + 11_2 = 1011_2;$$

$$\checkmark 111_2 + 1_2 = 1000_2.$$

✓ Выполним сложение $10111011 + 1111$ в двоичной системе счисления.

Для этого необходимо вспомнить правила сложения в столбик в десятичной системе и таблицу сложения для двоичной системы счисления.

Начинаем решать справа налево: $1+1=10$, 0 пишем, 1 переносим в следующий разряд; $1+1+1=11$, 1 пишем, 1 переносим в следующий разряд. Продолжаем, пока не сложим все числа или все разряды.

$$\begin{array}{r} 111111 \\ 10111011 \\ + \quad 1111 \\ \hline 11001010 \end{array}$$

■ Трои́чная система счисления

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

Правила сложения

$$\begin{array}{ll} 0 + 0 = 0 & 1 + 2 = 10 \\ 0 + 1 = 1 & 2 + 0 = 2 \\ 0 + 2 = 2 & 2 + 1 = 10 \\ 1 + 0 = 1 & 2 + 2 = 11 \\ 1 + 1 = 2 & \end{array}$$

ТАБЛИЦА СЛОЖЕНИЯ

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

Правила умножения

$$\begin{array}{ll} 0 \cdot 0 = 0 & 1 \cdot 2 = 2 \\ 0 \cdot 1 = 0 & 1 \cdot 1 = 1 \\ 0 \cdot 2 = 0 & 2 \cdot 1 = 2 \\ 1 \cdot 0 = 0 & 2 \cdot 2 = 11 \\ 1 \cdot 1 = 1 & \end{array}$$

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 нет перехода через разряд.

✓ Выполним сложение в троичной системе: $10210112_3 + 1111_3$.

$$\begin{array}{r} 10210112 \\ 111 \\ \hline 10212000 \end{array}$$

■ Восьмеричная система счисления

Аналогично можно написать правила сложения и умножения для дру-

гих 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.

В данных примерах действия выполняются в восьмеричной системе счисления:

- ✓ $1000_8 + 11_8 = 1011_8$;
- ✓ $111_8 + 1_8 = 112_8$.

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



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.

Восьмеричная система обычно используется в областях, связанных с использованием цифровых устройств, применяется при выставлении прав доступа к файлам в работе с битовыми флагами и масками.



Выполним сложение в восьмеричной системе: $15603421_8 + 1442_8$.

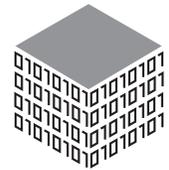
$$\begin{array}{r} 1 \\ 15603421 \\ + \quad 1442 \\ \hline 15605063 \end{array}$$

Примечание. В третьем столбце получилось значение 0, т. к. $4 + 4 = 8$, $8_{10} = 10_8$, поэтому записываем 0, а 1 переносим в следующий разряд.



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 ко-**

дированием. На компьютере можно закодировать следующую информацию:

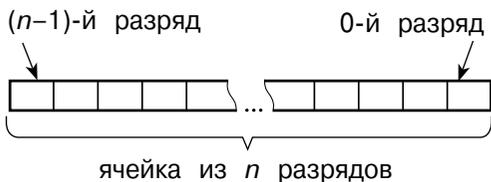
- ▲ числовую;
- ▲ текстовую;
- ▲ графическую;
- ▲ звуковую.

ПРЕДСТАВЛЕНИЕ ЧИСЛОВОЙ ИНФОРМАЦИИ

При переводе из k -ичной системы в десятичную мы нумеровали разряды справа налево. Любое число,

которое кодируется двоичным кодом, хранится в ячейке, состоящей из n разрядов.

В каждом разряде содержится 0 или 1.



На компьютере можно закодировать целые положительные и отрицательные вещественные числа с плавающей запятой. В компьютере используется двоичная система счисления, поэтому необходимо уметь переводить из десятичной системы счисления в двоичную. Рассмотрим порядок перевода целых положительных и отрицательных чисел. Для представления **целых чисел без знака** отводят 1, 2 байта информации, реже — 4 (при необходимости больше). При этом используются 8, 16 или 32 ячейки. Для подсчёта количества ячеек (N) используют формулу:

$$N = i \cdot 8,$$

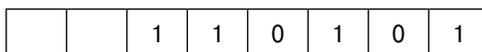
где i — количество байт информации.

✓ Как представляется число 53 в компьютере, если под него отводится 1 байт информации?

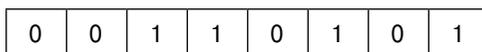
Переведём число 53 с помощью деления на 2 в двоичную систему:

$$53_{10} = 110101_2.$$

1 байт = 8 бит — 8 ячеек для хранения числа. Записываем число.



Остальные ячейки заполняем нулями. Получим **прямой код** числа.



Примечание. Если мы захотим под хранение числа 53 отвести 2 байта информации, нам понадобится 16 ячеек. В этом случае первые 10 ячеек, считая слева направо, будут заполнены нулями.

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

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

✓ Как представляется число -53 в компьютере, если под него отводится 1 байт информации?

Число $|-53|$ переведем с помощью деления на 2 в двоичную систему:

$$53_{10} = 110101_2.$$

1 байт = 8 бит — 8 ячеек для хранения числа. Записываем число.

		1	1	0	1	0	1
--	--	---	---	---	---	---	---

Остальные ячейки заполняем нулями.

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

Инвертируем числа (меняем 0 на 1 и 1 на 0).

1	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---

Добавляем к числу 1: $11001010 + 1 = 11001011$.

1	1	0	0	1	0	1	1
---	---	---	---	---	---	---	---

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 соответствует какой букве, используется специальная **кодовая таблица символов**.

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. Так, число 3 в двоичной системе имеет код 11, а изображаемый символ числа 3 кодируется как 110011.

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
54	110110	6
55	110111	7
56	111000	8
72	1001000	Н
73	1001001	И

Десятичный код	Двоичный код	Символ
74	1001010	J
75	1001011	K
76	1001100	L
77	1001101	M
78	1001110	N

Для русского языка используются кодировки CP-866, КОИ-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. К областям применения Unicode относятся компьютерные программы и мобильные приложения, информация в Интернете, записи в базах данных. Для кодирования одного символа информации используется 2 байта.

ПРЕДСТАВЛЕНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Примерно 80 % всей информации человек получает с помощью зрения. С появлением компьютера возникла необходимость обработки графической информации, включающей кодирование, просмотр и ре-

дактирование, хранение изображений и видео.

Пространство непрерывно, значит, в каждой его области содержится бесконечное количество точек, ко-

торое не может быть закодировано на компьютере, поскольку для этого потребуется невероятный объём памяти. Чтобы сделать процесс эффективным, используется дискретизация. **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;
- ▲ разрешение;
- ▲ глубина цвета.

■ Цветовая модель

Цветовой моделью называют математическую модель описания представления цветов в виде набора чисел (обычно из трёх или четырёх значений), называемых цветовыми компонентами или цветовыми координатами.

В компьютерной технике чаще всего используются следующие цветовые модели:

- ▲ **RGB** (Red, Green, Blue — красный, зелёный, синий) — применяется для вывода изображения на экраны мониторов и другие электронные устройства;
- ▲ **CMYK** (Cyan, 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) — используется в полиграфии для стандартной печати;

- ▲ **HSB** (Hue, Saturation, Brightness — тон, насыщенность, яркость) — является аналогом RGB, основана на её цветах, но отличается системой координат.

■ Разрешение

Разрешение — величина, определяющая количество точек на единицу площади. Чем большее количество точек приходится на единицу площади, тем более высокое разрешение получается. Высокое разрешение обеспечивает более точное представление оригинала.

■ Глубина цвета

Глубина цвета определяется количеством бит, которое используется для кодирования цвета одного пикселя. От глубины цвета зависит количество воспроизводимых цветов и размер файла. Единицей измерения глубины цвета является бит/пиксель.

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.

✓ Найдём количество цветов в палитре с глубиной цвета 16 бит.

Используем формулу $N = 2^i$ и вычислим значение $2^{16} = 65\,536$ цветов.

■ Информационный объём растрового графического изображения

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 $A \times B$ 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 бит 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 >	65 536	Подходит для цветовой модели RGB	39 Кбайт



Практические задания

39 Автоматическая фотокамера производит растровые изображения размером 600×400 пикселей. При этом объём файла с изображением не может превышать 90 Кбайт, упаковка данных не производится. Какое максимальное количество цветов можно использовать в палитре?

Решение:

Чтобы найти информационный объём растрового изображения, надо умножить количество пикселей в изображении на объём памяти V , необходимый для хранения цвета одного пикселя: $600 \cdot 400 \cdot V \leq 90 \cdot 2^{13}$ бит, откуда $V \leq 3,072$ бит = 3 бит. Значит, в изображении можно использовать не более $2^3 = 8$ цветов.

Ответ: 8.

40 Рассчитайте объём файла 100×200 пикселей, который содержит изображение в серых оттенках. Ответ запишите в килобайтах.

Решение:

Найдём количество пикселей в изображении: $100 \cdot 200 = 20\,000$ пикселей.

Зная, что глубина цвета для серых оттенков равна 8 бит/пиксель, вычислим: $20\,000 \cdot 8 \text{ бит/пиксель} = 160\,000 \text{ бит}$. Переведём в Кбайты, для чего сначала разделим полученное число бит на 8 для перевода в байты, затем на 1024, поскольку 1 Кбайт = 1024 байт. Получим:

$$160\,000 : 8 : 1024 = 19,5 \text{ Кбайт.}$$

Ответ: 19,5 Кбайт.

- 41** В результате преобразования растрового графического изображения количество цветов уменьшилось с 256 до 16. Как при этом изменится объём видеопамати, занимаемой изображением?

Решение:

Имеем: $N_1 = 256$ цветов, $N_2 = 16$ цветов. Используем формулы для определения количества цветов и расчёта информационного объёма:

$$V_1 = K \cdot i_1; N_1 = 2^{i_1}; V_2 = K \cdot i_2; N_2 = 2^{i_2}.$$

$$N_1 = 256 = 2^8; i_1 = 8 \text{ бит/пиксель.}$$

$$N_2 = 16 = 2^4; i_2 = 4 \text{ бит/пиксель.}$$

$$V_1 = K \cdot 8; V_2 = K \cdot 4. V_1 : V_2 = 8 : 4 = 2.$$

Ответ: объём графического изображения уменьшится в 2 раза.

- 42** Рисунок размером 128×256 пикселей занимает в памяти 24 Кбайта без учёта сжатия. Найдите максимально возможное количество цветов в палитре изображения.

Решение:

1) Найдём количество пикселей, используя для вычисления степени числа 2:

$$N = 128 \times 256 = 2^7 \cdot 2^8 = 2^{15}.$$

2) Объём файла в Кбайтах: $24 = 3 \cdot 2^3$.

3) Объём файла в битах: $3 \cdot 2^3 \cdot 2^{13} = 3 \cdot 2^{16}$ (т. к. 1 Кбайт = $2^{10} \cdot 2^3 = 2^{13}$ бит).

4) Глубина кодирования (количество битов, выделяемых на 1 пиксель): $3 \cdot 2^{16} : 2^{15} = 6$ бит/пиксель.

5) Максимальное возможное количество цветов: $2^6 = 64$.

Ответ: 64.

ПРЕДСТАВЛЕНИЕ ЗВУКОВОЙ ИНФОРМАЦИИ

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

Оцифровка звука — технология деления звука на небольшие временные шаги с последующей записью полученных значений в численном виде. Для кодирования звука необходимо выбрать следующие параметры:

- ▲ частоту звуковой дискретизации;
- ▲ глубину звука.

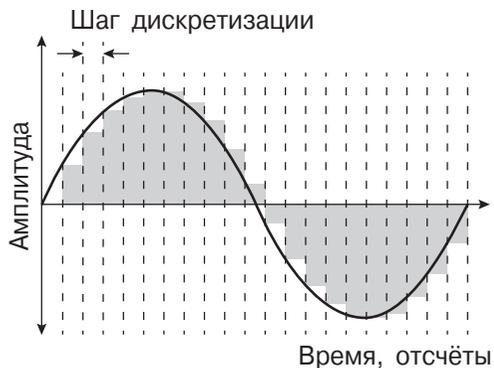
■ Частота звуковой дискретизации

Данная величина показывает количество измерений громкости звука за 1 с. Количество измерений может лежать в диапазоне от 8 до 48 кГц.

Герц (Гц или Hz) — единица измерения частоты периодических процессов (например, колебаний). 1 Гц

обозначает одно исполнение такого процесса за 1 с:

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

■ Глубина звука

Данная величина показывает количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука.

КАЧЕСТВО ОЦИФРОВАННОГО ЗВУКА

Радиотрансляция
Глубина кодирования: 8 бит. Частота дискретизации: до 8 кГц
Звучание CD-диска
Глубина кодирования: 16 бит. Частота дискретизации: до 48 кГц

Часто возникает необходимость оценить объём звукового файла, например для записи на диск.

Формула расчёта объёма звукового файла:

$$V = f \cdot t \cdot l \cdot n,$$

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).



Практические задания

- 43** Определите длительность звучания стереоаудиофайла, занимающего 1,3 Мбайт памяти при глубине звука 16 бит и частоте 48 кГц.

Решение:

$$V = 600 \text{ Кбайт} = 600 \cdot 1024 \cdot 8 \text{ бит} = 4\,915\,200.$$

$$f = 48 \text{ кГц} = 48\,000 \text{ Гц.}$$

$$l = 16 \text{ бит.}$$

Поскольку в условии говорится о стереофайле, значит, $n = 2$.

Подставим имеющиеся значения в формулу $V = f \cdot t \cdot l \cdot n$.

$$\text{Получим: } 4\,915\,200 = 48\,000 \cdot t \cdot 16 \cdot 2.$$

$$4\,915\,200 = 1\,536\,000 \cdot t.$$

$$t = 4\,915\,200 : 1\,536\,000 = 3,2 \text{ с.}$$

Ответ: 3,2 с.

- 44** Производилась двухканальная (стерео) звукозапись с частотой дискретизации 64 кГц и 24-битным разрешением. В результате был получен файл размером 120 Мбайт, сжатие данных не производилось. Определите приблизительно, сколько вре-

мени (в минутах) производилась запись. В ответе укажите ближайшее ко времени записи целое число, кратное 5.

Решение:

1) Переведём данные в единицы измерения, необходимые для применения формулы расчёта объёма звукового файла:

$$f = 64 \text{ кГц} = 64 \cdot 1000 = 64\,000 \text{ Гц} = 64 \cdot 10^3 \text{ Гц};$$

$$V = 120 \text{ Мбайт} = 120 \cdot 10^6 \text{ байт} = 120 \cdot 8 \cdot 10^6 \text{ бит.}$$

2) Используя формулу $V = f \cdot t \cdot l \cdot n$, найдём t : $t = \frac{V}{f \cdot l \cdot n}$, где

$l = 24$ бит, $n = 2$ (поскольку в условии говорится о стереофайле).

$$t = \frac{120 \cdot 8 \cdot 10^6}{64 \cdot 10^3 \cdot 24 \cdot 2} = 312,5 \text{ с.}$$

3) Переведём секунды в минуты: $312,5 : 60 = 5 \text{ мин } 12,5 \text{ с} \approx 5 \text{ мин.}$

Ответ: 5.

45 Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город Санкт-Петербург по каналу связи за 15 с. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 2 раза выше и частотой дискретизации в 2,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Москву. Пропускная способность канала связи с Москвой в 2 раза выше, чем канала связи с Санкт-Петербургом. Сколько секунд длилась передача файла в Москву? В ответе запишите только целое число.

Решение:

Объём файла прямо пропорционален разрешению файла и его частоте дискретизации, следовательно, объём файла во втором случае в $2 : 2,5 = 0,8$ раза больше. Длительность передачи обратно пропорциональна пропускной способности канала связи, откуда получаем длительность передачи файла во второй раз: $15 : 2 \cdot 0,8 = 6$.

Ответ: 6.



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).

Истина	Ложь
0	1
1	0



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. Логические выражения могут быть простыми и сложными.

Простое логическое выражение состоит из одного высказывания

и не содержит логических операций. В простом логическом выражении возможен только один результат — либо «истина», либо «ложь».

Сложное логическое выражение содержит высказывания, объединённые логическими операциями. По аналогии с понятием функции в алгебре сложное логическое выражение содержит аргументы, которыми являются высказывания. Истинность сложного высказывания можно определить, зная значение входящих в него высказываний.

ЛОГИЧЕСКИЕ ВЫСКАЗЫВАНИЯ. ИСТИННОСТЬ ВЫСКАЗЫВАНИЯ

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

верждение называется **логическим высказыванием**.

✓ «Шесть минус три равно три», «Париж — столица Франции» — **истинные** высказывания, «пять больше семи», «Луна — это звезда» — **ложные**.

Не всякое предложение является логическим высказыванием, поэтому не всегда есть смысл говорить о его ложности или истинности.

✓ Высказывание «Саша — молодёц» не является логическим высказыванием, потому что оценка «молодёц» субъективна: для кого-то Саша молодёц, а для кого-то — нет.

Различают простые (элементарные) высказывания, обозначаемые латинскими буквами (A , B , C , $D...$), и 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.

Обозначение результатов логических утверждений представлено в таблице.

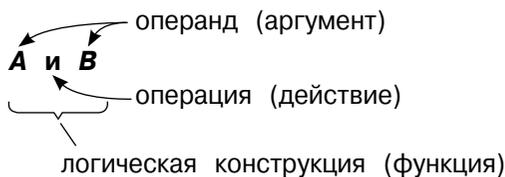
1	Да (истинно)	True	Истина
0	Нет (ложно)	False	Ложь

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

✓ Пусть высказывание A — «Саша любит конфеты», высказывание B — «Саша любит шоколад». Если говорится об одном и том же человеке, то оба эти высказывания можно объединить в одно: «Саша любит конфеты и шоколад» — и записать как A и B . Если оба высказывания истинны, то истинно и составное высказывание. Если же какое-нибудь из них ложно, то ложно и всё высказывание A и B .

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

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.

Операции различаются по количеству операндов, над которыми производится действие.

▲ **Унарная (одноместная) операция** — операция, которая применяется к одному операнду.

✓ Изменение знака числа. К A применим операцию изменения знака, получим $(-A)$.

▲ **Бинарная (двуместная) операция** — 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** — 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 вида $a ? b : c$, используемой в языках программирования.

ЛОГИЧЕСКИЕ ОПЕРАЦИИ

В устной речи мы сталкиваемся со сложными высказываниями, которые объединяют несколько логических выражений. Чтобы понять, что автор имел в виду, можно построить логическую конструкцию и упростить её.

✓ Фраза «Если часы неправильно показывают время, то можно не вовремя прийти на занятия» не меняет своей истинности, если мы заменим её на такую: «Если часы правильно показывают время, то можно вовремя прийти на занятия».

Чтобы научиться упрощать сложные фразы, рассмотрим основные логические операции и их таблицы истинности.



Таблицы истинности — таблицы с результатами каждой логической операции.

Логическое отрицание

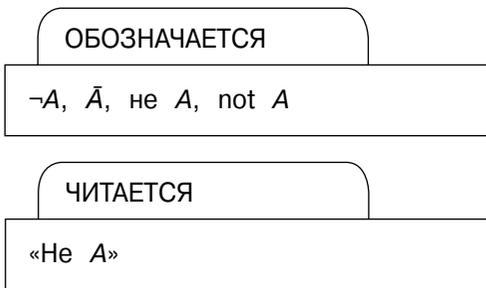
Логическое отрицание, или **инверсия** (лат. *inversion* — «переворачивание»), — унарная операция, в результате которой из данного высказывания (например, A) получается новое высказывание (не A), которое называется отрицанием исходного высказывания.



Схема операции логического отрицания (инвертор)

Порядок работы схемы: на вход подаётся значение A , после прохождения прямоугольника с операцией \neg на выходе получается значение $\neg A$.
 ✓ «Катя получила двойку», после применения отрицания получим: «Катя не получила двойку».

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.



- ✓ 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

Логическое умножение, или **конъюнкция** (лат. *conjunctio* — «соединение»), — бинарная операция,

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

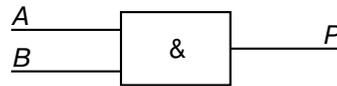
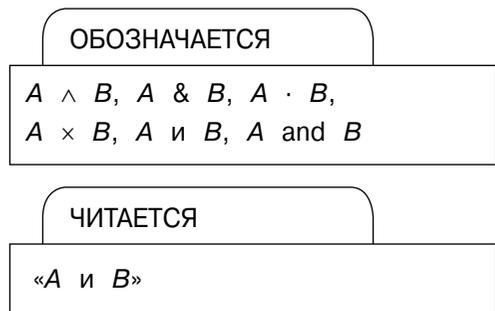


Схема операции конъюнкции (конъюнктор)

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

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



- ✓ «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



Суть конъюнкции легко запомнить с помощью следующего выражения: «Конъюнкция (высказывание $A \& B$) истинна только тогда, когда оба высказывания A, B истинны».

■ Логическое сложение

Логическое сложение, или **дизъюнкция** (лат. *disjunctio* — «разделение»), — 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

$A \vee B, A + B, A | B, A$ или B, A or B

ЧИТАЕТСЯ

« A или B »

✓ «Число x делится на 3 или на 5».

! Это высказывание будет истинным, если выполняются оба условия или хотя бы одно из условий.

ТАБЛИЦА ИСТИННОСТИ ОПЕРАЦИИ ДИЗЪЮНКЦИИ

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

■ Дизъюнкция строго разделительная

Дизъюнкция строго разделительная, т. е. **исключающее ИЛИ** (сложение по модулю 2), — бинарная операция, соединяющая два высказывания с помощью связки «или», употреблённой в исключаящем смысле.

ОБОЗНАЧАЕТСЯ

$A \oplus B, A \vee\vee B, A \underline{\vee} B, A$ or B

ЧИТАЕТСЯ

«Либо A , либо B »

✓ «Этот треугольник тупоугольный или остроугольный».

! Высказывание истинно, если выполняется одно из условий.



Суть операции «исключающее ИЛИ» легко запомнить с помощью следующего выражения: «Исключающее ИЛИ (высказывание $A \oplus B$) истинно только тогда, когда высказывания A и B имеют различные значения».

ТАБЛИЦА ИСТИННОСТИ ОПЕРАЦИИ
«ИСКЛЮЧАЮЩЕЕ ИЛИ»

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

■ Логическое следование

Логическое следование, или **импликация** (лат. *implicatio* — «тесно связываю»), — логическая операция, соединяющая два высказывания с помощью связки «если... то...».

ОБОЗНАЧАЕТСЯ

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

ЧИТАЕТСЯ

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

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

✓ «Если четырёхугольник — квадрат, то в него можно вписать окружность».

! Эта 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.

ТАБЛИЦА ИСТИННОСТИ
ОПЕРАЦИИ ИМПЛИКАЦИИ

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



Для операции импликации справедливо утверждение: «Результат операции ложен только тогда, когда предпосылка есть истина, а следствие — ложь».

■ Эквивалентность

Эквивалентность, или **двойная импликация** (равнозначность) (лат. *aequalis* — «равный», *valentis* — «имеющий силу»), — бинарная операция, позволяющая из двух высказываний A и B получить новое высказывание $A = B$.

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 »

Эквивалентность — это 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 ЭКВИВАЛЕНТНОСТИ

A	B	A ↔ B
0	0	1
1	1	1
1	0	0
0	1	0



Операция эквивалентности противоположна исключающему ИЛИ и имеет результат «истина» тогда и только тогда, когда значения переменных совпадают.

Зная значения простых высказываний, можно на основании таблиц истинности определить значения сложных высказываний. При этом важно знать, что для представления любой функции алгебры логики достаточно трёх операций: конъюнкции, дизъюнкции и отрицания.



Обозначения для логических связей (операций):

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

При решении математических выражений, состоящих из нескольких действий, в первую очередь необходимо задуматься о порядке дей-

ствий. Например, $2 + 2 \cdot 2$ не то же самое, что $(2 + 2) \cdot 2$. При работе с логическими конструкциями также важно сначала понять, какое

действие следует выполнить первым. Для этого существуют определённые правила.

Операции одинакового приоритета выполняются слева направо.

✓ Расставим порядок действий в следующих логических высказываниях.

$$\begin{array}{ccccccc} & 3 & 4 & 2 & 5 & 1 & \\ A \vee B & \rightarrow & C \& D & \leftrightarrow & \neg A & \\ & 4 & 2 & 3 & 5 & 1 & \\ A \vee (B \rightarrow C) & \& D & \leftrightarrow & \neg A & & \end{array}$$

Правила приоритетов для логических связей

1) Выражение в скобках

2) Логическое НЕ

3) Логическое И

4) Логическое ИЛИ

5) Импликация

6) Эквивалентность



Практические задания

46 Напишите наименьшее число x , для которого истинно высказывание:

$$(x > 25) \text{ И НЕ } (x \text{ чётное}).$$

Решение:

В данном выражении содержится две логические операции: НЕ и И. По приоритету выполним первую логическую операцию НЕ:

$$\text{НЕ } (x \text{ чётное}) = x \text{ нечётное.}$$

После выполнения первой операции имеем:

$$(x > 25) \text{ И } (x \text{ нечётное}).$$

Выражение является истинным, когда обе его части истинны одновременно:

$$(x > 25) \text{ И } (x \text{ нечётное}) = \text{ИСТИНА.}$$

истина истина

Наименьшим числом, для которого истинны оба полученных утверждения, является число 27.

Ответ: 27.

47 Напишите наименьшее число x , для которого ложно высказывание:

$$(x \leq 48) \text{ ИЛИ НЕ } (x \text{ нечётное}).$$

Решение:

В данном выражении имеется две логические операции: НЕ и ИЛИ. По приоритету выполним первую логическую операцию НЕ:

НЕ (x нечётное) = x чётное.

После выполнения первой операции получим:

$(x \leq 48)$ ИЛИ (x чётное).

Это выражение будет являться ложным, когда обе части его ложны одновременно:

$(x \leq 48)$ ИЛИ (x нечётное) = ЛОЖЬ.

ложь

ложь

Наименьшим числом, для которого истинны оба полученных утверждения, является число 50.

Ответ: 50.

48 Напишите наибольшее целое число x , для которого истинно высказывание:

НЕ ($x \leq 8$) И НЕ ($x \geq 15$) И (x чётное).

Решение:

Используя приоритет логической связки НЕ, перепишем выражение в виде:

$(x > 8)$ И ($x < 15$) И (x чётное).

Выражение будет являться истинным, когда все входящие в него высказывания истинны:

$(x > 8)$ И ($x < 15$) И (x чётное).

истина

истина

истина

Для первых двух высказываний ($x > 8$) и ($x < 15$) получаем числа 9, 10, 11, 12, 13, 14. Из них 10, 12, 14 являются чётными, значит, наибольшее чётное число, для которого всё высказывание будет истинным, — 14.

Ответ: 14.

49 Для какого из приведённых имён истинно высказывание:

НЕ (первая буква гласная) И (количество букв > 5)?

- 1) Мария
- 2) Екатерина
- 3) Дарья
- 4) Пелагея

В ответе запишите номер правильного варианта.

Решение:

Используя приоритет логической связки НЕ, перепишем выражение в виде:

(первая буква согласная) И (количество букв > 5).

Выражение будет истинным, когда оба высказывания истинны.

Первая буква согласная у имён Дарья (3) и Пелагея (4).

Количество букв больше 5 в этих именах только у варианта под номером 4 — Пелагея.

Ответ: 4.

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
Закон двойного отрицания: $\neg(\neg A) = A$	При двойном отрицании исходного суждения в итоге получится оно же

Название закона, формула	Содержание
Свойства констант: $A \vee 0 = A$ $A \& 0 = 0$ $A \vee 1 = 1$ $A \& 1 = A$	Отрицание лжи есть истина. Отрицание истины есть ложь
Закон идемпотентности: $A \& A = A$ $A \vee A = A$	Закон, дающий возможность исключить повторяющиеся суждения
Закон коммутативности: $A \& B = B \& A$ $A \vee B = B \vee A$	Перемена мест высказываний не влияет на результат рассуждения
Закон ассоциативности: $A \vee (B \vee C) = (A \vee B) \vee C$ $A \& (B \& C) = (A \& B) \& C$	Логическое сложение и умножение ассоциативно, т. е. при наличии в выражении лишь конъюнкции или лишь дизъюнкции можно опускать скобки
Законы дистрибутивности: $A \vee (B \& C) = (A \vee B) \& (A \vee C)$ $A \& (B \vee C) = (A \& B) \vee (A \& C)$	Правила раскрытия скобок при конъюнкции и дизъюнкции
Закон поглощения: для конъюнкции: $A \vee (A \& B) = A$; для дизъюнкции: $A \& (A \vee B) = A$	По предыдущему закону можно раскрыть скобки и убрать лишнее A
Законы де Моргана: $\neg(A \& B) = \neg A \vee \neg B$ $\neg(A \vee B) = \neg A \& \neg B$	Отрицание суждения « A и B » эквивалентно суждению «не A или не B ». Неверно, что A и B , если и только если неверно A и неверно B

При вычислении логического выражения по таблице истинности следует определить, сколько строк будет в таблице. Для этого необходимо посчитать количество разных аргументов и возвести в полученную степень число 2, т. к. имеется

два значения — 0 и 1. Например, для двух аргументов A и B получится 4 строки (2^2), а для функции с тремя аргументами x , y , z в таблице будет отведено 8 строк (2^3), для пяти аргументов понадобится 32 строки.



Практические задания

50 Составьте таблицу истинности для следующей функции:

$$F = (A \vee B) \& (\neg A \vee \neg B).$$

Решение:

- 1) Определим количество строк в таблице: $2^2 = 4$, поскольку имеется два разных аргумента A и B и у каждого по два значения — 0 и 1.
- 2) Количество столбцов: 5 действий и 2 аргумента — 7 столбцов.
- 3) Расставим порядок выполнения действий.

$$\begin{array}{cccccc} 1 & 5 & 2 & 4 & 3 & \\ (A \vee B) \& (\neg A \vee \neg B) \end{array}$$

- 4) Составим таблицу.

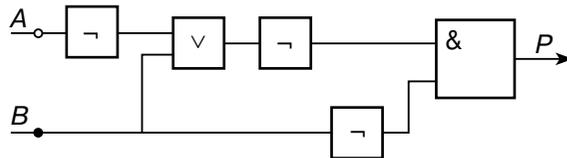
A	B	$A \vee B$	$\neg A$	$\neg B$	$\neg A \vee \neg B$	$(A \vee B) \& (\neg A \vee \neg B)$
0	0	0	1	1	1	0
0	1	1	1	0	1	1
1	0	1	0	1	1	1
1	1	1	0	0	0	0

51 Расставьте порядок действий и составьте схему для следующей функции:

$$\neg B \& \neg(\neg A \vee B).$$

Решение:

$$\begin{array}{cccccc} 3 & 5 & 4 & 1 & 2 & \\ \neg B \& \neg(\neg A \vee B) \end{array}$$



52 Логическая функция F задаётся выражением

$$\neg y \vee x \vee (\neg z \wedge w).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых

функция F ложна. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z, w .

?	?	?	?	F
0	0	0	1	0
0	0	1	1	0
1	0	1	1	0

В ответе напишите буквы x, y, z, w в том порядке, в котором идут соответствующие им столбцы.

Решение:

- 1) Для того чтобы функция была равна 0, необходимо, чтобы $\neg u, x$ и скобка были равны 0 одновременно.
- 2) Во втором столбце все значения равны 0, значит, в нём содержится переменная x ; в четвёртом столбце — переменная y , т. к. при нахождении отрицания y все значения обращаются в 0.
- 3) Рассмотрим скобку $(\neg z \wedge w)$. Она должна быть ложна, а логическое И принимает значение 0 тогда, когда хотя бы одно значение из двух равно 0. Это возможно, когда y столбца 3 берётся отрицание. Значит, третий столбец — это z .
- 4) Делаем вывод, что переменная w занимает первый столбец, а z , соответственно, третий.

Ответ: $wxzy$.

53 Сколько различных решений имеет уравнение

$$J \wedge \neg K \wedge L \wedge \neg M \wedge (N \vee \neg N) = 0,$$

где J, K, L, M, N — логические переменные?

В ответе не нужно перечислять все различные наборы значений J, K, L, M, N , при которых выполнено данное равенство. В качестве ответа нужно указать количество таких наборов.

Решение:

Выражение $(N \vee \neg N)$ истинно при любом N , поэтому имеем:

$$J \wedge \neg K \wedge L \wedge \neg M = 0.$$

Применим отрицание к обеим частям логического уравнения и используем закон де Моргана: $\neg(A \wedge B) = \neg A \vee \neg B$.

Получим: $\neg J \vee K \vee \neg L \vee M = 1$.

Логическая сумма равна 1, если хотя бы одно из составляющих её высказываний равно 1. Поэтому полученному уравнению удовлетворяют любые комбинации логических переменных, кроме случая, когда все входящие в уравнение величины равны 0. Каждая из четырёх переменных может быть равна либо 1, либо 0, поэтому всевозможных комбинаций $2 \cdot 2 \cdot 2 \cdot 2 = 16$. Следовательно, уравнение имеет $16 - 1 = 15$ решений.

Осталось заметить, что найденные 15 решений соответствуют любому из двух возможных значений логической переменной N , поэтому исходное уравнение имеет 30 решений.

Ответ: 30.

54 Логическая функция F задаётся выражением

$$(x \wedge \neg y) \vee (y \equiv z) \vee \neg w.$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F ложна. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных w , x , y , z . Все строки в представленном фрагменте разные.

П. 1	П. 2	П. 3	П. 4
	0		
1	0		0
1		0	0

В ответе напишите буквы w , x , y , z в том порядке, в котором идут соответствующие им столбцы (без разделителей). (П. 1 — переменная 1.)

Решение:

1) Функция будет равна 0 при условии, что обе скобки $(x \wedge \neg y)$, $(y \equiv z)$ и $\neg w$ ложны одновременно. Следовательно, значение w должно быть истинно, т. е. равно 1. Значит, переменной w соответствует первый столбец.

2) Первая скобка $(x \wedge \neg y)$ принимает значение 0 во всех случаях, кроме $x = 1, y = 0$; вторая скобка $(y \equiv z)$ принимает значение 0 в случае, когда одна из переменных — 0, а вторая — 1, y не равно z , поэтому переменной x соответствует столбец 4.

3) Так как y и z не равны друг другу, получаем следующую таблицу.

П. 1	П. 2	П. 3	П. 4
1	0	1	
1	0	1	0
1	1	0	0

Поскольку в таблице приведены фрагменты ложной функции, то значение x в первой строке равно 1, следовательно, из первой скобки можно сделать вывод, что переменная y соответствует второму столбцу, т. к. она не может быть равна 0.

4) Делаем общий вывод: переменная w занимает первый столбец, z — второй, y — третий, x — четвёртый.

Ответ: $wzyx$.

55 На числовой прямой даны два отрезка: $P = [150; 195]$ и $Q = [170; 205]$. Укажите наименьшую возможную длину такого отрезка A , что формула

$$(x \in P) \rightarrow (((x \in Q) \wedge \neg(x \in A)) \rightarrow \neg(x \in P))$$

истинна при любом значении переменной x , т. е. принимает значение 1 при любом значении переменной x .

Решение:

Раскроем первую и вторую импликации, получим: $\neg(x \in P) \vee (\neg((x \in Q) \wedge \neg(x \in A)) \vee \neg(x \in P))$.

Применяем законы де Моргана, получим: $\neg(x \in P) \vee \neg(x \in Q) \vee (x \in A)$. Выражения $\neg(x \in P)$ и $\neg(x \in Q)$ принимают значение 0 тогда, когда x принадлежит обоим отрезкам. Поэтому значения переменной такие, что $170 \leq x \leq 195$. Таким образом, получим отрезок A $[170; 195]$, длина которого равна 25.

Ответ: 25.

56 Элементами множества A являются натуральные числа. Известно, что выражение

$$\begin{aligned} & ((x \in \{3, 5, 7, 9, 11, 13\}) \rightarrow \\ & \rightarrow \neg(x \in \{5, 8, 11, 14\})) \vee (x \in A) \end{aligned}$$

истинно, т. е. принимает значение 1 при любом значении переменной x .

Определите наименьшее возможное значение суммы элементов множества A .

Решение:

Избавимся от импликации:

$$\neg(x \in \{3, 5, 7, 9, 11, 13\}) \vee \neg(x \in \{5, 8, 11, 14\}) \vee (x \in A).$$

По закону де Моргана:

$$\neg(x \in \{3, 5, 7, 9, 11, 13\}) \wedge (x \in \{5, 8, 11, 14\}) \vee (x \in A);$$

$$\neg(x \in \{5, 11\}) \vee (x \in A).$$

Данное выражение будет истинным при $A = \{5, 11\}$.

Отсюда сумма элементов равна 16.

Ответ: 16.

57 Обозначим через m & n поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, например, 14_{10} & $5_{10} = 1110_2$ & $0101_2 = 100_2 = 4$. Для какого наименьшего неотрицательного целого числа A формула

$$x \text{ \& } 51 \neq 0 \rightarrow (x \text{ \& } A = 0 \rightarrow x \text{ \& } 25 \neq 0)$$

тождественно истинна, т. е. принимает значение 1 при любом неотрицательном целом значении переменной x ?

Решение:

Формула состоит из импликации двух выражений. Импликация истинна во всех случаях, кроме случая $1 \rightarrow 0$. Раскроем импликацию по формуле, получим: $x \& 51 = 0 \vee (x \& A \neq 0 \vee x \& 25 \neq 0) = (x \& 51 = 0 \vee x \& 25 \neq 0) \vee x \& A \neq 0$.

Переведём числа 51 и 25 в двоичную систему.

$$51 = 110011_2$$

$$25 = 11001_2$$

Рассмотрим поразрядную конъюнкцию выражения $x \& 51 = 0$.

1	1	0	0	1	1	51
0	0	0/1	0/1	0	0	x

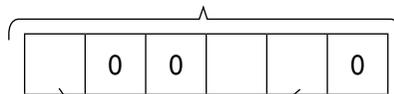
Рассмотрим поразрядную конъюнкцию выражения $x \& 25 = 0$.

0	1	1	0	0	1	25
0/1	0	0	0/1	0/1	0	x

Нельзя, чтобы $(x \& 51 = 0 \vee x \& 25 \neq 0)$ и $x \& A \neq 0$ были одновременно ложными.

Эти разряды одновременно не равны нулю

x (в двоичной системе)



Нельзя, чтобы выражение $x \& A = 0$ было истинно.

