

## Глава 2

# ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ<sup>1)</sup>

*Из курса информатики основной школы вам известно:*

- о позиционных и непозиционных системах счисления;
- понятия «основание системы счисления», «алфавит системы счисления»;
- как осуществляется перевод натурального числа из десятичной системы счисления в двоичную и из двоичной в десятичную;
- о восьмеричной и шестнадцатеричной системах счисления;
- как осуществляется перевод натурального числа из десятичной системы в восьмеричную, шестнадцатеричную и обратно;
- как перевести натуральное число из двоичной системы счисления в восьмеричную и шестнадцатеричную и обратно;
- об общих подходах к кодированию текстовой информации;
- о кодовых таблицах ASCII и Unicode, используемых для кодирования текстовой информации;
- о кодировании цвета и цветовых моделях RGB и CMYK, HSB и CMY;
- о глубине цвета;
- общее представление о растровой и векторной графике;
- о кодировании звука как одного из видов информации;
- об общих принципах кодирования звука.

## 2.1

### Различные системы счисления для представления данных

*Изучив эту тему, вы узнаете:*

- о позиционных системах счисления с произвольным основанием;
- как перевести вещественное число в различные системы счисления;
- как осуществляется перевод чисел из одной системы счисления в другую.

<sup>1)</sup> Примеры из главы 2 приведены в приложении 2 в конце части 2 учебника и в электронном приложении в авторской мастерской Н. В. Макаровой на методическом сайте издательства [metodist.Lbz.ru](http://metodist.Lbz.ru).

### Объектный подход к понятию «система счисления»

Давайте разберёмся, почему систему счисления называют системой. Во-первых, любая система состоит из элементов, каковыми в данном случае являются числа. Элементарными, неделимыми элементами системы счисления являются цифры — знаки алфавита. Во-вторых, должен выполняться системный принцип целостности, т. е. система счисления должна иметь характеристики, относящиеся к системе в целом, а не к отдельным её элементам. Этими характеристиками являются алфавит и основание системы счисления. Элементы системы должны находиться во взаимосвязях между собой. Такие взаимосвязи устанавливают правила образования чисел из цифр, правила выполнения действий над числами.

### Представление вещественного числа в системе счисления с любым основанием

Для записи чисел можно использовать систему счисления практически с любым основанием  $p > 1$ . Если основание  $p \leq 10$ , то для обозначения чисел используются арабские цифры от 0 до  $p - 1$ . Если  $p > 10$ , то десяти арабских цифр уже недостаточно, и тогда используют буквы латинского (английского) алфавита.

Для представления чисел в компьютере используются три системы счисления:

- *двоичная* (основание  $p = 2$ , алфавит — 0, 1). Цифры двоичного алфавита называются **битами** (от англ. *bit* — binary digit, двоичная цифра);
- *восьмеричная* (основание  $p = 8$ , алфавит — 0, 1, 2, ..., 7);
- *шестнадцатеричная* (основание  $p = 16$ , алфавит — 0, 1, 2, ..., 9 и буквы — А, В, С, D, Е, F, заменяющие числа 10, 11, 12, 13, 14, 15 соответственно).

Название системы счисления указывает на её основание и алфавит. Так, например, алфавит пятеричной системы счисления содержит 5 знаков (цифр): 0, 1, 2, 3, 4, а алфавит двадцатеричной системы — 20 знаков: десять арабских цифр и десять латинских букв.

Правило образования чисел в любой позиционной системе счисления едино и связано с разложением числа по степеням основания. В общем случае любое вещественное число  $N$  в позиционной системе счисления можно представить в следующем виде:

$$N_p = a_k \cdot p^k + a_{k-1} \cdot p^{k-1} + \dots + a_1 \cdot p^1 + a_0 \cdot p^0 + a_{-1} \cdot p^{-1} + \dots + a_{-n} \cdot p^{-n},$$

где  $p$  — основание системы счисления;  $k + 1$  — количество разрядов в целой части числа;  $n$  — количество разрядов в дробной части числа;  $a_i$  — цифры (символы алфавита) используемой системы счисления. Нижние индексы ( $i = -n, \dots, -1, 0, 1, \dots, k$ ) определяют разряд цифры в числе:

- положительные значения индексов — для целой части числа;
- отрицательные значения индексов — для дробной части числа.

С учётом этих обозначений запись числа  $N$  в любой позиционной системе счисления с основанием  $p$  имеет вид:

$$(a_k a_{k-1}, \dots, a_1 a_0, a_{-1} a_{-2}, \dots, a_{-n})_p.$$

По сути, запись числа — это сокращённое обозначение разложения его по степеням основания. При одновременном использовании нескольких систем счисления принято указывать в виде индекса основание системы счисления, в которой записано число.

**Примеры 2.1, 2.2.** См. приложение 2.



Используя разложение числа по степеням основания, можно перевести число из любой системы счисления в десятичную. Для этого надо вычислить сумму всех слагаемых.

**Пример 2.3.** См. приложение 2.



## Перевод вещественного числа из десятичной системы счисления в другую позиционную систему

Представим вещественное десятичное число в общем виде  $N, M$ . Здесь  $N$  — целая часть числа, а  $M$  — дробная часть. Для перевода десятичного числа в позиционную систему счисления с основанием  $p$  необходимо воспользоваться двумя правилами: одно определяет алгоритм перевода целой части числа, а другое — дробной части.

*Алгоритм перевода целой части числа:*

1. Целую часть числа ( $N$ ) разделить нацело на  $p$  с остатком. Записать частное и остаток.
2. Если частное больше  $p$ , разделить его на  $p$  и записать новое частное и остаток.
3. Повторять п. 2 до тех пор, пока частное не станет меньше  $p$ .  
В новой системе счисления целая часть будет записываться из последнего частного, которое будет цифрой старшего разряда,

и всех остатков, записанных в порядке, обратном порядку их получения.

*Алгоритм перевода дробной части числа:*

1. Дробную часть числа ( $M$ ) умножить на основание  $p$ .
2. Записать целую часть полученного числа в позицию с номером  $-1$ .
3. Выделить дробную часть полученного числа и умножить на  $p$ .
4. Выделить целую часть числа и записать её в следующий справа разряд дробной части.
5. Выполнять пункты 3–4 до тех пор, пока в дробной части не будет получен ноль или не будет достигнута требуемая точность, например 5 знаков после запятой.

Заметим, что при переводе числа из десятичной системы счисления в другую систему может получиться число с бесконечной дробной частью.



**Примеры 2.4, 2.5.** См. приложение 2.

## Родственные системы счисления

Наряду с двоичной системой счисления в компьютере используются ещё две — восьмеричная и шестнадцатеричная. Двоичную, восьмеричную и шестнадцатеричную системы называют **родственными**, поскольку их основания являются степенями числа 2 и существуют быстрые способы перевода чисел из одной родственной системы в другую. Родственными, к примеру, можно также назвать системы счисления с основаниями 3 и 9.

Для перевода числа из  $p$ -ичной системы счисления в  $m$ -ичную человеку удобнее сначала перевести  $p$ -ичное число в десятичное, а потом десятичное — в  $m$ -ичное.

Перевод чисел внутри родственных систем (в частности, с основаниями 2, 8 и 16) можно выполнять, минуя десятичную систему счисления. Рассмотрим табл. 2.1. В ней приведено соотношение чисел в системах счисления с основаниями 10, 2, 8 и 16.

Если внимательно посмотреть на столбцы 2 и 3, то можно увидеть, что все восьмеричные цифры (от 0 до 7) можно записать при помощи трёх двоичных разрядов. На этом основан быстрый перевод из восьмеричной системы в двоичную и наоборот. Для перевода восьмеричного числа в двоичное достаточно каждую цифру этого числа заменить двоичной триадой (три разряда) в соответствии с табл. 2.1, например:

$$734,46_8 = 111\ 011\ 100,100\ 110_2.$$

Таблица 2.1

**Соотношение десятичной, двоичной, восьмеричной  
и шестнадцатеричной систем счисления**

Номер столбца	1	2	3	4
Основание системы счисления	10	2	8	16
Число	0	0000	0	0
	1	0001	1	1
	2	0010	2	2
	3	0011	3	3
	4	0100	4	4
	5	0101	5	5
	6	0110	6	6
	7	0111	7	7
	8	1000	10	8
	9	1001	11	9
	10	1010	12	A
	11	1011	13	B
	12	1100	14	C
	13	1101	15	D
	14	1110	16	E
	15	1111	17	F

Для перевода двоичного числа в восьмеричное нужно воспользоваться следующим алгоритмом:

1. Разбить целую часть двоичного числа справа налево на группы по три цифры (триады). Если в последней слева группе окажется меньше трёх цифр, добавить слева незначащие нули.
2. Разбить дробную часть числа слева направо на триады. Если в последней справа триаде окажется меньше трёх цифр, добавить справа незначащие нули.
3. Каждой триаде сопоставить восьмеричную цифру, например:

$$1010,11111_2 = 001\ 010,111\ 110_2 = 12,76_8.$$

Подобным свойством обладают и шестнадцатеричные цифры. Из таблицы 2.1 (столбцы 2 и 4) видно, что все шестнадцатеричные цифры (от 0 до F) можно записать при помощи четырёх двоичных разрядов — тетрады.

Взаимный перевод из шестнадцатеричной системы в двоичную и обратно осуществляется аналогично переводу между восьмеричной и двоичной системами с той лишь разницей, что шестнадцатеричная цифра соответствует четырём двоичным разрядам, например:

$$18AE,37F_{16} = 0001\ 1000\ 1010\ 1110,0011\ 0111\ 1111_2;$$

$$11010101110,00101111_2 = 0110\ 1010\ 1110,0010\ 1111_2 = 6AE,2F_{16}.$$

Восьмеричная и шестнадцатеричная системы счисления используются для более компактной записи двоичных чисел.

Числа, записанные в любой системе счисления, можно сравнивать по тем же правилам, что и десятичные числа. Так, например,  $10011_2 > 1001_2$ , так как у первого числа больше разрядов в целой части. Если число разрядов в целой части одинаковое, то числа сравнивают поразрядно, от старшего разряда к младшему. Больше то, у которого больше первый несовпадающий разряд. Если число имеет нулевую целую часть, то сравнивают поразрядно слева направо (т. е. от старшего разряда к младшему) дробную часть. Например:  $53361_8 < 53371_8$ ,  $0,CD1B_{16} > 0,CD1A_{16}$ .

## Арифметические действия в разных системах счисления

При обучении счёту в первом классе вы учили таблицу сложения и таблицу умножения однозначных чисел от 0 до 9. Для многоразрядных чисел вы использовали алгоритм сложения и умножения в столбик.

Для системы счисления с любым основанием можно воспользоваться теми же правилами. Так, например, в двоичной системе счисления для сложения многоразрядных чисел надо использовать таблицу сложения от 0 до 1:

$$\begin{aligned} 0 + 0 &= 0; \\ 0 + 1 &= 1; \\ 1 + 0 &= 1; \\ 1 + 1 &= 10. \end{aligned}$$

Обратите внимание на то, что результатом сложения  $1 + 1$  является двузначное двоичное число  $10_2$ , которое соответствует десятичному числу  $2_{10}$ .

Сложение многоразрядных двоичных чисел выполняется, начиная с младшего разряда. Удобно использовать сложение в столбик, как и в десятичной системе счисления. Сложение двух единиц одного разряда даёт 0 в данном разряде и единицу переноса в старший разряд.

**Пример 2.6.** См. приложение 2.

Вычитание в компьютере заменяется сложением с дополнительным кодом (см. параграф 2.2), умножение — многократным сложением.

### Вопросы и задания

- 2.1. Чем характеризуется любая позиционная система счисления?
- 2.2. Что называется основанием системы счисления?
- 2.3. Что называется алфавитом системы счисления?
- 2.4. Какое основание имеет система счисления со следующим алфавитом:  $\{ * \# \$ \}$ ?
- 2.5. Сформулируйте правило определения наименьшего основания системы счисления, в которой может быть записано некоторое число, и приведите примеры применения этого правила.
- 2.6. Перечислите знаки алфавита для шестнадцатеричной системы счисления.
- 2.7. Является ли система счисления с основанием 4 родственной двоичной?
- 2.8. Являются ли системы счисления с основаниями 5 и 25 родственными?
- 2.9. Сколько двоичных разрядов необходимо для кодирования любой цифры восьмеричной системы счисления?
- 2.10. Сколько двоичных разрядов необходимо для кодирования любой цифры шестнадцатеричной системы счисления?
- 2.11. Опишите в общем виде алгоритм перевода:
  - произвольного положительного десятичного числа в двоичное и обратно;
  - произвольного положительного двоичного числа в восьмеричное и обратно;
  - произвольного положительного двоичного числа в шестнадцатеричное и обратно.
- 2.12. Приведите таблицу сложения цифр двоичной системы счисления.

### Задания для самостоятельной работы

- 2.1. Определите номера позиций, на которых расположена цифра 5 в числе 154418,25, и запишите их в порядке возрастания.
- 2.2. Переведите десятичное число 13,125<sub>10</sub> в двоичную систему счисления.
- 2.3. Переведите число 32,25<sub>10</sub> в восьмеричную систему счисления.

- 2.4. Преобразуйте представленные ниже числа к десятичной форме и запишите в порядке возрастания:  $1102_3$ ;  $134_5$ ;  $61_8$ ;  $A_{16}$ .
- 2.5. Расположите следующие числа в порядке возрастания:  $423_5$ ;  $B9_{16}$ ;  $102_{10}$ .
- 
- 2.6. В системе счисления с некоторым основанием десятичное число 38 записывается в виде 102. Укажите это основание.
- 2.7. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись десятичного числа 36 оканчивается на 4.
- 
- 2.8. Сколько разрядов будет содержать число 10011100110001 при переводе его в шестнадцатеричную систему счисления? Ответ дайте, не выполняя перевода.
- 2.9. Преобразуйте число  $111011,10011_2$  в шестнадцатеричную систему счисления.
- 2.10. Дано:  $X = D6_{16}$ ,  $Y = 336_8$ . Какое из чисел  $Z$ , записанных в двоичной системе, отвечает условию  $X < Z < Y$ ?  
1) 11010110; 2) 11000110; 3) 11011011; 4) 11011111.
- 2.11. Вычислите сумму чисел  $X$  и  $Y$ , если  $X = 555_8$ ,  $Y = E1_{16}$ . Результат представьте в двоичном виде.

### Практическая работа на компьютере

- 2.1. В табличном процессоре разработайте модель перевода числа из десятичной системы счисления в двоичную. На рисунке представлен примерный вид модели. Двоичная запись целой части и дробной части числа выделены фоном.

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1													Исходное число	568,785
2	<b>Перевод в двоичную форму целой части числа</b>													
3	Последовательное деление на 2													
4	Остаток от деления на 2 (двоичное число записывается из остатков)	0	1	0	0	0	1	1	1	0	0	0		
5	<b>Проверка правильности перевода</b>													
6	Номер позиции в числе N	10	9	8	7	6	5	4	3	2	1	0		
7	$2^N$	1024	512	258	128	64	32	16	8	4	2	1		
8	Получение целой части числа как суммы степеней числа 2	0	512	0	0	0	32	16	8	0	0	0	568	
9	<b>Разложение по степеням дробной части числа</b>													
10	Умножение дробной части на 2		1,57	1,14	0,28	0,56	1,12	0,24	0,48	0,96	1,92			
11	Двоичные цифры дробной части		1	1	0	0	1	0	0	0	1			
12	Дробная часть исходного числа	0,785	0,57	0,14	0,28	0,56	0,12	0,24	0,48	0,96	0,92			
13	Номер позиции дробной части числа		-1	-2	-3	-4	-5	-6	-7	-8	-9			

Для составления модели вам могут понадобиться встроенные функции ЦЕЛОЕ() и ОСТАТ(). О назначении и технологии применения функций вы можете прочитать в *Справке*.

2.2. В табличном процессоре разработайте модель вычисления суммы двоичных чисел. На рисунке представлен примерный вид модели.

	A	B	C	D	E	F	G	H	I	J	K
1	Сложение двоичных чисел										
2	Единица переноса	1	0	0	1	1	1	1	1	1	0
3	1-е слагаемое		1	0	0	1	1	1	0	0	1
4	2-е слагаемое		1	1	0	0	1	0	1	1	1
	Поразрядная сумма цифр										
5	(вспомогательная строка)	1	2	1	1	2	3	2	2	2	2
6	Сумма (двоичное число)	1	0	1	1	0	1	0	0	0	0

## 2.2

### Представление числовых данных

*Изучив эту тему, вы узнаете:*

- общие принципы представления данных в компьютере;
- алгоритмы представления чисел в разных форматах в компьютере.

#### Общие принципы представления данных

Вспомним, что понятие «данные» отличается от понятия «информация», хотя довольно часто их трактуют одинаково как совокупность каких-либо сведений. Информацией называют те сведения об объектах окружающей среды, их параметрах, свойствах и состояниях, которые уменьшают степень неопределённости, неполноты знаний о них. Из определения видно, что информация — понятие субъективное. Для одного человека некие сведения могут стать информацией, а для другого — нет, так как они уже ему известны. Данными называются сведения, которые зафиксированы в некоторой форме для их сохранения. Данные существуют независимо от того, знает о них человек или нет, может он понять эти данные или нет. Если человек осмысливает (обрабатывает) данные и при этом уменьшается неполнота его знаний по какой-либо теме, то данные превращаются в информацию. Все сведения, которые мы в каком-либо виде сохранили на компьютере, являются именно данными. Все хранимые и обрабатываемые на компьютере данные можно разделить на следующие виды: числовые, символьные (например, текстовые), изображения, звуковые и видеоданные.