Получается, что $A = 100010_2$. Это число наименьшее из возможных, при котором мы получим, что вся формула будет не ложна.

$$A = 100010_2 = 34_{10}$$

Ответ: 34.



Данную задачу можно решить несколькими способами. На с. 253 представлено решение на языке программирования Python.

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 $\exists!$.

Предикат (лат. *praedicatum* — «заявленное, упомянутое, сказанное») — утверждение, содержащее переменные и принимающее значение 1 или 0 (истинно или ложно) в зависимости от значений переменных.

✓ Утверждение « x делится на 9 на множестве целых положительных чисел» при $x = 9, 18, 27$ истинно, при $x = 8, 16, 33$ ложно.

Множество, на котором предикат принимает только истинные значения, называется **множеством истинности предиката** I_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...).

Предикатом в программировании является функция, которая принимает один аргумент и более и возвращает значения булева типа (от англ. *Boolean*, или *logical data type*).

Невыполнимый предикат — предикат, принимающий значение «ложь» при всех допустимых значениях переменной. Так, выражение $x^2 + 3 = 0$ при всех действительных значениях x не принимает значения «истина», иначе говоря, уравнение не имеет действительных корней.

Предикат A является **следствием** предиката B , если для любых значений переменных, при которых B — истина, A — тоже истина. Например, если число a делится на 9 (B), то число a делится также на 3 (A). Обратное утверждение не всегда будет являться истинным.

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 также будет являться 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
 ▲ **Квантор всеобщности.**

Обозначение: \forall .

Читается: «для любого...», «для каждого...», «для всех...» или «каждый...», «любой...», «все...».

▲ Квантор существования.

Обозначение: \exists .

Читается: «существует...» или «най-
дётся...».

▲ Квантор существования и единственности.

Обозначение: $\exists!$.

Выражение «существует точно одно
такое x , что...».

В математической логике приписыва-
ние 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 делит-
ся на 7;
- 3) все натуральные числа делятся
на 7.

Квантор будет иметь вид:

$$(\forall x \in N) P(x).$$



Практические задания

- 58** На числовой прямой даны два отрезка: $D = [155; 177]$ и $B = [111; 130]$. Укажите наименьшую возможную длину такого отрезка A , что формула

$$(x \in D) \Rightarrow ((\neg(x \in B) \wedge \neg(x \in A)) \Rightarrow \neg(x \in D))$$

тождественно истинна, т. е. принимает значение 1 при любом значении переменной x .

Решение:

- 1) Для удобства произведём замену:

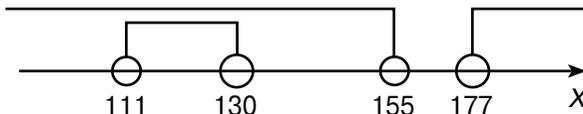
$$(x \in D) = D, (x \in B) = B, (x \in A) = A.$$

Следовательно, новая формула будет выглядеть так: $D \Rightarrow ((\neg B \wedge \neg A) \Rightarrow \neg D)$.

- 2) Произведём упрощение формулы, воспользовавшись определением знака следования:

$$D \Rightarrow ((\neg B \wedge \neg A) \Rightarrow \neg D) = \neg D \vee ((\neg B \wedge \neg A) \Rightarrow \neg D) = \neg D \vee B \vee A \vee \neg D = \neg D \vee B \vee A.$$

- 3) $\neg D \vee B \vee A = 1$.



- 4) Из графика и выражения $\neg D \vee B \vee A$ становится понятно, что A должен перекрывать пустой промежуток, где нет «крыш»; поскольку нужно найти минимальную длину для A , присвоим ей промежуток $[155; 177]$.
- 5) Длина $A = 177 - 155 = 22$.

Ответ: 22.

- 59** Для какого наибольшего целого числа A формула
 $((x \leq 9) \rightarrow (x \cdot x \leq A)) \wedge ((y \times y \leq A) \rightarrow (y \leq 9))$
 тождественно истинна, т. е. принимает значение 1 при любых целых неотрицательных x и y ?

Решение:

Для упрощения данной схемы необходимо:

- раскрыть импликацию по правилу $A \rightarrow B = \neg A \vee B$;
- заменить логическую сумму совокупностью, а логическое произведение — системой соотношений;
- определить значения параметра A , при котором следующая система совокупностей будет иметь решения для любых целых неотрицательных чисел:

$$\begin{cases} x > 9, \\ x^2 \leq A, \\ y^2 > A, \\ y \leq 9. \end{cases}$$

Заметим, что переменные не связаны между собой уравнением или неравенством, поэтому необходимо и достаточно, чтобы решениями первой совокупности были все неотрицательные x , а решениями второй совокупности — все неотрицательные y .

Решениями неравенства $x > 9$ являются числа 10, 11, 12... . Чтобы совокупность выполнялась для всех целых неотрицательных чисел, числа 0, 1, 2... 9 должны быть решениями неравенства $x^2 \leq A$. Значит, $A \geq 81$. Аналогично решениями неравенства $y \leq 9$ являются числа 0, 1, 2... 9. Следовательно, числа 10, 11, 12... должны быть решениями неравенства $y^2 > 10$. Поэтому $A < 100$. Получим: $A[81; 100)$. Искомое наибольшее целое значение параметра равно 99.

Ответ: 99.

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

Частичная интерпретация (операционная система)

>>>

>>>

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

▲ **Уровень 0.** Цифровой логический уровень — аппаратное обеспечение машины, включающее ряд логических элементов — вентиляей.

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.

▲ **Уровень 1.** 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.

▲ **Уровень 2.** Уровень архитектуры набора 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).

▲ **Уровень 3.** Уровень операционной системы. Это гибридный уровень: одна часть команд интерпретируется операционной системой, другая — микропрограммой.

▲ **Уровень 4.** Уровень ассемблера. Представляет собой символическую форму одного из языков более низкого уровня.

▲ **Уровень 5.** 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.

ОРГАНИЗАЦИЯ РАБОТЫ КОМПЬЮТЕРОВ

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

**Компьютер =
= аппаратная часть +
+ программное обеспечение.**

Аппаратное обеспечение 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* — «изде-

лие»), — набор устройств, из которых он состоит.

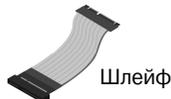
Основные аппаратные компоненты компьютера

Процессор

Память компьютера

Устройства ввода-вывода

Дополнительные компоненты, связывающие основные между собой



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.

Процессоры можно разделить на два типа: **мобильные**, применяемые в современных телефонах, планшетах, и **десктопные**, используемые 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.

В современных компьютерах используются микропроцессоры нового поколения, которые во много раз меньше устройств предыдущего поколения.

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.

Внутри процессора имеются специальные ячейки (регистры) для оперативного хранения обрабатываемых данных и некоторой служебной информации. Наиболее важными частями процессора являются устройство управления и арифметико-логическое устройство.

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.

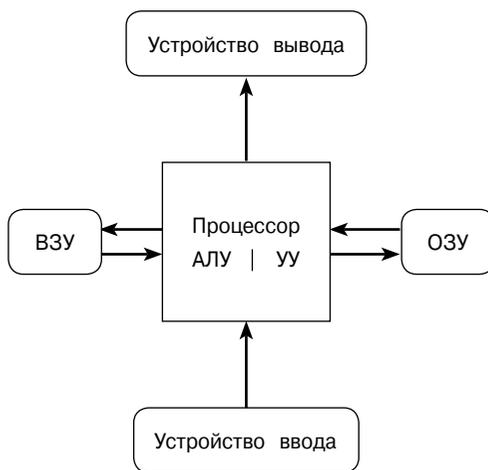


Схема Дж. фон Неймана. Стрелки отражают движение информации

■ Память компьютера



Оперативная память (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 память (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, Basic Input/Output System), он получает управ-

ление при выключении и сбросе (reset) системной платы.

Внешняя память (ВЗУ — внешнее запоминающее устройство) компьютера состоит из накопителей на жёстких магнитных дисках (HDD), дисководов чтения и записи оптических дисков (CD и DVD), а также карты памяти и флеш-накопителей.

Кэш-память — сверхбыстрая память, используемая процессором для временного хранения наиболее часто используемых данных.

■ Системная шина, контроллеры и другие устройства

Системная шина передаёт данные между функциональными блоками компьютера.

Контроллеры обеспечивают обмен информацией с внешними устройствами.

В состав материнской платы входит также **чипсет** (набор микросхем, обеспечивающих согласованную работу всех аппаратных средств компьютера) и ряд других подсистем, которые обеспечивают удобство и функциональность конкретной материнской платы (электропитания, мониторинга физических и электрических параметров).

Видеокарта (видеоадаптер, графическая карта) — устройство, преобразующее изображение, находящееся

в памяти компьютера, в видеосигнал для монитора.

Блок питания — устройство, обеспечивающее 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** (через такие порты информация передаётся по одному биту) и **параллельными** (несколько битов информации передаётся одновременно).

Устройства ввода информации

Клавиатура

Манипуляторы (мышь, трекбол)

Графический планшет

Сканер

Звуковая карта, микрофон

Игровые манипуляторы

Устройства вывода информации

Монитор

Принтер

Акустические колонки и наушники

Модем (устройство ввода/вывода)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Программное обеспечение (ПО), или *software* (англ. *soft* — «мягкий», *ware* — «изделие»), — набор программ, управляющих действиями компьютера при его использовании для решения задач. В современном русском языке активно применяется также термин «софт». ПО управляет аппаратной частью ПК. Удобство и универсальность ПО заключается

в его способности модифицироваться. Программы, способные запоминать информацию, делают вычислительные машины гибкими и легко адаптируемыми к различным условиям.

Рассмотрим подробнее основные элементы структуры программного обеспечения.



■ Системное ПО

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

Системное ПО можно разделить на базовое и сервисное. Базовое программное обеспечение, как правило, поставляется вместе с компьютером, а сервисное можно приобрести дополнительно.

Базовое ПО

Минимальный набор программных средств, обеспечивающих работу компьютера. В него входят операционная система, операционные оболочки (обычно текстовые и графические), сетевая операционная система.

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

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.

▲ **Драйверы** — компьютерные программы, с помощью которых другие программы получают доступ к аппаратному обеспечению устройства.
✓ Драйверы для звуковой карты, принтеров, сканеров.

▲ **Программы диагностики** работоспособности компьютера.
✓ Программы для тестирования оперативной памяти, жёстких дисков, процессора, определения типа ядра и объёма кэш-памяти.

▲ **Антивирусные программы** — программы, выполняющие задачи обнаружения компьютерных вирусов, а также нежелательных вредоносных программ, восстановления заражённых (модифицированных) такими программами файлов и предотвра-

щения заражения файлов или ОС вредоносным кодом.

✓ Антивирус Касперского, Dr. Web, ESET NOD32, Panda Security, Avast.

▲ **Программы обслуживания дисков** — программы, способные тестировать производительность, проверять кластеры, объединять разделы или восстанавливать винчестер.
✓ Проверка диска (Scan Disk), Advanced SystemCare, CCleaner и др.

▲ **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.
✓ 7-Zip, WinRAR, WinZip и др.

▲ **Программы обслуживания сети** — коммуникационные программы, организующие обмен информацией между компьютерами.
✓ Zabbix, NetTraffic, Wireshark.

■ Прикладное ПО

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

Приложения общего назначения

К данному типу относится комплект офисных приложений MS Office или LibreOffice (для macOS). Эти ком-

плекты предназначены для работы с текстом (текстовые редакторы), таблицами (электронные таблицы), пакеты для создания презентаций, СУБД (базы данных).

Приложения специального назначения

В число таких приложений входят браузеры — средства просмотра веб-страниц, графические редакторы, бухгалтерские системы типа 1С, веб-редакторы.

■ Инструментальное ПО

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 к аппаратным ресурсам и управления ими.

В более общем виде состав модулей ОС можно разделить на две части:

▲ **ядро** — модули, выполняющие основные функции ОС;

▲ **вспомогательные модули**, выполняющие вспомогательные функции ОС.

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.

Современные ОС имеют довольно сложную структуру, каждый её элемент выполняет определённые функции в компьютере.

Структура ОС

Программный модуль

Командный процессор

Драйверы

Графический интерфейс

Сервисные программы

Справочная система

Программный модуль

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

Командный процессор — часть операционной системы, обрабатывающая команды, вводимые с терминала или из командного файла, и запускающая задачи для их выполнения.

Команда — описание элементарной операции, которую должен выполнить компьютер или формальный исполнитель. У каждой команды есть свой двоичный код. Пользователь даёт команду запуска программы, проведения какой-либо операции над файлами (копирование, удаление, переименование), вывода документа на печать, а операционная система выполняет эту команду.

Драйверы

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.

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.

Сервисные программы

Сервисные программы, или **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.

Справочная система

Программное обеспечение отличается высокой сложностью, поэтому и в операционной системе, и в большинстве её приложений предусмотрены справочные системы.

Справочная система обеспечивает получение информации о функционировании ОС в целом и о работе её отдельных модулей.

■ Классификация ОС

По числу одновременно обслуживаемых пользователей

- ▲ Однопользовательские — одно- временно поддерживается не более одного сеанса пользователя.
- ▲ Многопользовательские — одно- временно поддерживается множе- ство сеансов пользователя.

По числу одновременно выполняемых процессов

- ▲ Однозадачные — не более одно- го работающего процесса.
- ▲ Многозадачные — множество работающих процессов.

По разрядности кода ОС

- ▲ 8-разрядные.
- ▲ 16-разрядные.
- ▲ 32-разрядные.
- ▲ 64-разрядные.

По назначению

- ▲ Универсальные — рассчитаны на решение любых задач пользователя.
- ▲ Специализированные ОС — ра- ботают 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 се- мейства Windows и системы класса UNIX — Linux и Mac OS).

■ Функции ОС

ОС является необходимой состав- ляющей программного обеспечения, без неё компьютер работать не мо- жет. К **базовым функциям ОС** от- носятся:

- ▲ планирование заданий;
- ▲ использование процессора;
- ▲ обеспечение программ средства- ми коммуникации и синхронизации;
- ▲ управление памятью;
- ▲ управление файловой системой;
- ▲ управление вводом-выводом;
- ▲ обеспечение безопасности.

Дополнительные функции ОС:

- ▲ многозадачность;
- ▲ защита системы, а также поль- зовательских данных и программ;
- ▲ разграничение прав доступа и многопользовательский режим ра- боты (аутентификация, авторизация).

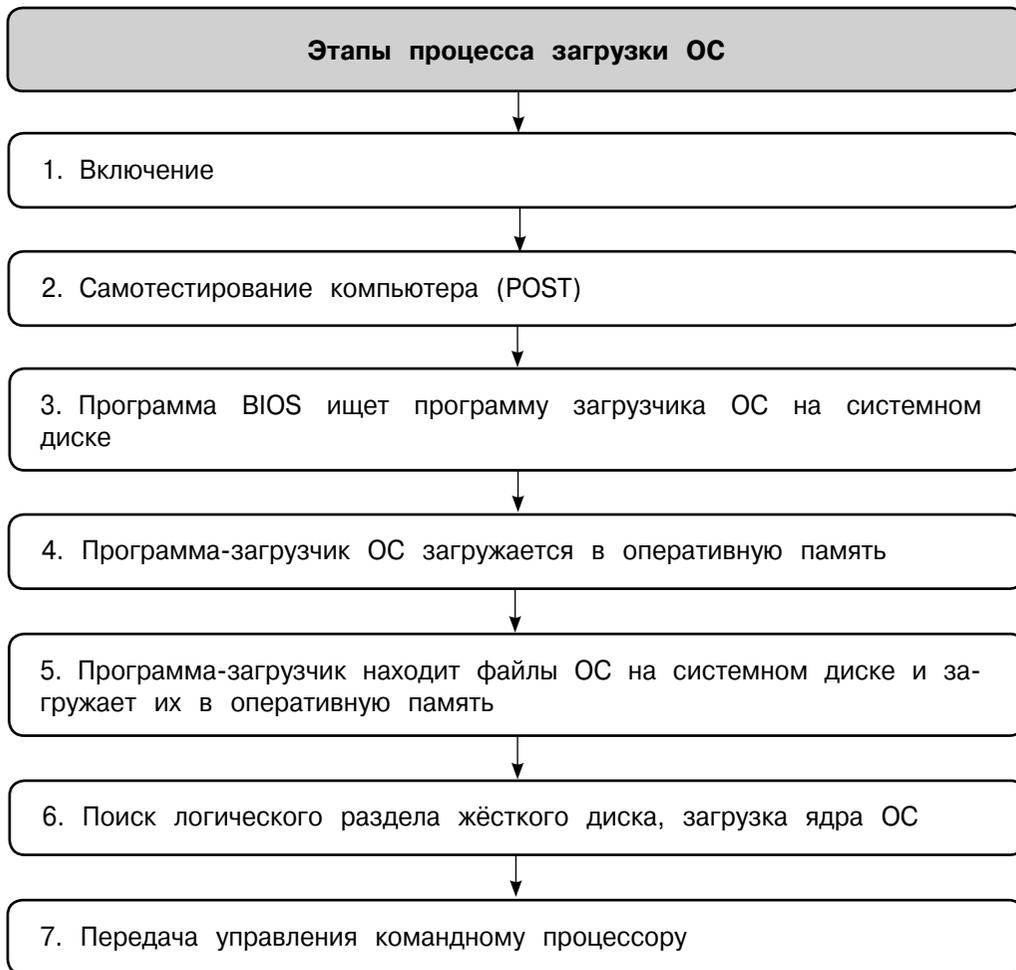
■ Загрузка ОС

Для понимания процесса загрузки ОС необходимо изучить следующие понятия и сокращения.

Системный диск — жёсткий, гиб- кий или лазерный диск, на котором находятся файлы операционной си- стемы и с которого производится её загрузка.

POST (Power-On Self-Test) — само- тестирование после включения.

BIOS (Basic Input/Output System) — базовая система ввода-вывода.



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.

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.

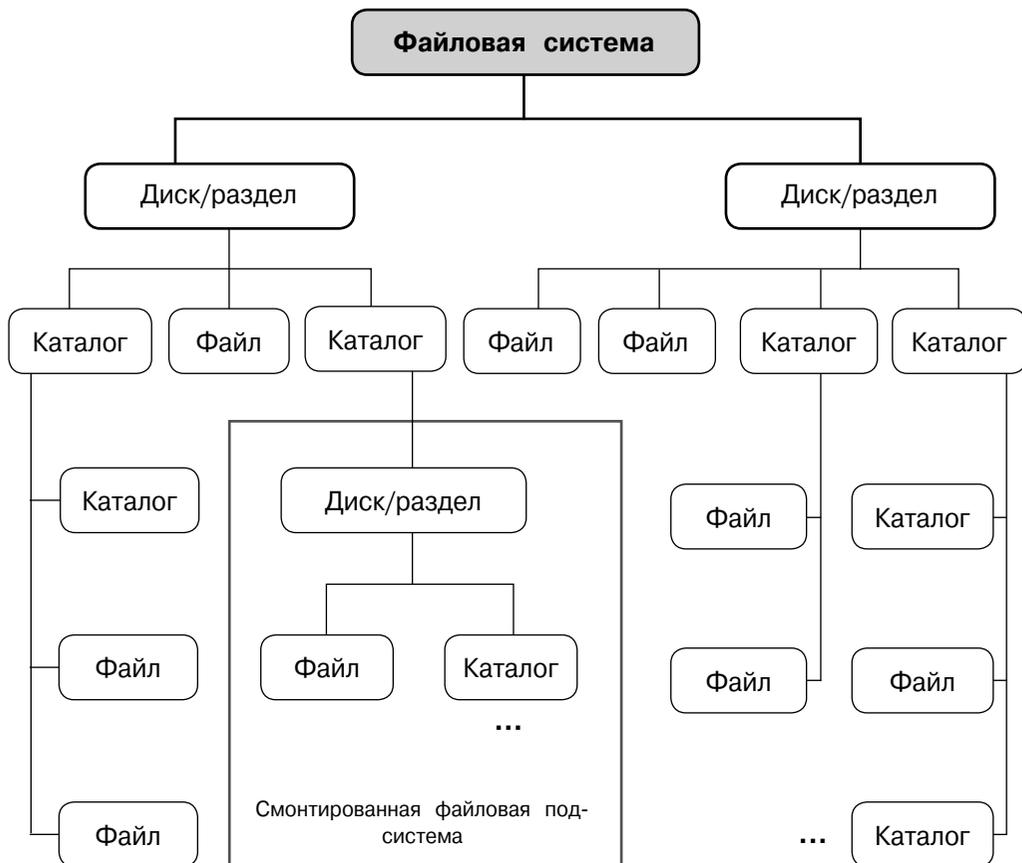
Каталоги — системные файлы, предназначенные для поддержки структуры файловой системы. Если на диске хранится много файлов, то для упрощения и удобства поиска их объединяют в папки по тематике. Одни папки могут находиться внутри других, в таком случае их называют вложенными каталогами или папками.

Файл — программа или данные, имеющие имя и хранящиеся в долговременной (внешней) памяти.

Для операционной системы файл — это просто последовательность байтов. Каждый файл имеет своё расширение, исходя из его назначения.

✓ Файл «12d.exe» имеет название «12d» и расширение «.exe» — служебный файл.

✓ Файл «Задачи 10 класс.docx» имеет название «Задачи 10 класс» и расширение «.docx» — текстовый файл.



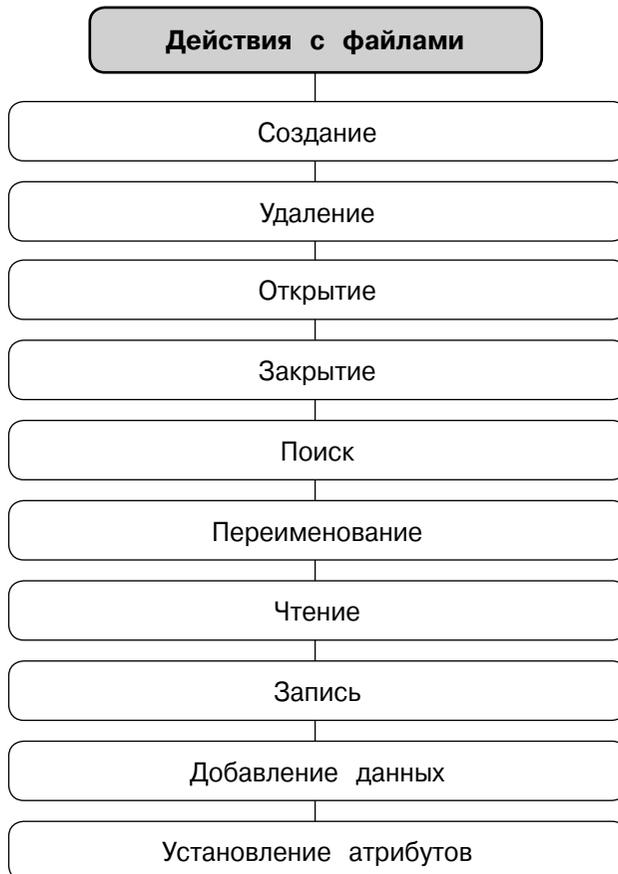
Тип файла	Расширение
Исполняемые файлы	.exe, .com
Текстовые файлы	.txt, .doc, .docx, .rtf
Файлы баз данных	.dbf, .mdb
Графические файлы	.bmp, .tif, .cdr, .jpg
Звуковые файлы	.mod, .wav, .wma
Видеофайлы	.avi, .mpg, .mp4
Форматы Internet	.html, .asp, .asf
Системные файлы	.sys, .bat

Каждой папке также присваивается имя, но, в отличие от файла, название папки пишется без расширения. Кроме имени, у файла имеются также иные данные. Все операционные системы связывают

с файлом другую информацию, например его размер, дату и время последней модификации. 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, ненулевое значение — заблокированный
Длина записи	Количество байтов в записи
Позиция ключа	Смещение ключа внутри каждой записи

Атрибут	Значение
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 для работы с текстом

Текстовые редакторы

Редакторы кодов программ

Текстовые процессоры

НИС

▲ **Текстовые редакторы.** Позволяют создавать, редактировать, сохранять текстовую информацию. Функция форматирования вида текста отсутствует.

✓ Стандартный Блокнот от Windows, Notepad, Keypad.

▲ **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 Editor, 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 он будет выводиться на печать, предоставляют широкие возможности для редактирования. Они позволяют изменять размер, вид и цвет шрифта, от-

ступы абзацев, параметры страниц, вставлять в текст таблицы, диаграммы, отдельные формулы, графические изображения и звуковые объекты.

✓ Microsoft Word, LibreOffice Writer, WPS Office Writer, МойОфис Текст.

▲ **Настольные издательские системы (НИС).** Расширенные программы, предназначенные для профессиональной издательской деятельности, позволяющие осуществлять компьютерную вёрстку

книг и периодических изданий — газет и журналов.

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.

✓ QuarkXPress, Adobe InDesign, Scribus, Microsoft Publisher, Apple Pages.

Что позволяют делать НИС

Компоновать (верстать) текст

Использовать всевозможные шрифты и полиграфические изображения

Обрабатывать графические изображения

Обеспечивать вывод документов высокого качества и др.

Осуществлять редактирование на уровне лучших текстовых процессоров

РАБОТА С ТЕКСТОВЫМ ДОКУМЕНТОМ

■ Основные элементы текстового документа

Символ — минимальная единица текстовой информации.

Слово — произвольная последовательность букв и цифр, ограниченная с двух сторон служебными символами (в вычислительной и иной программируемой технике применяется понятие «машинное слово»).

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 — элемент текстового документа, который 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.

■ Виды работы с текстовым документом

Открытие текстового документа

1. Найти файл на рабочем столе или в папке

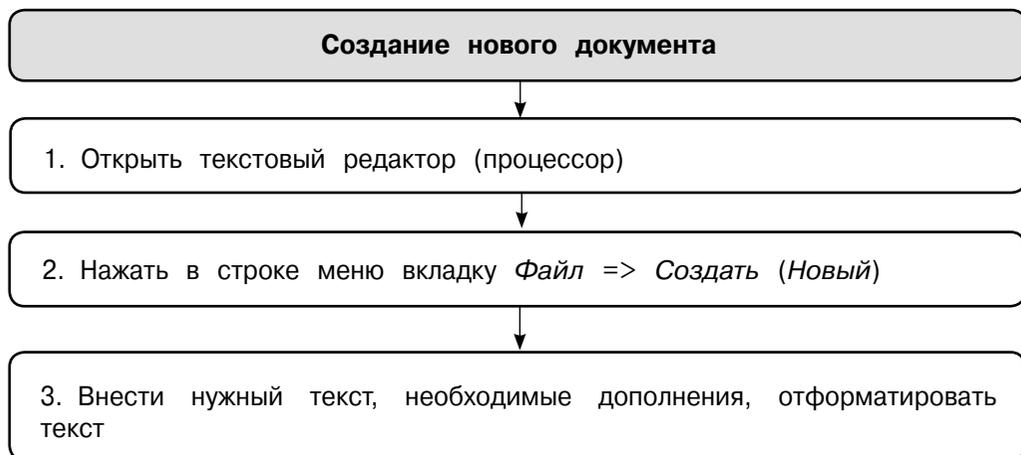
2. В зависимости от настроек кликнуть на нём один или два раза или вызвать контекстное меню правой кнопкой мыши и выбрать *Открыть*

Сохранение текстового документа

1. Нажать *Файл => Сохранить как*

2. Назвать документ любым именем и сохранить

3. Закрыть документ или продолжить редактирование



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

Редактирование

Редактирование предполагает внесение изменений в текст документа.

К редактированию можно отнести следующие операции:

- ▲ набор документа;
- ▲ копирование;
- ▲ перемещение частей текста;
- ▲ вырезание частей текста;
- ▲ исправление и т. п.

Форматирование

Процесс изменения характеристик документа или его частей.

К форматированию относятся следующие операции:

- ▲ изменение параметров абзацев;
- ▲ изменение параметров шрифта;

- ▲ изменение параметров заголовков;
- ▲ преобразование текстовой информации в таблицы;
- ▲ изменение колонтитулов, нумерация страниц.

При выборе конкретного стиля форматирования происходит автоматическое изменение ряда свойств документа, прописанное в коде данного стиля. Чаще всего в тексте приходится форматировать шрифт, для этого необходимо знать его параметры.

Параметры шрифта

- ▲ Тип шрифта (гарнитура). Это 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 символов, цвет фона символов.

▲ Эффекты: верхний и нижний индекс, зачёркивание, тень и т. д.

Средства поиска

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 ОС Windows — `Ctrl + F`.

Рассмотрим пример использования поисковых средств Microsoft Word для решения следующей задачи:

определить, сколько раз, не считая сносок, встречается слово «день» или «День» в предложенном тексте, при этом другие формы слова «день», такие как «полдень», «дни» и т. д., учитывать не следует.

Для решения задачи вызываем строку поиска с помощью сочетания клавиш `Ctrl + F`, вводим «день» с пробелами с обеих сторон слова, чтобы поисковая система не считала другие формы и слова, такие как «полдень» или «деньги». Число, отражающее количество употреблений заданного слова, появится под строкой поиска.

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 образцом или специально разработанным 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.

Если открыть шаблон, моментально создаётся его копия, которая имеет такое же расширение, как и у текстового документа: `.dotx`, `.dot`, `.dotm`.

Чаще всего шаблоны используются в различных организациях, где важно сохранить шапку документа, герб, водяной знак или другие отличительные черты. Шаблоны могут иметь определённую структуру — количество разделов, страниц или другие характеристики. Для предупреждения нежелательного изменения содержимого шаблона следует позаботиться о пароле. Шаблоны есть во многих текстовых редакторах и процессорах. Можно создать шаблон как новый документ или использовать готовый файл.



ПРОВЕРКА ОРФОГРАФИИ И ГРАММАТИКИ

Автоматическая проверка обеспечивает выявление ошибок в тексте непосредственно 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 *Правописание (Правописание и грамматика)* и т. п.

Автозамена служит для автоматического исправления ошибок или опечаток, которые наиболее часто возникают при наборе. В словарь автозамены заносятся ошибочные и соответствующие правильные варианты. При ошибочном наборе слова из этого словаря текстовый

редактор автоматически исправит его написание.

Словарь автозамены пополняется и редактируется. Пользователь мо-

жет самостоятельно настроить программу по своим предпочтениям, добавляя новые слова. 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.

С появлением персонального компьютера стали создаваться электронные словари, а затем и программы, способные переводить не только отдельные слова, но и целые тексты. Первые опыты их применения сложно было назвать корректными переводами, однако современные программы уже значительно продвинулись в решении задачи получить на выходе связный текст, адекватный исходному.

Области применения машинного перевода

Перевод письма

Чтение иностранных сайтов

Перевод инструкции для техники

Чтение профессиональной документации

Решение задач профессионального перевода

Программы машинного перевода пока не могут соперничать с человеком по качеству исполнения, хотя их использование значительно облегчает труд переводчика. 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, Abby 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 вычисления;
- ▲ обрабатывать и обобщать большие массивы числовых данных;
- ▲ упорядочивать по заданным критериям;
- ▲ создавать макеты страниц.

Вставка формулы

1. Нажать вкладку *Вставка* => *Формула* / *Вставить новую формулу* (в зависимости от редактора и его возможностей)

2. Вставить нужную формулу и значения для неё

Добавление таблицы

1. Нажать вкладку *Вставка* => *Таблица*

2. Выбрать нужное количество строк и колонок

Добавление простых рисунков

1. Нажать вкладку *Вставка* => *Рисунок, Фигура*

2. Выбрать рисунок из имеющихся на ПК или из Интернета. При необходимости выбрать из предложенного меню параметры цвета, заливки, изменения границ объекта и т. п.

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.

Добавление диаграммы

1. Выделить необходимые для диаграммы ячейки с данными

2. Нажать в строке меню вкладку *Вставка* => *Диаграмма*

3. Выбрать из предложенных вариантов нужную диаграмму

4. В созданной диаграмме изменить название и при необходимости отформатировать оси

ИСПОЛЬЗОВАНИЕ СИСТЕМ РАСПОЗНАВАНИЯ ТЕКСТОВ

Для ввода информации в компьютер часто используется сканер.

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 провести распознавание текста.

Преобразование документа в электронный вид происходит в три этапа, каждый из которых может выполняться программами как автоматически, так и под контролем пользователя.

Преобразование можно выполнить с помощью специальных программ. Наиболее распространённым инструментом для распознавания текста является ABBYY FineReader. Можно использовать также Adobe Acrobat DC, FreeOCR, gImageReader, ABBYY Screenshot Reader, Free Online OCR.

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

В зависимости от способа сканирования объекта и самих объектов сканирования различают несколько видов сканеров.

▲ **Планшетные.** Являются наиболее распространёнными, поскольку 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, единственными его плюсами являются дешевизна и мобильность, при этом он имеет массу недостатков: низкое разрешение, малая скорость работы, узкая полоса сканирования. Кроме того, возможны перекосы изображения, поскольку пользователю сложно перемещать сканер с постоянной скоростью.

▲ **Сканеры штрихкода.** Небольшие компактные модели для сканирования штрихкодов товара в магазинах.

Кроме вышеперечисленных выделяют ещё сканеры отпечатков пальцев, листопротяжные, книжные, планетарные, барабанные, слайд-сканеры, 3D-сканеры.

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

■ 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.



Для подробного знакомства с программой можно изучить руководство пользователя ABBYY FineReader на официальном сайте.

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, в каком 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.

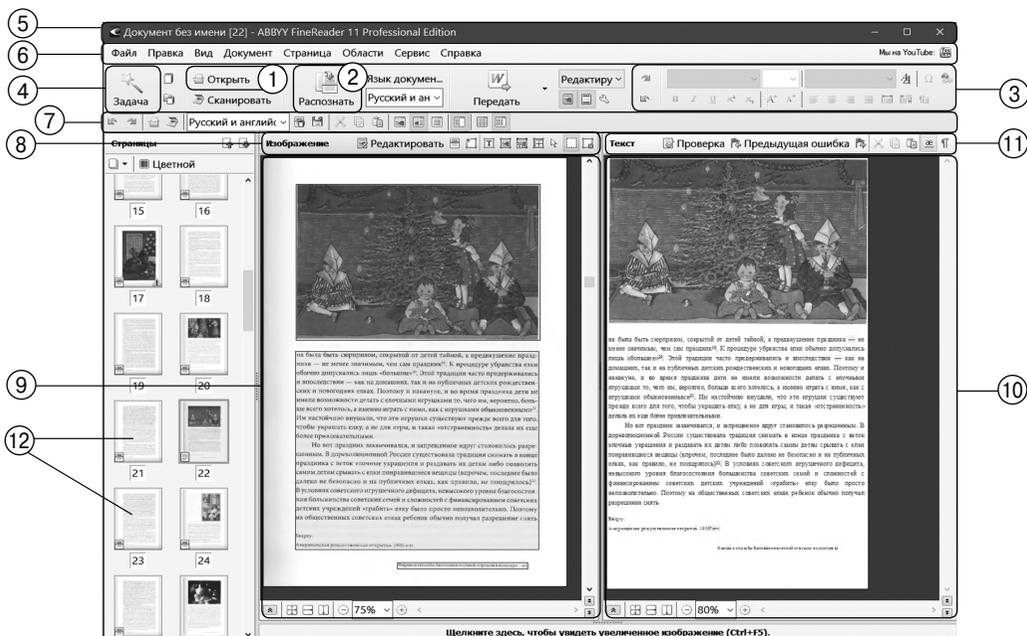
Если соперник не отступает, хозяин территории принимает более энергичные меры, танца вниз головой, бешено кусает ртом песок, словно желая сказать: «Если не уйдешь, я и тебя могу так отделить!» А если и это не устроило агрессора, тогда танцор разворачивается к нему широкой стороной тела и оттопыривает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

Вид текста после проверки орфографии

Редактирование полученного документа

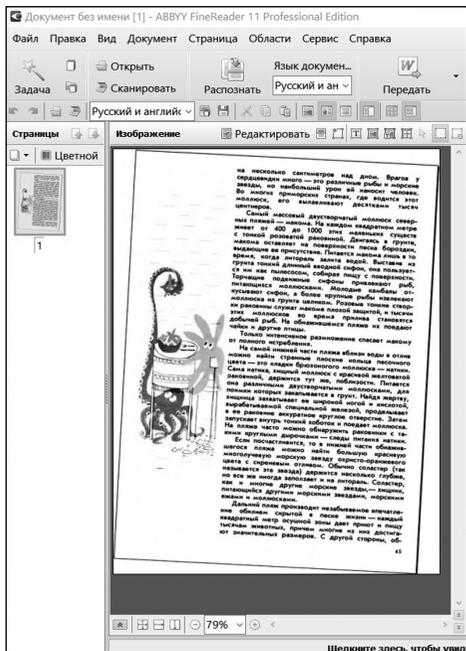
Все изображения, относящиеся к данному документу, могут быть скопированы, вырезаны, вставлены или повернуты с помощью соответствующей панели редактирования. Если изображение во время сканирования было немного повернуто, этот недостаток устраняется с помощью функции *Исправить перекос*, которая активизируется автоматически при преданализе изображения.

Функция *Фотокоррекция* позволяет исправить искажение строк, устранить размытие картинки.



Интерфейс ABBYY 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



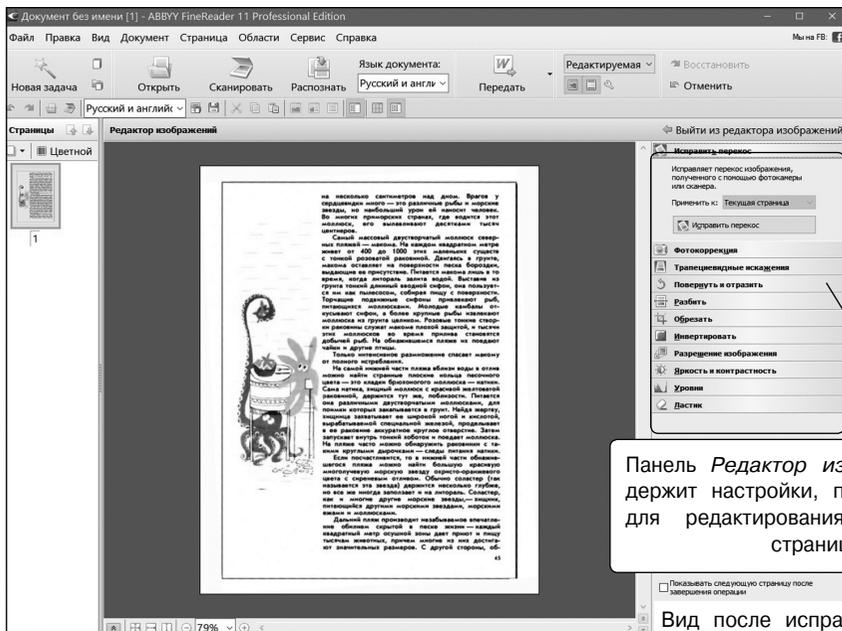
Вид до активизации кнопки *Исправить* перекося

Программа также допускает устранение трапециевидных 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

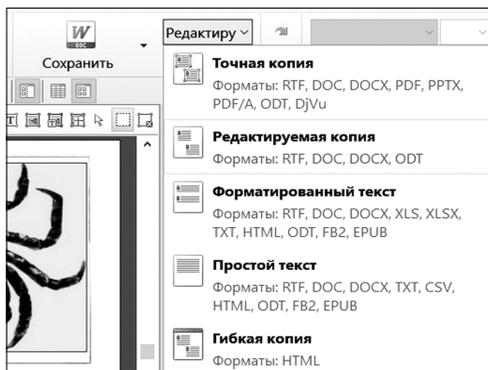
Редактирование всего текста осуществляется с помощью соот-



Панель *Редактор изображений* содержит настройки, предназначенные для редактирования объектов на странице

Вид после исправления перекося

ветствующей панели. Сначала необходимо выбрать режим редактирования (простой текст, точная копия, редактируемая копия, форматированный текст).