Для хранения и обработки данных в компьютере используется **принцип двоичного кодирования**. Это означает, что данные представляются в компьютере в виде двоичного кода, для записи которого используется двоичный алфавит: 0 и 1. Таким образом, в двоичном виде записываются не только числа, но и все другие виды данных. Это составляет суть **принципа однородности данных**.

Принцип однородности данных позволяет использовать единый подход к измерению **объёма данных** как количества знаков в двоичном коде. Наименьшей единицей измерения объёма данных является 1 бит, что соответствует одному знаку двоичного кода.

При измерении объёма данных пользуются укрупнёнными единицами. Единица 1 байт = 8 бит связана с кодированием символов в тексте.

В большинстве случаев объём данных измеряется тысячами, десятками тысяч и большим количеством байт. Поэтому для его измерения используются ещё более крупные единицы:

1 килобайт (Кбайт) =  $2^{10}$  байт = 1024 байт;

1 мегабайт (Мбайт) =  $2^{10}$  Кбайт =  $2^{20}$  байт;

1 гигабайт (Гбайт) =  $2^{10}$  Мбайт =  $2^{30}$  байт.

При измерении величин (ГОСТ 8.417–2002 «Единицы величин») для обозначения кратных единиц используют приставки кило-, мега-, гига- и др., обозначающие увеличение, кратное степени числа 10. Например, 1 кг = 1000 ( $10^3$ ) г. А в информатике исторически эти же приставки обозначают увеличение, кратное степени числа 2, а приставку Кило- записывают с прописной буквы. Например: 1 килобайт (Кбайт) =  $2^{10}$  байт. Это приводило к путанице при вычислении объёма данных.



В марте 1999 года Международная электротехническая комиссия ввела новый стандарт МЭК 60027–2, в котором описано именование единиц измерения двоичных данных. Приставки МЭК схожи с используемыми СИ <sup>1)</sup>: они начинаются на те же слоги, но второй слог у всех двоичных приставок — «би» (от англ. *binary* — двоичный).

После введения указанных стандартов стало возможно использовать для обозначения объёма данных как десятичные приставки (кратные степени 10), так и двоичные (кратные степени числа 2).

Стандарт был утвержден на международном уровне, но введённые названия практически не используются.

Приставки, используемые для обозначения объёма данных согласно упомянутым стандартам, приведены в табл. 2.2.

<sup>1)</sup> СИ — система интернациональная.

Несмотря на то что стандарт принят окончательно, его внедрение происходит довольно медленно. Цифровое сообщество уже привыкло к приставкам СИ, и даже новые операционные системы и приложения всё ещё продолжают их использовать. Поэтому и в этом учебнике используются привычные названия единиц измерения информации с приставками СИ (ГОСТ 8.417–2002).

Таблица 2.2

**Кратные единицы измерения объёма данных**

ГОСТ 8.417–2002			МЭК 60027–2		
Название некорректное с приставками СИ	Обозначение		Степень, используемая в информатике	Название	Символ Обозначение
	международное	русское			
Байт	B	Б (байт)	$2^0$	Байт	B
Килобайт	kB	КБ (Кбайт)	$2^{10}$	Кибибайт	KiB
Мегабайт	MB	МБ (Мбайт)	$2^{20}$	Мебибайт	MiB
Гигабайт	GB	ГБ (Гбайт)	$2^{30}$	Гибибайт	GiB
Терабайт	TB	Тбайт	$2^{40}$	Тебибайт	TiB
Петабайт	PB	Пбайт	$2^{50}$	Пебибайт	PiB
Эксабайт	EB	Эбайт	$2^{60}$	Эксбибайт	EiB
Зеттабайт	ZB	Збайт	$2^{70}$	Зебибайт	ZiB

Для каждого вида данных используются правила преобразования в двоичный код. При обработке данных программа распознает вид данных и применяет соответствующие виды обработки.

Использование двоичного алфавита для представления данных в компьютере объясняется удобством технической реализации двоичных знаков 0 и 1, которые интерпретируются двумя легко распознаваемыми состояниями: электрическими сигналами высокого и низкого напряжения, намагниченными и ненамагниченными участками магнитного материала, наличием или отсутствием электрического тока в цепи. Кроме того, как оказалось, реализовать обработку двоичных данных наиболее просто.

## Форматы представления чисел

Для правильной интерпретации чисел и выполнения машинных операций числа должны иметь одинаковую длину и быть записаны по одинаковым правилам, т. е. иметь одинаковый **формат**. Длина машинного представления числа кратна 1, 2, 4 или 8 байтам. Для хранения чисел в памяти компьютера<sup>1)</sup>, а также их обработки используются два формата: целочисленный и с плавающей точкой (точка — разделительный знак целой и дробной частей числа) — рис. 2.1.



Рис. 2.1. Форматы представления чисел в компьютере

**Целочисленный формат (формат с фиксированной точкой)** используется для представления в компьютере целых положительных и целых отрицательных чисел. Числа этого формата имеют длину 1, 2, 4 байта.

Однобайтовое представление применяется только для положительных целых чисел. В этом формате отсутствует знаковый разряд. Наибольшее двоичное число, которое может быть записано при помощи одного байта, равно  $11111111_2$ , что в десятичной системе счисления соответствует числу  $255_{10}$ .

Для положительных и отрицательных целых чисел обычно используются 2 или 4 байта, при этом старший бит выделяется под знак числа: 0 — плюс, 1 — минус.

При помощи кодирования чисел в двухбайтовом формате можно представить числа от  $-32768_{10}$  до  $32767_{10}$ .

*Внимание!* Если число выйдет за указанные границы, произойдёт **переполнение**. Поэтому при работе с большими целыми числами под них выделяют больше места, например 4 байта.

<sup>1)</sup> Часто это называют также внутренним (или машинным) представлением числа в памяти компьютера.

**Формат с плавающей точкой** используется для представления в компьютере действительных (вещественных) чисел. Числа с плавающей точкой размещаются, как правило, в 4 или 8 байтах.

### Представление целого положительного числа

Для представления целого положительного числа в компьютере используется следующий алгоритм:

1. Перевести число в двоичную систему счисления.
2. Проверить, что заданное число попадает в диапазон чисел, которые можно закодировать требуемым форматом.
3. Дополнить результат слева нулями в пределах выбранного формата, т. е. до получения 8-, 16- или 32-битовой длины. Последний разряд слева является знаковым, в положительном числе он равен 0.

**Пример 2.7.** См. приложение 2.



### Представление целого отрицательного числа

Для представления целого отрицательного числа в компьютере используется *дополнительный код*. Такое представление позволяет заменить операцию вычитания числа операцией сложения с дополнительным кодом этого числа. Знаковый разряд целых отрицательных чисел всегда равен 1. Для представления целого отрицательного числа в компьютере используется следующий алгоритм:

1. Перевести число без знака в двоичную систему счисления.
2. Проверить, что заданное число попадает в диапазон чисел, которые можно закодировать требуемым форматом.
3. Дополнить результат слева нулями в пределах выбранного формата, т. е. до длины 2 или 4 байта.
4. Получить **обратный код** двоичного числа. Для этого нули в числе заменить единицами, а единицы — нулями.
5. К полученному коду прибавить 1.

Полученное число и есть **дополнительный код** положительного числа, который является представлением отрицательного числа в компьютере.

Для проверки правильности перевода можно сложить два противоположных числа. В сумме должен получиться 0. При сложении положительного числа с его дополнительным кодом получаются нули во всех разрядах. При этом в старшем слева разряде возникает единица переноса, которая отбрасывается.

**Примеры 2.8, 2.9.** См. приложение 2.



## Представление вещественного (действительного) числа

Формат с плавающей точкой используется для представления дробных чисел, а также очень больших или, наоборот, очень маленьких чисел в компьютере.

Любое вещественное число может быть представлено в **экспоненциальной** (или **нормализованной**) форме, например:

$$16000000_{10} = 0,16 \cdot 10^8;$$

$$-0,0000156_{10} = -0,156 \cdot 10^{-4}.$$

В общем виде число в нормализованной форме можно записать в виде произведения:

$$R = s \cdot m \cdot P^n,$$

где знак  $s = \pm 1$  отражает знак числа,  $m$  — мантисса,  $P$  — основание системы счисления,  $n$  — целое число, называемое **порядком**.

Порядок  $n$  указывает, на какое количество позиций и в каком направлении должна сместиться в мантиссе точка (запятая), отделяющая дробную часть от целой для получения обычной формы записи числа. Мантисса, как правило, **нормализуется**, т. е. представляется в виде правильной дроби в диапазоне  $0,1 < m < 1$ . Целая часть у мантиссы всегда равна нулю, первый знак после запятой всегда больше нуля.