Выбор режима редактирования



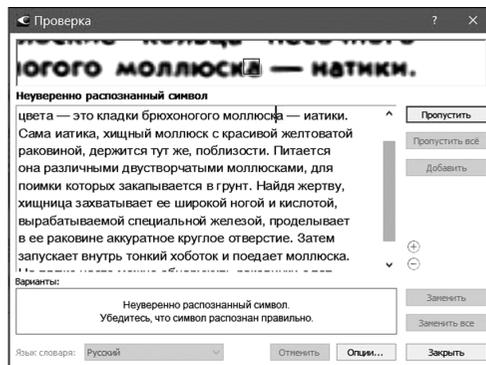
Панель *Текст* позволяет выявлять и исправлять ошибки в распознанном тексте



Панель редактирования текста

Пользователь может изменить шрифт текста, его размер, включить такие эффекты, как курсив, подчёркивание, выделение жирным шрифтом. Существуют возможности при необходимости выравнивать текст (четы-

ре варианта), изменить направление письма (справа налево), включить режим редактирования гиперссылок.



Пример работы в режиме проверки орфографии

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

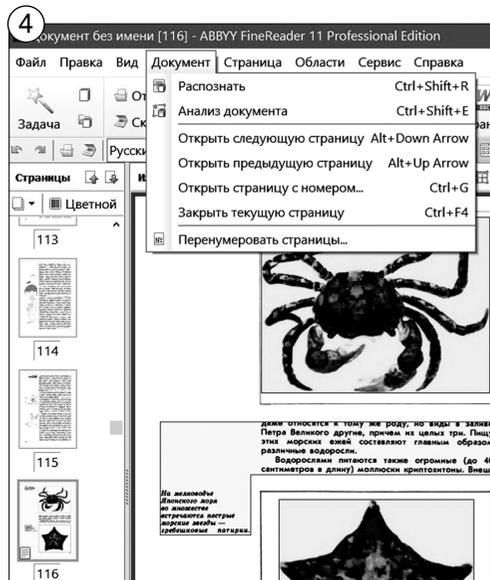
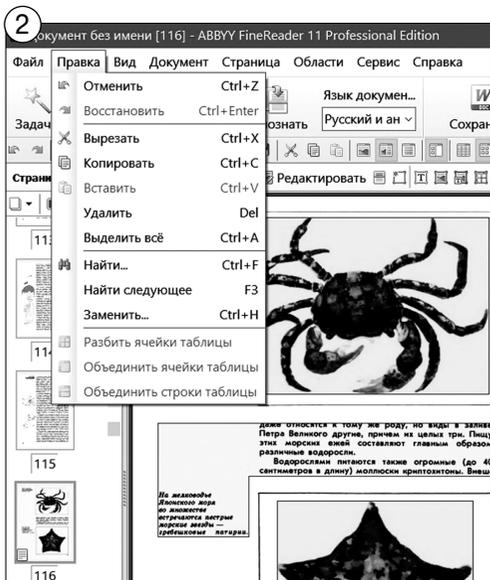
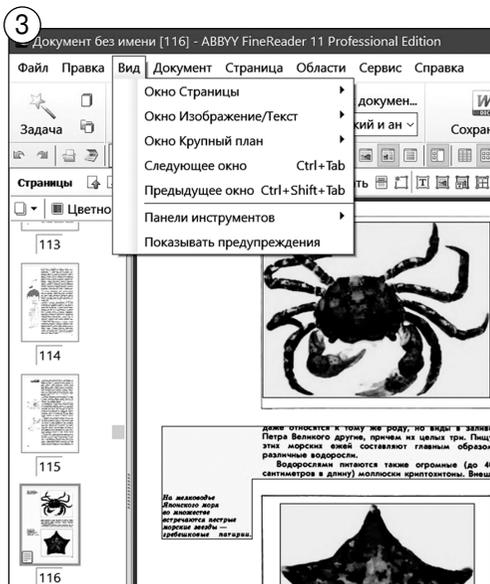
Изменение структуры документа

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

Работа с функционалом строки меню

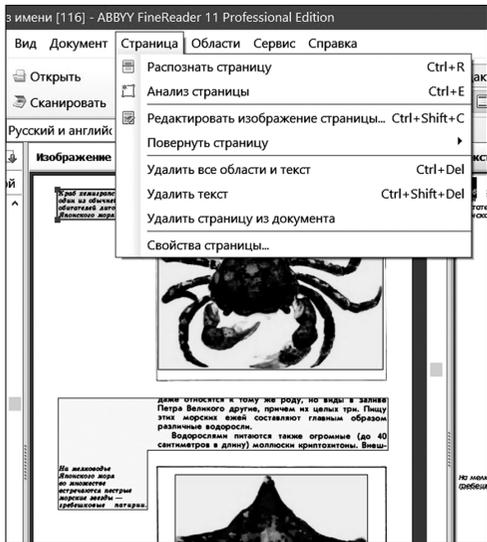
▲ Закладка *Файл* (1, с. 130) позволяет производить действия с самим файлом: сохранять, печатать, передавать и т. д.

▲ Закладка *Правка* (2, с. 130) помогает вставлять элементы, удалять, копировать, находить символы.

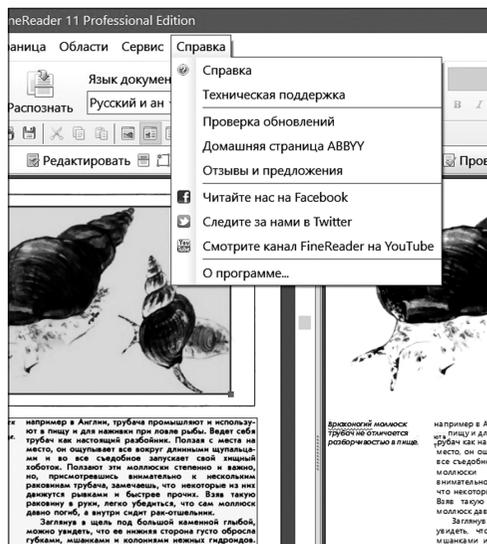


▲ Закладка *Вид* (3) отвечает за внешний вид и расположение окон *Изображение*, *Текст* и *Крупный план*, позволяет выбрать и настроить панели инструментов.

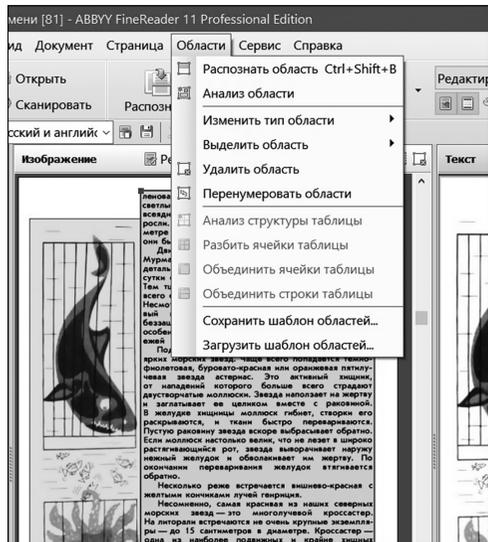
▲ Закладка *Документ* (4) позволяет запустить анализ и распознавание документа, перейти к предыдущей или следующей странице, а также к странице с указанным номером.



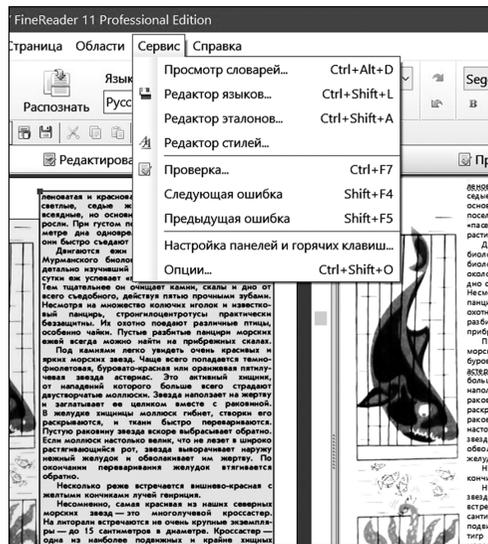
▲ Закладка **Страница** даёт возможность распознавать, анализировать страницу, редактировать отдельные области.



▲ Закладка **Справка** открывает справочные материалы.

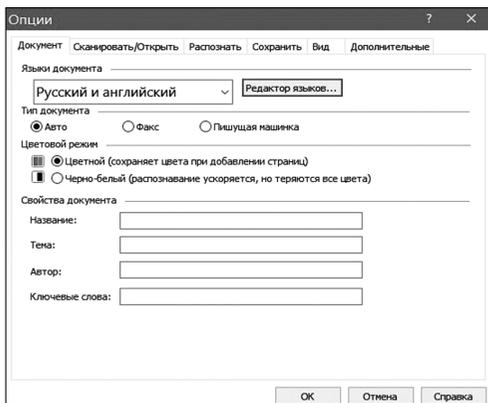


▲ Закладка **Области** помогает изменять тип области с изображения на таблицу, текст, картинку или штрихкод.

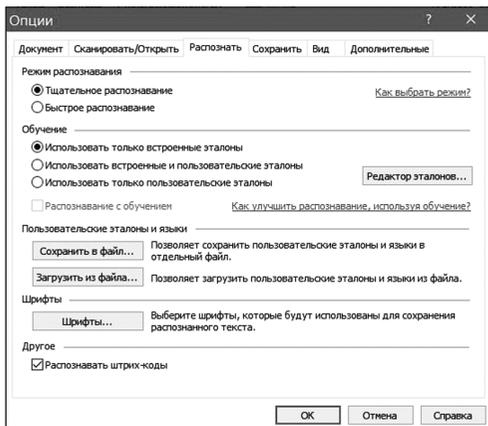


▲ Закладка **Сервис** позволяет использовать словари, редактор языков и проверку.

Рассмотрим подробнее раздел *Опции* в подменю *Сервис*.

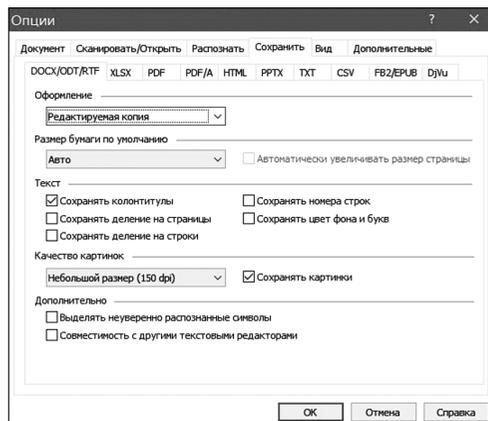


Вкладка *Документ* применяется, чтобы определить язык распознаваемого документа, тип документа, цветность, задать атрибуты конечного файла (указать название, тему и автора документа, ввести ключевые слова).

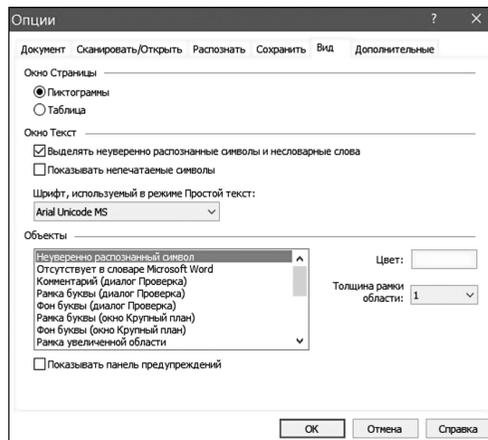


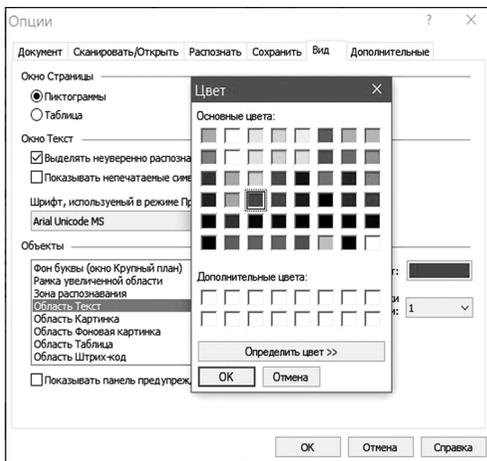
Вкладка *Распознать* определяет режим распознавания (быстрый или тщательный), а также позволяет

применять пользовательский эталон для обучения программы.

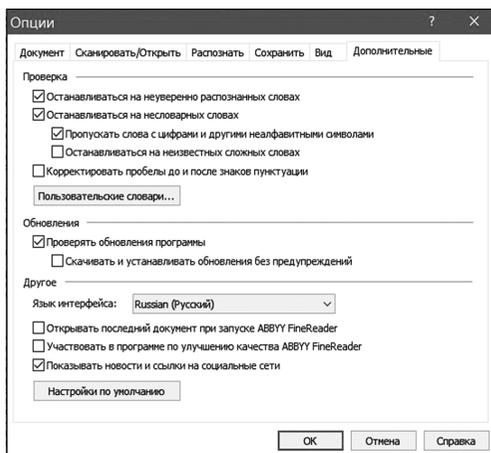


Вкладка *Сохранить* позволяет выбрать режим сохранения в файл типа *.rtf, *.doc, *.docx, *.xls, *.html, *.pptx, *.csv, *.txt или *.pdf. Здесь же пользователь выбирает оформление документа (форматируемый текст, точная копия, простой текст) и размер бумаги. Допускается сохранение текста вместе с колонтитулами и делением на страницы и строки. Для каждого формата имеются отличия в настройках.





Вкладка *Вид* 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.



Вкладка *Дополнительные* содержит список шрифтов, которые используются для отображения рас-

познанного 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.

Работа с функционалом панели быстрого доступа



Панель быстрого доступа

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.

Работа в разных режимах расшифровки

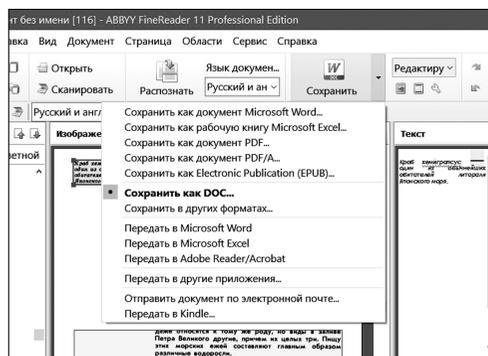
Помимо редактирования формата отсканированной страницы, пользователь может самостоятельно выделять области с текстом, картинками и таблицы, а затем распознавать

обработанную страницу. В некоторых условиях ручной режим определения типа блока может значительно повысить качество обрабатываемого документа. 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.

Сохранение файла

Обработанный документ может быть сохранён в формате Word, Excel или Acrobat Reader.



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.

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.

ФОРМАТЫ ГРАФИЧЕСКИХ ОБЪЕКТОВ

Компьютерная графика — технология создания и обработки графических изображений с помощью аппаратных и программных средств компьютера.

Компьютерная графика бывает **двухмерной** (обозначается 2D; от англ. *two dimensions* — «два измерения») и **трёхмерной** (3D; от англ. *three dimensions* — «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.

Способы хранения изображений	
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
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, Picture Man Painter, PhotoFinish, Paint	CorelDRAW, Adobe Illustrator, MS Draw

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.

■ Форматы растровых изображений

▲ **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. Фор-

мат 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

▲ **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 применяется в технических областях: конструкторской, авиационной и др.

▲ **SVG.** Формат предусматривает работу с двумерной векторной гра-

фией. 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 отображения деталей. 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

Способы получения цифровых графических объектов

Копирование готовых изображений с цифровой фотокамеры, из Интернета

Создание новых изображений с помощью программного обеспечения

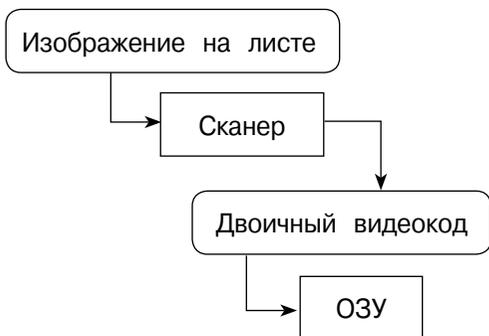
Ввод графических изображений с помощью сканера

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

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, CorelDRAW, OpenOffice.org и др.

Ввод графических изображений, существующих на бумажных носителях, с помощью сканера происходит в следующем 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.

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, форматирования и вывода изображений, свои аппаратные и программные средства.

Основные способы обработки изображения

Изменение масштаба изображения

Изменение размера с помощью обрезки

Коррекция изображения:

- ▲ изменение контраста, яркости, насыщенности;
- ▲ задание параметров резкости или размытия

Работа с файлом в графическом редакторе

1. Открыть графический редактор

>>>

2. Найти вкладку *Файл* => *Открыть*. В появившемся окне выбрать нужный файл

3. Провести необходимые изменения в файле

4. Сохранить в требуемом формате

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 составляет 90 дБ, работающего перфоратора — 100 дБ, шестла листьев — 10 дБ.

Разность между максимальной и минимальной интенсивностью звука называется **динамическим диапазоном**, для человеческого уха она составляет 140 дБ.

Звуки разделяются на **естественные** (речь, шум моря, свист ветра, звук скрипки) и **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 бит цифрового кода является звуковым сигналом в цифровой форме.

■ Форматы звуковых файлов

Применительно к звуковым файлам принято различать форматы **без потери качества (lossless-формат)** и **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 ин-

формации, и его полное восстановление становится невозможным.

■ Воспроизведение и запись звука

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

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**.

Компьютерная запись звука

1. Подключить источник сигнала к порту звуковой карты компьютера

2. Настроить звуковую карту — выбрать источник и уровень сигнала. Для этого может быть использована стандартная программа *Регулятор громкости*, входящая в состав ОС Windows (для других операционных систем следует использовать соответствующие им программы)

>>>

3. При выборе команды *Параметры => Свойства* откроется окно *Свойства*, в котором нужно выбрать переключатель *Запись*, указать нужный вход (Microphone или Line-in) и установить необходимый уровень записи

4. Далее можно использовать любой звуковой редактор

Для создания и редактирования цифровых звукозаписей применяются специальные программы — **звуковые редакторы**, например Adobe

Audition, Sound Forge. Они позволяют записывать, воспроизводить и синтезировать звук, а также проводить его сложную обработку.

Создание нового файла в звуковом редакторе

1. Открыть звуковой редактор

2. Выбрать команду *File => New (Файл => Создать/Новый)*

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.

✓ Речь с микрофоном в нестудийных условиях с приемлемым качеством можно записать с такими параметрами: частота дискретизации 8000 Гц, 8 бит, 1 канал

4. Для прослушивания звукового файла подключить к линейному выходу звуковой карты колонки или наушники

■ Программы, работающие с аудиофайлами

▲ **Аудиоконвертеры.** Программы, которые переводят аудиофайлы из одного формата в другой.

✓ FormatFactory, Free Audio Converter, Mobile Media Converter.

▲ **Аудиоплагины.** Программы, позволяющие расширить возможности проигрывателей и редакторов.

✓ Vocoder, Stereo Tool, WhiteCap.

▲ **Программы для записи звука.**

В таких программах аудиопоток может записываться как с микрофона, так и со звуковой карты компьютера.

✓ NanoStudio, Free Audio Recorder, Echo, Pro Tools.

▲ **Микшеры, синтезаторы.** Набор программ, благодаря которым пользователь может самостоятельно

создавать музыку, а также редактировать, микшировать и записывать аудиофайлы.

✓ Adobe Audition CC, Virtual DJ Studio, PianoFX STUDIO.

▲ **Аудиоредакторы.** Программы, предназначенные для редактирования аудиофайлов любых форматов.

✓ Audacity, MP3Gain, Sibelius, MAGIX Music Maker, Adobe Audition CC.

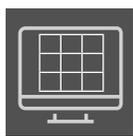
▲ **Аудиоплееры и аудиопроигрыватели.**

Программы для воспроизведения музыкальных файлов.

✓ Winamp, iTunes, Winyl Player, AIMP.

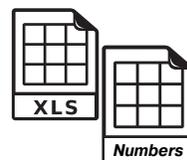
▲ **Аудиокодеки.** Программы, с помощью которых можно воспроизводить и редактировать звуковые файлы.

✓ Windows Essentials Codec Pack, FLAC for Windows, LameXP, MP3 Encoder.



ТЕХНОЛОГИЯ ОБРАБОТКИ ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

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



ТАБЛИЧНЫЙ ПРОЦЕССОР

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

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

Электронная таблица позволяет хранить большое количество исходных данных, результатов, а также связей между ними (математических или логических соотношений). При изменении исходных данных с ис-

пользованием специальных формул все результаты автоматически пересчитываются. Электронная таблица является основным объектом обработки табличным процессором. Она, как и таблица в текстовом процессоре MS Word, состоит из столбцов и строк, на пересечении которых образуются ячейки. В ячейках ЭТ могут храниться числа, тексты и формулы для их обработки.

Табличные процессоры являются полезными при ведении учёта семейных доходов и расходов, заполнении налоговой декларации и пр.

Предназначение табличных процессоров

Автоматизация расчётов

Моделирование различных ситуаций

Отсеивание изменений получаемых результатов

Выбор наиболее подходящего варианта из множества решений задачи

Удобная реализация для проведения бухгалтерских и статистических расчётов

Выполнение расчётов прочности конструкции

Ведение учёта товаров в магазине

Можно использовать табличные процессоры в учебной деятельности для решения математических и экономических задач, обработки результатов исследований, при выполнении практических и лабораторных работ по химии и физике, в разработке плана кредитования и пр.

Современные табличные процессоры:

▲ Excel (англ. *excel* — «преобладать, превосходить») — входит в па-

кет программ Microsoft Office, один из самых популярных сегодня;

▲ Calc (англ. *calculator* — «вычислитель, калькулятор») — входит в пакет программ StarOffice;

▲ GNumeric (англ. *GNU* — проект по созданию программного обеспечения свободного распространения, *numeric* — «числовой») — находится в свободном распространении;

▲ Numbers (англ. *numbers* — «числа») — разработан Apple Inc. как часть пакета iWork.

Функции табличного процессора

Ввод данных в ячейки электронных таблиц, их редактирование и форматирование

Построение диаграмм и графиков на основе данных, которые содержатся в ячейках электронных таблиц

Поиск данных в соответствии с указанными условиями

Вычисление новых результатов при изменении данных или формул для их обработки

Обработка числовых данных по формулам, которые могут содержать арифметические выражения, математические, логические, статистические и другие функции

Отбор данных, которые удовлетворяют определённым условиям (сортировка данных)

Печать данных ЭТ, диаграмм и графиков

В табличных процессорах возможно создание гипертекстовых ссылок с целью перехода в другие файлы, хранящиеся на компьютере, в ло-

кальной сети или сети Интернет. Данные и диаграммы можно сохранять как отдельную веб-страницу.

ОБЪЕКТЫ ТАБЛИЧНОГО ПРОЦЕССОРА EXCEL И ИХ СВОЙСТВА

Рассмотрим работу табличного процессора на примере MS Excel (см. рис. ниже).

Каждый лист книги имеет имя, которое указывается на ярлычке листа. Если на листах размещаются электронные таблицы, то такие листы по умолчанию имеют имена Лист1, Лист2 и т. д. Лист можно переименовать двойным кликом.

Основные объекты

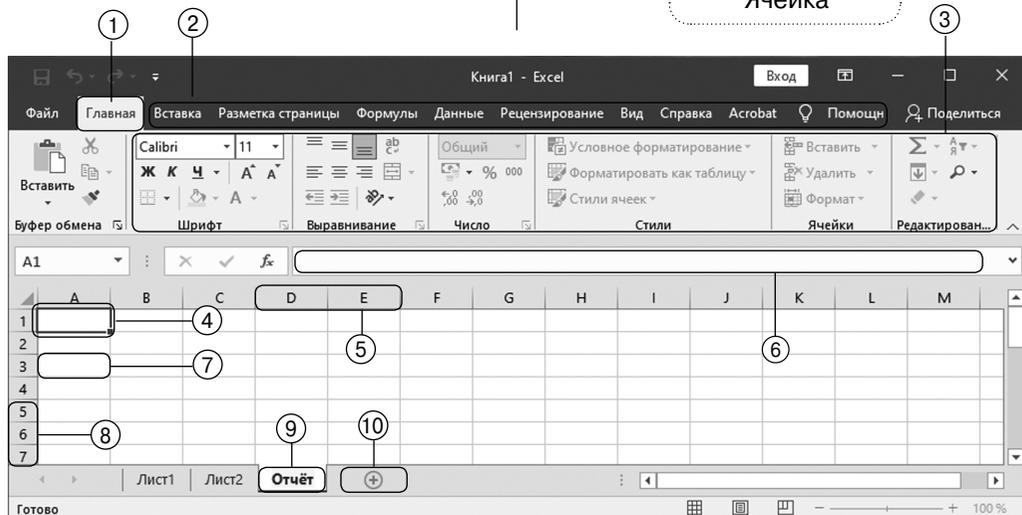
Электронная книга

Лист

Электронная таблица

Строка и столбец

Ячейка



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

Если листы предназначены для размещения только диаграмм, они по умолчанию имеют имена Диаграмма1, Диаграмма2 и т. д.

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.

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 связанного диапазона задаётся адресами двух ячеек, расположенных в противоположных углах диапазона. При этом адреса ячеек разделены двоеточием.

Для выделения целого столбца следует щёлкнуть левой кнопкой мыши по его названию (латинской букве), для выделения строки — щёлкнуть 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 B, — B:B.

Чтобы указать адрес несвязного диапазона ячеек, нужно указать адреса его связанных частей, разделив их точкой с запятой.

Ячейка или диапазон	Использование
Ячейка на пересечении столбца B и строки 12	B12
Диапазон ячеек: столбец A, строки 12—22	A12:A22
Диапазон ячеек: строка 15, столбцы B—E	B15:E15
Все ячейки в строке 5	5:5
Все ячейки в строках с 2-й по 10-ю	2:10
Все ячейки в столбце B	B:B
Все ячейки в столбцах с H по P	H:P
Диапазон ячеек: столбцы A—X, строки 10—20	A10:X20

■ 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.

В каждой ячейке по мере необходимости можно изменить следующие параметры:

- ▲ размер, тип и цвет шрифта;
- ▲ величина ячеек;
- ▲ заливка;
- ▲ границы;
- ▲ формат;
- ▲ выравнивание текста.

Создание электронной книги

1. Найти на рабочем столе ярлык программы или найти программу с помощью меню *Пуск*

2. Один или два раза в зависимости от настроек кликнуть по значку программы Excel

3. На панели инструментов найти кнопку меню *Файл => Создать*

Сохранение файла

1. На панели инструментов найти кнопку меню *Файл => Сохранить*

2. В появившемся окне указать имя файла, тип файла, выбрать диск и папку, в которой будет храниться рабочая книга, и нажать кнопку *Сохранить*

3. При сохранении файла под другим именем использовать команду *Файл => Сохранить как*

4. Для форматирования документа открыть файл и произвести изменения, затем сохранить изменения

■ 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.

✓ Найдём максимальное значение температуры, указанной в ячейках B1—B120.

	A	B	C	D	E
1	1 июня	24	24		
2	2 июня	25			
119			
120	1 октября	12			

Для поиска максимального значения температуры воспользуемся

Частями формул в Excel могут выступать:

- ▲ знаки математических и логических операций;
- ▲ ссылки на другие ячейки;
- ▲ функции;
- ▲ числа или текст.

формулой `=МАКС(B1:B120)` в ячейке C1 (или в другой свободной ячейке). Нажимаем кнопку *Enter* (ввод) или просто переходим в свободную ячейку. В ячейке с формулой появится нужный ответ.

	A	B	C	D	E	F	G	H
1	1 июня	24	=МАКС(B1:B120)					
2	2 июня	25						
119						
120	1 октября	12						
121								
122								
123								
124								

Аргументы функции

МАКС

Число1 B1 = 24

Число2 B120 = 12

Число3 = число

Введение формулы

1. Поставить курсор в нужную ячейку

2. Нажать знак «=»

3. Ввести нужную формулу

4. Нажать *Enter*. В ячейке появится результат вычислений

✓ Чтобы умножить значение ячейки В4 на 0,2, необходимо выполнить следующие действия: в строке формул поставить знак равенства, щёлкнуть по ячейке В4, поставить знак умножения, написать 0,2 и нажать *Enter*.

	A	B	C	D
1	Петров	10		
2	Сидоров	18		
3	Иванов	25		
4	Харламов	15	=B4*0,2	
5				

	A	B	C	D
1	Петров	10		
2	Сидоров	18		
3	Иванов	25		
4	Харламов	15	3	
5				

Ниже приведены некоторые операции программы 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")
SIN (функция)	Синус угла в радианах	=SIN(E4)
Квадратный корень	Находит квадратный корень из числа	=КОРЕНЬ(A4+B4)
МИН (функция)	Находит наименьшее значение из выбранных элементов	=МИН(A3:C3;A8:C8)
МАКС (функция)	Находит наибольшее значение из выбранных элементов	=МАКС(A3:C8)
СЧЁТ (функция)	Определяет количество чисел в выбранном промежутке	=СЧЁТ(A3:C3;A8:C8)



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.

▲ **База данных.** Ядро банка данных, структура, позволяющая хранить информацию об объектах в строго упорядоченном виде.

▲ **Системы управления базами данных.** Создание баз данных, 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.

▲ **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:

- связи между файлами;
- модель представления данных;
- информация о пользователях (о разграничении доступа, парольной защите).

Иногда роль словаря данных берёт на себя СУБД.

▲ **Администратор банка данных.** Обеспечивает выполнение следующих функций:

- 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:

- сетевой администратор;
- программисты;
- статисты;
- технический персонал.

ТИПЫ МОДЕЛЕЙ БАЗ ДАННЫХ

В зависимости от связей, которые устанавливаются между единицами информации, базы данных делятся на реляционные (табличные), иерархические, сетевые, объектно-ориентированные, многомерные и гибридные. Последние могут включать элементы всех перечисленных систем.

■ Реляционная модель

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

✓ MS Access, Fox Pro, Paradox, dBase.

+ Преимущества

- ▲ Простота организации.
- ▲ Удобство.
- ▲ Наглядность.

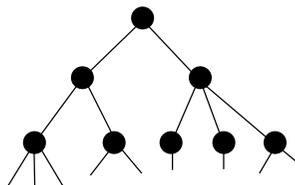
- ▲ Высокая скорость обработки.
- ▲ Простота реализации.

— Недостатки

- ▲ Строгость записи.
- ▲ Наличие 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**

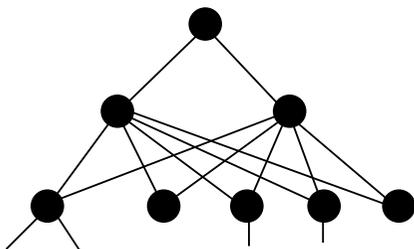
- ▲ Наглядность структуры.
- ▲ Эффективное использование памяти.
- ▲ Небольшое время обработки данных.

⊖ **Недостаток**

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.

⊕ **Преимущества**

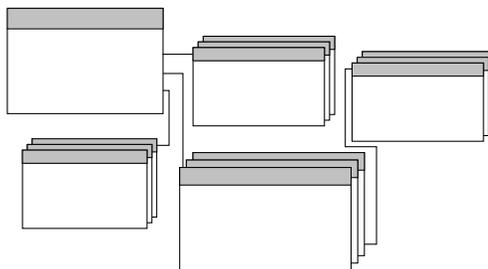
- ▲ Учёт разнообразия связей между объектами.
- ▲ Небольшие затраты памяти.

⊖ **Недостатки**

- ▲ 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.

⊕ **Преимущество**

Простота представления объектов со сложными связями.

⊖ **Недостатки**

- ▲ Большая сложность понятий.
- ▲ Невысокая скорость обработки.
- ▲ Неудобная обработка данных.

■ **Многомерная модель**

Данная модель возникла параллельно с реляционной, поскольку они похожи по своей структуре. Её можно представить как трёхмерную модель в виде куба, а в случае, когда необходимо использовать более трёх измерений, — в виде гиперкуба.

✓ InterSystems Cache, OpenQM, Rocket U2.

+ **Преимущества**

- ▲ Компактное размещение базы информации на диске.
- ▲ Небольшое время обработки информации.

- **Недостатки**

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

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

ТАБЛИЧНЫЕ БАЗЫ ДАННЫХ

Наиболее простой и наглядной 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.

Табличные базы данных состоят из двумерных таблиц.

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).

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

Поле — место для основных элементов в базе данных. От выбора типа поля в настройках зависит тип расположенных в нём элементов. В поле задаётся информация об операциях, производимых над размещёнными элементами.

Свойства поля:

- ▲ размер — выражается в знаках (или символах);
- ▲ имя — должно быть уникальным для каждого поля;
- ▲ подпись — отображается в заголовке столбца;
- ▲ формат — устанавливает формат данных.

■ Типы данных, находящиеся в поле

▲ **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-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 медиа-файлы.

▲ **Счётчик.** Для данных, которые могут изменять свои значения в результате наращивания.

▲ **Поле MEMO.** В таком поле может быть текст размером до 65 535 символов.



Практическое задание

1

Ниже приведены фрагменты таблиц базы данных победителей городских предметных олимпиад.

Школа	Фамилия
№ 10	Иванов
№ 10	Петров
№ 10	Сидоров
№ 50	Кошкин
№ 150	Ложкин
№ 150	Ножкин
№ 200	Тарелкин
№ 200	Мискин
№ 250	Чашкин

Фамилия	Предмет	Диплом
Иванов	физика	I степени
Чашкин	математика	III степени
Сидоров	физика	II степени
Кошкин	история	I степени
Ложкин	физика	II степени
Ножкин	история	I степени
Тарелкин	физика	III степени
Петров	история	I степени
Мискин	физика	I степени

Учитывая, что победителями считаются дипломанты первой степени, определите, сколько различных школ имеют победителей олимпиады по физике.

Решение:

- 1) Находим во второй таблице фамилии победителей олимпиады (диплом I степени): Иванов, Кошкин, Ножкин, Петров, Мискин.
- 2) Отбираем тех из них, кто получил диплом по физике: Иванов, Мискин.
- 3) Находим из первой таблицы, что Иванов и Мискин учатся в разных школах.

Ответ: 2.

ИСПОЛЬЗОВАНИЕ ИНСТРУМЕНТОВ ПОИСКОВЫХ СИСТЕМ (ФОРМИРОВАНИЕ ЗАПРОСОВ)

Наиболее важными функциями при работе с базами данных являются фильтр информации, сортировка, поиск и запрос.

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.

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.

СПЕЦИАЛЬНЫЕ СИМВОЛЫ И ИХ ЗНАЧЕНИЯ

*	Означает любую последовательность символов произвольной длины, в том числе может задавать и пустую последовательность
?	Вместо данного знака может быть один (в том числе пустой) произвольный символ, который допускается использовать
!	Является исключающим знаком, т. е. исключает из поиска те символы, которые указаны за ним в квадратных скобках
[]	Включает в поиск определённую последовательность символов, стоящих в скобках
-	Позволяет включать в поиск любой символ, отвечающий выбранному диапазону

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.



Практические задания

2

Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, среди которых также могут встречаться следующие символы: «?» (вопросительный знак) — означает ровно один произвольный символ; «*» (звёздочка) — означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

В каталоге находятся 6 файлов:

- mustard.map;
- mustard.mp3;
- catarsis.mp4;
- vitarcon.mp4;
- taras.mp3;
- star.mp3.

Ниже представлено 8 масок. Сколько среди них таких, которым соответствуют ровно 4 файла из данного каталога?

tar.mp*	*?tar?*mp?	?*tar*.mp?*	*t*r*?.m?p*
???*???.mp*	???*???.m*	*a*.a*	*s*.mp*

Решение:

Рассмотрим каждую маску.

Маске *tar*.mp* соответствуют 5 файлов: все, кроме первого, т. к. в каждом файле есть tar и его расширение начинается сочетанием mp, а у первого файла — ma.

Маске *?tar?*mp? соответствуют 3 файла: mustard.mp3, catarsis.mp4, vitarcon.mp4; файлы taras.mp3, star.mp3 не подходят, поскольку у них нет первого или последнего символа после tar.

Маске ?*tar*.mp?* соответствуют 4 файла: mustard.mp3, catarsis.mp4, vitarcon.mp4, star.mp3.

Маске *t*r*?.m?p* соответствует 1 файл: mustard.map.

Маске ???*???.mp* соответствуют 3 файла: mustard.mp3, catarsis.mp4, vitarcon.mp4.

Маске ???*???.m* соответствуют 4 файла: mustard.map, mustard.mp3, catarsis.mp4, vitarcon.mp4.

Маске *a*.a* соответствует только 1 файл: mustard.map.

Маске *s*.mp* соответствуют 4 файла: mustard.mp3, catarsis.mp4, taras.mp3, star.mp3.

Ответ: 3 маски, которым соответствуют 4 файла из данного каталога.

3 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а логической операции «И» — символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

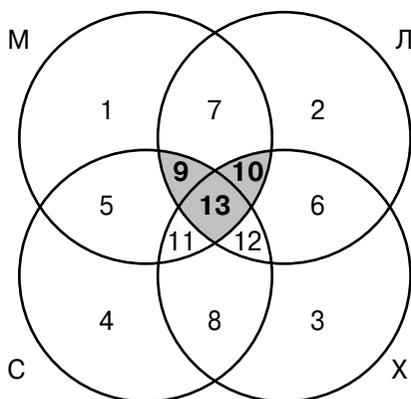
Запрос	Найдено страниц (тыс.)
Леннон & Маккартни & Старр	1100
Леннон & Маккартни & Харрисон	1300
Леннон & Маккартни & Старр & Харрисон	1000

Какое количество страниц (в тыс.) будет найдено по запросу «(Леннон & Маккартни & Старр) | (Леннон & Маккартни & Харрисон)»?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Решение:

Способ 1. Решим эту задачу с помощью кругов Эйлера. Для этого нарисуем четыре пересекающихся круга, т. к. у нас в таблице четыре различных поисковых запроса. Каждую область пересечения пронумеруем, получим 13 областей.



Количество запросов в данной области будем обозначать N_i . Наша цель — найти количество запросов в выделенной области.

$$N_9 + N_{13} + N_{10}$$

Тогда из таблицы находим, что:

$$N_9 + N_{13} = 1100,$$

$$N_{13} + N_{10} = 1300,$$

$$N_{13} = 1000.$$

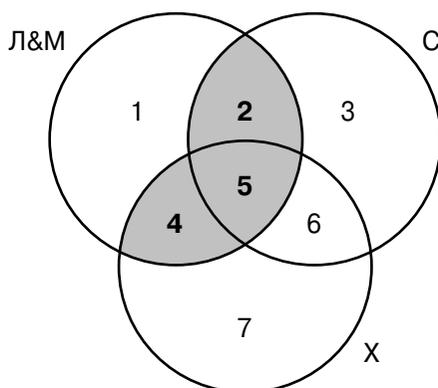
Сложим первое и второе уравнения:

$$N_9 + 2N_{13} + N_{10} = 2400.$$

Чтобы найти количество страниц по запросу «(Леннон & Маккартни & Старр) | (Леннон & Маккартни & Харрисон)», вычтем из правой и левой частей уравнения N_{13} . Получим:

$$N_9 + N_{13} + N_{10} = 1400.$$

Способ 2. Приведём другое решение.



Можно немного упростить решение, если вместо множеств, соответствующих запросам «Леннон» и «Маккартни», использовать множество «Леннон & Маккартни».

Тогда необходимо найти сумму:

$$N_2 + N_4 + N_5.$$

Из таблицы находим:

$$N_2 + N_5 = 1100,$$

$$N_4 + N_5 = 1300,$$

$$N_5 = 1000.$$

Сложим первое и второе уравнения:

$$N_2 + N_5 + N_4 + N_5 = 2400.$$

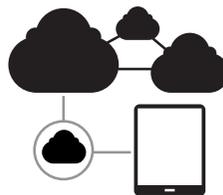
Отсюда $N_2 + N_4 + N_5 = 2400 - N_5 = 1400$.

Ответ: 1400.



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.

Современные телекоммуникационные технологии основаны на использовании глобальных компьютерных сетей. Для передачи данных посредством телекоммуникационных технологий применяется специальное программное обеспечение.

Использование средств телекоммуникационных технологий

Все сферы коммерческой и производственной деятельности

Ведение отчётности в учреждениях

Осуществление контроля работы предприятий

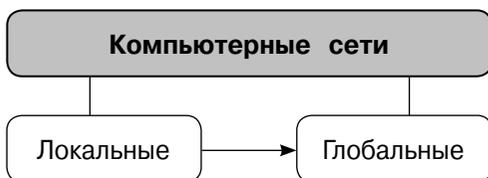
Дистанционное обучение

Личные цели и др.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СРЕДСТВ

■ Компьютерные сети

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



Локальная сеть (англ. *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 (англ. *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 — гло-

бальная компьютерная сеть. Каждый компьютер в сети имеет адреса трёх уровней.

■ Адресация в сети Интернет

Уровни компьютерных адресов в сети

1) MAC-адрес

2) IP-адрес

3) Адрес домена

MAC-адреса назначаются производителями оборудования и являются уникальными.

Связь между IP-адресами и доменными именами осуществляет международная база данных, основанная на использовании DNS-серверов (*Domain Name System*, или DNS, — доменная система имён).

IP-адрес является аналогом почтового адреса. Любой IP-адрес состоит из четырёх десятичных чисел со значением от 0 до 255, разделённых точками.

✓ IP-адрес: 198.168.70.36.



В настоящее время внедряют IPv6-адреса. С их помощью можно записать огромное количество вариантов. Адреса IPv6 отображаются как восемь четырёхзначных шестнадцатеричных чисел (т. е. групп по четыре символа), разделённых двоеточием. Запись адреса IPv6 выглядит следующим образом: 2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d.

Адрес домена состоит из имён и точек. Такой адрес указывает на название страны и организации, которая предоставляет 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 присвоено двухбуквенное обозначение):

✓ «RU» — Россия, «РФ» — Российская Федерация, «UA» — Украина, «DE» — Германия;

▲ общие домены:

✓ «COM» — коммерческие организации, «NET» — предприятия, свя-

занные с развитием сетей и телекоммуникационных технологий, «ORG» — некоммерческие организации.

Адрес документа в Интернете (англ. *Uniform Resource Locator*, или URL) состоит из следующих частей:

▲ протокол передачи данных; может быть двух видов:

- http (для веб-страниц) или защищённый протокол https;
- ftp (для передачи файлов);

▲ символы-разделители «:» и «//», отделяющие название протокола от остального адреса;

▲ доменное имя сайта (или IP-адрес);

▲ каталоги на сервере, где располагается файл; они разделяются знаком «/»;

▲ имя файла.



http://informs.org.ru/main/katalog/zadania.html

1 2 3 4 5 6 7 8 9 10

Составляющие адреса:

1 — имя протокола сетевой службы; определяет тип сервера HTTP (протокол передачи гипертекста);

2 — разделители;

3 — полное доменное имя сервера;

4 — путь поиска веб-документа;

5 — имя веб-сервера;

6 — домен верхнего уровня «org»;

7 — имя национального домена «ru»;

8 — каталог main на компьютере;

9 — каталог katalog в каталоге main;

10 — конечная цель поиска — файл zadania.html.

■ Линии связи

Сеть Интернет состоит из компьютерных узлов и каналов связи. К узлам связи подключаются персональные компьютеры, а услуги абонентам предоставляют провайдеры.

Линия связи — приспособление, используемое для объединения в сеть отдельных компьютеров.

Все объекты в сети соединены каналами связи. В качестве каналов связи могут использоваться радиоволны, оптоволокно, электрические, а также телефонные кабели.

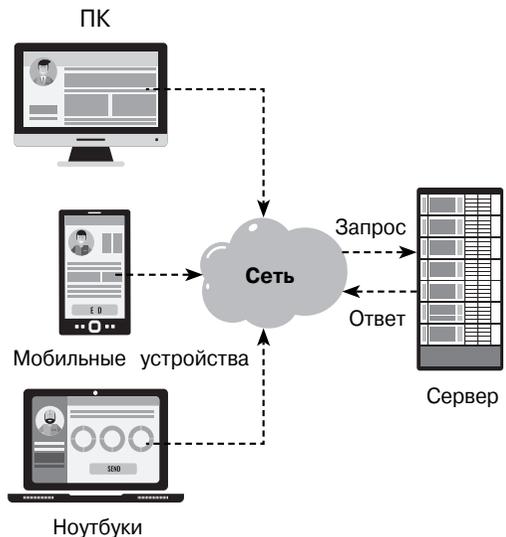
Сервер — компьютер, ресурсами которого пользуются другие компьютеры в сети.

Рабочая станция — компьютер, который пользуется сетью.

Сеть функционирует благодаря **сетевым операционным системам**,

отвечающим за приём и передачу информации по сети, и **сетевым приложениям**, к которым можно отнести браузер.

Браузер — компонента, необходимая для отображения страницы в Интернете.



Пример клиент-серверной архитектуры

Характеристики каналов связи

Пропускная способность, т. е. скорость обмена данными

Надёжность — способность канала связи передавать информацию без искажений

Стоимость

Возможность усовершенствования



Практические задания

4 Доступ к файлу **domashka.html**, находящемуся на сервере **obr.org**, осуществляется по протоколу **https**. Фрагменты адреса файла закодированы цифрами от 1 до 7. Запишите в ответе последовательность этих цифр, кодирующую адрес указанного файла в сети Интернет.

- 1) obr.
- 2) /
- 3) org
- 4) ://
- 5) html
- 6) domashka.
- 7) https

Решение:

Поскольку файл находится на сервере, в данном случае в сети Интернет, сначала определим протокол, зная, что в адресе он всегда находится на первом месте. Выбираем протокол под номером 7 — https.

После протокола всегда следует разделитель «://» (4). Получим:

https://

7 4

Далее всегда следует имя сервера, по условию задачи это obr.org. Отсюда:

https://obr.org

7 4 1 3

Файл в адресе всегда находится на последнем месте. По условию задания файл имеет имя domashka.html, ему соответствуют цифры 6 и 5, отсюда:

https://obr.org... domashka.html

7 4 1 3... 6 5

Между именем файла и названием сервера должен находиться символ «/» (2). Получим:

https://obr.org/domashka.html

7 4 1 3 2 6 5

Ответ: 7413265.

5 Доступ к файлу **zoo.bmp**, находящемуся на сервере **zoo32.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

- А) zoo
- Б) /
- В) ru
- Г) ://
- Д) http
- Е) .bmp
- Ж) zoo32.

Решение:

Поскольку файл находится на сервере, в данном случае в сети Интернет, сначала определим протокол. Выбираем название протокола под буквой Д: http.

После протокола всегда следует разделитель «://» (Г). Получим:

http://

Д Г

Далее всегда следует имя сервера, по условию задачи это zoo32.ru (пункты Ж и В). Отсюда имеем:

https://zoo32.ru

Д Г Ж В

Файл в адресе всегда находится на последнем месте. По заданию файл имеет имя zoo.bmp, ему соответствуют пункты А и Е:

https://zoo32.ru... zoo.bmp

Д Г Ж В... А Е

Между именем файла и названием сервера должен находиться символ «/» (Б). Получаем адрес:

https://zoo32.ru/zoo.bmp

Д Г Ж В Б А Е

Ответ: ДГЖВБАЕ.

6 Файл **cats.pptx** был выложен в Интернете по адресу **ftp://mycats.ru/cats.pptx**. Потом его переместили в каталог **dom** на сайте **presentation.edu**, доступ к которому осуществляется по протоколу **http**. Имя файла не изменилось.

Фрагменты нового и старого адресов файла закодированы цифрами от 1 до 9. Запишите последовательность этих цифр, кодирующую адрес файла в сети Интернет после перемещения.

- 1) http:/
- 2) mycats
- 3) cats
- 4) presentation
- 5) .edu
- 6) ftp:/
- 7) /
- 8) .pptx
- 9) dom

Решение:

Сначала указывается протокол (как правило, ftp или http), по условию это http, потом разделитель «:/», имя сервера, затем символ «/», далее указывается каталог и снова символ «/», название файла находится в конце адреса.

Пункт 1 http:/ не содержит второго символа «/», поэтому нужно добавить пункт 7 «/». Теперь записываем сайт, на который перенесли файл: presentation.edu (пункты 4 и 5), после этого названия используем символ «/», далее указываем каталог dom и имя файла cats.pptx.

Получим: http://presentation.edu/dom/cats.pptx

Ответ: 174579738.

7 На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Сотрудники обозначили эти фрагменты буквами А, Б, В и Г.

2.14	16	.69	8.151
А	Б	В	Г

Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

Решение:

IP-адрес представляет собой четыре разделённых точками числа, каждое из которых не больше 255.

____.____.____.____

Рассмотрим число под буквой А: после 2.14 нельзя поместить 16 (Б), но можно .69 (В) или 8.151 (Г); перед 2.14 можно расположить только 16. Получим два варианта: 162.14.69 или 162.148.151. Первый вариант не подходит, т. к. после 69 нельзя поставить 8, значит, единственно возможным является второй вариант: 162.148.151.69. Следовательно, правильный ответ: БАГВ.

Ответ: БАГВ.

Поисковые запросы

Для поисковых запросов можно использовать операции конъюнкции (&) или дизъюнкции (|).

▲ Операция И (&) в поисковом запросе всегда ограничивает поиск (уменьшает количество страниц в выдаче). Это значит, что в ответ на запрос «апельсин И мандарин» поисковый сервер выдаст меньше страниц, чем на запрос «апельсин», потому что будет искать страницы, на которых присутствуют оба указанных слова.

▲ Операция ИЛИ (|) в поисковом запросе всегда расширяет поиск (увеличивает количество страниц в выдаче). Это значит, что в от-

вет на запрос «апельсин ИЛИ мандарин» поисковик выдаст больше страниц, чем на запрос «мандарин», потому что будет искать страницы, на которых присутствует хотя бы одно из этих слов (или сразу оба слова).



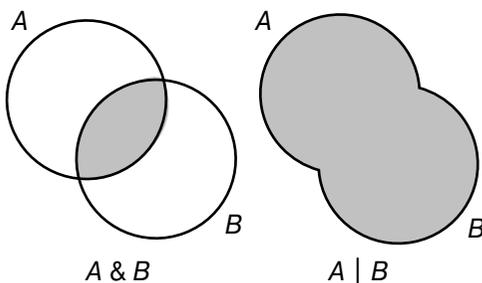
▲ Чем больше в запросе имеется операций И, тем меньше результатов в выдаче.

▲ Чем больше в запросе операций ИЛИ, тем больше результатов в выдаче.

Операция И представляется как пересечение двух или нескольких множеств.

Операция ИЛИ представляется как объединение двух или нескольких множеств.

Заштрихованная область при объединении больше, чем при пересечении.



Формула для определения количества итоговых запросов:

$$N_A + N_B = N_{A \& B} + N_{A | B},$$

где N_A — количество результатов для запроса A , N_B — количество результатов для запроса B , $N_{A \& B}$ — количество результатов для запроса $A \& B$, $N_{A | B}$ — количество результатов для запроса $A | B$.



Практические задания

8

Ниже приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу.

Для обозначения логической операции ИЛИ в запросе используется символ «|», а для обозначения логической операции И — символ «&».

- 1) Коты & Собаки & Мыши
- 2) Коты | Собаки
- 3) Коты | Мыши | Собаки
- 4) Коты & Собаки

Решение:

Рассмотрим количество операций в запросах:

- 1) 2 И
- 2) ИЛИ
- 3) 2 ИЛИ
- 4) И

С учётом того что операция И уменьшает количество результатов, а ИЛИ увеличивает, на первое место следует поставить запрос под номером 1 с наибольшим количеством операций И. Затем пойдёт запрос под номером 4, поскольку там имеется одна операция И. Остаются операции ИЛИ, которые увеличивают количество результатов:

сначала следует номер 2 с одной операцией ИЛИ, наибольшее количество страниц выдаст запрос под номером 3 с двумя ИЛИ.

Ответ: 1423.

9 В языке запросов поискового сервера для обозначения логической операции ИЛИ используется символ «|», а для обозначения логической операции И — символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (тыс.)
Крокодил Аллигатор	750
Крокодил	220
Крокодил & Аллигатор	80

Какое количество страниц (в тыс.) будет найдено по запросу «Аллигатор»?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Решение:

Способ 1. Используем формулу для определения количества итоговых запросов: $N_A + N_B = N_{A \& B} + N_{A|B}$.

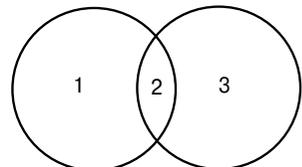
Подставим в формулу известные значения и найдём неизвестное: $220 + N_B = 750 + 80$, отсюда $N_B = 610$.

Способ 2. Представим таблицу в виде кругов Эйлера. Пусть Крокодил — круг 1, Аллигатор — круг 3. Тогда найдём количество элементов N в областях 2 и 3: $N_2 + N_3$. Из таблицы известно:

$$N_1 + N_2 + N_3 = 750,$$

$$N_1 + N_2 = 220,$$

$$N_2 = 80.$$



Теперь найдём N_3 : $N_3 = 750 - 220 = 530$.

Таким образом, по запросу «Аллигатор» будет найдено $N_2 + N_3 = 80 + 530 = 610$ тыс. страниц.

Ответ: 610.

10 В языке запросов поискового сервера для обозначения логической операции ИЛИ используется символ «|», а для обозначения логической операции И — символ «&».

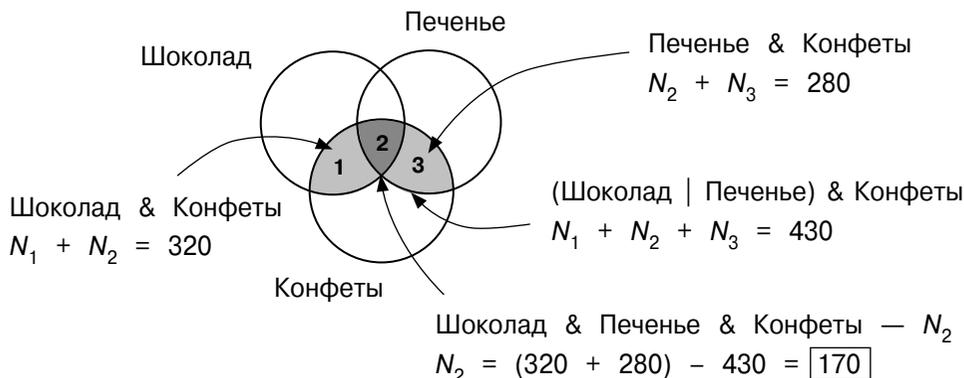
В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Ключевое слово	Количество сайтов, для которых данное слово является ключевым
Шоколад & Конфеты	320
Печенье & Конфеты	280
(Шоколад Печенье) & Конфеты	430

Сколько сайтов будет найдено по запросу «Шоколад & Печенье & Конфеты»?

Решение:

Используем круги Эйлера.



Ответ: 170.

11

В языке запросов поискового сервера для обозначения логической операции ИЛИ используется символ «|», а для обозначения логической операции И — символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Ключевое слово	Количество сайтов, для которых данное слово является ключевым
Альфа & Омега	260
Омега & (Альфа Гамма)	467
Альфа & Омега & Гамма	131

Сколько сайтов будет найдено по запросу «Омега & Гамма»?

Решение:

Заметим, что «& Омега» встречается в каждом запросе. Значит, эту составляющую можно сократить и воспользоваться формулой для определения количества итоговых запросов: $N_A + N_B = N_{A \& B} + N_{A|B}$.

Подставим в формулу известные значения и найдём неизвестное: $260 + N_B = 467 + 131$, отсюда $N_B = 338$.

Ответ: 338.

ИНСТРУМЕНТЫ СОЗДАНИЯ ИНФОРМАЦИОННЫХ ОБЪЕКТОВ ДЛЯ ИНТЕРНЕТА

Информационный объект — объект предметной области в информационной системе, определяющий её структуру, атрибуты, целостность данных и поведение.

Простые информационные объекты: звук, изображение, текст, число.

Комплексные (структурированные) информационные объекты: элемент, база данных, таблица, гипертекст, гипермедиа. Информационные объекты отображаются на веб-страницах.

Веб-страница — документ, созданный в формате html, который может

включать в себя видео, фото, текст, графику, гиперссылки.

Гиперссылка (или просто ссылка) — текст, фотография, иконка или другой элемент, при нажатии на который открывается другая веб-страница. Если несколько страниц имеют общий интерфейс, оформление, общие гиперссылки, а также если они имеют расположение на одном веб-сервере, то их можно назвать **веб-сайтом**.

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.

Веб-разработка — динамичный процесс создания сайтов, который позволяет вносить изменения в их содержимое. 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.

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.).

Знание этого языка становится всё более актуальным. Его основы необходимы для корректного и эстетичного оформления писем электронной почты, при создании группы в социальных сетях и т. д.

В настоящее время востребован язык HTML5, который не является продолжением языка разметки гипертекста, а представляет собой новую платформу, предназначенную для создания сайтов, веб-приложений, использующих аудио, видео, графику, анимацию и др. В HTML5 реализовано множество новых синтаксических особенностей, разработанных для упрощения создания и управления графическими и мультимедийными объектами в сети без необходимости использования сторонних плагинов. Для использования языка нужно знать и применять дескрипторы, которые называются также тегами. Именно с помощью тегов создаётся документ.

■ Основные компоненты и структура документа HTML

▲ **Тег (tag)**. 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.

Парный тег начинается с открывающего и заканчивается закрывающим тегом. Символ «/» в парных тегах означает его окончание.

✓ Тег <p>...</p> парный: <p> означает, что абзац начался, </p> — абзац закончился.

✓ Тег одиночный. У него нет парного ему закрывающего тега.

▀ **Атрибут (аргумент).** Изменяет тег.

✓ Выравнивание абзаца, изменение цвета шрифта и т. п.

▀ **Значение.** Значения присваиваются атрибутам и определяют вносимые изменения. Значения могут быть текстовыми, а также числовыми, как, например, ширина и высота изображения.

✓ <p style="font-family:verdana; color:red;">

Тег <p> означает начало абзаца, атрибут style присваивает значению тексту: красный цвет и шрифт Verdana.

HTML-документ имеет расширение .html. На сайте обязательно должна быть страница index.html — главная страница сайта. Далее приведена основная структура документа HTML.

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

▀ Элемент <!DOCTYPE> предназначен для указания типа текущего документа.

▀ <html> показывает начало и конец кода.

▀ <head> описывает то, что находится в документе, и как это следует выводить.

▀ <title> показывает начало и конец основного заголовка.

▀ <body> содержит весь контент, отображаемый на веб-странице в браузере.

Примеры тегов

▀ от <h1></h1> до <h6></h6> — шесть уровней заголовков;

▀ <p></p> — абзац;

▀
 — переход на другую строку;

- ▶ `<hr>` — горизонтальная линия;
- ▶ `` и его атрибуты `face`, `size`, `color` — гарнитура, размер, цвет шрифта;
- ▶ `` — нумерованный список;
- ▶ `` — элемент списка;
- ▶ `` — маркированный список, атрибут `TYPE` задаёт вид маркера: `disc` — диск, `square` — квадрат, `circle` — окружность;
- ▶ `` — вставка изображения;
- ▶ `<video> </video>` — вставка видео;
- ▶ `<link>` — установка связи текущего документа с внешним ресурсом.

Атрибуты тегов

- ▶ `src` — исходный URL-адрес файла изображения;
- ▶ `align` — способ выравнивания изображения;
- ▶ `border` — видимая рамка вокруг изображения;
- ▶ `width` — ширина изображения;
- ▶ `height` — высота изображения;
- ▶ `loop` — число повторений видео;
- ▶ `start` — условие запуска видео.

■ Таблицы стилей

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

По методам добавления стилей в документ различают внутренние, глобальные и внешние стили.

Внутренние стили определяются атрибутом `style` конкретных тегов, он действует только на элементы, определённые этими тегами.

```
✓ <p style="color:green">
  Абзац с текстом зеленого
  цвета</p>
```

Глобальные стили CSS размещаются в теге `<style>...</style>`, расположенном, в свою очередь, в `<head>...</head>`.

```
✓ <style type="text/css">
    p {color:#808080;}
</style>
```

В приведённом выше примере результатом будет серый цвет текста во всех абзацах.

Внешние стили определяются в отдельном файле с расширением `css`. Внешние стили позволяют всем страницам сайта выглядеть единообразно.

Для подключения требуемой таблицы стилей используется элемент `<link>` в блоке `<head>` документа HTML5.

```
✓ <head>
    link...
....
</head>
```

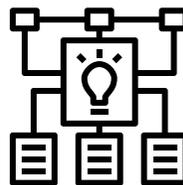
```
✓ <link href="style.css"
  rel="stylesheet">
```

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ



АЛГОРИТМЫ И АЛГОРИТМИЗАЦИЯ

Ежедневно, сами того не замечая, мы сталкиваемся с алгоритмами, которые помогают нам эффективно и качественно решать различные задачи. Понятие алгоритмизации неразрывно связано с программированием, т. к. знание теории алгоритмов применяется всеми программистами для решения практических задач.



ИСТОРИЧЕСКАЯ СПРАВКА

Информатика тесно связана с математикой. Можно сказать, что с алгоритмами математики люди были знакомы уже несколько тысячелетий назад. Например, алгоритм Евклида (нахождение наибольшего общего делителя двух целых чисел) открыт более 2000 лет назад. Слово «алгоритм» произошло от имени великого персидского учёного Мухаммада аль-Хорезми, описавшего четыре арифметических действия в начале IX в. При переводе его трудов на латинский язык имя учёного перевели как *Algorithmi*. Считается, что так

и возник термин «алгоритм». Постепенно значение слова расширилось, и математики стали применять его не только к вычислительным, но и к другим операциям.

Многие философы и математики внедряли данное понятие в свои труды, например философ Н. Орем применил его в математическом трактате «Вычисление пропорций» (*Algorismus proportionum*) около 1360 г. В 1684 г. Г. Лейбниц в сочинении впервые использовал слово «алгоритм» (*Algorithmo*) в более ши-

роком смысле: как систематический способ решения проблем дифференциального исчисления. Впоследствии понятие алгоритма использовал выдающийся математик XVIII в. Л. Эйлер, его работа называлась «Использование нового алгоритма для решения проблемы Пелля».

Математики решали задачи, используя интуитивное понимание алгоритма, сама же теория алгоритмов появилась только в начале 30-х гг. XX в. Понятие «алгоритм» было определено в работах Д. Гильберта, А. Чёрча, Э. Поста, А. Тьюринга, А. Маркова. Они представили разные модели алгоритмов, которые оказались эквивалентными в том смысле, что проблемы любого класса, разрешимые в одной модели, разрешимы и в другой.

■ Машина Тьюринга

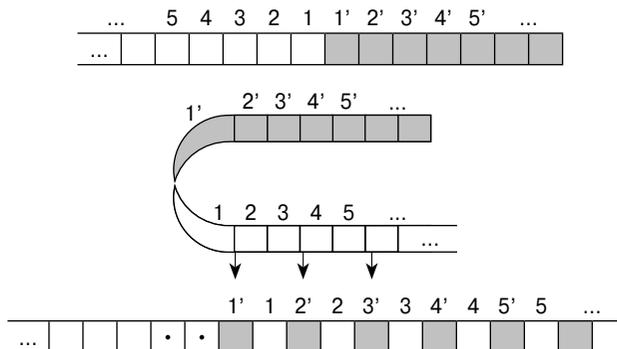
Машина Тьюринга — модель универсального исполнителя алгоритмов обработки символьных последова-

тельств, состоящая из центра управления и ряда ячеек. Управляющее устройство машины имеет возможность двигаться по ряду ячеек и считывать все значения, которые им присвоены.

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 буква R — передвинуть впра-

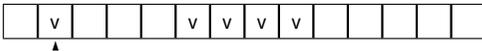


Схематичное изображение машины Тьюринга, работающей на полубесконечной ленте

во, буква N в команде оставляет устройство на месте.

■ 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

■ 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 описывается 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.

КОМАНДЫ И 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
$n v m$	Запись метки в текущую пустую клетку и переход к выполнению команды с номером m
$n \updownarrow m$	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

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.

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)

✓ Пусть $A = \{b, c\}$ — алфавит. Рассмотрим следующую схему нормального алгоритма в A :

$$\begin{cases} c \rightarrow \mu \\ b \rightarrow b \end{cases}$$

Преобразуйте строки $scbcb$, cb , cc , $bbc b$, $bc b c$, используя данный алгоритм.

Решение:

Всякое слово V в алфавите A , содержащее хотя бы одно вхождение буквы c , алгоритм преобразовывает в слово, получающееся из V вычёркиванием в нём самого левого (первого) вхождения буквы c . Пустое слово он перерабатывает в пустое. Получим:

$$\begin{aligned} scbcb &\rightarrow cbcb, & cb &\rightarrow b, & cc &\rightarrow c, \\ bbc b &\rightarrow bbb, & bc b c &\rightarrow bbc. \end{aligned}$$

ОСНОВНЫЕ ПОНЯТИЯ

Алгоритмизация — метод описания систем или процессов путём составления алгоритмов их функционирования.

Алгоритмом называется точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату.

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.

Рассмотрим алгоритм на примере задачи.

✓ Даша решила вскипятить чайник. Опишите, какие действия она должна выполнить.

- 1) Подойти к плите.
- 2) Посмотреть, сколько в чайнике воды.
- 3) Если воды мало или она отсутствует, то налить воду и перейти

к пункту 4, если много, то перейти к пункту 4.

- 4) Включить плиту.
- 5) Поставить чайник на нужную конфорку.
- 6) Подождать, пока вода закипит.
- 7) Выключить чайник.

В данном случае Даша — исполнитель, который выполняет определённую последовательность команд. Эту последовательность можно визуализировать, т. е. представить в графическом виде с помощью блок-схемы.

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

СВОЙСТВА И СПОСОБЫ ПРЕДСТАВЛЕНИЯ АЛГОРИТМОВ

▲ **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 алгоритма

Словесно-формульный

Графический

Операторный

Словесно-формульный способ — представление алгоритма в виде текстовых записей и (или) формулы.

Графический способ описания алгоритма — представление алгоритма в виде графической схемы, рисунка. В программировании для написания определённой программы часто используют алгоритмы, пред-

ставленные блок-схемой, и после используют операторный способ.

Рассмотрим примеры словесно-формульного и графического алгоритмов.

✓ Требуется решить линейное уравнение $ax + b = 0$, $a \neq 0$ (без этого условия уравнение не будет линейным). Математической моделью решения этой задачи является формула $x = \frac{-b}{a}$. 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 $x = c$ ».

Правила построения алгоритмов на языке блок-схем

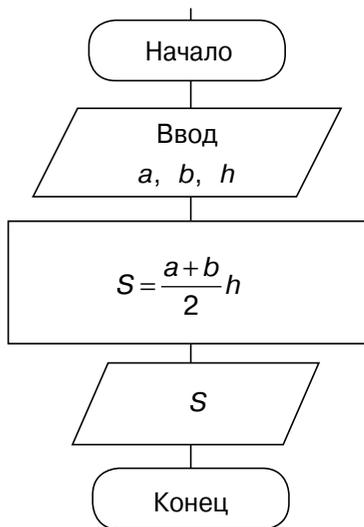
1. Блок-схема строится сверху вниз

2. В любой блок-схеме имеется один элемент, соответствующий началу, и один элемент, соответствующий концу

3. Должен существовать хотя бы один путь из начала блок-схемы к любому элементу

4. Должен существовать хотя бы один путь от каждого элемента блок-схемы в её конец

✓ Пример графического способа записи алгоритма для вычисления площади S трапеции, где a , b — основания трапеции, h — высота трапеции.



Операторный способ — описание алгоритма на языке программирования (алгоритмическом языке).

✓ Пример описания на алгоритмическом языке.

```

алг
нач
вывод ("Привет ученик!")
кон
  
```

Данный алгоритм выводит на экран текст «Привет ученик!».

В дальнейшем мы подробнее рассмотрим построение алгоритмов для решения задач с использованием различных языков программирования.

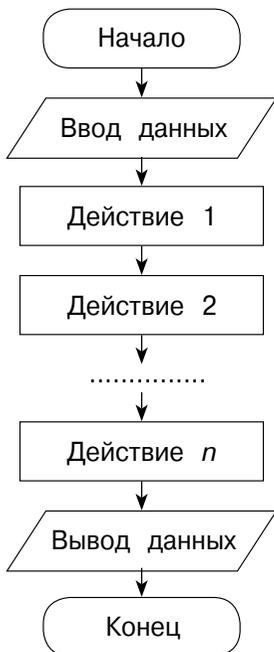
ВИДЫ АЛГОРИТМОВ

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 четыре основных вида алгоритмов: **линейный** (шаги выполняются последовательно друг за другом), **разветвляющийся** (порядок выполнения шагов изменяется в зависимости от условия), **циклический** (определённая после-

довательность шагов повторяется несколько раз) и **рекурсивный**.

■ 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 графической, так и в текстовой форме.



Так выглядит общая блок-схема линейного алгоритма. На с. 182 приведена блок-схема линейного алгоритма для нахождения площади трапеции.

Линейный алгоритм, записанный в текстовой форме, можно использовать при решении задач.

✓ У исполнителя Зубчика две команды, которым присвоены номера:
 1 — умножь на 3;
 2 — вычти b (b — неизвестное натуральное число).

Первая команда увеличивает число на экране в 3 раза, вторая уменьшает его на b . Известно, что программа 21122 переводит число 6 в число 21.

Определите значение b .

Решение:

Составим и решим уравнение:

$$(6 - b) \cdot 3 \cdot 3 - 2b = 21,$$

$$54 - 9b - 2b = 21,$$

$$11b = 54 - 21,$$

$$b = 3.$$

Ответ: 3.

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



Практические задания

1 У исполнителя Нолика две команды, которым присвоены номера:

1 — прибавь 1;

2 — умножь на c (c — неизвестное натуральное число; $c \geq 2$).

Выполняя первую из них, Нолик увеличивает число на экране на 1, а выполняя вторую, умножает это число на c . Программа для исполнителя Нолика — это последовательность номеров команд. Известно, что программа 11211 переводит число 3 в число 62. Определите значение c .

Решение:

Имеем программу: 11211. Значит, выполняем следующие действия (алгоритм):

- 1) 1 — прибавляем 1 к изначальному числу;
- 2) 1 — прибавляем 1 к полученному в шаге 1 числу;
- 3) 2 — умножаем на c полученное в шаге 2 число;
- 4) 1 — прибавляем 1 к полученному в шаге 3 числу;
- 5) 1 — прибавляем 1 к полученному в шаге 4 числу.

Изначально дано число 3, значит, после выполнения первых двух команд $3 + 1 + 1$ получим число 5.

Далее 5 умножаем на c и снова выполняем команды с прибавлением: $5c + 1 + 1$.

Составим и решим уравнение:

$$5c + 2 = 62,$$

$$5c = 60,$$

$$c = 12.$$

Ответ: 12.

2 Исполнитель Крестик перемещается на координатной плоскости, оставляя след в виде линии. Крестик может выполнять команду **Сместиться на** (a, b) (где a, b — целые числа),

перемещающую Крестика из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Если числа a, b положительные, значение соответствующей координаты увеличивается, если отрицательные — уменьшается.

Крестику был дан для исполнения следующий алгоритм:

«Повтори 3 раза

Сместиться на (1, 1) Сместиться на (2, 2) Сместиться на (1, -3)

Конец»

Какую команду надо выполнить Крестику, чтобы вернуться в исходную точку, из которой он начал движение? В ответе запишите две координаты через запятую.

Решение:

Команда **Повтори 3 раза** означает умножение последующих параметров на 3.

Предположим, что Крестик начал движение с начала координатной плоскости ($x = 0, y = 0$), хотя он мог начать с любой и на ответ это не влияет. Исходя из данного предположения, рассчитаем его перемещение по осям ox и oy .

По оси ox :

$3 \cdot (1 + 2 + 1) = 12$ (начало с 0) \Rightarrow по оси ox он попадает в точку 12.

По оси oy :

$3 \cdot (1 + 2 - 3) = 0$ (начало с 0) \Rightarrow по оси $oy = 0$.

Этот алгоритм можно заменить на команду **Сместиться на (12, 0)**.

Поскольку по заданию Крестику нужно вернуться в исходную точку, ему необходимо выполнить команду, обратную данной, т. е. команду **Сместиться на (-12, 0)**.

Ответ: -12, 0.

3 Исполнитель Крестик перемещается на координатной плоскости, оставляя след в виде линии. Крестик может выполнять команду **Сместиться на (a, b)** (где a, b — целые числа), перемещающую Крестика из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Если числа a, b

положительные, значение соответствующей координаты увеличивается, если отрицательные — уменьшается.

Крестику был дан для исполнения следующий алгоритм:

«Повтори 4 раза

Команда1 Сместиться на (3, 2) Сместиться на (2, 1)

Конец

Сместиться на (-12, -8)»

После выполнения этого алгоритма Крестик вернулся в исходную точку. Какую команду надо поставить вместо команды *Команда1*? В ответе запишите только координаты через запятую.

Решение:

Команда **Повтори 4 раза** означает умножение последующих параметров на 4 (до команды **Конец**). Предположим, что Крестик вышел из координаты (0, 0).

Найдём неизвестные координаты, исходя из алгоритма. Вместо неизвестной команды подставим x и y :

$$4 \cdot (x + 3 + 2) + (-12) = 0 \text{ (т. к. он вернулся в исходную позицию),}$$

$$4x + 12 + 8 - 12 = 0,$$

$$4x = -8,$$

$$x = -2.$$

Найдём вторую координату y :

$$4 \cdot (y + 2 + 1) + (-8) = 4y + 8 + 4 - 8,$$

$$4y = -4,$$

$$y = -1.$$

Получили значения -2 и -1.

Ответ: -2, -1.

4 У исполнителя Винтика три команды, которым присвоены номера:

1 — вычти 5;

2 — возведи в квадрат;

3 — прибавь 2.

Первая из них уменьшает число на экране на 5, вторая возводит его во вторую степень, третья прибавляет 2. Исполнитель работает только с натуральными числами.

Составьте алгоритм получения из числа 4 числа 49, содержащий не более 6 команд. В ответе запишите только номера команд.

Например, 21311 — это алгоритм: возведи в квадрат, вычти 5, прибавь 2, вычти 5, вычти 5, который преобразует число 6 в 23. Если таких алгоритмов более одного, то запишите любой из них.

Решение:

Для решения задачи необходимо найти любой алгоритм. Заметим, что 49 — это квадрат числа 7, поэтому из числа 4 нам нужно получить число 7, используя команды 1 и 3.

$$4 + 2 - 5 + 2 + 2 + 2 = 7$$

Далее командой 2 возводим число 7 в квадрат и получаем 49.

Запишем алгоритм: 313332.

Ответ: 313332.

5 Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) заменить (v, w)

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w .

Б) нашлось (v)

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактора. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка при этом не изменяется.

Какая строка получится в результате применения приведённой ниже программы к строке, в которой 71 раз подряд повторяется группа цифр 1133? В ответе запишите полученную строку.

```

НАЧАЛО
ПОКА нашлось (13) ИЛИ нашлось (333)
  ЕСЛИ нашлось (13)
    ТО заменить (13, 3)
  ИНАЧЕ заменить (333, 3)
КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ

```

Решение:

Рассмотрим одну группу цифр: 1133, произведём операции, заданные в условии задачи.

Условие цикла истинно (ПОКА **нашлось** (13)), и условие в цикле истинно (ЕСЛИ **нашлось** (13)), значит, делаем замену: в первой группе цифр 1133 заменяем 13 на 3:

1133 => 133.

Аналогично после второй итерации цикла получаем:

133 => 33.