Аналогично можно получить нормализованную форму двоичного числа, например:

$$1010001,00111_2 = 0,101000100111_2 \cdot 10_2^{111}.$$

В этой записи  $10_2$  означает основание системы счисления (2), записанное в двоичной форме. Степень основания 111 означает число 7, записанное в двоичной форме. Степень показывает, на сколько позиций надо переместить запятую вправо, чтобы получить обычную форму записи числа. В примере

$$0,00000010111_2 = 0,10111_2 \cdot 10_2^{-110}$$

степень  $-110$  означает отрицательное число  $-6$ , которое показывает, на сколько позиций надо переместить запятую влево, чтобы получить обычную форму записи числа.

Для представления в компьютере действительного числа обычной точности с плавающей точкой используется 2 байта (16 бит) или 4 байта (32 бита), для чисел двойной точности — 8 байтов (64 бита). 1-й бит слева всегда означает знак числа. Остальные биты распределяются для хранения мантиссы и порядка.



Рассмотрим представление вещественных чисел в четырёхбайтовом виде. Для определённости перенумеруем биты от 0 справа и до 31-го. Тогда 31-й бит хранит знак числа. В битах с 24-го по 30-й хранится порядок числа в машинном представлении. И наконец, в битах с 0-го по 23-й хранится мантисса.

Для машинного представления порядка отведено 7 разрядов, что позволяет закодировать 128 чисел. Первые 64 числа — от 0-го до 63-го — предназначены для кодирования отрицательного порядка, а числа с 64-го до 127-го — для кодирования неотрицательного порядка (табл. 2.3). Так, если порядок числа равен  $-6$  (см. пример ранее), то его двоичный код будет соответствовать числу 58 ( $64 - 6 = 58$ , см. табл. 2.3), записанному в двоичной семибитовой записи, т. е. числу  $0111010_2$ .

Таблица 2.3

### Сопоставление машинного и математического порядков числа

<b>Машинный порядок</b>	0	1	2	3	...	62	63	64	65	...	126	127
<b>Математический порядок</b>	-64	-63	-62	-61	...	-2	-1	0	1	...	62	63

Тогда алгоритм представления числа в четырёхбайтовом виде таков:

1. Перевести исходное число в двоичное.
2. Добавить в дробной части справа нули так, чтобы получить 24-разрядное число.
3. Получить нормализованную форму числа — определить мантиссу и порядок числа.
4. Записать 32-битовое число: знак (1 бит), порядок (7 бит), мантисса (24 бита).

Для более компактной записи полученное машинное представление числа переводят в шестнадцатеричное число.

Заметим, что, поскольку мы переводим мантиссу в двоичную форму как целое число, может возникнуть проблема переполнения. Так, если для мантиссы отводится 24 знака в четырёхбайтовом представлении, то максимальное 24-битовое число, которое можно записать в двоичной форме, — это  $2^{24} - 1 = 16\,777\,215$ . Если мантисса выходит за эти пределы, необходимо число округлить.

**Пример 2.10.** См. приложение 2.



**Вопросы и задания**

- 2.13. Какие два формата используются для представления чисел в компьютере?
- 2.14. Сколько байт используют для представления целых чисел?
- 2.15. Какие числа обычно кодируют одним байтом?
- 2.16. Как кодируется знак числа в машинном представлении?
- 2.17. Опишите алгоритм представления целого положительного числа в компьютере.
- 2.18. Как получается обратный код двоичного числа?
- 2.19. Как получается и для чего используется дополнительный код двоичного числа?
- 2.20. Опишите алгоритм представления целого отрицательного числа в компьютере.
- 2.21. Где располагается порядок числа в четырёхбайтовом представлении вещественного числа в компьютере?

**Задания для самостоятельной работы**

- 2.12. Представьте число  $481_{10}$  в 2-байтовом целочисленном формате. Ответ представьте 16-ричным числом.
- 2.13. Представьте число  $1020_{10}$  в 2-байтовом целочисленном формате. Ответ представьте 16-ричным числом.
- 2.14. В 2 байтах представлено целое положительное число в формате с фиксированной точкой  $0000110001100001$ . Что это за число? Ответ представьте в десятичном виде.
- 2.15. В 2 байтах представлено целое положительное число в формате с фиксированной точкой  $0010010010001110$ . Что это за число? Ответ представьте в десятичном виде.



- 2.16. Представьте число  $-540010$  в 2-байтовом целочисленном формате. Ответ представьте 16-ричным числом.
- 2.17. В 2 байтах представлено целое отрицательное число в формате с фиксированной точкой  $1000000100000101$ . Что это за число? Ответ представьте в десятичном виде.
- 2.18. Выполните представление вещественного двоичного числа  $0,0001100011101_2$  числа в 4-байтовом формате с плавающей точкой. Ответ представьте в 16-ричном виде.
- 2.19. Выполните представление вещественного двоичного числа  $-0,00000000000001100011_2$  числа в 4-байтовом формате с плавающей точкой. Ответ представьте в 16-ричном виде.
- 2.20. Выполните представление вещественного числа  $401,625_{10}$  в 4-байтовом формате с плавающей точкой. Ответ представьте в 16-ричном виде.
- 2.21. Выполните представление вещественного числа  $-67,03125_{10}$  в 4-байтовом формате с плавающей точкой. Ответ представьте в 16-ричном виде.

- 2.22. В 2 байтах представлено число 011111000100010 в формате с плавающей точкой. Под порядок выделено 4 разряда. Запишите соответствующее десятичное число.
- 2.23. Переведите число  $-0,09375_{10}$  в двоичную систему счисления и представьте его в 2-байтовом формате с плавающей точкой, при условии что под мантиссу выделен 1 байт.
- 2.24. Представьте число  $-2001_{10}$  в 2-байтовом целочисленном формате. Ответ представьте 16-ричным числом.
- 2.25. В 2 байтах представлено целое отрицательное число в формате с фиксированной точкой 1000 1101 0000 1100. Что это за число? Ответ представьте в десятичном виде.
- 2.26. Выполните представление вещественного двоичного числа  $0,000001001100101_2$  числа в 4-байтовом формате с плавающей точкой. Ответ представьте в 16-ричном виде.
- 2.27. Выполните представление вещественного двоичного числа  $0,000000000000000010011101_2$  числа в 4-байтовом формате с плавающей точкой. Ответ представьте в 16-ричном виде.
- 2.28. Выполните представление вещественного числа  $1800,1875_{10}$  в 4-байтовом формате с плавающей точкой. Ответ представьте в 16-ричном виде.
- 2.29. Выполните представление вещественного числа  $-511,5_{10}$  в 4-байтовом формате с плавающей точкой. Ответ представьте в 16-ричном виде.
- 2.30. Переведите число  $10,0625_{10}$  в двоичную систему счисления и представьте его в 2-байтовом формате с плавающей точкой, при условии что под мантиссу выделен 1 байт.

## 2.3

### Представление текстовых данных

*Изучив эту тему, вы узнаете:*

- о структуре кодовых таблиц ASCII и Unicode;
- о вычислении объема текстового файла.

#### Кодовые таблицы символов

Текстовые (символьные) данные, так же как и числовые, для хранения и обработки на компьютере кодируются в двоичном алфавите. Нажатие любой алфавитно-цифровой клавиши на клавиатуре приводит к тому, что в компьютер посылается сигнал в виде двоичного числа, представляющего собой одно из значений кодовой таблицы. **Кодовая таблица** — это внутреннее представление символов в компьютере. На начальном этапе развития вычисли-

тельной техники на разных компьютерах использовались разные коды для кодирования символов. Но производители поняли, что необходимо использовать единый стандарт для кодирования хотя бы международных символов. Устройства памяти в те времена имели ограниченный объём. Поэтому надо было придумать такую кодировку, чтобы, с одной стороны, можно было закодировать все известные международные символы, а с другой — чтобы код символа был как можно более коротким. В качестве стандарта была принята **кодовая таблица ASCII** (*American Standard Code for Informational Interchange* — американский стандартный код информационного обмена). Согласно этой таблице, для хранения двоичного кода одного символа выделен 1 байт = 8 бит. Учитывая, что каждый бит принимает значение 1 или 0, количество возможных сочетаний единиц и нулей длиной 8 бит равно  $2^8 = 256$ . Значит, с помощью 1 байта можно получить 256 двоичных кодовых комбинаций и отобразить с их помощью 256 различных символов. Эти коды и составляют таблицу ASCII (табл. 2.4). Стандартными являются коды первой половины таблицы: это коды международных символов — прописные и строчные буквы английского алфавита, цифры, знаки. Первые 32 символа являются управляющими и предназначены в основном для передачи команд управления. Их назначение может варьироваться в зависимости от программных и аппаратных средств.

Вторая половина таблицы выделена для кодирования национальных алфавитов и других символов. В разных странах могут использоваться различные варианты второй половины кодовой таблицы.

Рассмотрите таблицу ASCII. Строки и столбцы в ней обозначены цифрами шестнадцатеричной системы счисления. Код конкретного символа состоит из двух цифр. На первом месте указывается номер столбца, на втором — номер строки. Например, латинская буква «S» в таблице ASCII представлена шестнадцатеричным кодом 53, что соответствует двоичному коду 01010011. При выводе буквы «S» на экран компьютер выполняет декодирование — на основании этого двоичного кода создает изображение символа.