По условию имеем 71 группу цифр, значит, после двух итераций они превратятся в 142 цифры 3.

Теперь используем условие ИНАЧЕ **заменить** (333, 3). Из 142 цифр каждые три заменяем на одну. Производим цикл снова до тех пор, пока не сможем производить дальнейшие операции.

$142 : 3 = 47 \text{ троек} + 1 \text{ цифра } 3 \text{ (остаток)} = 48 \text{ цифр } 3$

$48 : 3 = 16 \text{ троек}$

$16 : 3 = 5 \text{ троек} + 1 \text{ цифра } 3 \text{ (остаток)} = 6 \text{ цифр } 3$

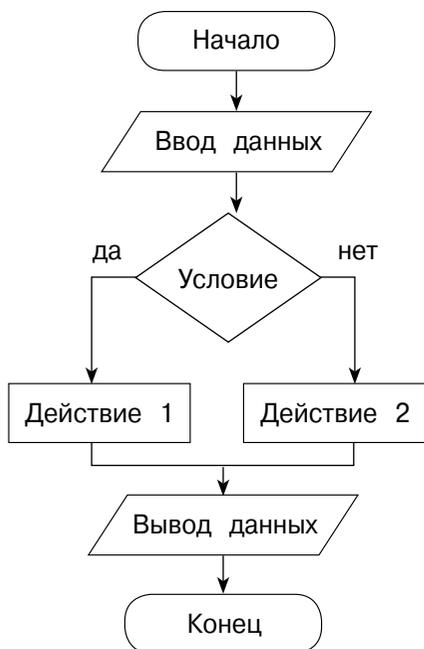
$6 : 3 = 2 \text{ тройки}$, цикл завершён, операций больше произвести невозможно.

Ответ: 33.

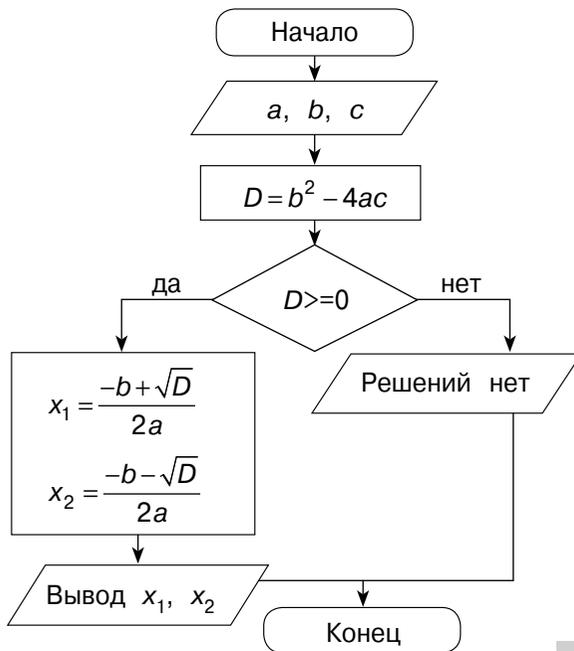
■ Разветвляющийся алгоритм

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>.



✓ Пример разветвляющегося алгоритма в виде блок-схемы для решения квадратного уравнения $ax^2 + bx + c = 0$.



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

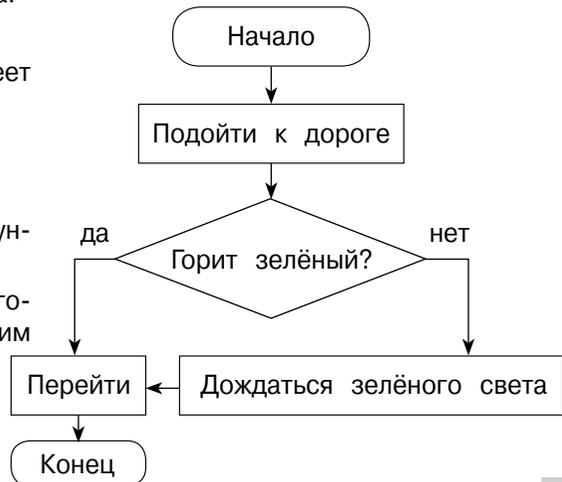
✓ Составьте алгоритм для перехода дороги при наличии светофора.

Соответствующая блок-схема.

Решение:

Алгоритм в текстовом виде имеет следующий вид.

- 1) Подходим к светофору.
- 2) Смотрим, какой горит свет.
- 3) Если зелёный, переходим к пункту 5.
- 4) Если красный, ждём, пока загорится зелёный, а потом переходим к пункту 5.
- 5) Переходим дорогу.



Практические задания

6 На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .
2. К этой записи дописываются справа ещё два разряда по следующему правилу: складываются все цифры двоичной записи; если
 - а) сумма нечётная, к числу дописывается 11,
 - б) сумма чётная, к числу дописывается 00.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R . Укажите такое наибольшее число M , которое не превышает 135 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе счисления.

Решение:

Переведём число 135_{10} в двоичную систему: $135_{10} = 10001111_2$. Убираем два последних символа и получаем число N в двоичной системе счисления, которое подаётся на вход, — 10001_2 . Следуя алгоритму,

нужно проверить чётность единиц. Количество единиц чётное, значит, нужно добавить два нуля: 10000100_2 . Переводим из двоичной системы счисления в десятичную по алгоритму: $128 + 4 = 132$. 132 не превышает 135 и является наибольшим из возможных результатов.

Ответ: 132.

7 Автомат получает на вход трёхзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также вторая и третья цифры исходного числа.

2. Полученные два числа записываются друг за другом в порядке возрастания (без разделителей).

Пример. Исходное число: 456. Суммы: $4 + 5 = 9$; $5 + 6 = 11$. Результат: 911.

Укажите наибольшее число, в результате обработки которого автомат выдаст число 1216.

Решение:

Сумма двух цифр всегда меньше или равна 18. Разобьём число 1216 на два числа: 12 и 16. Чтобы число было наибольшим, необходимо, чтобы первый разряд был 9.

Возможны варианты разложения чисел: $12 = 9 + 3$, тогда $16 = 3 + 13$, что нам не подходит, т. к. 13 не является цифрой, или $16 = 9 + 7$, тогда $12 = 7 + 5$. Составляем число: 975.

Ответ: 975.

■ Циклический алгоритм

Циклический алгоритм — описание действий, которые должны повторяться указанное число раз или до тех пор, пока не будет выполнено заданное условие. В представленном на с. 192 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

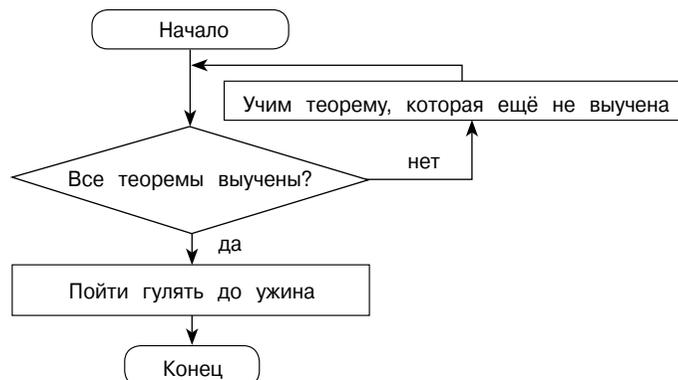
4. Произвести необходимые действия, записать математические формулы

5. Изменить переменную на величину шага

6. Проверить условие окончания (или повторения) цикла

7. Перейти к пункту 5, если цикл не закончен

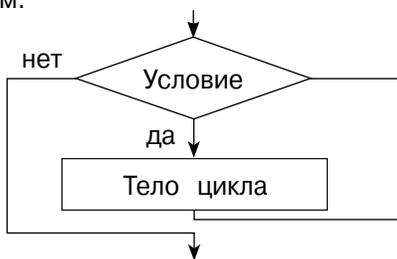
✓ Пример циклического алгоритма, представленного блок-схемой.



Существует классификация циклических алгоритмов.

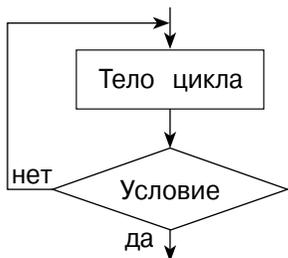
▲ **Цикл с предусловием (цикл ПОКА).** Цикл, который выполняется, пока истинно некоторое условие, указанное до его начала. Это условие проверяется до выполнения тела цикла, поэтому, если условие ложно, тело цикла может быть не выполнено ни разу.

✓ Блок-схема цикла с предусловием.

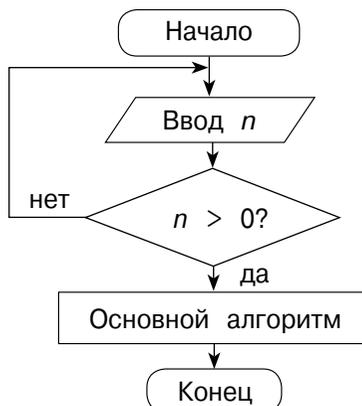


▲ **Цикл с постусловием (цикл ДО).** Цикл, в котором условие проверяется после выполнения тела цикла. Отсюда следует, что тело всегда выполняется хотя бы один раз. Этот цикл предписывает повторять команды тела цикла до тех пор, пока не выполнится условие в его заголовке (т. е. пока оно остаётся ложным).

✓ Блок-схема цикла с постусловием.

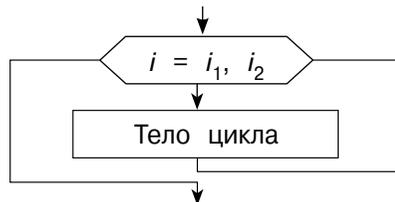


✓ Введите целое положительное число (n), определите, чётное ли оно.



▲ **Цикл с параметром (цикл ДЛЯ).** Имеет также название «цикл со счётчиком» — цикл, в котором переменная изменяет значение от заданного начального до конечного с некоторым шагом, и для каждого значения этой переменной тело цикла выполняется один раз.

✓ Блок-схема цикла с параметром.



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.

✓ Подсчитайте сумму чисел от 1 до N .

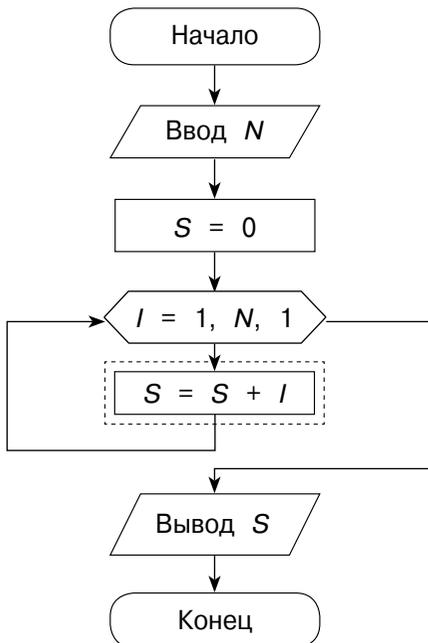
Решение:

Имеется переменная I , которая меняется от 1 до N включительно. Начальное значение $I = 1$, конечное значение $I = N$, которое мы задаём в начале программы.

S — итоговое значение, которое меняется каждый раз при изменении переменной. Тип — целочисленный.

Параметр цикла — переменная I , значение которой изменяется от 1 до N с шагом 1. При каждом прохождении цикла к значению S прибавляется значение I .

Опишем это с помощью циклического алгоритма.



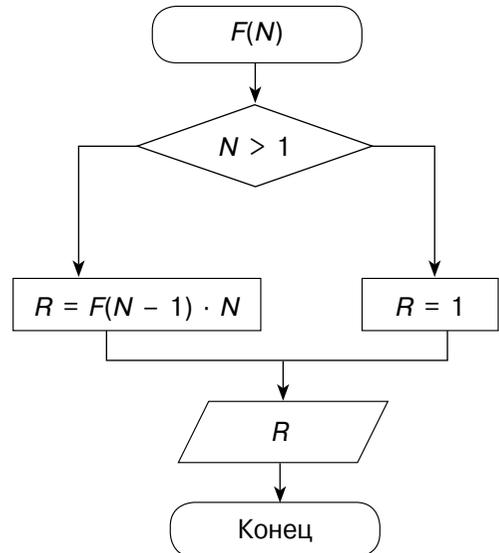
■ Рекурсивный алгоритм

Рекурсия — определение объекта через обращение к самому себе.

Алгоритм, в котором прямо или косвенно содержится ссылка на него же как на вспомогательный алгоритм, называется **рекурсивным**.

Рекурсивный метод в программировании предполагает разработку решения задачи, основываясь на свойствах рекурсивности отдельных объектов или закономерностей. При этом исходная задача сводится к решению аналогичных подзадач, которые являются более простыми.

✓ Пример рекурсивного алгоритма, представленного блок-схемой.



✓ Последовательность чисел задаётся рекуррентным соотношением:

$$F(1) = 2,$$

$$F(2) = 1,$$

$F(n) = F(n - 2) + F(n - 1)$ при $n > 2$, где n — натуральное число.

Чему равно 9-е число в последовательности Люка?

В ответе запишите только натуральное число.

Решение:

Для определения 9-го числа в последовательности найдём все предыдущие значения с 3-го по 8-е.

$$F(3) = F(1) + F(2) = 3,$$

$$F(4) = F(2) + F(3) = 4,$$

$$F(5) = F(3) + F(4) = 7,$$

$$F(6) = F(4) + F(5) = 11,$$

$$F(7) = F(5) + F(6) = 18,$$

$$F(8) = F(6) + F(7) = 29,$$

$$F(9) = F(7) + F(8) = 47.$$

Девятое число в последовательности Люка равно 47.

Ответ: 47.



Практические задания

8 Алгоритм вычисления значений функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(n) = 1 \text{ при } n \leq 2;$$

$$F(n) = 2 \cdot F(n - 1) + F(n - 2) \text{ при } n > 2.$$

Чему равно значение функции $F(7)$? В ответе запишите только натуральное число.

Решение:

Последовательно находим:

$$F(1) = 1,$$

$$F(2) = 1,$$

$$F(3) = 2 + 1 = 3,$$

$$F(4) = 6 + 1 = 7,$$

$$F(5) = 14 + 3 = 17,$$

$$F(6) = 34 + 7 = 41,$$

$$F(7) = 82 + 17 = 99.$$

Ответ: 99.

9 Алгоритм вычисления значений функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(1) = 1, F(2) = 2, F(3) = 3,$$

$$F(n) = F(n - 3) \cdot n \text{ при } n > 3.$$

Чему равно значение функции $F(7)$? В ответе запишите только натуральное число.

Решение:

Исходя из условия алгоритма, находим значения F при $n > 3$:

$$F(4) = 4,$$

$$F(5) = 10,$$

$$F(6) = 18,$$

$$F(7) = 28.$$

Ответ: 28.

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 для каждого игрока

3. Указание условий выигрыша

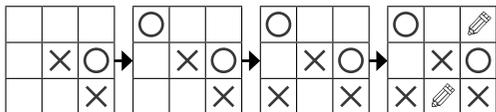
Рассмотрим самую простую игру — крестики-нолики.

Начальные условия игры: имеется поле 3×3 и две фигуры — крестик и нолик, у каждого игрока своя фигура.

Правила: игрок с фигурой крестик начинает игру; игроки по очереди ставят свои фигуры на поле в любую из девяти клеток.

Условия выигрыша: если один из игроков составит позицию из трёх своих фигур по горизонтали, вертикали или диагонали, то он выигрывает.

Например, первый игрок ставит крестик в середину поля, а второй ставит нолик не в угловую клетку, в этом случае игроком-крестиком достигается победа.



Выигрышная стратегия крестика при неверном ходе соперника

Если же после хода крестика в центральную клетку второй игрок первым ходом поставит нолик в угловую клетку, то будет ничья. При безошибочной игре обоих участников в результате всегда получается ничья. Другие примеры игр: шашки, шахматы, нарды,

«5 в линию», го (игра из Китая), «Монополия» и др.

Для победы в любой из этих игр необходимо составить выигрышный алгоритм.

Методы нахождения выигрышной стратегии:

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

В задачах подобного типа запись (6, 22) обозначает, что в одной совокупности находится 6 объектов, а в другой — 22.

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, кто имеет выигрышную стратегию, недостаточно

5. Привести доказательства, что указанная стратегия приводит к выигрышу. Просто назвать стратегию выигрышной недостаточно



Практические задания

10 Два игрока, Паша и Даша, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в два раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 30 камней. У каждого игрока, чтобы делать ходы, имеется неограниченное количество камней.

Игра завершается в тот момент, когда камней в куче становится не менее 20. Если при этом в куче оказалось не более 30 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его соперник. Например, если в куче было 17 камней и Паша удвоит это количество, то игра закончится и победителем будет Даша, поскольку число камней превысит 30. В начальный момент в куче было S камней, $1 \leq S \leq 19$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания.

1. а) При каких значениях числа S Паша может выиграть в один ход? Укажите все такие значения и соответствующие ходы Паши.
- б) У кого из игроков есть выигрышная стратегия при $S = 18, 17, 16$? Опишите выигрышные стратегии для этих случаев.
2. У кого из игроков есть выигрышная стратегия при $S = 9, 8$? Опишите соответствующие выигрышные стратегии.
3. У кого из игроков есть выигрышная стратегия при $S = 7$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах — количество камней в позиции.

Решение:

1. а) Паша может выиграть, если $S = 19$ или $S = 10, 11, 12, 13, 14, 15$. При $S = 19$ первым ходом нужно добавить в кучу один камень, удвоить камни Паша не может, т. к. $2 \times 19 = 38$, он проиграет, поскольку $30 < 38$. При остальных указанных значениях S нужно удвоить количество камней, в этом случае мы получим число больше 20, что означает победу игрока.

б) При $S = 16, 17$ или 18 удваивать мы не можем, т. к. получим число больше 30, что означает поражение. Единственный возможный ход — добавление в кучу одного камня.

При $S = 18$ после первого хода Паши (прибавление одного камня) в куче станет 19 камней. В этой позиции ходящий (Даша) выигрывает (см. п. 1а): при $S = 18$ Паша (игрок, который должен ходить первым) проигрывает. Поэтому выигрышная стратегия есть у Даши.

При $S = 17$, после того как Паша своим первым ходом добавит один камень, в куче станет 18 камней. В этой позиции ходящий (Даша) проигрывает (см. выше): при $S = 17$ Паша (игрок, который должен ходить первым) выигрывает. Выигрышная стратегия есть у Паши.

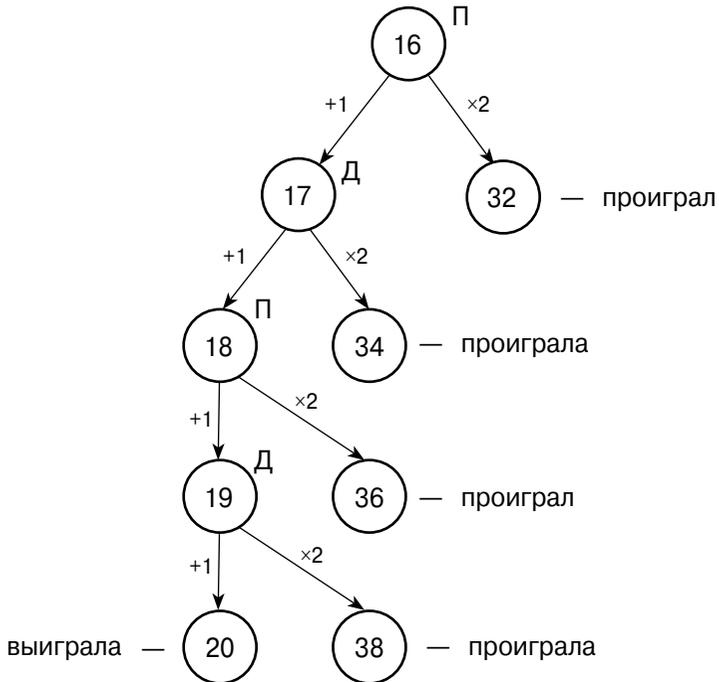
Решение можно представить в виде таблицы.

Ход	Вариант 1-го хода	Результат
Ход 1, Паша	$17 + 1 = 18$	нет результата
Ход 1, Даша	$18 + 1 = 19$	нет результата
Ход 2, Паша	$19 + 1 = 20$	победил

Ход	Вариант 2-го хода	Результат
Ход 1, Паша	$17 \cdot 2 = 34$	проиграл
Ход 1, Даша	$18 \cdot 2 = 36$	проиграла
Ход 2, Паша	$19 \cdot 2 = 38$	проиграл

При $S = 16$ выигрышная стратегия есть у Даши. Действительно, если Паша первым ходом удваивает количество камней, то в куче становится 32 камня, и игра сразу заканчивается выигрышем Даши. Если Паша добавляет один камень, то в куче становится 17 камней. Как мы уже знаем, в этой позиции игрок, который должен ходить (т. е. Даша), выигрывает.

Объяснение можно записать в виде дерева.



2. При $S = 9$ или 8 выигрышная стратегия есть у Паши. Она состоит в том, чтобы удвоить количество камней в куче и получить кучу, в которой будет соответственно 18 или 16 камней. В обоих случаях игрок, который будет делать ход (теперь это Даша), проигрывает (см. п. 1б).

3. При $S = 7$ выигрышная стратегия есть у Даши. После первого хода Паши в куче может стать либо 8 , либо 14 камней. В обеих этих позициях выигрывает игрок, который будет делать ход (теперь это Даша).
Случай $S = 8$ рассмотрен в п. 2, случай $S = 14$ рассмотрен в п. 1а.

Ответ:

1. а) Паша может выиграть, если $S = 19$ или $S = 10, 11, 12, 13, 14, 15$. При $S = 19$ первым ходом нужно добавить в кучу один камень, при остальных указанных значениях S нужно удвоить количество камней.

б) При $S = 18$ после такого хода Паши в куче станет 19 камней. В этой позиции ходящий (т. е. Даша) выигрывает (см. п. 1а): при $S = 18$ Паша (игрок, который должен ходить первым) проигрывает. Выигрышная стратегия есть у Даши.

При $S = 17$, после того как Паша своим первым ходом добавит один камень, в куче станет 18 камней. В этой позиции ходящий (т. е. Даша) проигрывает (см. выше): при $S = 17$ Паша (игрок, который должен ходить первым) выигрывает. Выигрышная стратегия есть у Паши.

При $S = 16$ выигрышная стратегия есть у Даши. Действительно, если Паша первым ходом удваивает количество камней, то в куче становится 32 камня, и игра сразу заканчивается выигрышем Даши. Если Паша добавляет один камень, то в куче становится 17 камней. Как мы уже знаем, в этой позиции игрок, который должен ходить (т. е. Даша), выигрывает.

2. При $S = 9$ или 8 выигрышная стратегия есть у Паши. Она состоит в том, чтобы удвоить количество камней в куче и получить кучу, в которой будет соответственно 18 или 16 камней. В обоих случаях игрок, который будет делать ход (теперь это Даша), проигрывает (см. п. 1б).

3. При $S = 7$ выигрышная стратегия есть у Даши. После первого хода Паши в куче может стать либо 8, либо 14 камней. В обеих этих позициях выигрывает игрок, который будет делать ход (теперь это Даша). Случай $S = 8$ рассмотрен в п. 2, случай $S = 14$ рассмотрен в п. 1а.

11

Два игрока, Саша и Костя, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Саша. За один ход игрок может:

- добавить в любую кучу один камень;
- увеличить количество камней в любой куче в два раза.

Игра завершается в тот момент, когда суммарное количество камней в двух кучах становится не менее 63. Победителем считается игрок, сделавший последний ход, т. е. первым получивший такую позицию, при которой в кучах будет 63 камня и более. В начальный момент в первой куче было 5 камней, во второй — S камней, $1 \leq S \leq 57$.

Выполните следующие задания.

1. а) При каких значениях числа S Саша может выиграть в один ход? Укажите все такие значения и соответствующие ходы Саши.

б) Известно, что Костя выиграл своим первым ходом после неудачного первого хода Саши. Укажите минимальное значение S , когда такая ситуация возможна.

2. Укажите такое значение S , при котором у Саши есть выигрышная стратегия, причём одновременно выполняются два условия:

- Саша не может выиграть за один ход;
- Саша может выиграть своим вторым ходом независимо от того, как будет ходить Костя.

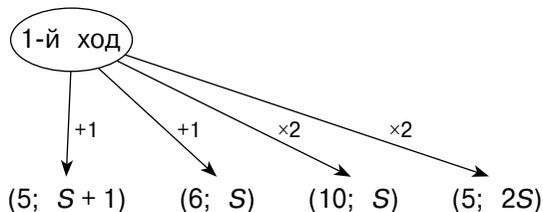
3. Укажите значение S , при котором одновременно выполняются два условия:

- у Кости есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Саши;
- у Кости нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Кости.

Решение:

1. Существует четыре варианта развития для первого хода: или прибавляем 1 к любой куче, или умножаем на 2 количество камней в любой куче.



а) Саша может выиграть в один ход, если:

- 1) $5 + S + 1 \geq 63$, $S \geq 57$
- 2) $6 + S \geq 63$, $S \geq 57$
- 3) $10 + S \geq 63$, $S \geq 53$
- 4) $5 + 2S \geq 63$, $2S \geq 58$, $S \geq 29$

$S \geq 29$, но по условию $S \leq 57$

Отсюда $29 \leq S \leq 57$.

Значит, при $S = 29, 30 \dots 57$ Саша выиграет первым ходом, увеличив количество камней в куче в два раза (при $S = 57$ можно добавить

в кучу один камень). При меньших значениях S за один ход нельзя получить суммарно в кучах 63 камня и более.

б) $S = 15$. Своим первым ходом Саша ошибётся и умножит количество камней в куче в два раза $\Rightarrow (5, 30)$. Костя домножит количество камней в куче на 2 и победит.

2. $S = 26$ или 28. В этих случаях Саша не может выиграть первым ходом, но при верном первом ходе выиграет вторым. При $S = 26$: $(5, 26) \rightarrow (10, 26)$, затем любой ход Кости — и Саша домножает количество камней в большей куче в два раза и выигрывает. При $S = 28$: $(5, 28) \rightarrow (6, 28)$, затем любой ход Кости — и Саша домножает количество камней в большей куче в два раза и выигрывает.

3. Возможные значения S : 25, 27. При таких S вне зависимости от ходов Саши Костя выиграет либо первым, либо вторым ходом.

Ответ:

1. а) При $S = 29, 30 \dots 57$.
- б) $S = 15$, дважды умножая на 2, получаем: $2 \cdot 2 \cdot 15 + 5 = 65$.
2. $S = 26$ или 28.
3. $S = 25$ или 27.

12

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) два камня или увеличить количество камней в куче в три раза. Например, пусть в одной куче 10 камней, а в другой — 7 камней; такую позицию в игре будем обозначать $(10, 7)$. Тогда за один ход можно получить любую из четырёх позиций: $(12, 7)$, $(30, 7)$, $(10, 9)$, $(10, 21)$. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 68. Победителем считается игрок, сделавший последний ход, т. е. первым получивший такую позицию, когда в кучах всего будет 68 камней и более.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях (6, 36), (7, 35), (9, 34) выигрышная стратегия есть у Пети. Чтобы выиграть, ему достаточно утроить количество камней во второй куче.

Выполните следующие задания.

1. Для каждой из начальных позиций (6, 21), (7, 20) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

2. Для каждой из начальных позиций (4, 20), (6, 20) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

3. Для начальной позиции (5, 20) укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии. Постройте дерево всех партий, возможных при указанной выигрышной стратегии. Представьте дерево в виде рисунка или таблицы.

Решение:

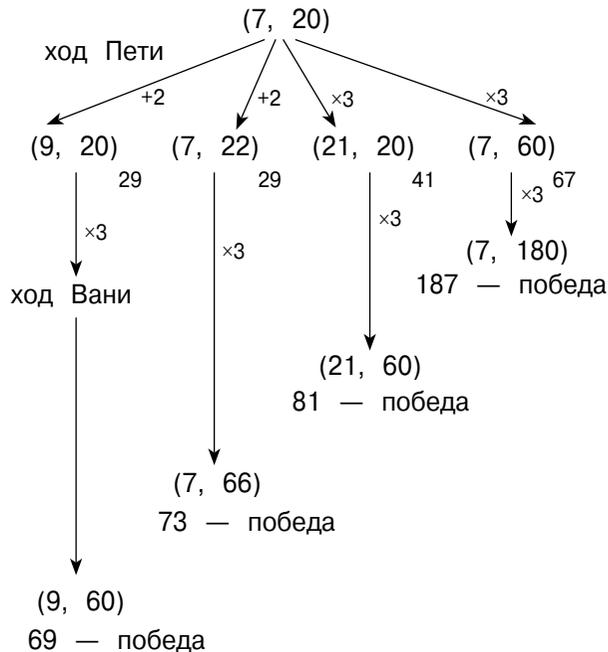
1. В начальной позиции (6, 21) выигрышная стратегия есть у Пети. Он своим первым ходом увеличивает количество камней во второй куче в три раза и побеждает.

$$\begin{array}{c} \times 3 \\ (6, 21) \\ \downarrow \end{array}$$

$$\downarrow$$

(6, 63) 69 камней — победа Пети

В начальной позиции $(7, 20)$ выигрышная стратегия есть у Вани. Своим первым ходом Петя может добавить в одну из куч два камня или увеличить количество камней в одной из куч в три раза, получив позиции $(9, 20)$, $(7, 22)$, $(21, 20)$, $(7, 60)$. Независимо от хода Пети, Ваня своим первым ходом увеличивает количество камней во второй куче в три раза и побеждает.



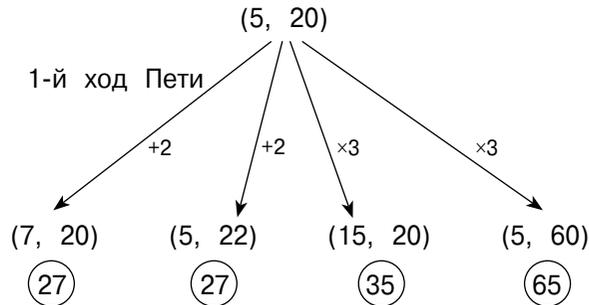
2. В начальной позиции $(4, 20)$ выигрышная стратегия есть у Пети. Он своим первым ходом должен получить позицию $(6, 20)$, добавив в первую кучу два камня. Ваня своим первым ходом может получить позиции $(8, 20)$, $(18, 20)$, $(6, 22)$ и $(6, 60)$. Во всех случаях Петя увеличивает количество камней во второй куче в три раза и выигрывает своим вторым ходом.

В начальной позиции $(6, 20)$ выигрышная стратегия есть у Вани. Эта позиция разобрана выше. В этом случае побеждает игрок, который будет ходить вторым (в данном случае это Ваня).

3. Рассмотрим начальную позицию $(5, 20)$.

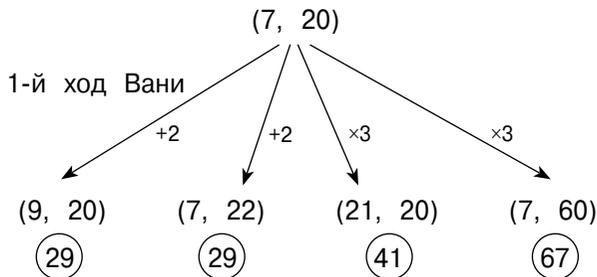
Выигрышная стратегия есть у Пети, если своим первым ходом он получит позицию $(7, 20)$, поскольку в остальных случаях Ваня своим пер-

вым ходом, следующим после хода Пети, утраивает камни в большей куче и выигрывает.



Рассмотрим позицию $(7, 20)$.

В данной стратегии любой ход Вани не приводит к выигрышу. В каждой из этих позиций Петя утраивает количество камней во второй куче и выигрывает своим вторым ходом.



Ответ:

1. В начальной позиции $(6, 21)$ выигрышная стратегия есть у Пети. В начальной позиции $(7, 20)$ выигрышная стратегия есть у Вани.
2. В начальной позиции $(4, 20)$ выигрышная стратегия есть у Пети. В начальной позиции $(6, 20)$ выигрышная стратегия есть у Вани.
3. В начальной позиции $(5, 20)$ выигрышная стратегия есть у Пети.

13

Два игрока, Миша и Вова, играют в следующую игру. Слова в этой игре — просто цепочки букв, они не должны быть осмысленными словами русского языка. Игра заключается в том, что игроки составляют слово из набора, приписывая по очереди буквы к концу составляемого слова, т. е. справа.

При этом каждое промежуточное слово должно быть началом одного из заданных слов. Выигрывает тот, кто получит одно из заданных слов целиком. Первый ход делает Миша, т. е. Миша пишет первую букву составляемого слова.

Пример. Заданный набор слов: {АНТАРКТИДА, АНТРАЦИТ, АБАРА, АБАЖУР, БББ, БАОБАБ, БАР}. Первым ходом Миша пишет Б (он мог написать Б или А). Вова в ответ дописывает А и получает БА (он мог ещё получить ББ). Вторым ходом Миша получает БАР и выигрывает.

В заданиях используются следующие понятия. Стратегия игрока — правило, указывающее игроку ход, который он должен сделать. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Стратегия игрока называется выигрышной, если игрок выигрывает в любой партии, разыгранной в соответствии с этой стратегией, как бы ни играл противник.

Множество всех партий, которые могут получиться при данной стратегии, представляется в виде дерева, называемого деревом всех партий для заданной стратегии. В узлах дерева — позиции игры; на рёбрах — ходы, которые переводят одну позицию в другую; корень дерева — начальная позиция игры. Дерево всех партий для данной стратегии можно описать с помощью рисунка или таблицы.

Выполните следующие задания.

1. а) Укажите, у кого есть выигрышная стратегия при исходном наборе слов {ГДЕЖЗИКЛ, КЛМНБВГ}. Опишите эту стратегию.

б) Укажите, у кого есть выигрышная стратегия при исходном наборе слов {ДВАДВА...ДВА, ПОРАПОРА...ПОРА} (в первом слове ДВА повторено 33 раза, т. е. его длина 99 букв; во втором слове ПОРА повторено 22 раза, т. е. его длина 88 букв). Опишите эту стратегию.

2. В задании 1а поменяйте местами две буквы в более коротком слове так, чтобы теперь выигрышная стратегия была у другого игрока. Напишите полученный набор слов; опишите выигрышную стратегию.

3. Рассмотрим набор слов {СТОЛ, СТОЛЕТИЕ, СПОРТ, КОЛЕСО, КОЛБА, КАК}. У кого из игроков есть выигрышная стратегия для этого набора? Приведите в виде рисунка или таблицы дерево всех партий, возможных при этой стратегии.

Решение:

1. а) Выигрышная стратегия есть у Миши. Первым ходом он пишет К. В конце игры у игроков должно получиться слово КЛМНБВГ. Длина слова равна 7 (нечётная), поэтому последнюю букву напишет Миша.

б) Выигрышная стратегия есть у Миши. Первым ходом он пишет Д. После этого у игроков должно получиться слово ДВАДВА...ДВА. В этом слове 99 букв. Миша напишет 1-ю, 3-ю, 5-ю и т. д. буквы, т. е. все нечётные буквы. Поэтому 99-ю букву напишет Миша и, следовательно, выиграет.

2. В слове КЛМНБВГ нужно поставить на первое место букву Г. Например, можно поменять местами первую и последнюю буквы; получится слово ГЛМНБВК.

Полученный набор слов: {ГДЕЖЗИКЛ, ГЛМНБВК}. Теперь оба слова начинаются с буквы Г, и Миша должен написать Г. Вова пишет Д, получается ГД. Теперь в конце игры получится слово ГДЕЖЗИКЛ. Длина слова — 8 букв. Поэтому выиграет Вова.

3. Выигрышная стратегия есть у Вовы. Миша первым ходом может поставить либо букву С, либо букву К. В зависимости от первой поставленной буквы Вове следует действовать в соответствии со стратегией, изображённой на рисунке выше. При любом ходе Миши у Вовы есть выигрышная стратегия. Дерево всех партий для этой стратегии показано на следующем рисунке. Подчёркнуты позиции, в которых партии заканчиваются.



Ответ:

1. а) Выигрышная стратегия есть у Миши.
- б) Выигрышная стратегия есть у Миши.

2. Нужно в слове КЛМНБВГ поставить на первое место букву Г. Например, можно поменять местами первую и последнюю буквы; получится слово ГЛМНБВК.
3. У Вовы есть выигрышная стратегия.

14

Два игрока, Петя и Ваня, играют в следующую игру. Задан некоторый набор символьных цепочек (слов), в котором ни одно слово не является началом другого. Игра начинается с пустой строки, в конец которой игроки по очереди дописывают буквы по одной букве за ход так, чтобы полученная цепочка на каждом шаге была началом одного из заданных слов. Первый ход делает Петя. Выигрывает тот, кто первый составит слово из заданного набора.

Выполните следующие задания.

1. а) Определите, у кого из игроков есть выигрышная стратегия для набора слов {КОБРАГИБУС, КОБИМУЛЮС}.
 б) Определите, у кого из игроков есть выигрышная стратегия для набора слов {ПИРАТ...ПИРАТ, ЗОЛОТО...ЗОЛОТО}. В первом слове 155 раз повторяется слово ПИРАТ, а во втором — 14 раз повторяется слово ЗОЛОТО.
2. В наборе слов, приведённом в задании 1а, поменяйте местами две буквы в любом слове так, чтобы выигрышная стратегия была у другого игрока.
3. Дан набор слов {ВОРОНА, ВОЛК, ВОЛНА, КРОНА, КРОКУС, КРОКОДИЛ}. У кого из игроков есть выигрышная стратегия?

Решение:

1. а) Выигрышную стратегию имеет Ваня. Первые три буквы слов одинаковые, соответственно, на четвёртом ходу будет определяться, какое слово будут составлять игроки. Поскольку это ход Вани, он выберет для себя выигрышную стратегию — слово «КОБРАГИБУС». Делаем вывод: слова с чётным количеством букв — выигрывает Ваня, с нечётным — Петя.
- б) Выигрышную стратегию имеет Петя. Петя задаёт развитие игры, определяя, какое слово будут записывать игроки, поэтому он начнёт

с буквы П. Слово ПИРАТ имеет нечётное количество букв и повторяется нечётное количество раз; из пункта 1а мы знаем, что при нечётном количестве символов в цепочке побеждает Петя.

2. Мы знаем, что в пункте 1а выиграет Ваня, соответственно, должны сделать так, чтобы выиграл Петя. Для этого достаточно заменить первую букву в любом слове на любую другую, тогда Петя своим первым ходом определит, какое слово будет написано, соответственно, он начнёт писать слово с нечётным количеством символов.

3. Выигрышная стратегия есть у Вани. Вне зависимости от того, с какой буквы начнёт Петя, Ваня будет иметь выигрышную стратегию. В случае если первая буква будет В, вне зависимости от второго хода Пети Ваня напишет либо слово ВОЛК (если Л), либо ВОРОНА (если Р). Если первая буква будет К, то своим вторым ходом Ваня выберет букву К (КРОК...) и вне зависимости от ходов Пети выиграет, т. к. слова КРОКУС и КРОКОДИЛ имеют чётное количество символов.

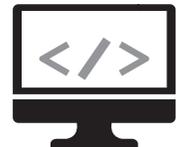
Ответ:

1. а) Выигрышную стратегию имеет Ваня.
- б) Выигрышную стратегию имеет Петя.
2. Заменить первую букву в любом слове.
3. Выигрышная стратегия есть у Вани.



ПРОГРАММИРОВАНИЕ

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



Прежде чем приступить к написанию программы, необходимо освоить теорию алгоритмов и языки программирования, каждый из которых отличается особым набором

команд и синтаксисом. В данном пособии мы рассмотрим несколько языков программирования, познакомимся с их основами, потренируемся писать несложные программы.

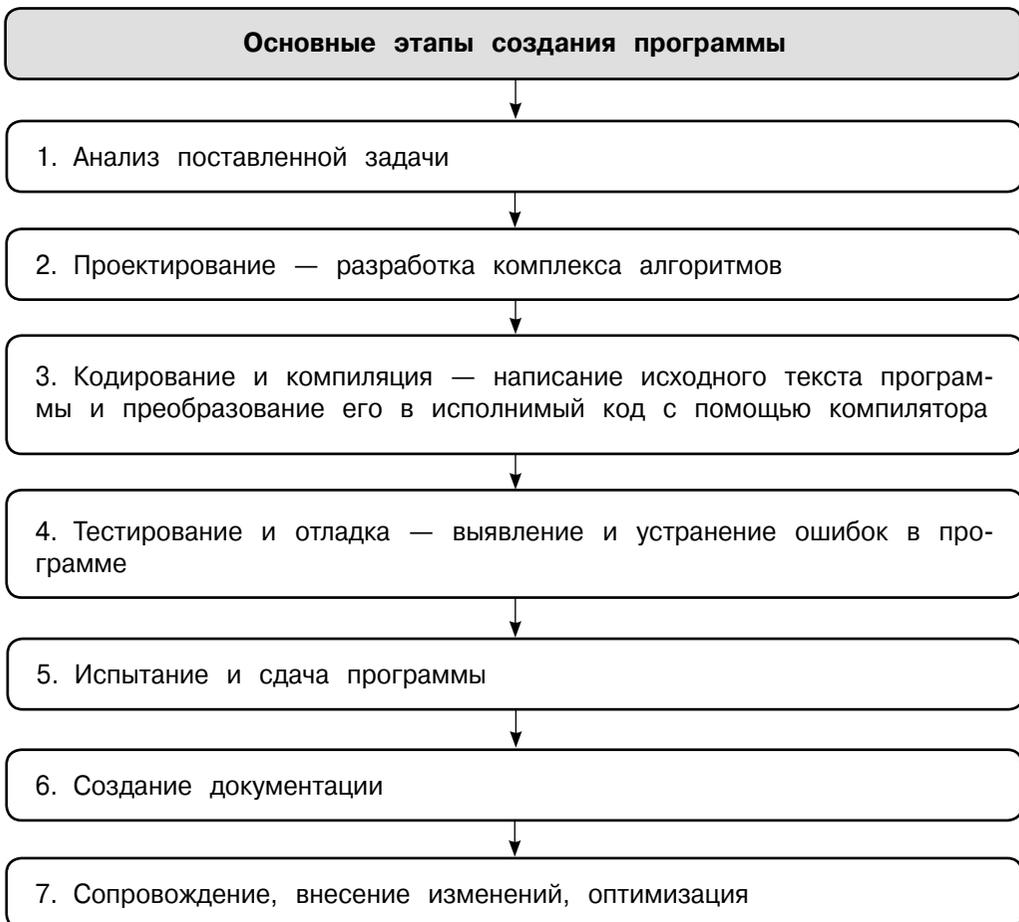
Программирование включает:

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

Для написания решения конкретной задачи, будь то решение квад-

ратного уравнения или разработка приложения, необходимо выполнить определённые действия. Создание программы предполагает поэтапное использование специального цикла и уверенное владение языками программирования.

Условно цикл написания программы, начиная с анализа поставленной задачи и заканчивая оптимизацией готового продукта, можно разбить на семь этапов.



КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Языки программирования представляют собой формальные искусственные системы — специальные языки, которые, в отличие от естественных, сконструированы целенаправленно.

В настоящее время существует более 2000 различных языков и модификаций, которые можно условно разделить по определённым критериям.

По степени зависимости от аппаратных средств

- ▲ Языки низкого уровня.
- ✓ Автокод, Assembler.
- ▲ Языки высокого уровня.
- ✓ Fortran, Pascal, Ada, Basic.

По принципам программирования

- ▲ Процедурные.
- ✓ Assembler, Fortran, Basic, Pascal, Ada.
- ▲ Непроцедурные, включающие объектно-ориентированные.
- ✓ Prolog, Langin, C++, Visual Basic, Java.

По ориентации на класс задач

- ▲ Универсальные.
- ✓ Algol, PL/1, Simula, Basic, Pascal.
- ▲ 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.

Алфавит — разрешённый к использованию набор символов, с помощью которого могут быть образованы слова и величины данного языка.

Словарь — все используемые слова языка программирования.

Грамматика — свод правил и законов 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.

Взаимодействие синтаксических и семантических правил определяют основные понятия языка, такие как операторы, идентификаторы, константы, переменные, функции, процедуры и т. д.

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.

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;
- ▲ программная документация;
- ▲ 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

Любые данные в языках программирования характеризуются своим типом. **Тип данных** — множество

значений, которые могут принимать данные, и множество операций, применимых к этим значениям.

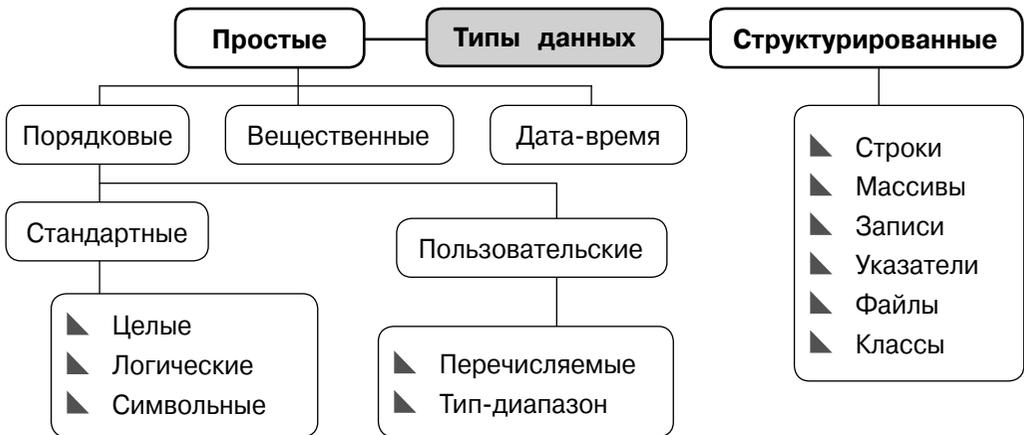
✓ Данные разных типов: 56 — целое число, 26.07.2020 — дата.

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.

Рассмотрим структурированные типы данных подробнее, поскольку они имеют свою специфику реализации и применения.



СТРУКТУРИРОВАННЫЕ ТИПЫ ДАННЫХ

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

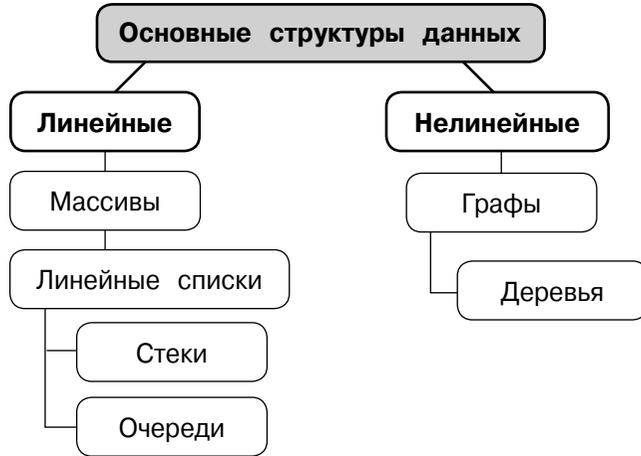
Основным понятием в структурах данных является узел.

Узел (элемент структуры данных) — единица хранения данных, несущая в себе ссылки на связанные с ней элементы (узлы). Каждый узел содержит две составляющие: фак-

тические данные, которые в нём хранятся (это могут быть данные любого типа), и указатель (или ссылку) на следующий узел в последовательности.

В **линейных структурах** данных элементы образуют последовательность или линейный список, обход узлов будет также линейен.

В **нелинейных структурах** данных обход узлов нелинейный и данные непоследовательны.



ГРАФЫ

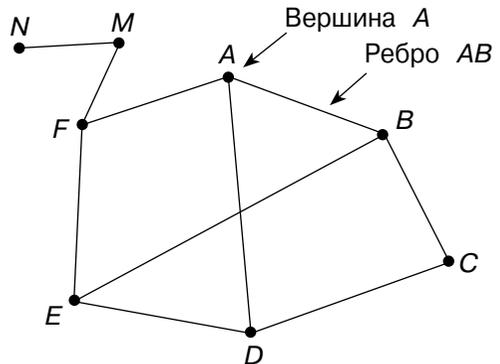
Граф — непустое множество точек и множество (возможно, и пустое) отрезков кривой, оба конца которой принадлежат заданному множеству точек. Точки графа называются **вершинами**, а отрезки — **рёбрами**.

Вершины, последовательно соединённые рёбрами, называются **маршрутом**. Маршрут должен иметь только одно направление. Маршрут, в котором рёбра не пересекаются, называют **путём**. Если путь замкнут, он является **циклом**.

Если при отсутствии некоторого ребра граф делится на несколько компонент, то это ребро называется **мостом**.

Вершина графа, которая имеет только одно ребро, является **висячей**.

Путь, который проходит через каждую вершину, причём единожды, называют **гамильтоновым путём**.



Составные части графа:
 $ABCD$, $ABCDE$, $ABEDC$, $ADCBE$ — маршруты, причём $ADCBE$ не является путём; вершина A — начало маршрута $ABCD$, D — конец маршрута $ABCD$; $ABCDA$, $ABCDEFA$ — циклы; FM — мост; N — висячая вершина

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.

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). Если же цикл включает в себя все рёбра графа по одному разу, такой цикл называется **эйлеровой линией**.

■ Виды графов и их свойства

По количеству рёбер и вершин:

- ▲ нулевые;
- ▲ пустые;
- ▲ полные.

По количеству компонент связности:

- ▲ связные;
- ▲ несвязные.

По расположению на плоскости:

- ▲ планарные;
- ▲ плоские;
- ▲ непланарные.

По направленности:

- ▲ ориентированные;
- ▲ неориентированные.

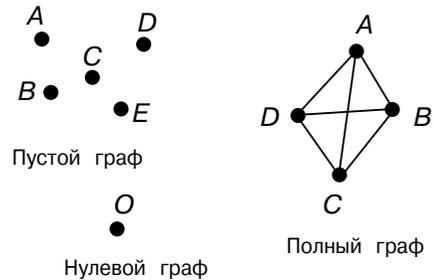
Другие виды графов:

- ▲ взвешенные;
- ▲ псевдографы.

Граф называется **нулевым**, если он состоит из одной точки.

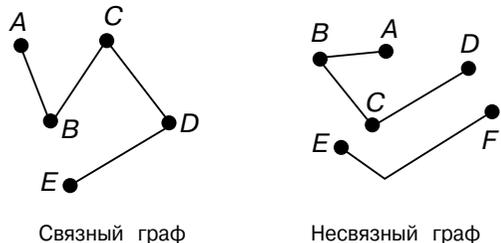
Граф называется **пустым**, если в нём нет ни одного ребра.

Граф называется **полным**, если каждые две его различные вершины соединены одним и только одним ребром.



Граф называется **связным**, если у него одна компонента связности.

Граф называется **несвязным**, если у него больше одной компоненты связности.



Граф, который можно уложить в одной плоскости, называется **плоским**.

Свойства связных и несвязных графов

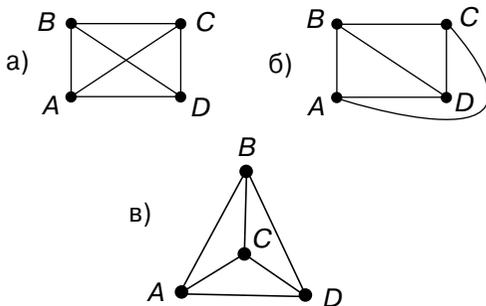
Если в связном графе все степени вершин чётные, то существует эйлеров путь, причём начало пути совпадает с его концом

Если в связном графе степень двух вершин нечётная, то существует эйлеров путь, причём начало и конец пути не совпадают

Если в графе более двух вершин с нечётной степенью, то эйлерова пути не существует

Граф называют **планарным**, если его можно изобразить на плоскости таким образом, что никакие два ребра, за исключением выходящих из общей вершины, не имеют общих точек (т. е. не пересекаются).

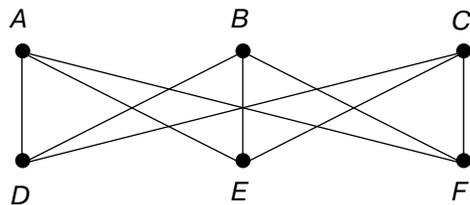
У всякого планарного графа имеется плоское представление.



Планарный граф и его плоское представление

На рисунке а изображён граф, в котором имеется пересечение рёбер. Если ребро AC перебросить

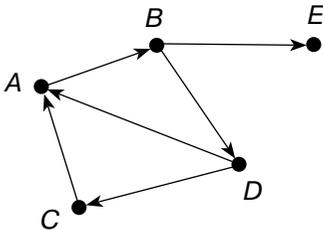
через вершину B (рис. б) и перерисовать граф в другом виде, получим граф $ABCD$ (рис. в), в котором нет пересечения рёбер. Таким образом, можно утверждать, что граф, изображённый на рисунке а, планарный, а его плоское представление — плоский граф на рисунке в. В случае отсутствия возможности подобного преобразования граф не будет являться планарным.



Непланарный граф

На рисунке выше приведён пример графа, который не имеет плоского представления. Никакое перемещение рёбер не позволит превратить его в граф без их пересечения.

Граф, имеющий рёбра с заданной направленностью, называется **ориентированным**.

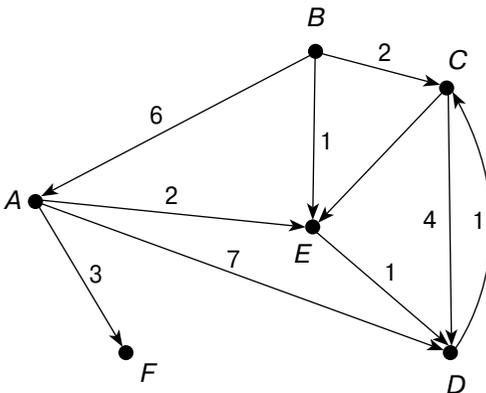


Ориентированный граф

Граф, ни одному ребру которого не присвоено направление, называется **неориентированным**.

Взвешенным (или орграфом) называется граф, некоторым элементам которого (вершинам, рёбрам или дугам) сопоставлены числа.

Чаще всего числа сопоставляют рёбрам или дугам. Это важно при составлении алгоритмов.

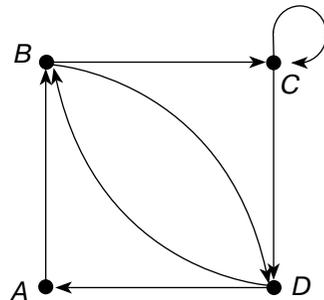


Взвешенный ориентированный граф

✓ Например, есть необходимость купить новую материнскую пла-

ту для компьютера. Обратившись в магазин возле дома, придётся заплатить 50 000 рублей, в центре города — 70 000 рублей. Если бы данный алгоритм был написан с помощью взвешенного графа, то дуга, ведущая к покупке платы возле дома, была бы короче.

Псевдограф — граф, в котором допускается наличие петель и существование более одного ребра между двумя вершинами.

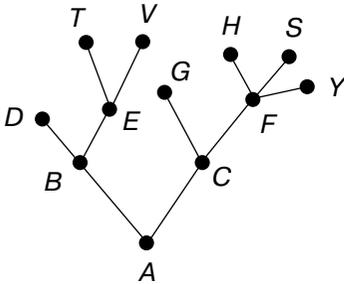


Псевдограф

■ 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

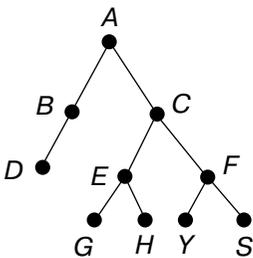
Для каждой пары вершин дерева существует единственный путь, соединяющий их.

Всякое ребро в дереве является мостом. Действительно, после удаления любого ребра дерево распадается на два.

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

Типовые операции над элементами дерева:

- ▲ добавление элемента в дерево;
- ▲ удаление элемента из дерева;
- ▲ обход дерева;
- ▲ поиск элемента в дереве.

Способы представления графа

В программировании довольно сложно представить граф в виде рисунка, поэтому существует множество способов для задания графа.

1) **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 $Sm [A, B] = 1$, в противном случае $Sm [A, B] = 0$.

Матрицу можно представлять в виде таблицы, где в каждой ячейке есть значение 0 или 1.

2) Матрица инцидентности графа.

Количество строк в этой матрице соответствует числу вершин, а количество столбцов — числу рёбер.

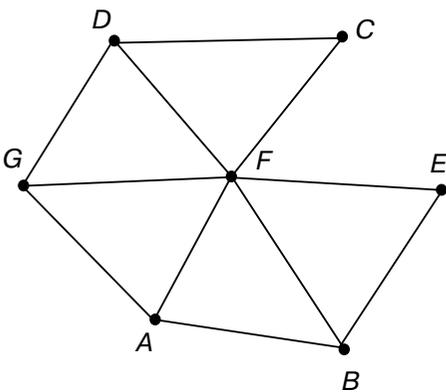
3) Список рёбер.

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>].

4) Списки смежности. Для каждой вершины будет указан список всех смежных с ней вершин (для орграфа — список вершин, являющихся концами исходящих дуг).

✓ <номер_начальной_вершины>: <мера_смежных_вершин>.



На рисунке изображена схема дорог Н-ского района, в таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет. Каждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам A и G на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

Решение:

Граф схемы дорог задан матрицей (таблицей) 7×7 . Сопоставим населённые пункты графа и населённые пункты в таблице. Из F ведут шесть дорог. Таким образом, F — 3. Из городов C и E ведут только две дороги — 4 и 5. Заметим, что из пунктов A и G нет дороги в населённые пункты C и E, следовательно, 1 и 2 — это B и D. Остаются пункты 6 и 7, значит, они соответствуют A и G.

Ответ: 67.

	1	2	3	4	5	6	7
1			*	*			*
2			*		*	*	
3	*	*		*	*	*	*
4	*		*				
5		*	*				
6		*	*				*
7	*		*			*	



Практические задания

15 Между населёнными пунктами *A, B, C, D, E, F* построены дороги, протяжённость которых (в километрах) приведена в таблице.

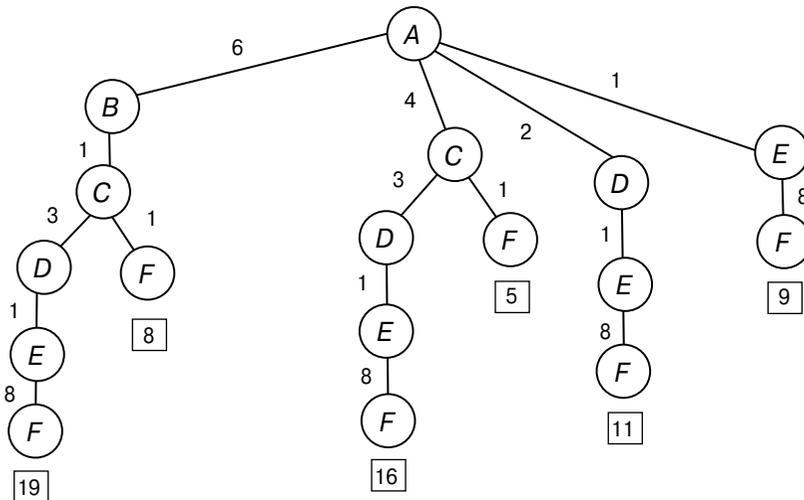
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>A</i>		6	4	2	1	
<i>B</i>	6		1			
<i>C</i>	4	1		3		1
<i>D</i>	2		3		1	
<i>E</i>	1			1		8
<i>F</i>			1		8	

Определите длину кратчайшего пути между пунктами *A* и *F*. Передвигаться можно только по дорогам, протяжённость которых указана в таблице. В ответе запишите только число.

Решение:

Таблица симметрична относительно главной диагонали, поэтому достаточно рассмотреть только нижнюю или только верхнюю часть.

Построим дерево протяжённости дорог, на ветвях будем отображать расстояния.



По дереву можно определить, что длина кратчайшего пути равна 5 км.

Ответ: 5.

- 16** Даша живёт на станции Арбузная, а учится на станции Яблочная. Чтобы вовремя успевать утром на уроки, ей необходимо ехать по самой короткой дороге. Проанализируйте таблицу и укажите длину кратчайшего пути от станции Арбузная до станции Яблочная.

	Арбузная	Васильки	Сельская	Яблочная	Ежевичная
Арбузная		1			1
Васильки	1			5	
Сельская				1	2
Яблочная		5	1		7
Ежевичная	1		2	7	

Решение:

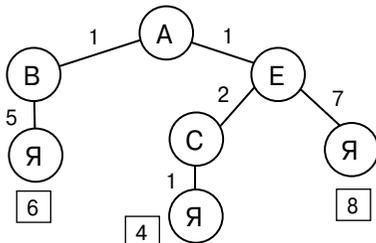
Найдём все варианты маршрутов от станции Арбузной до Яблочной и выберем самый короткий. Построим дерево протяжённости дорог, на ветвях будем отображать расстояния.

От станции Арбузная (А) можно попасть в пункты Васильки (В), Ежевичная (Е).

Из пункта В можно попасть на станцию Яблочная (Я).

Из пункта Сельская (С) можно попасть в пункты Я, Е.

Из пункта Я можно попасть в пункт Е.



Самый короткий путь: А—Е—С—Я.
Длина маршрута составляет 4 км.

Ответ: 4.



Если в город S из города A можно добраться только из городов X , Y и Z , то количество различных путей из города A в город S равно сумме числа различных путей из A в X , из A в Y и из A в Z , т. е.:

$$N_S = N_X + N_Y + N_Z,$$

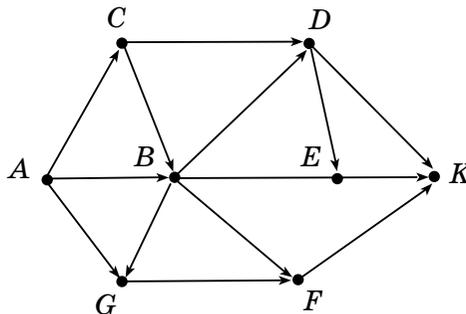
где N_S — количество путей из города A в город S (от вершины A к вершине S на схеме).

Задачи такого типа удобнее решать с конца, т. е. начиная с последнего города.

17

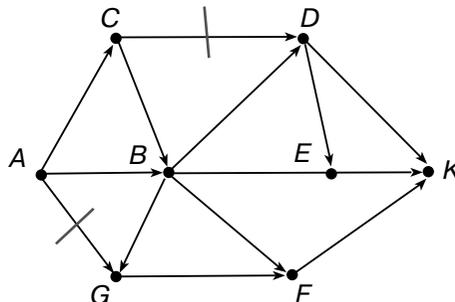
На рисунке представлена схема дорог, связывающих города A , B , C , D , E , F , G и K . По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Сколько существует различных путей из города A в город K , проходящих через город B ?



Решение:

Способ 1. В условии сказано, что путь должен проходить через город B , поэтому сначала вычеркнем дороги, которые не проходят через B . К ним относятся дороги из C в D и из A в G .



Решим задание с конца, рассмотрим сначала город K . В K можно попасть из трёх городов — D , E и F ; запишем это так: $K = D + E + F$.

Аналогично рассмотрим города D , E и F :

$D = B$ (путь из C в D не учитываем, т. к. его вычеркнули),

$E = D + B$,

$F = B + G$.

$D = B$,

$C = A = 1$,

$B = C + A$,

$D = B$,

$G = B$ (путь из A в G не учитываем).

Теперь возвращаемся, подставляя найденные значения:

$B = C + A = 2$,

$G = B = 2$,

$D = B = 2$,

$F = B + G = 2 + 2 = 4$,

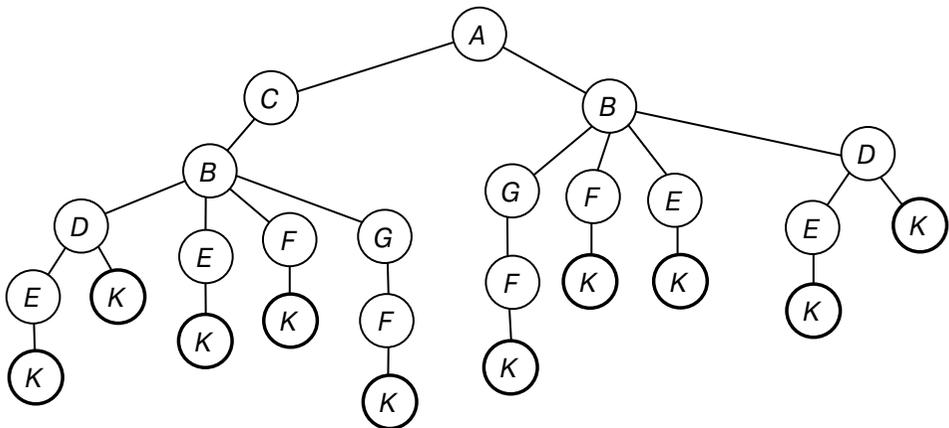
$E = D + B = 2 + 2 = 4$.

Поскольку нас интересуют пути, проходящие через город B , исключаем те дороги, которые минуют B :

$K = D + E + F = 2 + 4 + 4 = 10$.

Способ 2 (построение дерева).

Исключив дороги, которые не ведут через город B , построим дерево.

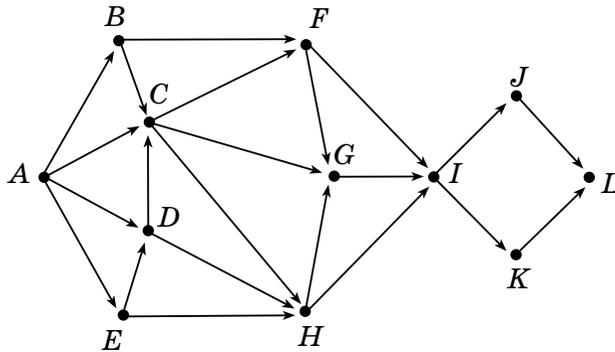


Посчитаем количество путей в K , получим 10.

Ответ: 10.

18

На рисунке представлена схема дорог, связывающих города $A, B, C, D, E, F, G, H, I, J, K, L$. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города A в город L , проходящих через город G , но не проходящих через город J ?



Решение:

Рассмотрим пути из A в L . Количество путей, ведущих в каждый пункт, равно сумме путей, ведущих в пункты, из которых можно попасть в данный. Буквами $B, C, D...$ будем обозначать количество путей из A в соответствующий пункт.

Если путь не должен проходить через какой-то город, то при подсчёте сумм не будем учитывать этот город. А если город, наоборот, обязательно должен лежать на пути, тогда для городов, в которые из нужного города идут дороги, в суммах надо брать только этот город. В данной задаче следует обязательно пройти через город G , но не нужно проходить через город J .

Посчитаем последовательно количество путей до каждого из городов:
 $A = 1$.

$B = A = 1$ — из A в B идёт только один путь.

$E = A = 1$ — из A в E идёт тоже только один путь.

$D = A + E = 1 + 1 = 2$. В пункт D идут две стрелки, значит, количество путей складывается из суммы путей A и E .

$C = A + B + D = 1 + 1 + 2 = 4$. В пункт C входят три стрелки, поэтому сумма $A + B + D$ — это количество путей в C .

$F = B + C = 1 + 4 = 5$. В пункт F идут две стрелки — из пунктов C и B , значит, складываем результаты C и B .

$H = C + D + E = 4 + 2 + 1 = 7$. В пункт H идут три стрелки из пунктов C , D , E , их сумма — количество путей в H .

$G = C + F + H = 4 + 5 + 7 = 16$. В пункт G ведут три стрелки из пунктов C , F , H , их сумма — количество путей в G .

Если посмотреть на схему дорог, то $I = G = 16$.

$J = I = 16$, но пункт J не должен быть учтён в поиске.

$K = I = 16$.

Таким образом, количество путей, проходящих через город G , но не проходящих через город J , равно $L = K = 16$.

Ответ: 16.

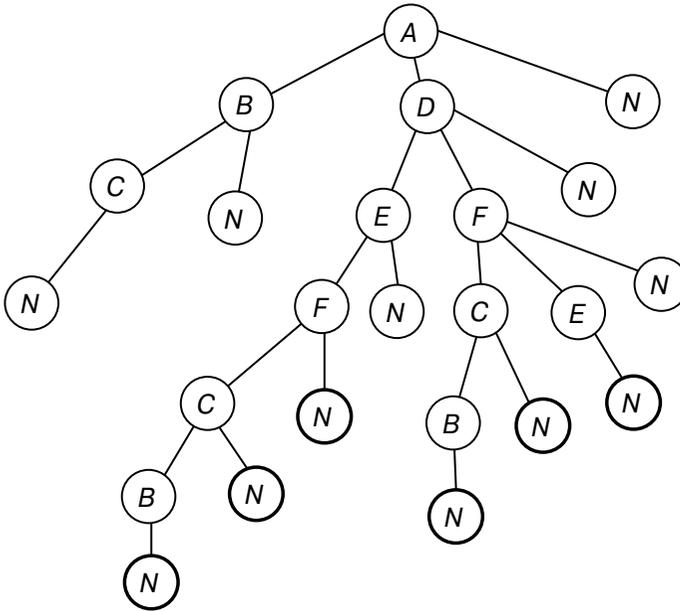
- 19** Между населёнными пунктами A, B, C, D, E, F, N построены дороги с односторонним движением. В таблице указана протяжённость каждой дороги (отсутствие числа в таблице означает, что прямой дороги между пунктами нет).

	A	B	C	D	E	F	N
A		3		5			14
B			2				8
C		2					7
D					1	4	4
E						1	5
F			12		1		9
N							

Сколько существует таких маршрутов из A в N , которые проходят через пять и более населённых пунктов? Два раза проходить через один пункт нельзя.

Решение:

Построим граф-дерево, учитывая табличные данные.



Посчитаем количество только тех маршрутов, которые проходят через пять и более пунктов, получим число 6.

Ответ: 6.

МАССИВ

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

Массивы используются для обработки большого количества однотипных 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.

Элемент массива — отдельная переменная, входящая в массив.

Индекс элемента массива — номер элемента в этом массиве.

Размерность массива — количество элементов, которое содержит массив. Массив, состоящий из некоторого ряда элементов, называется **векторным** или **одномерным**. Если массив включает несколько рядов, он называется **матричным** или **многомерным**.

✓ [1, 2, 5, 7, 8] — векторный массив, состоящий из пяти элементов.

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

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

Если два массива и более описаны одинаково, то с ними можно проводить следующие операции.

▲ **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** из одного массива в другой. Для выпол-

нения действия необходимо, чтобы массивы, участвующие в данной операции, были описаны одинаково.

▲ **Поиск в массиве элементов**, удовлетворяющих некоторым условиям. 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** (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. Так, для массива A [12, 34, 4, 1] элемент {1} будет минимальным, а элемент {34} — максимальным.

▲ **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 алгоритмов

▲ **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 отсортированных данных. Алгоритм ведёт себя естественно, если учитывает эту характеристику входной последовательности и работает лучше.

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

Сортировка обмена (пузырьком)

Идея метода: сортировка массива от элемента a_0 к элементу a_i с попарным изменением местами двух элементов массива при определённых условиях.

▲ 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 вверху 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 вверх.
- 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.

Сортировка выбором

Идея метода: создавать отсортированную последовательность путём присоединения к ней одного элемента за другим в правильном порядке. Строим готовую последовательность, начиная с левого конца массива.

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 ↔ 4

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

шаг 2: 3 ↔ 9

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

шаг 3: 4 ↔ 7

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

шаг 4: 6 ↔ 6

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

шаг 5: 7 ↔ 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)

Этот вид сортировки является наиболее применяемым и эффективным.

Идея метода: разделение массива на подмассивы и сортировка этих подмассивов.

- 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 $a[i]$, — впра-

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

$\leq a[i]$	$a[i]$	$\geq a[i]$
-------------	--------	-------------

3) Если в подмассиве более двух элементов, рекурсивно запускаем для него ту же процедуру. Произво-

дим эту операцию для обоих подмассивов.

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

Сортировка слиянием

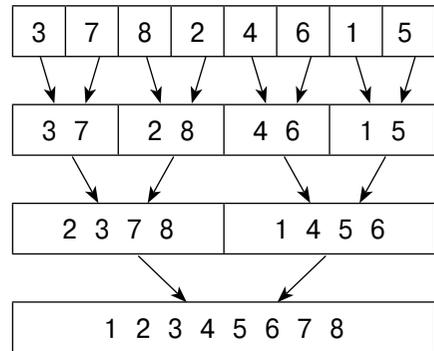
Идея метода: вместо разделения по опорному элементу, как в принципе сортировки вставками, массив просто делится пополам.

Массив разделён до последовательностей длины один

Слияние до упорядоченных пар

Слияние пар в упорядоченные четвёрки

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 мо-

гут выступать разные части поля данных. Ключи разных элементов списка могут совпадать.

Существует несколько разновидностей списков, для которых определены дополнительные правила работы с начальным и конечным элементами.

По способу связи элементов различают линейные и циклические списки.

Операции, выполняемые над списками

Начальное формирование списка (создание первого элемента)

Добавление элемента в конец списка

Чтение элемента с заданным ключом

Вставка элемента в заданное место списка (до или после элемента с заданным ключом)

Удаление элемента с заданным ключом

Упорядочивание списка по ключу

Копирование списка

Поиск заданного элемента

Объединение нескольких списков в один

Разбиение одного списка на несколько отдельных списков

Кольцо (кольцевой список) — циклический список, имеющий дополнительную связь между последним и первым элементами (первый узел является следующим для последнего).

У некоторых видов списков добавление, удаление и доступ к значениям элементов всегда осуществляются

только в крайних узлах (первом или последнем). К ним относятся стеки и очереди.

Стек — 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 сте-**

ка. Иногда стек называют списком LIFO (*Last In — First Out* — «последним вошёл, первым вышел»).

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.

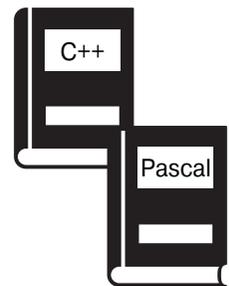
Очередь — разновидность линейного списка, в котором все включения элементов производятся на одном его конце, а удаление узлов (и обычно всякий доступ) — на другом. Эти узлы называются началом и концом очереди.

Иногда очередь называют списком FIFO (*First In — First Out* — «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

В данном пособии рассматриваются основы четырёх языков программирования: школьный (учебный) алго-

ритмический язык, Pascal (Паскаль), C++ (Си++), Python (Питон).



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

- 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.

▲ C++

- 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.

▲ Python

- 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.

▲ Pascal

- 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: «:=», «..», «<>», «<=>», «>=», «{ }».



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

алг — алгоритм
сим — символьный тип
лит — литерный тип
цел — целый тип
вещ — вещественный тип
лог — логический тип
длин — длина
рез — результат
нач — начало
кон — конец
таб — таблица, массив
нц — начало цикла
кц — конец цикла
дано **всё**
надо **или**
для **если**

Алгоритмический язык

да **то**
нет **иначе**
и **до**
ввод **пока**
вывод

Pascal

and — логическое И
mod — остаток от деления
array — массив
not — логическое НЕ
begin — начало блока
or — логическое ИЛИ
case — вариант
of — из
const — константа
object — объект
div — деление нацело
procedure — процедура
go to — переход на...
program — программа
do — выполнять
repeat — повторять
string — строка
else — иначе
then — то
end — конец блока
file — файл
type — тип
for — для

Pascal

until — до
 function — функция
 if — если
 var — переменная
 interrupt — прерывание
 while — пока
 with — с

C++

struct — структура
 switch — выбор
 typedef — определение типа
 union — совмещение
 void — пустой
 while — пока

C++

auto — автоматический
 break — прервать
 case — вариант
 char — символьный тип
 const — константа
 continue — продолжить
 default — прочие
 do — цикл
 double — двойной
 define — определить
 else — иначе
 elif — иначе если
 float — плавающий тип
 for — для
 goto — переход
 if — если
 int — целочисленный тип
 include — включать
 long — длинный
 return — возвращать
 short — короткий
 not — нет
 static — статический

Python

False — ложь
 True — правда
 and — логическое И
 break — выход из цикла
 class — пользовательский тип, состоящий из методов и атрибутов
 continue — переход на следующую итерацию цикла
 def — определение функции
 del — удаление объекта
 elif — в противном случае, если
 else — если
 except — перехватить исключение
 for — цикл for
 from — импорт нескольких функций из модуля
 global — позволяет сделать значение переменной, присвоенное ей внутри функции, доступным и за пределами этой функции
 if — если
 import — импорт модуля
 in — проверка на вхождение

Python

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

РАЗДЕЛИТЕЛИ ЯЗЫКА

Разделители — специальные символы, которые используются в служебных целях от организации текста программы до определения указаний компилятору языка. Ниже рассмотрим только те разделители, которые могут понадобиться для написания простых программ, а также в ходе выполнения заданий ЕГЭ.