Коды символов составлены таким образом, чтобы можно было выполнять естественную сортировку символьных данных — сначала цифры, потом англоязычные символы (сначала прописные, затем строчные), и наконец, национальные символы (сначала прописные, затем строчные). Есть также определённая логика в расположении знаков в кодовой таблице. Важно запомнить, что коды международных символов начинаются на цифры от 0 до 7, а национальные коды — на цифры от 8 до F.



Внимательно проанализируйте таблицу ASCII-кодов (приложение 1 в части 2 учебника, табл. 2.4). Вы увидите, что прописная и соответствующая ей строчная буквы расположены на одной строке через столбец, т. е. номера столбцов кодов символов различаются на 2 единицы. Например, шестнадцатеричный код буквы «E» — 45, «e» — 65, «H» — 48, «h» — 68, «Z» — 5A, «z» — 7A. Зная это, можно по коду одного символа определить код другого символа.

**Пример 2.11.** См. приложение 2.

Цифры кодируются по стандарту ASCII в двух случаях — при вводе/выводе и когда они встречаются в тексте. Если цифры используются для кодирования числовых данных и использования их в вычислениях, то они преобразуются в двоичный код в соответствии с правилами, рассмотренными в предыдущих темах.

Для сравнения рассмотрим два варианта кодирования числа 45.

При *использовании в тексте* это число потребует для своего представления 2 байтов, поскольку каждая цифра будет представлена своим кодом в соответствии с таблицей ASCII. В шестнадцатеричной системе код будет выглядеть как 3435, в двоичной системе — 0011010000110101.

При *использовании в вычислениях* код этого числа будет получен по правилам кодирования числовых данных, и если для его представления будет выбран 1 байт, то число будет иметь вид 00101101.

Развитие вычислительной техники привело к появлению устройств хранения данных большого объема. И появилась возможность использовать единую кодировку для различных алфавитов, используемых в разных странах и разных видах деятельности. В настоящее время широко распространена двухбайтовая **кодировочная таблица Unicode**. Эта кодировка поддерживается в большинстве операционных систем, во всех современных браузерах и многих программах.

Стандарт Unicode (приложение 1) явился результатом сотрудничества Международной организации по стандартизации (ISO) с ведущими производителями компьютеров и программного обеспечения. В мире существует 6700 живых языков, но только 50 из них являются официальными языками государств. Письменностей используется около 25, что делает возможным создание универсального стандарта.

Как оказалось, для кодирования этих письменностей достаточно 16 бит (2 байта) на символ, т. е. диапазона от 0000 до FFFF. Стандарт ASCII занимает в кодовом пространстве диапазон от 0000 до 00FF. Каждой письменности выделен свой блок кодов. На сегодняшний день кодирование всех живых офици-

альных письменностей считается завершённым — распределено около 29 000 позиций из 65 535 возможных.

Россия — многонациональное государство. Стандарт Unicode 3.2 поддерживает следующие языки: алтайский, башкирский, бурятский, калмыцкий, коми, корякский, марийский, нанайский, ненецкий, осетинский, русский, татарский, тувинский, удмуртский, хакасский, чувашский, эвенкийский, эвенский и др.

В последнее время консорциум Unicode приступил к кодированию остальных письменностей нашей планеты, которые представляют какой-либо интерес, — письменностей мёртвых языков, выпавших из современного обихода, китайских иероглифов, искусственно созданных алфавитов и т. п. Для представления такого разнообразия языков 16-битового кодирования недостаточно, и Unicode приступил к освоению 21-битового пространства кодов (000000–10FFFF), которое разбито на 16 зон, названных плоскостями.

### Объём текстовых данных

Для вычисления объёма текстовых данных надо знать количество байт, выделенных для кодирования символа ( $b$ ), и количество символов в тексте. Обычно текстовый документ характеризуется количеством страниц ( $p$ ), средним количеством строк на странице ( $r$ ) и средним количеством символов в строке ( $s$ ). Тогда оценить объём текстового документа в байтах можно по формуле

$$V = b \cdot p \cdot r \cdot s.$$

Для большого объёма можно использовать укрупнённые единицы измерения данных.



**Пример 2.12.** См. приложение 2.



### Вопросы и задания

- 2.22. Как расшифровывается на русском языке аббревиатура ASCII?
- 2.23. Какой кодовый диапазон отводится для символов русского алфавита в таблице кодов ASCII?
- 2.24. Сколько бит содержит код каждого символа в кодировке ASCII?
- 2.25. Объясните, почему в ASCII-коде можно закодировать 256 различных символов.
- 2.26. Какие символы закодированы кодировкой Unicode?
- 2.27. Как получить код символа по таблице ASCII?
- 2.28. ASCII-код какого символа больше: «U» или «u»? Почему?
- 2.29. Как вычислить объём текстовых данных?



- 2.30. При упорядочении по возрастанию тестовых данных «100» и «50» какое будет расположено раньше?
- 2.31. В каком направлении предполагается развивать кодовую таблицу Unicode?

### Задания для самостоятельной работы

- 2.31. Запишите код числа 0,5 как текстовой строки в кодовой таблице ASCII и как числа в 2-байтовом формате с плавающей точкой, в котором на порядок отводится 4 разряда. Представьте обе записи 16-ричными числами.
- 2.32. Определите в байтах объём сообщения «Москва — столица России», закодированного с помощью 16-битового Unicode.
- 2.33. В таблице ниже представлена часть кодовой таблицы ASCII:

Символ	1	2	f	g	n	F	G
Десятичный код	49	50	102	103	110	70	71
Шестнадцатеричный код	31	32	66	67	6E	46	47

Определите шестнадцатеричный код символа «N».

- 2.34. Известно, что информационный объём брошюры составляет 40 Кбайт. Брошюра состоит из 20 страниц, а каждая страница содержит в среднем 32 строки. Определите среднее количество символов в строке на одной странице брошюры.
- 2.35. По таблице ASCII определите, в каком порядке будут расположены символьные строки после сортировки по возрастанию. В ответе укажите последовательность номеров строк, соответствующую правильному упорядочению:
- 1) [input]
  - 2)  $10 * N = 40$
  - 3)  $5 * N = 20$
  - 4) (input)
- 2.36. Считая, что каждый символ кодируется в Unicode, определите информационный объём в байтах следующей фразы:  
Экзамен — лотерея, выиграть в которой может только тот, кто имеет представление о предмете!
- 2.37. Определите в байтах объём сообщения  
Санкт-Петербург — интеллектуальная и культурная столица нашей Родины.  
закодированного с помощью кодовой таблицы ASCII.
- 2.38. Найдите объём текстового сообщения в битах, если сообщение состоит из 10 символов и каждый символ закодирован ASCII-кодом.

- 2.39. Во сколько раз изменится информационный объём сообщения при его преобразовании из кодировки Unicode (таблица кодировки содержит 65536 символов) в кодировку ASCII (таблица кодировки содержит 256 символов)? В ответе укажите, уменьшится или увеличится и во сколько раз.
- 2.40. Количество листов в брошюре — 14. Каждый лист содержит в среднем: строк — 42; символов в строке — 72. Текст имеет кодировку ASCII. Поместится ли брошюра на флеш-память, если на ней осталось 100 Кбайт свободного места? В ответе укажите в виде положительного числа в байтах, сколько останется свободного места на флеш-памяти, если брошюра поместится, или в виде отрицательного числа, сколько не хватает байт, если брошюра не поместится.

## 2.4

### Представление графических данных

*Изучив эту тему, вы узнаете:*

- как вычислить глубину цвета и оценить объём графического изображения;
- какие форматы используются при хранении графических файлов;
- о разрешении сканера и цифрового фотоаппарата и оценке качества графического изображения;
- об оценке объёма векторного изображения.

Создавать и хранить графические объекты в компьютере можно двумя способами — как растровое или как векторное изображение. Для каждого из типов изображения используется свой способ кодирования.

**Растровое изображение** представляет собой совокупность точек — **пикселей**, каждый пиксель окрашен в определённый цвет. **Цветовая палитра** — набор цветов, используемых для создания рисунка.

Каждый цвет в палитре кодируется двоичным кодом. Число бит, используемых для кодирования цвета, называется **глубиной цвета**.

При кодировании цветовой палитры ставится задача определить, какое наименьшее число бит потребуется для кодирования каждого из цветов. Эта задача напрямую связана с задачей вычисления объёма графического изображения. Чтобы определить глубину цвета в палитре, можно действовать по следующему алгоритму:

1. Перенумеровать цвета палитры десятичными порядковыми числами, начиная с 0.
2. Перевести в двоичный код последний (максимальный) порядковый номер. Сколько двоичных разрядов будет в этом числе, столько бит и потребуется для кодирования цветов палитры.

Рассмотрим, как кодируются некоторые цветовые палитры.