▲ Алгоритмический язык

Пробел — разделяет слова;
 табуляция — позволяет оформить структуру программы;
 , запятая — разделение величин в строке, перечисление элементов;
 : двоеточие — используется в спецсимволах (например, оператор присваивания);
 () скобки — используются для ввода параметров команды;
 перевод строки (enter) — переводит на новую строку.

▲ Pascal

Пробел — разделяет слова;

. точка — обозначение конца программы;
 ; точка с запятой — разделяет описание и операторы;
 табуляция — позволяет оформить структуру программы;
 , запятая — разделение величин в строке, перечисление элементов;
 : двоеточие — используется в спецсимволах (например, оператор присваивания);
 () скобки — используются для ввода параметров команды;
 перевод строки (enter) — переводит на новую строку.

▲ C++

Пробел — разделяет слова;
 ; точка с запятой — признак конца инструкции (команды);
 : двоеточие — используется для указания базового класса при наследовании классов, задаёт список инициализации для конструктора;
 , запятая — разделение величин в строке, перечисление элементов;

() скобки — используются для ввода параметров команды;
 табуляция — отступы в блоках;
 '\n' — символ перехода на новую строку.

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

В каждом языке программирования изначально определена своя структура. Она устраняет проблему сложности разработки, делая её понятной для пользователя, повышает надёжность работы программы, сокращает срок тестирования и разработки.

Структура программы — искусственно выделенные программистом взаимодействующие части программы.

Иногда для написания программы требуется что-то прокомментировать. Для этого в каждом языке используются особые символы. Обычно комментарии размещаются до конца строки, новая строка идёт уже вне комментария.

Ниже представлены структуры программ с использованием рассматриваемых здесь языков.

Алгоритмический язык

алг <название алгоритма> (аргумент и результат)

дано | условия применимости алгоритма

надо | цель алгоритма

нач описание промежуточных величин

последовательность команд (тело программы)

кон

Дано и **надо** используются для понимания выполняемой задачи. После знака | можно писать комментарии, они не будут отображаться.

Pascal

Program <название программы>;

const ...; (константы)

var...; (переменные)

(процедуры и функции)

begin

... (тело программы)

end.

{ } или (* *) — комментарии

▲ C++

```
#include <iostream> // подключение заголовочного файла с помощью директивы include
using namespace std; //использование пространства имён std
int main() //функция main, внутри скобок идёт набор инструкций
{
    тело программы
    return 0;
}
// после двойной наклонной строки // можно писать любые однострочные комментарии
/*
    многострочный комментарий
*/
```

▲ Python

Любая Python-программа состоит из последовательности лексем (допустимых символов), записанных в определённом порядке и по определённым правилам.

Этот язык не содержит операторных скобок (`begin...end` в Pascal или `{..}` в C++), вместо этого блоки выделяются отступами — пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием.

Количество пробелов в отступах произвольно, однако по договорённости равняется четырём пробелам. При этом отступ всего блока должен быть одинаковым.

Несколько строк кода с одинаковым отступом формируют отдельный блок кода. Благодаря такой системе значительно повышается читаемость кода, у исполнителя вырабатывается привычка писать понятно и структурированно.

`#` — комментарий в программе идёт до конца строки.

ИДЕНТИФИКАТОРЫ

С помощью **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 делает программу более читабельной. Во-вторых,

каждый язык программирования накладывает на имена ряд ограничений.

Действие идентификатора в программе может быть локальным или глобальным. **Глобальный идентификатор** объявляется вне функций и доступен во всей программе. **Локальный идентификатор** объявляется внутри определённой функции и доступен только в её рамках.

▲ 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.

▲ 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.

▲ C++

⊕ Можно использовать:

- латинские буквы;
- цифры;
- символы тире `-` и нижнее подчёркивание `_`;
- первым 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.

▲ 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.

Правильно подобранные идентификаторы значительно упрощают программистам и аналитикам понимание порядка работы программы, а также помогают исправить или расширить исходный код, чтобы использовать его для текущей или новой задачи.

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.

Атрибуты, связанные с понятием переменной

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

Числовые переменные и константы записываются в виде конкретного числа, они бывают двух типов: целые и вещественные.

Целый тип — положительные и отрицательные целые числа.

✓ -5897, +39, 20.

Вещественный тип — числа, имеющие целую и дробную части. Существует две формы 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 фор-

мы 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 тип — 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 тип — один символ. Заключается в кавычки.

Дата/время — тип записи даты или времени.

✓ 12.12.2020.

Логический тип предполагает вывод значения «истина» или «ложь» в разных интерпретациях (0 или 1, true или false).

В каждом языке программирования различные типы переменных и констант имеют свои обозначения.

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, $summ$ В программу внесли 3 переменные целочисленного типа с именами i, mn, $summ$. ✓ $n2:=34$ $n2$ 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>✓ $var\ a, b: integer;$ $H: real;$ $s1, s2: string$ В программу внесены несколько переменных. Первая строка: объявление целочисленных переменных a и b; вторая строка: объявление вещественной переменной H; третья строка: объявление строковых переменных $s1$, $s2$. ✓ $a2=45;$ $ret='Даша';$ $n=True;$</p>

Тип переменных и констант	Примеры
	<p>Переменным присвоены их значения соответствующего типа: переменной <code>a2</code> — значение целого типа, переменной <code>ret</code> — значение строкового типа, переменной <code>n</code> — значение логического типа</p>
C++	
<p><code>char</code> — символьный <code>short</code> — целые (-32768...32767) <code>int</code> — целое число <code>long</code> — длинное целое число <code>float</code> — вещественное число <code>bool</code> — логическое значение</p>	<p>✓ <code>const int N=15;</code> Объявление целочисленной константы <code>N</code> со значением 15.</p> <p>✓ <code>const char simb='f';</code> Константа символьного типа, в которой хранится символ <code>f</code>. Записывается в одинарных кавычках.</p> <p>✓ <code>int x, y;</code> Переменные <code>x</code> и <code>y</code> целочисленного типа.</p> <p>✓ <code>char str[10];</code> Пример объявления строки, где <code>str</code> — имя переменной, <code>10</code> — размер массива, <code>9</code> символов и ноль-терминатор.</p> <p>Строки в <code>C++</code> представляются как массивы элементов типа <code>char</code>, заканчивающиеся ноль-терминатором. <code>\0</code> — символ ноль-терминатора.</p> <p>Символьные строки состоят из набора символьных констант, заключённых в двойные кавычки. При объявлении строкового массива необходимо учитывать наличие в конце строки ноль-терминатора и отводить под него дополнительный байт</p>
Python	
<p><code>int</code> — целый <code>float</code> — вещественный</p>	<p>Не нужно указывать тип переменной во время присвоения значения.</p>

Тип переменных и констант	Примеры
<p>str — строковый bool — логический</p>	<p>Python — язык с динамической типизацией, поэтому на этапе присвоения интерпретатор «понимает», что переменная <code>hello</code> имеет тип <code>str</code> (<i>string</i> — строка). При этом далее можно присвоить этой переменной значение <code>123</code>, и её тип автоматически сменится на <code>int</code> (<i>integer</i> — число).</p> <p>✓ <code>a, b, c = 1, 'hello', 12</code></p> <p>Можно в одной строке присваивать разным переменным разные значения. Данная выше запись равнозначна следующей:</p> <pre>a = 1 b = 'hello' c = 12</pre>

! 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.

✓ `random (N)` — генерирует случайное число 0-N.

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	<code>exp(x)</code>	<code>exp(x)</code>
Случайное число	<code>rnd(x)</code>	<code>random(N)</code>

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>

Название оператора	Форма записи функции
	C++
Косинус	$\cos(x)$
Тангенс	$\tan(x)$
Натуральный логарифм	$\log(x)$
Показательная	$\exp(x)$
Случайное число	$\text{random}(z)$ — генерирует случайное число от 0 до $z-1$

Название оператора	Форма записи функции
	Python
Модуль числа	$\text{abs}(x)$
Корень квадратный	$\text{sqrt}(x)$
Деление нацело	$x // y$
Остаток от деления нацело	$x \% y$
Синус	$\sin(x)$
Косинус	$\cos(x)$
Тангенс	$\tan(x)$
Натуральный логарифм	$\log(x)$
Показательная	$\exp(x)$
Случайное число	$\text{randint}(a,b)$ — генерирует случайное число от a до b ; $\text{uniform}(a,b)$ — вещественное число от a до b



Чтобы применить функцию $\text{ctg}(x)$, необходимо использовать формулу:

$$\text{ctg}(x) = \frac{1}{\text{tg}(x)}$$

ОПЕРАТОРЫ И ОПЕРАЦИИ

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 «+».

Во всех изучаемых языках программирования арифметические операции записываются одинаково.

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

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

! Важно помнить, что при делении целого числа на целое можно получить дробное. 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 получится нецелое 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

! 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.

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

Название оператора	Форма записи			
	Алгоритмический язык	Pascal	C++	Python
Присваивание	$:=$	$:=$	$=$	$=$
Ввод	ВВОД	ReadLn Read	$\text{cin} \gg$	input
Вывод	ВЫВОД	WriteLn Write	$\text{cout} \ll$	print

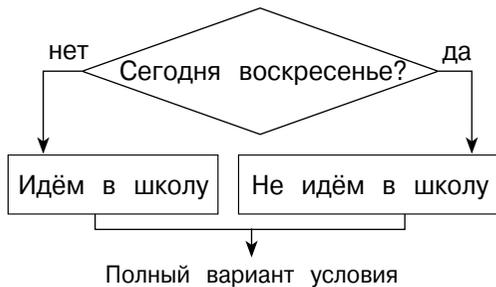
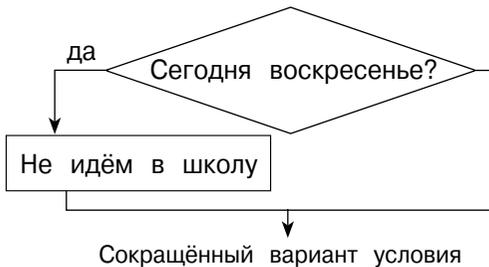
Операторы условия

В зависимости от ситуации процесс может пойти по одной или другой ветви, а может быть усложнён несколькими условиями. Рассмотрим определённую ситуацию. Для того

чтобы узнать, надо ли сегодня идти в школу, следует посмотреть на день недели. Условие можно представить в одном из видов:

- ▶ если сегодня воскресенье, то мы не идём в школу;
- ▶ если сегодня воскресенье, то мы не идём в школу, иначе — идём.

Блок-схемы для этого алгоритма представлены ниже.



Чтобы написать в программе определённые условные конструкции, используют английские слова: if — если, then — то, else — иначе.

Реализация условных конструкций в разных языках программирования имеет свои особенности.

Pascal

Сокращённый вариант:

```
if условие then
    оператор;
```

Полный вариант:

```
if условие then
    оператор
else
    оператор;
```

Алгоритмический язык

Сокращённый вариант:

```
если условие
    то
        действия
всё
```

Полный вариант:

```
если условие
    то
        операторы
    иначе
        операторы
всё
```

Python

Полный вариант:

```
if условие 1:
    операция
elif условие 2:
    операции
else:
    условие 3
```

При необходимости можно использовать конструкцию `elif`

Замена полного варианта на сокращённый:

```
if X:
    A = Y
else:
    A = Z

можно заменить на
A = Y if X else Z
```

C++

Сокращённый вариант:

```
if (условие)
    оператор;
```

Полный вариант:

```
if (условие){
    блок операций;
} else {
    блок операций;
}
```

Операторы отношения

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

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**.

В главе «Алгоритмы и алгоритмизация» мы познакомились с понятием цикла, узнали о циклах с предусловием, постусловием и со счётчиком (параметром), составили алгоритм для нахождения суммы чисел от 1 до N . Рассмотрим, как можно реализовать цикл в языках программирования. Для этого изучим конструкции различных видов циклов.

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

Цикл с предусловием**Алгоритмический язык**

```

нц
  пока условие
    действия
кц

```

Pascal

```

while выражение do
begin
  Внутренний оператор;
end;

```

C++

```

while (Условие) {
  блок операций;
}

```

Python

```

while условие:
  операторы
else:
  операторы

или без else:
while условие:
  операторы

```

Цикл с параметром**Алгоритмический язык**

```

нц для параметр от а до b шаг h
  действия
кц

```

Pascal

```

for управляющая переменная :=исходное значение
to конечное значение
do оператор

```

C++

```

for (Инициализация; Условие; Модификация) {
  блок операций;
}

```

Python

```

for <переменная> in <список> :
  операторы 1...N

```

Цикл с постусловием**Алгоритмический язык**

```

нц
  действия
до условия
кц

```

Pascal

```
repeat
оператор;
until логическое выражение;
```

C++

```
do {
    блок операций;
} while (Условие);
```

Python

В этом языке программирования циклы с постусловием отсутствуют, но решение можно записать в таком виде:

```
while True:
    if not условие:
        break
```



Практические задания

- 20** Обозначим через $m \& n$ поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, $14_{10} \& 5_{10} = 1110_2 \& 0101_2 = 100_2 = 4$. Для какого наименьшего неотрицательного целого числа A формула

$$x \& 51 \neq 0 \rightarrow (x \& A = 0 \rightarrow x \& 25 \neq 0)$$

тождественно истинна (т. е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

Решение:

```
For A in range(0, 1000):
    k=0
    For x in range(0, 1000):
        if x&51==0 or (x&A!=0 or x&25!=0):
            k=k+1
    if k==1000:
        print(A)
```

Ответ: 34.



Данную задачу можно решить несколькими способами. На с. 97 представлено решение посредством рассуждения с использованием элементов алгебры логики.

21 Ниже приведена программа, записанная на нескольких языках программирования.

Pascal	Python
<pre>var m, t: integer; begin readln(m); readln(t); if (m < 8) and (t < 10) then writeln(«YES») else writeln(«NO») end.</pre>	<pre>m = int(input()) t = int(input()) if (m < 8) and (t < 10): print(«YES») else: print(«NO»)</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main(){ int m, t; cin >> s; cin >> t; if (m < 8 && t < 10) cout << «YES» << endl; else cout << «NO» << endl; return 0; }</pre>	<pre>алг нач цел m, n ввод m ввод n если m < 8 и n < 10 то вывод «YES» иначе вывод «NO» все кон</pre>

Было проведено девять запусков программы, при которых в качестве значений переменных вводились следующие пары чисел (m , t):

(-10, 9); (5, 6); (19, -10); (6, 10); (9, 9); (8, 10); (-11, 5); (6, 8); (7, 16).

Сколько было запусков, при которых программа напечатала «YES»?

Решение:

Программа напечатает «YES», если для пары чисел будут одновременно выполнены условия: переменная m меньше 8, переменная t меньше 10. Значит, было четыре запуска, при которых программа напечатала «YES». В качестве значений переменных m и t в этих случаях вводились следующие пары чисел: $(-10, 9)$; $(5, 6)$; $(-11, 5)$; $(6, 8)$.

Ответ: 4.

22

В программе `:=` обозначает оператор присваивания, знаки `+`, `-`, `·` и `/` — соответственно операции сложения, вычитания, умножения и деления. Правила выполнения операций и порядок действий соответствуют правилам арифметики.

Определите значение переменной b после выполнения алгоритма:

```
a := 7
b := 4
a := 2·a + 8·b
b := a/2·b
```

В ответе укажите одно целое число — значение переменной b .

Решение:

Разберём построчно данный алгоритм:

$a := 7$ — переменной a присваивается значение 7;

$b := 4$ — переменной b присваивается значение 4;

$a := 2 \cdot a + 8 \cdot b$ — переменной a присваивается значение $2 \cdot a + 8 \cdot b = 2 \cdot 7 + 8 \cdot 4 = 14 + 32 = 46$;

$b := a/2 \cdot b$ — переменной b присваивается значение $a / 2 \cdot b = 46 / 2 \cdot 4 = 92$.

Ответ: 92.

23

Ниже приведена программа, записанная на четырёх языках программирования.

Python

```

m = int(input())
n = int(input())
A = int(input())
if (m > A) or (n > 13):
    print(«YES»)
else:
    print(«NO»)

```

Pascal

```

var m,n,A: integer;
begin
  readln(m);
  readln(n);
  readln(A);
  if (m > A) or (n > 13)
  then
writeln ('YES')
  else
writeln ('NO')
end.

```

Алгоритмический язык

```

алг
нач
цел m, n, A
ввод m
ввод n
ввод A
если m > A или n > 13
  то вывод «YES»
  иначе вывод «NO»
все
кон

```

C++

```

#include <iostream>
using namespace std;
int main() {
  int m, n, A;
  cin >> m;
  cin >> n;
  cin >> A;
  if (m > A) or (n > 13)
    cout << «YES» << endl;
  else
    cout << «NO» << endl;
  return 0;
}

```

Было проведено девять запусков программы, при которых в качестве значений переменных m и n вводились следующие пары чисел:

(14, 2); (12, 12); (-11, 12); (3, -2); (-11, -10); (6, -5); (3, 8); (9, 10); (1, 14).

Укажите наименьшее целое значение параметра A , при котором для указанных входных данных программа напечатает «NO» восемь раз.

Решение:

Программа печатает «YES», **если** $m > A$ или $n > 13$, т. е. должно быть истинно хотя бы одно условие. В последнем случае программа напечатает «YES» независимо от A , т. к. $14 > 13$ — истина.

Программа напечатает «NO», если переменная n будет меньше или равна 13, а переменная m будет меньше или равна A .

Рассмотрим первое условие. Для выполнения запуска со значением «NO» восемь раз необходимо, чтобы первое условие было ложным во всех запусках (или во всех, кроме последнего).

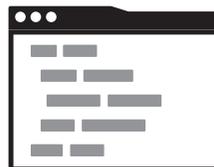
Заметим, что при $A = 1$ программа напечатает «NO» два раза, а при $A = 14$ — восемь раз. Таким образом, правильный ответ — 14.

Ответ: 14.



ЗАДАЧИ НА ЗАПИСЬ И АНАЛИЗ АЛГОРИТМОВ

Изучив основные элементы языков программирования, можно приступить к решению задач. Основным и в большинстве случаев необходимым этапом решения сложной задачи является оформление блок-схемы. Это позволяет значительно облегчить процесс составления программы. Рассмотрим случаи, наиболее часто встречающиеся на экзамене.





Практические задания

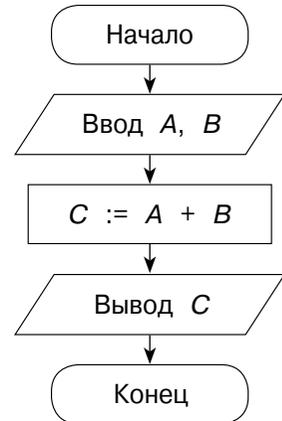
24 Ввести с клавиатуры два целых числа и найти их сумму.

Решение:

Начинаем решение с составления блок-схемы. Перед нами алгоритм с линейной структурой.

Далее, имея графическое представление задачи в виде блок-схемы, приступаем к написанию программы.

Представим решение на каждом из рассматриваемых в данном пособии языков.



Алгоритмический язык

```

алг суммаЧисел
  цел a, b, S
нач
  ввод a,b
  S=a+b
  вывод S
кон
  
```

Pascal

```

program qq;
var a, b, c: integer;
begin
  writeln ('Введите два целых числа');
  read ( a, b );
  c := a + b;
  writeln ( a, '+', b, '=', c );
end.
  
```

C++

```

#include <iostream>
using namespace std;
int main() {
  int a, b, c;
  cout << "Введите два целых числа\n";
  cin >> a >> b;
  c = a + b;
  std::cout << c << '\n';
  return 0;
}
  
```

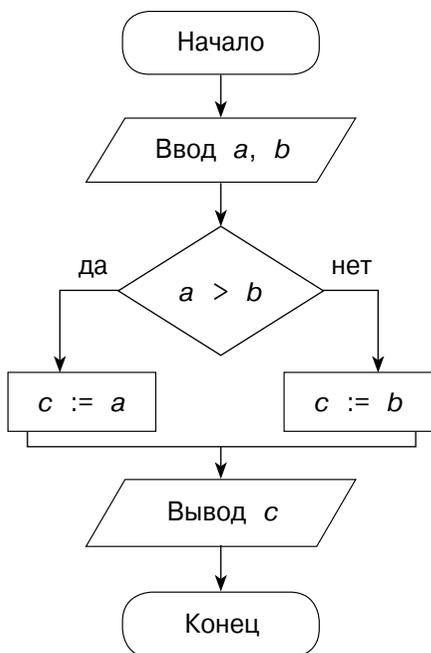
Python

```
print ( "Введите два числа:" )
a = int ( input() )
b = int ( input() )
c = a + b
print( a, "+", b, "=", c, sep="" )
```

25 Ввести с клавиатуры два неравных числа и вывести то, которое больше.

Решение:

Сделаем блок-схему. Перед нами алгоритм с условием, алгоритм выбора.



Алгоритмический язык

```
алг
цел a, b, c
нач
    ввод a, b
    если a>b
        то c:=a
        иначе c:=b
    все
    вывод c
кон
```

Pascal

```
program BD;
var A, B, C:integer;
begin
cls;
read(A, B);
if A>B then C:=A else C:=B;
write('Большее число ', C)
end.
```

C++

```
#include <iostream>
using namespace std;
int main() {
    int a, b, c;
    cout << "Введите два целых числа\n";
    cin >> a >> b;
    if (a > b) {
        c = a;
    } else {
        c = b;
    }
    cout << c << '\n';
    return 0;
}
```

Python

```
a=int (input () )
b=int (input () )
if a>b:
    c = a
else:
    c = b
print ("c=", c, sep=(' '))
```

- 26** Написать программу на школьном алгоритмическом языке по нахождению суммы квадратов чисел от 1 до N .

Решение:

алг Сумма квадратов (арг цел n , рез цел S)
дано | $n > 0$
надо | $S = 1*1 + 2*2 + 3*3 + \dots$
 + $n*n$
нач цел i
 ввод n ; $S:=0$
 нц для i **от** 1 **до** n
 $S:=S+i*i$
 кц
 вывод "S = ", S
кон

- 27** На четырёх языках записан алгоритм, который получает на вход число x и печатает два числа: a и b . Укажите наименьшее из таких чисел, при вводе которых алгоритм сначала печатает 18, а потом 0.

Решение:

Представим описание программы. Используем a , b , x — переменные целого типа (int). Выводим x . Присвоим $a=0$, $b=1$.

Python	Pascal
<pre>x=int(input()) a=0 b=1 while x>0: a=a+(x % 10) b=b*(x % 10) x=x//10 print(a) print(b)</pre>	<pre>var x,a, b:integer; begin readln(x); a:=0; b:=1; while x>0 do begin a:=a+(x mod 10); b:=b*(x mod 10); x:= x div 10; end; writeln(a);write(b); end.</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { int x, a, b; cin >> x; a = 0; b = 1; while (x>0) { a = a + (x % 10); b = b * (x % 10); x = x / 10; } cout << a << '\n' << b << '\n'; return 0; }</pre>	<pre>алг нач цел x, a, b ввод x a:=0; b:=1 нц пока x>0 a:=a+mod(x,10) b:=b*mod(x,10) x:=div(x,10) кц вывод a, нс b кон</pre>

Пока x больше 0, переменной a присвоим значение « a + остаток от деления x на 10», а переменной b присвоим значение « b , умноженное на остаток при делении на 10». Переменной x присвоим значение «целая часть x при делении на 10». Как только $x=0$ (т. е. не будет целой части при делении на 10), то цикл прекращается. Программа выведет на экран значения a и b .

Необходимо, чтобы вывелось 18, потом 0.

При первом проходе: $a=0+\text{mod}(x, 10)=\text{mod}(x, 10)$, $b=1*\text{mod}(x, 10)=\text{mod}(x, 10)$.

Искомое число не может быть однозначным, т. к. $a=18$. Если число многозначное, после первого прохода a и b будут равны, а x будет иметь значение « x без последней цифры». Однако по условию 18 и 0 не равны. Значит, следует обратиться к трёхзначным числам и найти среди них минимальное число, удовлетворяющее условию.

После второго прохода b должно обратиться в 0. Это значит, что средняя цифра числа — 0, а остаётся неизменным, x становится однозначным.

После третьего прохода b не меняется, т. к. любое число при умножении на 0 даёт 0, к a прибавляется первая цифра числа x .

Таким образом, a получается сложением трёх цифр, одно из которых — 0. По условию задачи $a=18$, что возможно тогда, когда $18 = 9 + 9$, т. е. оба числа равны 9. Таким образом, $x = 909$ — наименьшее.

Ответ: 909.

28 Определите, при каком наименьшем введённом значении переменной m программа выведет число 60. Для вашего удобства программа представлена на четырёх языках программирования.

Python

```
m = int(input())
n = 28
while m < 2023:
    m = m * 2
    n = n + 4
print(n)
```

Pascal

```
var m, n: integer;
begin
    readln(m);
    n := 28;
    while m < 2023 do
        begin
            m := m * 2;
            n := n + 4;
        end;
    writeln(n)
end.
```

Алгоритмический язык	C++
алг нач цел n, m ввод m $n := 28$ нц пока $m < 2023$ $m := m \cdot 2$ $n := n + 4$ кц вывод n кон	<pre>#include <iostream> using namespace std; int main() { int m; cin >> m; int n = 28; while (m < 2023) { m = m * 2; n = n + 4; } cout << n << endl; return 0; }</pre>

Решение:

Для того чтобы программа вывела число 60, к числу $N = 28$ необходимо прибавить 32, т. е. цикл должен выполняться $32 : 4 = 8$ раз. Поскольку в каждой итерации цикла число m умножается на 2, то итоговое значение $m = m \cdot 2^8$. Поделим 2023 на 2^8 и округлим до большего целого, получим 8. Значит, наименьшее возможное входное число m — это 8.

Ответ: 8.

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;
- ▲ результатом операции над массивами является массив;
- ▲ в одном массиве могут храниться и обрабатываться данные только одного типа.

■ Инициализация массива

▲ Алгоритмический язык

Объявление массивов

Массив целых чисел:

	Тип элемента	Имя	Начальный индекс	Конечный индекс
	↓	↓	↓	↓
целтаб		A	[1:5]	
цел		N=5		

Размер через константу:

целтаб A [1:N]

Ввод с клавиатуры:

```
нц для I от 1 до N
    вывод " A[" , i , "] = "
    ввод A [i]
кц
```

Обработка каждого элемента:

```
нц для I от 1 до N
    A[i] := A+1
кц
```

Вывод на экран:

```
    вывод " Массив A " , нс
нц для I от 1 до N
    вывод A [i] , " "
кц
```

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
```

▲ Python

Ввод массива с клавиатуры

Объявление массива:

```
Количество элементов в массиве
↓
N=10 # размер массива
A=[0]*N # выделили память для
↑ массива
Имя 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 экран

Как список:

```
print(A)
```

В строчку через пробел:

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

Пробел после вывода очередного числа

Вариант:

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

Разбить список на элементы

▲ Pascal

Объявление массива:

Имя
↓
var A: array [1..5] of integer;

Начальный индекс
↓
Конечный индекс
↓
Тип элемента
↓

Через константу:

```
const N=5;
var A: array [1..N] of integer;
```

Ввод с клавиатуры:

```
for i:=1 to N do begin
    write ('a[', i, ']=') ;
    read ( a[i] );
end;
```

Поэлементные операции:

```
for i:=1 to N do a[i]:=a[i]+1;
```

Вывод на экран:

```
writeln ('Массив A :');
for i:=1 to N do
    write (a[i] :4);
```

▲ 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]<< ' ';
```

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 массив не упорядочен, то для поиска определённого элемента необхо-

димо 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 по-

ложения элементов в заданном массиве согласно поставленным условиям. Рассмотрим наиболее часто используемые способы сортировки на примерах с применением школьного алгоритмического языка.

Сортировка обмена

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.

```

✓ алг Обменная_сортировка (арг
цел n, арг рез вещь таб A[1:n] )
  дано | A - массив размерности n
  надо | упорядочить массив по возрастанию
  нач
    цел i, j
  вещь Tmp
  нц для i от 2 до n
    нц для j от n до 1
      если A[j]<A[j-1]
        то Tmp:=A[j];
        A[j]:=A[j-1];
        A[j-1]:=Tmp
      все
    кц
  кц
кон

```

Сортировка прямым включением (вставкой)

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.

```

✓ алг Сортировка_вставкой
(арг цел n, арг рез вещь таб A[1:n] )
  дано | A - массив размерности n
  надо | упорядочить массив по возрастанию
  нач
    цел i, j
  вещь Tmp
  нц для i от 2 до n
    Tmp :=A[j]; j :=i-1;
    нц пока j>= 1 и A[j]>Tmp
      A[j+1] :=A[j]
      j :=i-1
    кц
    A[j+1] :=Tmp
  кц
кон

```

Сортировка выбором

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.

✓ **алг** Сортировка_выбором (**арг цел** n , **арг рез вещь таб** $A[1:n]$)
дано | A – массив размерности n
надо | упорядочить массив по возрастанию

нач
 цел i, k
вещ Min
нц для i **от** 1 **до** $n-1$
 $Min := A[j]; k := i$
 нц для j **от** $i+1$ **до** n
 если $Min > A[j]$
 то $Min := A[j]; k := j$
 все
 кц
 $A[k] := A[i]; A[i] := Min$

кц
кон



Практические задания

29 Дан фрагмент программы, обрабатывающий двухмерный массив A размером $n \times n$.

Python	Pascal
<pre>for i in range(1, n): for j in range(i+1, n+1): if A[i,1] < A[j,1]: k=A[i,1] A[i,1]=A[j,1] A[j,1]=k</pre>	<pre>for i:= 1 to n-1 do for j:=i+1 to n do if A[i,1] < A[j,1] then begin k:=A[i,1]; A[i,1]:=A[j,1]; A[j,1]:=k; end;</pre>
Алгоритмический язык	C++
<pre>нц для i от 1 до $n-1$ нц для j от $i+1$ до n если $A[i,1] < A[j,1]$ то $k:=A[i,1]$ $A[i,1]:=A[j,1]$ $A[j,1]:=k$ кц кц</pre>	<pre>for (int i = 1; i <= n-1; ++i) { for (int j = i+1; j <= n; ++j) { if (A[i][1] < A[j][1]) { k=A[i][1]; A[i][1]=A[j][1]; A[j][1]=k; } } }</pre>

Что происходит в данном фрагменте программы?

Решение:

В двойном цикле переменная i изменяется от 1 до $n-1$, а j — от 1 до n (на 1 шаг больше). Это означает, что с элементом $A[j, 1]$ будут по очереди сравниваться все элементы столбца 1, кроме последнего. Они выстраиваются от самого большого до самого маленького сверху вниз, т. е. по убыванию.

Рассмотрим данный фрагмент программы.

```
...k:=A[i,1];
A[i,1]:=A[j,1];
A[j,1]:=k;
```

Эта операция меняет местами два элемента первого столбца, значит, все изменения происходят только в нём. Таким образом, в программе упорядочивается первый столбец массива по убыванию.

30 Значения элементов двухмерного массива $A[1...100, 1...100]$ задаются с помощью следующего фрагмента программы.

Pascal

```
for i:=1 to 100 do
  for k:=1 to 100 do
    if i > k then
      A[i,k] := i
    else A[i,k] := -k;
```

Чему равна сумма элементов массива после выполнения фрагмента программы?

Решение:

Из данной программы получим массив, который для удобства можно представить в виде таблицы. Если рассмотреть симметричность таблицы, то относительно главной диагонали ячейки будут взаимно уничтожаться, потому что ниже диагонали все числа положительные, а выше — отрицательные.

-1	-2	-3	-4	...	-100
2	-2	-3	-4		-100
3	3	-3	-4		-100
4	4	4	-4		-100
...		
100	100	100	100	...	-100

Останется только найти сумму главной диагонали, в которой все числа отрицательные. Для этого, зная, что от 1 до 100 находится 100 чисел, сгруппируем их попарно, чтобы каждая пара в сумме давала 100.

Таких пар будет 49, и два числа (-50 и -100) останутся без пары:

$$-1 - 2 - 3 - 4 - 5 - \dots - 98 - 99 - 100 = -(1 + 99) - (2 + 98) - \dots - (49 + 51) - 50 - 100 = -100 \cdot 49 - 150 = -4900 - 150 = -5050.$$

31 Найдите сумму чётных элементов массива, используйте язык программирования Python.

Решение:

Чтобы определить чётность элемента, надо знать определение чётного числа: чётным является число, которое даёт остаток 0 при делении его на 2.

Способ 1

```

summa=0
for x in A:
    if x%2 ==0:
        summa+=x
print (summa)
    
```

Способ 2

```

B=[x for x in A
    if x%2 ==0]
print (sum (B))
    
```

32 Вычислите произведение чётных элементов массива. Запишите решение на языке программирования C++.

Решение:

Примечание. Напомним, что текст после символа // — комментарии, которые можно не вносить в решение.

```
#include <iostream>
using namespace std;
int func(int* x, int n) { // функция произведения
чётных элементов
    int p = 1; // начальное значение произведения
    for ( int i = 0; i<n; i++)
    {
        if (x[i] % 2 == 0) // чётность означает, что остаток
от деления на 2 равен 0
            p*= x[i]; // то же самое, что p = p * x[i];
    }
    return p;
}
// Главная функция
int main() {
    int a[5]; // объявлен массив a из 5 элементов
    int pr;
    // Ввод элементов массива
    for ( int i = 0; i<5; i++)
    {
        int x;
        cin >> x;
        a[i] = x; // a[i] — адрес i-го элемента массива
    }
    pr = func(a, 5); // вычисление произведения
    cout << pr << '\n';
    return 0;
}
```

33 Найдите минимальный элемент по его индексу. Напишите решение на языке программирования Pascal.

Решение:

Примечание. Напомним, что текст внутри скобок {} — комментарии, которые можно не вносить в решение.

```
var a: array[1..10] of integer; {ввод массива из 10 элементов типа integer}
i, min, iMin: integer;
begin
    randomize; {функция выбора произвольных элементов}
    for i:=1 to 10 do
    begin
        a[i]:=random(10); {интервал [0,9] — 10 элементов }
        write(a[i], ' '); {выводит элементы на экран}
    end;
    min=a[1]
    iMin=1
    for i := 2 to 10 do
        if a[i]<a[iMin] then {поиск минимального элемента}
        begin
            min := a[i];
            iMin := i;
        end;
    writeln('Минимальный элемент: A[' , iMin, ']=' , min); {выводит минимальный элемент на экран}
end.
```

34 На четырёх языках программирования записан рекурсивный алгоритм S.

Решение можно представить в виде дерева. Стрелками указано направление вывода чисел на экран.

При выполнении первого действия процедура $S(1)$ выведет число 1. Далее процедура $S(1)$ вызовет процедуру $S(n + 1)$, в результате выполнения которой на экране появится число $n + 1 = 2$. Процедура $S(2)$ вызовет процедуру $S(3)$, которая выведет на экран число 3 и вызовет процедуру $S(4)$, которая выведет на экран число 4 и вызовет процедуру $S(5)$, которая выведет на экран число 5.

Поскольку имеется условие, что если $n < 5$, то программа вернётся к процедуре $S(4)$, которая начнёт выполнять следующий шаг своего алгоритма, значит, обратится к процедуре $S(n + 3) = S(7)$. Процедура $S(7)$ выведет на экран число 7.

Далее управление вернётся к процедуре $S(3)$. Рассуждая аналогично, приходим к выводу, что процедура $S(3)$ дополнительно выведет на экран число 6, процедура $S(2)$ — число 5.

Последним действием процедуры $S(1)$ будет вызов процедуры $S(n + 3) = S(4)$, которая выведет на экран числа 4, 5, 7.

Таким образом, на экране будут числа 1, 2, 3, 4, 5, 7, 6, 5, 4, 5, 7. Их сумма равна 49.

Ответ: 49.

ИСПРАВЛЕНИЕ ОШИБОК В ПРОГРАММЕ

Существует три аспекта проверки программы:

- ▲ правильность;
- ▲ эффективность реализации;
- ▲ вычислительная сложность.

Эти аспекты дают ответ на вопрос, насколько хорошо и правильно работает алгоритм данной программы.

При написании программы часто возникают ошибки в написании служебного слова, постановке знаков и т. п. Существует опреде-

лённый порядок выявления таких недостатков.

Ошибки могут быть допущены на всех этапах решения задачи. К характерным ошибкам относятся:

- ▲ неправильная постановка задачи — верное решение неверно сформулированной задачи;
- ▲ неверный алгоритм — выбор алгоритма, приводящего к неточному, неэффективному решению задачи;
- ▲ ошибки при выполнении операций — слишком большое число

(переполнение), деление на ноль, извлечение квадратного корня из отрицательного числа и т. п.;

▲ ошибки анализа — неполный учёт ситуаций, которые могут возникнуть, логические ошибки;

▲ семантические ошибки — неправильный порядок выполнения операций;

▲ синтаксические ошибки — нарушение правил, определяемых языком программирования (например, пропуск двоеточия в некоторых языках);

▲ ошибки ввода/вывода — неверное считывание входных данных, задание форматов, отсутствие некоторых данных;

▲ ошибки в данных — неправильное определение возможного диапазона изменения 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).

Поиск ошибок в программе

1. **Просмотр.** Текст программы просматривается на предмет обнаружения ошибок и расхождений с алгоритмом. Для этого следует посмотреть 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

Если после тестирования были обнаружены какие-то неточности или ошибки в программе, то процесс можно перезапустить с этапа отлад-

ки. Затем провести тестирование снова. Этапы 4 и 5 следует проводить до полного устранения ошибок в программе.



Практические задания

35 Дано целое положительное число N . Необходимо найти и вывести количество и сумму цифр, кратных 3, в десятичной записи N . Если цифр, кратных 3, в числе нет, нужно вывести слово «нет».

Например, для $N = 6610$ нужно вывести числа: 3 (в числе три кратные цифры) и 12 ($6 + 6 + 0 = 12$), а для $N = 1457$ — слово «нет». Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Последовательно выполните следующее.

- 1) Напишите, что выведет эта программа при вводе $N = 3335$.
- 2) Приведите пример трёхзначного числа N , содержащего хотя бы одну цифру, кратную 3, при вводе которого программа выведет верный ответ.
- 3) Найдите в программе все ошибки (известно, что их не больше двух) и исправьте их. Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде. Достаточно указать ошибки и способ их исправления для одного языка программирования.

! Обратите внимание на то, что надо исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые.

Ниже эта программа приведена на четырёх языках программирования: Python, Pascal, C++, алгоритмическом языке.

Python	Pascal
<pre>n = int(input()) k = 0 s = 0 while n>10: d = n % 10 if d % 3 == 0: k += 1 s += d n = n // 10 if s > 0: print(k,s) else: print("нет")</pre>	<pre>var n, d, k, s: integer; begin readln(n); k := 0; s := 0; while n > 10 do begin d := n mod 10; if d mod 3 = 0 then begin k := k + 1; s := s + d end; n := n div 10 end; if s > 0 then write(k, ' ', s) else write('нет') end.</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { int n; cin >> n; int k = 0; int s = 0; while (n > 10) { int d = n % 10; if (d % 3 == 0) { ++k; s += d; } n /= 10; } if (s > 0) { cout << k << ' ' << s; } else { cout << "нет"; } return 0; }</pre>	<pre>алг нач цел n, d, k, s ввод n k := 0 s := 0 нц пока n > 10 d := mod(n,10) если mod(d,3)=0 то k := k + 1 s := s + d все n := div(n,10) кц если s > 0 то вывод k, ' ', s иначе вывод нет все кон</pre>

Решение:

- 1) При вводе $N = 3335$ программа выведет числа 2 и 6.
- 2) Программа выводит верный ответ, если первая цифра исходного числа не кратна 3 и в числе есть ненулевые цифры, кратные 3. Примеры таких трёхзначных чисел: 119, 261, 769.
- 3) Программа содержит две ошибки.

Ошибка 1. Неверное условие цикла. Первая цифра числа не проверяется, и, если она кратна 3, получается неверный результат.

Ошибка 2. Неверное условие наличия цифр, кратных 3. Вместо количества цифр проверяется их сумма. В результате в ситуации, когда в числе есть нули, но нет других цифр, кратных 3, будет выведен неверный ответ.

Пример исправления

Ошибка 1

Для каждого языка программирования нужно найти строку с ошибкой:

```
n > 10
```

Заменяем её исправленным вариантом:

```
n > 0
```

Ошибка 2

Ошибочная строка:

```
если s > 0 то
```

Исправленная строка:

```
если k > 0 то
```

- 36** Ниже на четырёх языках программирования записаны две рекурсивные функции: F и G .

Python

```
def F(n):
    if n > 2:
        return F(n-1)+ G(n-2)
    else: return 1
def G(n):
    if n > 2:
        return G(n-1) + F(n-2)
    else: return 1
```

Pascal	C++
<pre>function F(n: integer): integer; begin if n > 2 then F := F(n - 1) + G(n - 2) else F := 1; end; function G(n: integer): integer; begin if n > 2 then G := G(n - 1) + F(n - 2) else G := 1; end;</pre>	<pre>int F(int n) { if (n > 2) return F(n-1) + G(n-2); else return 1; } int G(int n) { if (n > 2) return G(n-1) + F(n-2); else return 1; }</pre>

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
  то
    знач := F(n - 1) + G(n - 2)
  иначе
    знач := 1
  все
кон
алг цел G(цел n)
нач
  если n > 2
  то
    знач := G(n - 1) + F(n - 2)
  иначе
    знач := 1
  все
кон
```

Чему будет равно значение, вычисленное при выполнении вызова F(7)?