Для чёрно-белой палитры глубина цвета равна 1 бит, так как в ней только 2 цвета и их можно закодировать двоичными цифрами — 0 (чёрный) и 1 (белый). Если палитра содержит 8 цветов, то для кодирования цвета понадобится 3 бита. У 16-цветной палитры глубина цвета 4 бита, у 256-цветной — 8 бит (1 байт). В таблице 2.6 представлены коды 16-цветной палитры.

Таблица 2.6

Кодирование 16-цветной палитры

Цвет	Яркость	Красный	Зелёный	Синий
Чёрный	0	0	0	0
Синий	0	0	0	1
Зелёный	0	0	1	0
Голубой	0	0	1	1
Красный	0	1	0	0
Фиолетовый	0	1	0	1
Коричневый	0	1	1	0
Белый	0	1	1	1
Серый	1	0	0	0
Светло-синий	1	0	0	1
Светло-зелёный	1	0	1	0
Светло-голубой	1	0	1	1
Светло-красный	1	1	0	0
Светло-фиолетовый	1	1	0	1
Жёлтый	1	1	1	0
Белый	1	1	1	1



Цвет используется не только в графических изображениях. При создании веб-страниц цветом кодируются фон страницы, символы, заливка ячеек, рядов и таблиц и пр. Для описания цвета различных элементов веб-страниц используются элементы языка разметки гипертекста (описание веб страниц) HTML `bgcolor = "xxxxxx", color = "xxxxxx", text = "xxxxxx"` и т. д. В кавычках задаются шестнадцатеричные значения интенсивности цветовых компонентов в 24-битовой RGB-модели, т. е. базовых цветов — красного (*red*), зелёного (*green*) и синего (*blue*). Интенсивность каждого базового цвета кодируется 1 байтом (8 битами), что позволяет закодировать  $2^8 = 256$  оттенков цвета. Шестнадцатеричные коды оттенков изменяются от 00 до FF. Всего такая палитра позволяет закодировать  $2^{24} = 16\,777\,216$  оттенков цвета. Невозможно придумать название каждому оттенку, поэтому необходимо знать коды основных цветов (табл. 2.7). Так, например, «чистому» зелёному цвету соответствует код 00FF00.

Таблица 2.7

Коды основных цветов

Чёрный	000000
Белый	FFFFFF
Красный	FF0000
Зелёный	00FF00
Синий	0000FF
Оттенки серого	Коды всех составляющих одинаковы, например 0F0F0F

Чем меньше код цвета, тем цвет ближе к чёрному.

Если глубина цвета равна  $n$  бит, то можно закодировать  $2^n$  цветов.

Как правило, растровое графическое изображение является прямоугольным, его ширина —  $a$ , высота —  $b$  (в пикселях). Количество пикселей в изображении  $K = a \cdot b$ .

Информационный объём графического изображения (в битах):

$$V = a \cdot b \cdot n.$$

Полученный объём графического изображения в битах можно перевести в байты, килобайты и т. д.

**Пример 2.13.** См. приложение 2.

В настоящее время существует более двух десятков форматов растровых графических файлов. Самые популярные — BMP,



GIF, TIFF, JPEG, PCX. Есть файлы, которые кроме статических изображений могут содержать анимационные клипы и/или звук, например GIF, PNG, AVI, SWF, MPEG, MOV. Важной характеристикой этих файлов является способность представлять содержащиеся в них данные в сжатом виде.

**BMP** (*Bitmap Picture* — точечный рисунок) — формат Windows, поддерживается всеми графическими редакторами, работающими под управлением этой операционной системы. В этом формате изображение кодируется без сжатия. Формат поддерживает 2-, 16- и 256-цветную и 24-битовую (RGB) палитры.

**GIF** (*Graphics Interchange Format*) — формат предназначен для сохранения растровых изображений с количеством цветов не более 256, использует алгоритм сжатия данных за счёт уменьшения количества цветов в палитре.

**JPEG** (*Joint Photographic Experts Group*) — формат предназначен для компактного хранения многоцветных изображений фотографического качества. Файлы этого формата имеют расширение *jpg* или *jpeg*.

В отличие от формата GIF в формате JPEG используется алгоритм сжатия с потерями данных, благодаря чему достигается очень большая степень сжатия (от единиц до сотен раз).

Обычно графическое изображение имеет вид прямоугольника и измеряется в сантиметрах по горизонтали и вертикали. Как оценить объём изображения в этом случае? Для этого надо знать, с помощью какого устройства было получено изображение. Если с помощью сканера, то надо знать разрешение, заданное при сканировании. Разрешение задаётся в **dpi** (*dot per inch* — точек на дюйм). Точная величина одного дюйма 2,54 см, но для оценки объёма изображения используют величину 2,5 см.

**Пример 2.14.** См. приложение 2.

Для того чтобы оценить объём фотографии, полученной с помощью цифрового фотоаппарата, надо знать разрешение матрицы — основного чувствительного элемента этого устройства, которое обычно указывается в виде произведения максимального количества точек по горизонтали на максимальное количество точек по вертикали. Если перемножить эти величины и разделить на 1 000 000, то получится разрешение в мегапикселях. Эта величина даёт возможность оценить качество получаемого изображения в зависимости от его размера. Для этого надо вычислить разрешение в dpi. Обычно хорошим считается разрешение от 300 dpi. Приемлемое качество можно получить при 150 dpi.

**Пример 2.15.** См. приложение 2.

**Векторное изображение** представляет собой совокупность графических примитивов: прямоугольников, овалов, прямолинейных отрезков и кривых линий — и геометрических фигур более сложной формы. Каждый графический примитив характеризуется набором геометрических параметров, связанных между собой математическими формулами, которые задают правило построения фигуры. Кроме того, для каждого примитива указывается тип границы — сплошная, пунктирная, штрихпунктирная и пр., толщина и цвет, а замкнутые фигуры дополнительно характеризуются типом заливки (цвет и узор). Поскольку набор примитивов в программной среде фиксирован, формулы построения заложены в код программы в виде подпрограмм с соответствующими алгоритмами. Поэтому код векторного изображения включает коды параметров фигур, а кодирование выполняется различными способами, зависящими от прикладной среды. Поэтому информационный объём векторного изображения существенно меньше, чем код аналогичного растрового.



*Внимание!* Информационный объём (или просто объём) графического изображения и размер графического файла, в котором хранится это изображение (даже если не используется сжатие), не совпадают, так как в файле хранятся также некоторые дополнительные сведения об изображении — цветовая палитра, способ кодирования и пр.



### Вопросы и задания

- 2.32. В каких единицах измеряется глубина цвета?
- 2.33. Как вычислить глубину цвета?
- 2.34. Как кодируется цвет на веб-страницах?
- 2.35. Как вычислить информационный объём растрового изображения?
- 2.36. Как кодируется векторное изображение?
- 2.37. Какая характеристика сканера позволяет судить о качестве получаемого при сканировании графического изображения?
- 2.38. Какая характеристика цифрового фотоаппарата позволяет судить о качестве получаемого при сканировании графического изображения?
- 2.39. Какие форматы графических файлов вы знаете?



### Задания для самостоятельной работы



- 2.41. Для выделения текстовой фразы на веб-странице используется тег (элемент языка HTML) `<font>` `</font>` с атрибутом `color="#XXXXXX"`, где в кавычках задаются шестнадцатеричные зна-

чения интенсивности цветовых компонент в 24-битовой RGB-модели. Выберите из предложенных вариантов код цвета, который является оттенком зелёного цвета:

- 1) 0B0000;      2) 00G100;      3) 0000F0;      4) 000F00.

- 2.42. Для кодирования цвета текстовой гиперссылки на веб-странице используется атрибут `link="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битовой RGB-модели. Какой цвет будет иметь гиперссылка на странице, заданной тэгом `<body link="#0000FF">?` Выберите номер правильного ответа:

- 1) белый;      2) зелёный;      3) красный;      4) синий.

- 
- 2.43. Определите информационный объём в килобайтах 256-цветного растрового изображения, размер которого  $1024 \times 768$  пикселей.
- 2.44. Цветное растровое изображение, использующее 16 цветов, преобразовано в чёрно-белое. Как изменится его информационный объём?
- 2.45. Как изменится информационный объём растрового изображения, если первоначально количество цветов в палитре было 256, а после преобразования — 16?
- 2.46. Информационный объём 256-цветного растрового изображения равен 1 Кбайт. Из какого количества пикселей он состоит?
- 2.47. Информационный объём растрового изображения размером  $1024 \times 640$  пикселей составляет 640 Кбайт. Определите количество цветов в палитре изображения.
- 2.48. Графический режим поддерживает 32 цвета. Какое минимальное количество разрядов необходимо для кодирования цветов?
- 2.49. Чёрно-белое растровое изображение имеет размер  $1024 \times 768$  пикселей. Определите его информационный объём.
- 2.50. Размер графического файла формата BMP складывается из дополнительных данных фиксированного размера 54 байта и объёма собственно растрового изображения, которое имеет размер  $32 \times 32$  пикселя в цветовой палитре, состоящей из  $2^{16}$  цветов. Вычислите в байтах размер графического файла.
- 2.51. В палитре художника всего «семь цветов радуги». Какое минимальное количество бит (глубина цвета) потребуется для кодирования одного цвета палитры?
- 2.52. В процессе преобразования растрового изображения количество цветов в палитре уменьшили с  $65\,536$  ( $2^{16}$ ) до 256, а его длину и ширину уменьшили пропорционально в 3 раза. Как изменился его объём?
- 2.53. Фотография размером  $10 \times 10$  см была отсканирована с разрешением 400 dpi при глубине цвета 24 бита. Определите в наиболее подходящих единицах измерения информационный объём полученного растрового изображения. Для вычислений используйте соотношение 1 дюйм = 2,5 см.