Решение:

Рассмотрим, что выводит функция, начиная с 1.

В первой строке каждой программы определяем функцию F. В следующих строчках, начиная с если $n > 2$ (или `if n > 2`), видим условия для нахождения функции F в зависимости от параметра n. Если $n > 2$, то мы выводим значение $F(n) = F(n - 1) + G(n - 2)$, если же n меньше или равно 2, то выводим значение F, равное 1. Аналогично с функцией G(n): мы выводим $G(n) = G(n - 1) + F(n - 2)$, если $n > 2$, и если n меньше или равно 2, то $G(n) = 1$.

При вызове функции F(7) необходимо знать значения F(6) и G(5), поскольку $7 > 2$. Для этого нужно найти значения F(5) и G(4) и т. д., пока не будут известны все элементы.

$$F(7) = F(6) + G(5) = 8 + 5 = 13;$$

$$F(6) = F(5) + G(4) = 5 + 3 = 8;$$

$$F(5) = F(4) + G(3) = 3 + 2 = 5;$$

$$F(4) = G(2) + F(3) = 1 + 2 = 3;$$

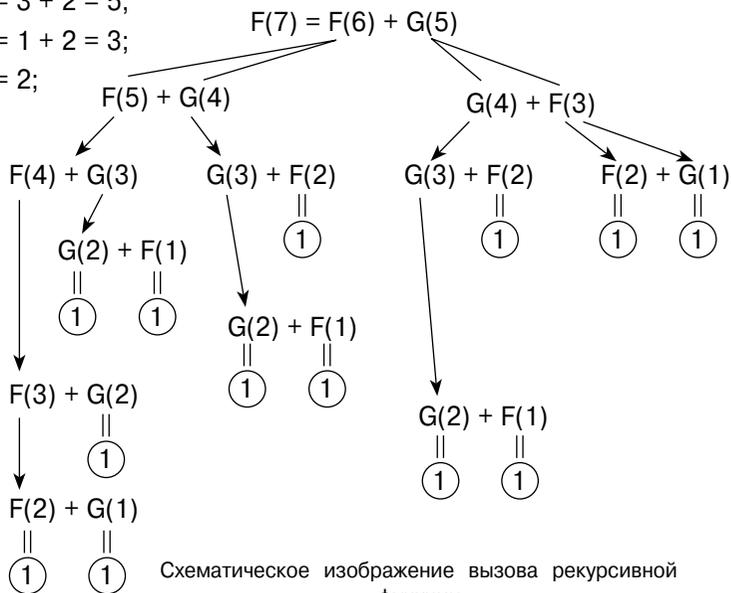
$$F(3) = G(1) + F(2) = 2;$$

$$F(1) = 1;$$

$$G(1) = 1;$$

$$G(2) = 1;$$

$$F(2) = 1.$$



Схематическое изображение вызова рекурсивной функции

Ответ: 13.

37 На обработку поступает натуральное число, не превышающее 10^9 . Нужно написать программу, которая выводит на

экран минимальную чётную цифру этого числа. Если в числе нет чётных цифр, требуется на экран вывести «Нет».

Алгоритмический язык	C++
<pre> алг нач цел N, digit, minDigit ввод N minDigit := mod(N,10) нц пока N > 0 digit := mod(N,10) если mod(digit, 2) = 0 то если digit < minDigit то minDigit := digit все все N := div(N,10) кц если minDigit = 0 то вывод "нет" иначе вывод minDigit все кон </pre>	<pre> #include <iostream> using namespace std; int main() { int N, digit, minDigit; cin >> N; minDigit = N % 10; while (N > 0) { digit = N % 10; if (digit % 2 == 0) if (digit < minDigit) minDigit = digit; N = N / 10; } if (minDigit == 0) cout << "нет" << endl; else cout << minDigit << endl; return 0; } </pre>
Python	Pascal
<pre> N = int(input()) minDigit = N % 10 while N > 0: digit = N % 10 if digit % 2 == 0: if digit < minDigit: minDigit = digit N = N // 10 if minDigit == 0: print("нет") else: print(minDigit) </pre>	<pre> var N,digit,minDigit: longint; begin readln(N); minDigit := N mod 10; while N > 0 do begin digit := N mod 10; if digit mod 2 = 0 then if digit < minDigit then minDigit := digit; N := N div 10; end; if minDigit = 0 then writeln('нет') else writeln(minDigit) end. </pre>

При написании программы были допущены ошибки. Известно, что в тексте программы можно исправить ровно две строки так, чтобы она стала работать правильно. Найдите допущенные программистом ошибки и исправьте их. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Решение:

Первая ошибка: неверная инициализация ответа (переменная `minDigit`).

Алгоритмический язык

Строка с ошибкой:
`minDigit := mod(N,10)`

Верное исправление:
`minDigit := 10`

C++

Строка с ошибкой:
`minDigit = N % 10;`

Верное исправление:
`minDigit = 10;`

Python

Строка с ошибкой:
`minDigit = N % 10`

Верное исправление:
`minDigit = 10`

Pascal

Строка с ошибкой:
`minDigit := N mod 10;`

Верное исправление:
`minDigit := 10;`

Вместо 10 может быть использовано любое целое число больше 8.

Вторая ошибка: неверная проверка отсутствия чётных цифр.

Алгоритмический язык

Строка с ошибкой:
если `minDigit = 0` **то**

Верное исправление:
если `minDigit = 10` **то**

C++

Строка с ошибкой:
`if (minDigit == 0)`

Верное исправление:
`if (minDigit == 10)`

Python

Строка с ошибкой:
`if minDigit == 0:`

Верное исправление:
`if minDigit == 10:`

Pascal

Строка с ошибкой:
`if minDigit = 0 then`

Верное исправление:
`if minDigit = 10 then`

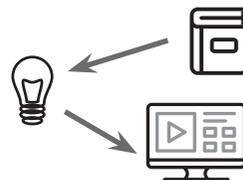
Вместо 10 может быть другое число больше 8, которое было положено в `minDigit` при исправлении первой ошибки, или проверка, что `minDigit > 8`.

ИНФОРМАЦИОННАЯ ДЕЯТЕЛЬНОСТЬ ЧЕЛОВЕКА



ЭТАПЫ РАЗВИТИЯ ИНФОРМАЦИОННОГО ОБЩЕСТВА

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



Этапы развития информационной деятельности

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.)

Исходя из этих этапов, можно утверждать, что появление и развитие новых знаний увеличивается

в геометрической прогрессии. Если первое удвоение общего количества знаний на планете произо-

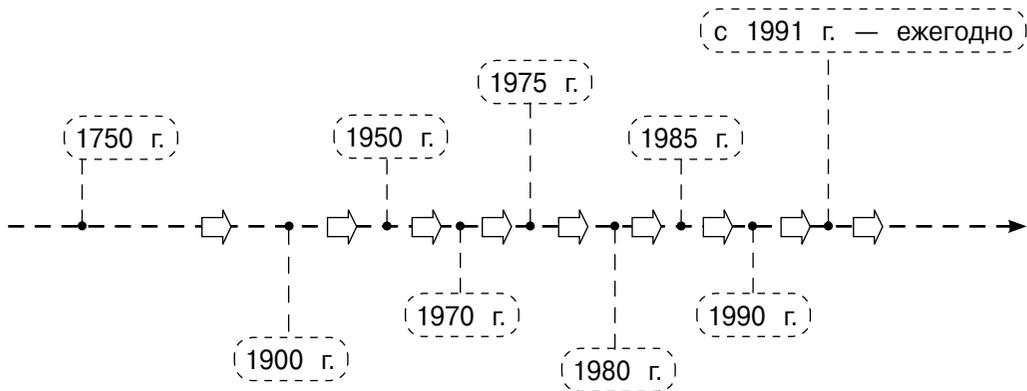
шло за период от рубежа нашей эры до 1750 г., то второе удвоение случилось уже за 150 лет, к началу XX столетия, третье — за 50 лет — к 1950 г. В дальнейшем объёмы знаний удваивались ещё более стремительными темпами: до 1970 г. — на протяжении 10 лет, после 1970 г. — каждые 5 лет, а с 1991 г. — ежегодно. Данный процесс можно наглядно представить схемой (см. внизу страницы), где стрелка ⇨ обозначает удвоение количества знаний.

Такое стремительное ускорение в развитии вызвало необходимость где-то хранить информацию и передавать её. Решению данной проблемы способствовало изобретение электронно-вычислительных машин (ЭВМ) и персональных компьютеров, создание телекоммуникационной инфраструктуры (баз данных и сетей разных типов).

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 задач, обработки огромных баз данных, работы телекоммуникационных сетей (Интернет).





ПРОФЕССИОНАЛЬНАЯ ИНФОРМАЦИОННАЯ ДЕЯТЕЛЬНОСТЬ

Профессиональная деятельность — вид трудовой деятельности человека, владеющего комплексом специальных теоретических знаний и практических навыков, которая является основным источником дохода.



Информационная деятельность человека — любая деятельность, связанная с процессами получения, преобразования, накопления и передачи информации.

Профессиональная деятельность всегда преследует определённую цель и предусматривает решение конкретных задач.

Цель профессиональной деятельности — предполагаемый результат, обеспечивающий развитие личности и её жизнедеятельность.

Задачи профессиональной деятельности — это этапы достижения цели. Например, целью информационной деятельности является обеспечение необходимой информацией определённых групп людей.

Основные цели информационной деятельности человека:

▲ налаживание сотрудничества с зарубежными информационными системами;

▲ удовлетворение спроса на информацию и потребности в ней;
▲ реализация информационного потенциала человека, предприятия или страны.

Основные задачи, решаемые в процессе профессиональной 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 надёжной защиты всех категорий информационных работников, прав собственности на информационную продукцию и производителей;

▲ ускорение процесса формирования и взаимодействия системы баз и банков данных.

ПРИМЕНЕНИЕ ТЕХНИЧЕСКИХ СРЕДСТВ И ИНФОРМАЦИОННЫХ РЕСУРСОВ В ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Профессии	Технические средства (ТС) / Информационные ресурсы (ИР)
Средства массовой информации	
Журналисты	ТС: телевидение, радио, телекоммуникации, компьютеры, компьютерные сети ИР: Интернет, электронная почта, библиотеки, архивы
Сфера связи: почта, телеграф, телефония	
Почтовые служащие, инженеры	ТС: телеграф, телефон, компьютерные сети ИР: базы данных
Наука	
Учёные	ТС: телекоммуникации, компьютеры, компьютерные сети ИР: библиотеки, архивы, базы данных, экспертные системы, Интернет
Техника	
Инженеры	ТС: телекоммуникации, компьютеры, компьютерные сети ИР: системы автоматизированного проектирования (САПР), библиотеки, патенты, базы данных, экспертные системы, Интернет
Медицина	
Медицинские работники	ТС: информационные системы, компьютеры, компьютерные сети ИР: базы данных, экспертные системы, архивы, Интернет
Финансы и экономика	
Бухгалтеры, экономисты	ТС: информационные системы, телекоммуникации, компьютеры, компьютерные сети ИР: базы данных, экспертные системы, архивы, Интернет
Управление	
Менеджеры	ТС: информационные системы, телекоммуникации, компьютеры, компьютерные сети ИР: базы данных, экспертные системы

Профессии	Технические средства (ТС) / Информационные ресурсы (ИР)
Образование	
Преподаватели	ТС: информационные системы, телекоммуникации, компьютеры, компьютерные сети ИР: библиотеки, Интернет
Искусство	
Писатели, художники, музыканты, дизайнеры	ТС: компьютеры, устройства ввода и вывода информации, системы мультимедиа, аудио- и видеосистемы, телекоммуникации, компьютерные сети ИР: галереи, библиотеки, музеи, Интернет

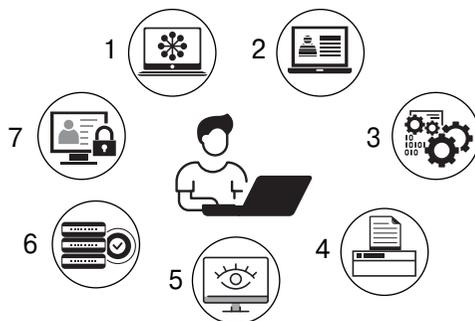
СИСТЕМНОЕ АДМИНИСТРИРОВАНИЕ

В настоящее время технические средства, в том числе компьютеры, планшеты и др., находятся на рабочем месте у специалиста любой профессии. 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 (на рисунке — 1);

- ▲ описание информационной среды (2);
- ▲ настройка операционной среды (3);
- ▲ настройка сетевой печати (4);
- ▲ мониторинг сети и управление сетевыми ресурсами (5);
- ▲ архивирование и восстановление данных сети (6);
- ▲ обеспечение безопасности (7).



Инсталляция — процесс установки программного обеспечения на компьютер, во время которого админи-

стратор отвечает на ряд вопросов в диалоговом режиме.

Описание информационной среды:

- ▶ создание каталогов пользователей;
- ▶ назначение прав доступа по отношению к каталогам и файлам;
- ▶ назначение атрибутов каталогам и файлам.

Настройка операционной среды:

- ▶ инсталляция конфигурационных файлов рабочих станций;
- ▶ разработка системных и пользовательских процедур подключения к сети;
- ▶ управление рабочими станциями.

Настройка сетевой печати предполагает описание:

- ▶ объектов печати (очередей, принтеров, серверов печати);
- ▶ форм печати (размера бумаги);
- ▶ конфигураций заданий на печать.

Мониторинг сети — оценка её характеристик производительности и надёжности. С помощью специальных программных пакетов можно реализовать:

- ▶ 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 очень 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

Интернет может быть как полезным помощником в работе различных организаций, так и средством, оказывающим отрицательное влияние

на работу сотрудников. Кроме того, Интернет — платный ресурс, потребление информации из которого требует средств.

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.

Чтобы избежать описанных выше проблем, следует предпринимать определённые меры.

▲ **Вести учёт потребления интернет-ресурсов.** Знание сотрудников о том, что каждое их действие в глобальной сети контролируется,

уменьшит потребление трафика и предотвратит нецелевое расходование средств учреждения. При необходимости можно без труда узнать, какие именно ресурсы посещаются пользователями в рабочее время.

▲ **Установить управление доступом в Интернет.** В данном случае указывается, когда, на какие ресурсы и кто из сотрудников может иметь доступ. Это позволяет сократить потери рабочего времени и повысить безопасность.

▲ **Установить защиту сети** от утечки информации, атак хакеров и сетевых вирусных программ.



ИНФОРМАЦИОННЫЕ РЕСУРСЫ

Традиционными видами общественных ресурсов являются материальные, сырьевые (природные), энергетические, трудовые, финансовые. В дополнение к этому в связи с информатизацией общества в современном мире появляются также информационные ресурсы.



Всякий ресурс, кроме информационного, после использования исчезает. Информационный же может изменяться, дополняться, становиться источником ещё большего знания, поэтому со временем значимость

информационных ресурсов возрастает.

В законе РФ «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 не-

сколько 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, ИХ КЛАССИФИКАЦИЯ

Национальные информационные ресурсы — информационные ресурсы, созданные и широко используемые в конкретном государстве. Понятие сформировалось около 25 лет назад в ответ на растущую зависимость развитых стран от объёмов информации, уровня развития средств её передачи и обработки.

Один из вариантов классификации информационных ресурсов — по форме представления и по назначению. Ресурсы Интернета можно разделить по этой классификации следующим образом: сервисная информация, библиографическая информация, материалы телеконференций, программное обеспечение и т. д.

Рассмотрим классификацию по источникам формирования информационных ресурсов.

- ▲ Библиотечные ресурсы.
- ▲ Архивные ресурсы.
- ▲ Научно-технические ресурсы.
- ▲ Правовая информация.

▲ Информация государственных (властных) структур.

▲ Отраслевая информация.

▲ Финансовая и экономическая информация.

▲ Информация о природных ресурсах.

▲ Информация предприятий и учреждений.

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.

Архивы хранят материалы, связанные с историей и культурой страны. Объёмы материалов огромны и обычно накапливаются быстрее, чем их удаётся обрабатывать.

Во всех развитых странах существуют специализированные системы

научно-технической информации.

Они включают многочисленные специальные издания, патентные службы и т. д. Информация такого рода часто является дорогостоящим товаром.

Своды законов, кодексы, нормативные акты, другие виды **правовой**

информации — основа функционирования любого государства.

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.

РЕСУРСОСБЕРЕЖЕНИЕ

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

В настоящее время вопросы ресурсосбережения приобретают особую актуальность. Ресурсосбережение рассматривается в узком смысле как мероприятия по изысканию резервов на основе снижения отходов и потерь. Сущность ресурсосберегающей деятельности заключается в комплексном использовании ресурсов, максимальном устранении всех видов потерь, как можно более полном вовлечении в хозяйственный оборот вторичных материальных и энергетических ресурсов. Центральными звеньями ресурсосбережения являются экономика, техника, технология и экология, поскольку ресурсосберегающий подход предполагает реализацию

целого комплекса задач, охватывающих эти четыре области знаний.

▲ **Экономическая задача.** Определение эффективных форм организации производства, постоянный учёт наличия, движения и расходования ресурсов, управление затратами, внедрение прогрессивных стимулов экономии ресурсов, политики ценообразования и сбыта.

▲ **Техническая задача.** Научно обоснованный выбор ресурсоэкономичных технических средств на стадиях производства и эксплуатации с оптимальными показателями долговечности, безотказности, ремонтпригодности и сохраняемости.

▲ **Технологическая задача.** Разработка безотходных и малооперационных технологий, обеспечивающих при минимальном потреблении ресурсов формирование требуемых качественных характеристик производимой продукции.

▲ **Экологическая задача.** Установление гармоничного взаимодействия агропромышленного производства с окружающей средой на основе

восстановления почвенного плодородия, энергоресурсов, водного баланса и минеральных ресурсов.

РЫНОК ИНФОРМАЦИОННЫХ РЕСУРСОВ И УСЛУГ

Обилие информационных ресурсов и возможность их представления в цифровом виде привели к появлению развитого, стремительно расширяющегося рынка. 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 потреби-

тели (покупатели). В качестве товара может 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.



К информационной сфере относятся:

▲ научно-информационные центры;

▲ информационные сети;

▲ библиотеки и архивы;

▲ средства массовой информации;

▲ в целом индустрия создания, хранения, обработки и распространения информации.

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

Структура экономической информационной системы

- Информационное обеспечение
- Техническое обеспечение
- Лингвистическое обеспечение
- Организационное обеспечение
- Правовое обеспечение

Информационная экономика как образовательная дисциплина изучает

закономерности использования информационных факторов как компонентов экономической системы. Информационная экономика базируется на инновационных экономических процессах.

Основные задачи информационной экономики решаются на макро- и микроуровне.

▲ **Макроуровень.** Происходит выделение наиболее перспективной отрасли для развития её информационной сферы, что определяется готовностью отрасли к новым переменам и интенсивному развитию.

▲ **Микроуровень.** На местах происходит создание алгоритмов, которые позволяют получить средства и возможности из микроуровня на развитие информационных ресурсов, что обеспечивает появление новых рабочих мест.

Стадии развития информационной экономики

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

За последние десятилетия в условиях современной экономики произошло возникновение и развитие информационного бизнеса, ставшего частью современной экономической информационной системы.

Информационный бизнес — направленная на получение дохода деятельность в сфере создания и коммерческого распространения информационных продуктов, технологий и услуг, т. е. бизнес в информационной сфере. В качестве примеров такого бизнеса можно выделить:

- ▲ осуществление электронной коммерции (услуги Интернета, телекоммуникации и т. п.);
- ▲ продвижение соответствующей продукции и услуг в сфере по-

требления (оптовая и розничная торговля в информационной сфере, посредничество, кредитование, консультационные и маркетинговые услуги, страхование средств вычислительной техники и т. п.).

К достоинствам данного вида бизнеса следует отнести отсутствие материального продукта: его не нужно хранить и доставлять покупателям. Основным недостатком считается большое количество конкурентов. По мере развития информационного бизнеса возникает необходимость его регулирования со стороны государства, которое контролирует экономические процессы и при необходимости вмешивается в их течение, в том числе на законодательном уровне.



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, можно разделить на несколько групп:

- ▲ проблемы, связанные с разработкой морального кодекса для IT-профессионалов и обычных поль-

зователей, чья работа связана с компьютерной техникой;

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

▲ проблемы, связанные с компьютерными преступлениями (преимущественно правовая сфера).

Понятие **сетевой этикет** появилось с возникновением первой сетевой услуги — электронной почты. Важно соблюдать правила сетевого этике-

та, чтобы не попасть в неприятную ситуацию или не нарушить правовые нормы, а также чтобы не создавать 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

Всегда заполняйте поле «Тема» в своём письме

Цитируйте отрывки письма, если это необходимо для ответа

Используйте прописные буквы только по необходимости

Не вставляйте в электронное письмо файлы большого объёма

Используйте подпись, в которой содержатся данные об авторе письма

Используйте смайлики только в неофициальной корреспонденции

Правила этикета для общения в чате, форуме, телеконференции

Будьте тактичным и корректным в своих высказываниях

Выбирайте себе псевдоним (или ник), не оскорбляющий других участников чата

Обращаясь к кому-либо, пишите его ник в начале вашей фразы

Не используйте много прописных букв и восклицательных знаков

Не злоупотребляйте смайликами

Уходя из чата, не забудьте попрощаться с собеседниками



ИНФОРМАЦИОННОЕ ПРАВО

С ростом количества информационных ресурсов появилась необходимость в их охране и защите. В последнее время увеличивается число компьютерных преступлений, связанных с хищениями конфиденциальной информации и материальных средств.



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

В настоящее время происходит быстрое накопление информации, информационный обмен, информационное взаимодействие как в пределах одного государства, так и на

международном уровне, поэтому необходимость правового регулирования выходит на передний план.

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.

Защита информации — деятельность, которая направлена на пре-

дотвращение 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, соблюдение коммерческой тайны персоналом).

Юридические средства защиты информации — совокупность законов и правовых актов, регулирующих информационную деятельность.

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

Перечислим названия и основные положения правовых актов, регулирующих информационную деятельность.

▲ Закон «О правовой охране программ для ЭВМ и баз данных» регламентирует юридические вопросы, связанные с авторскими правами на программные продукты и базы данных.

▲ Закон «Об авторском праве и смежных правах» разделяет следующие права на информационные ресурсы:

- **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.

▲ Закон «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 собствен-

нику предоставляется право самостоятельно в пределах своей компетенции устанавливать режим защиты информационных ресурсов и доступа к ним.

▲ В Уголовном кодексе РФ имеется раздел «Преступления в сфере компьютерной информации», который содержит:

- ст. 272 «О неправомерном доступе к компьютерной информации»;
- ст. 273 «Создание, использование и распространение вредоносных программ для ЭВМ»;
- ст. 274 «Нарушение правил эксплуатации ЭВМ, систем ЭВМ или сети ЭВМ».

Данный раздел предусматривает наказание:

- ▲ за неправомерный доступ к компьютерной информации;
- ▲ создание, использование и распространение вредоносных программ для ЭВМ;
- ▲ умышленное нарушение правил эксплуатации ЭВМ и их сетей.

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. Эти средства обеспечивают:

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



ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Информационная безопасность — совокупность мер по защите информационной среды общества и человека.



Информационная безопасность — механизм, который обеспечивает:

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

Термин «информационная безопасность» применим к деятельности не только человека, но и компаний. С развитием информационных

технологий и компьютеризацией экономики безопасность становится одним из важнейших вопросов в деятельности компании, поскольку информация — один из самых ценных активов любого предприятия, и она должна быть защищена надлежащим образом.

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

Угрозы безопасности в информационной сфере

Естественные: катаклизмы, не зависящие от человека: пожары, ураганы, наводнения, удары молнии и т. д.

Искусственные:

- непреднамеренные (совершаются людьми по неосторожности или незнанию);
- преднамеренные (хакерские атаки, в том числе компьютерные вирусы, противоправные действия конкурентов 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;
- ▲ неконтрольное распространение.

Достаточно часто источником подобных нежелательных воздействий являются 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 вычислительные сети с целью нарушения работы про-

грамм, порчи файлов и каталогов, создания всевозможных помех в работе компьютера. Программ-вирусов в настоящее время существует огромное количество. Они постоянно видоизменяются, дополняются и адаптируются под текущее программное обеспечение.

Меры профилактики воздействия вирусных программ:

- ▲ резервное копирование (создание копий файлов и системных областей жёстких дисков);
- ▲ исключение использования случайных и неизвестных программ;
- ▲ ограничение доступа к информации, в частности защита носителя информации (диска, флеш-карты) во время копирования файлов;
- ▲ использование специальных антивирусных программ.



Антивирусная программа — специализированная программа для обнаружения и блокировки компьютерных

вирусов, нежелательных вредоносных программ и восстановления заражённых (модифицированных) такими

программами файлов. В функции средств антивирусной защиты входит также профилактика — предотвращение заражения файлов или

операционной системы вредоносным кодом.

✓ Kaspersky, Dr.Web, Avast, NOD32, AVG.

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И МАШИННОЕ ОБУЧЕНИЕ

Человечество постоянно стремится создавать новые способы автоматизации различных видов деятельности, в итоге значительную часть работы в современном мире выполняют всевозможные механизмы, принимая на себя в том числе самые сложные и ответственные обязанности, которые ранее мог исполнять только человек.

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.

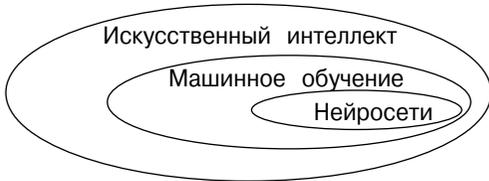
Искусственный интеллект — способность компьютера обучаться, принимать решения и выполнять действия, свойственные человеческому интеллекту.

Машинное обучение — технология искусственного интеллекта, которая предоставляет вычислительным системам возможность автоматически учиться и совершенствоваться на основе опыта без явного программирования.

Искусственные нейронные сети — математические модели, созданные по аналогии с биологическими нейронными сетями. ИНС способны моделировать и обрабатывать нелинейные отношения между входными и выходными сигналами.

Область применения искусственного интеллекта пересекается со многими другими областями, включая математику, статистику, теорию вероятностей, физику, обработку сигналов, машинное обучение, компьютерное зрение, психологию, лингвистику, науку о мозге и др. ИИ является мощным средством обработки данных и может находить решения сложных задач быстрее, чем традиционные алгоритмы, написанные программистами.

Недостатком искусственного интеллекта является этическая сторона прозрачности данных (информации): ИИ не даёт возможности изучить причины выбора им того или иного решения. С точки зрения структурной организации машинное обучение является разделом искусственного интеллекта, нейросети — одним из видов машинного обучения.



Схематическое представление связи искусственного интеллекта, машинного обучения и нейросетей

В настоящее время машинное обучение неспособно полностью заменить умственный труд, и в ближайшем будущем это вряд ли произойдёт, но ведётся активная работа в данном направлении.

С одной стороны, машинное обучение — дисциплина, находящаяся на пересечении математической статистики, численных методов оптимизации, теории вероятностей, а также дискретного анализа. С помощью её методов происходит решение задачи извлечения знаний из данных, которой занимается ещё только формирующаяся область — интеллектуальный анализ данных (*DataMining*).

С другой стороны, машинное обучение нацелено на создание систем, способных адаптироваться к решению различных задач без явного кодирования алгоритма, т. е. систем, способных самообучаться, а также улучшать показатели своей работы с помощью 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.

Предположим, вы хотите сдать экзамен по информатике и будете готовиться самостоятельно, используя множество видов литературы и задачников. Сначала вам нужно найти все темы, которые будут представлены на экзамене, и освоить их, а также прорешать на каждую тему множество задач, проверить эти решения и ответы. Компьютерная программа изучит ваши начальные знания и навыки, предложит вам определённый тест, найдёт в них пробелы и научится предлагать те задачи, в которых у вас имеются проблемы, а также сможет рекомендовать почитать нужную теорию. Для каждого человека будет разработано индивидуальное направление в изучении информатики. Но для того чтобы создать подобную программу, необходимо вначале её обучить, заложив нужные параметры: данные, характеристики (свойства, признаки) и алгоритм.

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 значительному 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. Для обучения машины часто используют «учителя». В качестве него

выступает обычно специалист по машинному обучению (программист), реже — специально созданная обучающая программа. Обучение без «учителя» заключается в том, что машине предлагается множество данных, которые она должна отсортировать, предварительно выявив некие критерии сортировки.

Практическое применение искусственного интеллекта

Банковская сфера: 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



ЭКСПЛУАТАЦИЯ КОМПЬЮТЕРНОГО РАБОЧЕГО МЕСТА

Правильная организация рабочего места чрезвычайно важна для обеспечения комфортной работы и предотвращения вреда здоровью.



Оборудование рабочего места влияет на органы опорно-двигательной системы.

Правильная рабочая поза позволяет избежать перенапряжения мышц, способствует лучшему кровотоку и дыханию. При работе, связанной с использованием оборудования в компьютерном классе и в домашних условиях, необходимо соблюдать следующие правила.

▲ Сидеть следует прямо (не сутулясь), опираясь спиной о спинку кресла. Прогнать спину в поясничном отделе нужно не назад, а, наоборот, немного вперёд.

▲ Подберите удобный стул: колени должны быть на уровне бёдер или немного ниже. При таком положении ног не возникает напряжения мышц.

▲ Нельзя скрещивать ноги, класть ногу на ногу — это нарушает циркуляцию крови из-за сдавливания сосудов. Лучше держать обе стопы на подставке или полу.

▲ Не располагайте рядом с монитором блестящие и отражающие свет предметы, не размещайте монитор напротив окна, чтобы избежать попадания солнечных лучей на монитор.

▲ Необходимо сохранять прямой угол (90°) в области локтевых, тазобедренных и голеностопных суставов.

▲ Экран монитора должен находиться на расстоянии 60—70 см от глаз пользователя, но не ближе 50 см с учётом размера символов, а верх монитора — располагаться немного выше уровня глаз. Поверхность экрана должна быть чистой и без световых бликов: периодически протирайте монитор от пыли и грязи.

▲ При работе с мышью рука не должна оставаться на весу.

▲ При работе за компьютером важно располагать клавиатуру и экран монитора непосредственно перед собой.

Справочное издание
анықтамалық баспа

Для старшего школьного возраста
мектеп жасындағы ересек балаларға арналған

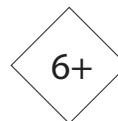


НАГЛЯДНЫЙ СПРАВОЧНИК ДЛЯ ПОДГОТОВКИ К ОГЭ И ЕГЭ

Тимофеева Елена Викторовна

ИНФОРМАТИКА

(орыс тілінде)



Ответственный редактор *А. Жилинская*
Ведущий редактор *Т. Судакова*
Выпускающий редактор *А. Проценко*
Художественный редактор *Е. Брынчик*

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

Во внутреннем оформлении
использованы иллюстрации:
Alexandr III, alexdndz, Anton Shaparenko, Becris,
davooda, eveleen, Gazlast, iconvectorstock, Jo-
vanovic Dejan, LineTale, phipatbig, puruan, r2dpr,
Sir.Vector, stockakia, VasutinSergey, VikiVector /
Shutterstock.com
Используется по лицензии от Shutterstock.com

В коллаже на обложке и титуле
использованы иллюстрации:
niiavko, Iuliia Shepeleva / Shutterstock.com
Используется по лицензии от Shutterstock.com

ООО «Издательство «Эксмо»
123308, 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
Өндірууі: «ЭКМО» АҚБ Баспасы,
123308, Ресей, қала Мәскеу, Зорге көшесі, 1 үй, 1 нұмарат, 20 қабағ, офис 2013 ж.
Тел.: 8 (495) 411-68-86.
Home page: www.eksmo.ru E-mail: info@eksmo.ru
Тауар белгісі: «Эксмо»
Интернет-магазин: www.book24.kz
Интернет-магазин: www.book24.kz
Интернет-дүкен: www.book24.kz
Импортер в Республику Казахстан ТОО «РДЦ-Алматы».
Қазақстан Республикасындағы импорттаушы «РДЦ-Алматы» ЖШС.
Дистрибутор и представитель по приему претензий на продукцию,
в Республике Казахстан: ТОО «РДЦ-Алматы»
Қазақстан Республикасында дистрибутор және өнім бойынша арыз-талаптарды
қабылдаушының өкілі «РДЦ-Алматы» ЖШС.
Алматы қ., Домбровский көш., 3-а, литер Б, офис 1.
Тел.: 8 (727) 251-59-90/91/92; E-mail: RDC-Almaty@eksmo.kz
Өнімнің жарамдылық мерзімі шектелмеген.
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
Өндірген мемлекет: Ресей. Сертификация қарастырылған

В электронном виде книги издательства вы можете
купить на www.litres.ru

ЛитРес:
один клик до книг



ЧИТАЙ
ГОРОД

ISBN 978-5-04-179033-2



9 785041 790332 >

ПРИСОЕДИНЯЙТЕСЬ К НАМ!



eksmo.ru

Мы в соцсетях:

eksmo
 eksmo.ru

Дата изготовления / Подписано
в печать 13.01.2023. Формат 70x90¹/₁₆.
Печать офсетная. Бумага офсетная
НЕ ПУХЛАЯ. Усл. печ. л. 22, 17.
Тираж экз. Заказ

book 24.ru

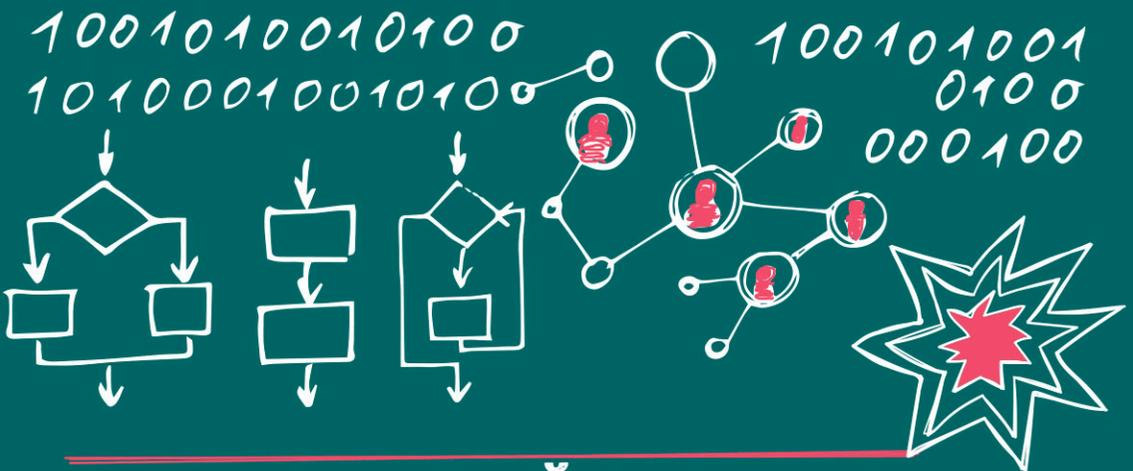
Официальный
интернет-магазин
издательской группы
“ЭКМО-АСТ”

НАГЛЯДНЫЙ СПРАВОЧНИК ДЛЯ ПОДГОТОВКИ К ОГЭ и ЕГЭ

ИНФОРМАТИКА



ВЕСЬ ШКОЛЬНЫЙ КУРС В ДОСТУПНОМ
ИЗЛОЖЕНИИ С НАГЛЯДНЫМИ ПРИМЕРАМИ
И ЗАДАНИЯМИ ДЛЯ ТРЕНИРОВКИ.



С ПОМОЩЬЮ ЭТОЙ КНИГИ ВЫ:

-  без проблем усвоите даже сложные для понимания темы;
-  проверите и закрепите свои знания;
-  сэкономите время и получите отличный результат.

ISBN 978-5-04-179033-2



9 785041 790332 >