- 2.54. Информационный объём растрового изображения составляет 200 байт. Определите глубину цвета, если размер изображения равен  $20 \times 40$  пикселей.
- 2.55. Вычислите информационный объём растрового изображения размером  $128 \times 128$  пикселей с палитрой, содержащей 16 градаций серого цвета.
- 2.56. Определите информационный объём растрового изображения в байтах, если глубина цвета составляет 4 бита, а размер изображения —  $40 \times 50$  пикселей.
- 2.57. Растровое изображение имеет размер  $400 \times 400$  пикселей и создано с глубиной цвета 8 бит. Инструментом *Надпись* графического редактора Paint на изображение нанесли поясняющий текст «Структура компьютера». Как изменился объём изображения?
- 2.58. Экранные обои представляют собой рисунок в 24-разрядном формате BMP. Вычислите, какой объём (Кбайт) на носителе займёт этот рисунок, если экран содержит  $600 \times 800$  точек.



### Поисково-исследовательская работа

- 2.1. Найдите в Интернете информацию о параметрах цифровых камер современных мобильных телефонов. Оцените, какого размера фотографию можно отпечатать со снимка, чтобы она была качественной (не менее 300 dpi).
- 2.2. Найдите в Интернете информацию о разрешении современных цифровых фотоаппаратов. Оцените, какого размера фотографию можно отпечатать со снимка, чтобы она была качественной (не менее 300 dpi). Дайте рекомендацию, какой цифровой фотоаппарат следует использовать, если вы планируете печатать портретные снимки (размером  $30 \times 40$  см<sup>2</sup> и выше).

## 2.5

### Представление звуковых данных

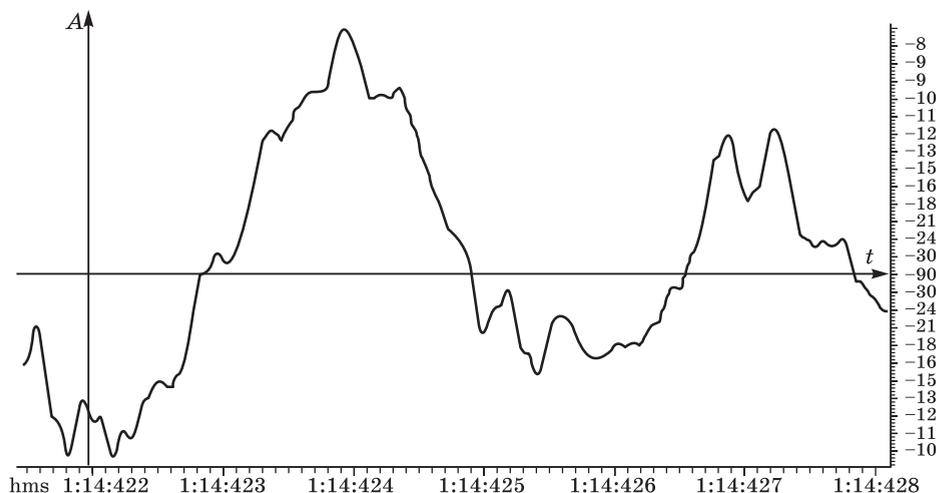
*Изучив эту тему, вы узнаете:*

- о графике звуковой волны и его отображении в звуковых редакторах;
- о дискретизации аналогового звукового сигнала и глубине кодирования звука;
- об оценке информационного объёма звукового сигнала;
- какие форматы используются при хранении звуковых файлов.



Из курса физики вы знаете, что звук — это колебательное движение частиц упругой среды, распространяющееся в виде волн в газообразной, жидкой или твёрдой средах. Чистое звуковое колебание соответствует звучанию одной ноты. Его можно из-

образить в виде синусоиды определенных частоты и амплитуды. Амплитуда колебания определяет силу звука (громкость), а частота колебания — высоту звука. Речь, музыка и другие звуки окружающего мира представляют собой сумму нескольких синусоид различных амплитуд и частот. Графически звук можно изобразить в виде функции, показывающей зависимость амплитуды  $A$  от времени  $t$  (рис. 2.2).

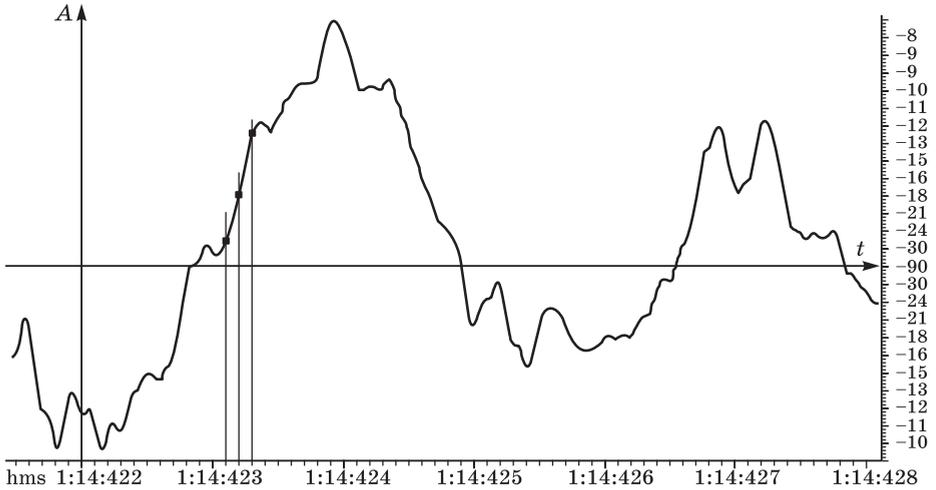


**Рис. 2.2.** Пример звуковой записи, представленной в одном из звуковых редакторов

Способ цифрового кодирования звука состоит в следующем: временную ось графика, соответствующего некоторому звуку, разбивают на маленькие участки и измеряют значения амплитуды колебания в точках разбиения (рис. 2.3). Числовые значения амплитуд переводят в двоичный код с заданной точностью. Качество цифрового кодирования звука зависит от величины участков разбиения (реально это десятитысячные доли секунды) и точности кодирования амплитуды.

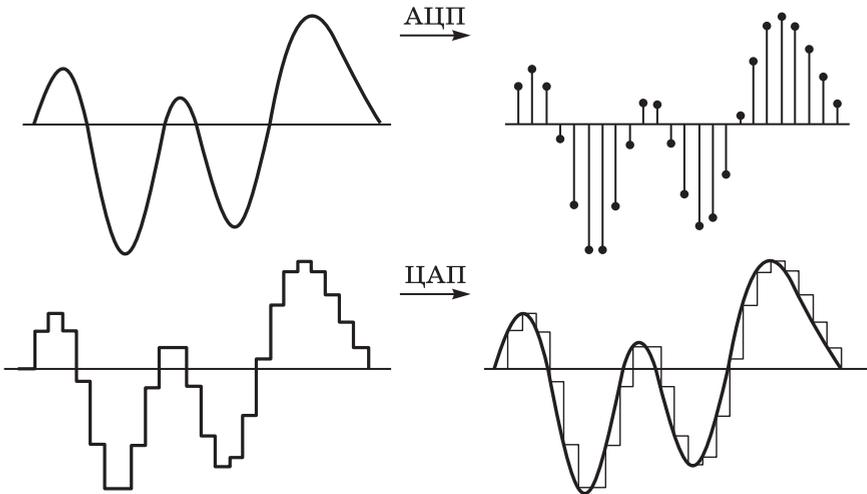
Если преобразовать звук в электрический сигнал (например, с помощью микрофона), то получится изменяющееся с течением времени напряжение. Для компьютерной обработки такой непрерывный (аналоговый) сигнал необходимо преобразовать в последовательность двоичных чисел. Для этого напряжение измеряется через равные промежутки времени, полученные значения аналогового сигнала (дискретные отсчеты) преобразуются в двоичный код, последовательность кодов записывается в память компьютера. Этот процесс называется **дискретизацией** (или **оцифровкой**),

а устройство, выполняющее его, — **аналого-цифровым преобразователем (АЦП)**.



**Рис. 2.3.** Отсчёты звукозаписи

Для того чтобы воспроизвести закодированный таким образом звук, нужно выполнить обратное преобразование (для чего служит **цифро-аналоговый преобразователь (ЦАП)**), а затем восстановить получившийся ступенчатый сигнал (рис. 2.4).



**Рис. 2.4.** Преобразование звукового сигнала в дискретный (АЦП) и обратно (ЦАП)

Количество бит, отводимое на кодирование одного отсчёта звукового сигнала, называют **глубиной кодирования звука**. Современные звуковые карты обеспечивают 16-, 32- или 64-битовую глубину кодирования звука.

По теореме Котельникова, **частота дискретизации** (т. е. количество отсчётов за секунду) должна превосходить наивысшую записываемую частоту сигнала как минимум вдвое. Человек слышит звук с частотами примерно от 20 Гц до 20 000 Гц. Чем выше частота дискретизации и чем больше разрядов отводится для каждого отсчёта, тем точнее будет представлен звук. Но, естественно, увеличится и размер звукового файла. Поэтому в зависимости от характера звука, требований, предъявляемых к его качеству и объёму занимаемой памяти, выбирают некоторые компромиссные значения. Например, при записи на компакт-диски используются 16-битовые отсчёты при частоте дискретизации 44 032 Гц. Стандарт на компакт-диски разрабатывался в расчёте на весь слышимый диапазон. А для работы только с речевыми сигналами достаточно 8-битовых отсчётов при частоте дискретизации 8 кГц. Чтобы можно было понять человеческую речь, достаточно передавать звук с частотой около 3000 Гц (это примерно соответствует характеристикам телефонных линий связи).

Естественность восприятия человеком звука зависит от способа записи и воспроизведения звука техническими устройствами. Это особенно важно, когда требуется передать одновременное звучание многих источников, например шум ветра, волн, дождя и шелест листьев или звучание множества инструментов в оркестре. Простейшей является одноканальная (моно) звукозапись. Это соответствует восприятию звука одним ухом. Стереозапись является двухканальной (по одному каналу на каждое ухо). Соответственно, файл со стереозвуком будет в два раза больше, чем с монозвучием. Есть и другие более сложные способы записи звука.

Пусть  $D$  — частота дискретизации (Гц),  $T$  — время звучания (или записи) звука (с),  $N$  — глубина кодирования звука (бит),  $K$  — количество каналов записи. Тогда объём звуковых данных будет вычисляться по формуле

$$V = D \cdot T \cdot N \cdot K \text{ (бит).}$$

Данная формула позволяет оценить объём цифровой звукозаписи без сжатия. Следует также учитывать, что звуковой файл кроме собственно звуковых данных содержит ещё и некоторые дополнительные сведения.

**Пример 2.16.** См. приложение 2.



Описанный способ кодирования звуковой информации универсален, он позволяет представить любой звук и преобразовывать его самыми разными способами. Кроме того, таким методом можно оцифровывать не только звуковые, но и любые другие аналоговые сигналы, например результаты измерений, выполненных аналоговыми приборами.

Звуковые файлы имеют несколько форматов. Наиболее популярные из них MIDI, WAV, MP3.

**MIDI** (*Musical Instrument Digital Interface*) — формат, изначально предназначенный для управления музыкальными инструментами. В настоящее время используется как основной стандарт для кодирования нотной записи для электронных музыкальных инструментов и компьютерных модулей синтеза.

**WAV** (*waveform*) — основной формат на системах Windows для хранения обычного несжатого звука в виде цифрового представления исходного звукового колебания или звуковой волны. Файлы этого формата достаточно большие по объёму.

**MP3** (*MPEG-1 Audio Layer 3*) — один из цифровых форматов хранения сжатой звуковой информации. Он обеспечивает более высокое, чем прочие, качество кодирования.



### Вопросы и задания

- 2.40. Чем отличается непрерывный сигнал от дискретного?
- 2.41. Что такое частота дискретизации и на что она влияет?
- 2.42. Опишите процесс оцифровки звука.
- 2.43. Как восстанавливается звук по двоичному коду?
- 2.44. Что такое глубина кодирования звука?
- 2.45. Как вычислить объём звукозаписи?
- 2.46. Какие звуковые форматы вы знаете?



### Задания для самостоятельной работы

- 2.59. Оцените в мегабайтах с точностью до двух знаков после запятой объём 1 минуты стереозвукозаписи, кодируемой без сжатия, если значение сигнала фиксируется 44 032 раза в секунду и записывается 16-битовым кодом.
- 2.60. Оцените в мегабайтах с точностью до двух знаков после запятой объём 1 минуты одноканальной (моно) звукозаписи, кодируемой без сжатия, если значение сигнала фиксируется 22 016 раз в секунду и записывается 8-битовым кодом.
- 2.61. Объём монофонического звукового файла длительностью 1,5 минуты составляет 967,5 Кбайт. С какой частотой дискретизации осуществлялась оцифровка, если каждый отчёт кодировался 8 битами?
- 2.62. Объём стереофонического звукового файла составляет 16,13 Мбайт. Какова длительность звучания, если оцифровка осуществлялась с

частотой дискретизации 44 032 Гц и каждый отчёт кодировался 32 битами?

- 2.63. Во сколько раз отличается объём звукового файла стереозаписи с 16-битовой глубиной и частотой дискретизации 44 032 Гц от объёма звукового файла этой же записи, выполненной с 8-битовой глубиной и частотой дискретизации 11 008 Гц?
- 2.64. Когда была изобретена технология записи звука на компакт-диск (CD), стали решать, какова должна быть стандартная ёмкость диска. После того как группа разработчиков провела опрос, выяснилось, что самым популярным классическим произведением в Японии в те времена была 9-я симфония Бетховена, которая длилась 72–73 мин. Поэтому было решено, что компакт-диск должен быть рассчитан на 74 мин стереозвучания. Поскольку в аудио-CD применялась прямая оцифровка звука (без сжатия), была утверждена ёмкость компакт-диска 650 Мбайт. Какая глубина кодирования звука использовалась при оцифровке?
- 2.65. Какая глубина кодирования звука использовалась, если оцифрованная двухканальная звукозапись длительностью 2 минуты при частоте дискретизации 22 016 Гц занимает объём 20 640 Кбайт?

### Поисковая работа

- 2.1. Найдите в Интернете информацию об учёном Котельникове и о теореме его имени.

### Практическая работа на компьютере

- 2.3. Откройте музыкальный файл в звуковом редакторе, в котором отображается график звукозаписи (см. рис. 2.2). Сохраните звукозапись в разных форматах. Оцените объём звукового файла и его качество. Сделайте вывод о соотношении качества звука и размера файла. Какой формат, по-вашему, даёт приемлемое звучание?
- 2.4. Для небольшого фрагмента музыкального файла попробуйте сделать самостоятельную кодировку 3-5 отсчётов звуковой записи. Параметры кодирования выберите самостоятельно.
- 2.5. Сделайте звукозапись на диктофоне своего мобильного телефона. Найдите файл звукозаписи и попробуйте оценить, с какой частотой дискретизации произведено кодирование звука.

## 2.6

### Представление видеоданных

**Изучив эту тему, вы узнаете:**

- каковы общие подходы к представлению видеоданных в компьютере;
- какие форматы используются при хранении видеоданных.

Когда говорят о видеозаписи, прежде всего имеют в виду движущееся естественное изображение на экране телевизора или монитора. Не будем здесь рассматривать мультипликацию.

Преобразование оптического изображения в последовательность электрических сигналов осуществляется видеокамерой. Эти сигналы несут информацию о яркости и цветности отдельных участков изображения. Данные сохраняются на носителе в виде изменения намагниченности видеоленты (аналоговая форма) или последовательности кодовых комбинаций (слов) электрических импульсов (цифровая форма).

Процесс превращения непрерывного сигнала в набор кодовых слов называется **аналого-цифровым преобразованием**. Это сложный процесс, состоящий из:

- **дискретизации**, когда непрерывный сигнал заменяется последовательностью мгновенных отсчётов через равные промежутки времени;
- **квантования**, когда величина каждого отсчёта заменяется округлённым значением ближайшего уровня квантования;
- **кодирования**, когда значениям уровней квантования, полученным на предыдущем этапе, сопоставляются их порядковые номера в двоичном виде.

Видеозапись в несжатом виде можно трактовать как последовательность фотографий, сделанных в реальном времени через очень малые промежутки времени. Качество этих фотографий и видеозаписи в целом зависит от характеристик устройства записи. Объём видеозаписи складывается из объёмов отдельных графических изображений.

Современную видеозапись невозможно представить без звука. Поэтому в большинстве случаев видеофайл содержит не только видеоданные, но и звуковые данные. Наиболее распространёнными форматами для хранения видеозаписи являются AVI и MPEG.

**AVI** (*Audio Video Interleave*) — формат несжатого видео. Это наиболее ресурсоёмкий формат с минимальной потерей данных.

**MPEG** (*Moving Picture Expert Group*) — формат, предназначенный для сжатия звуковых и видеофайлов для их загрузки или пересылки, например через Интернет. Разработан Экспертной группой кинематографии, которая занимается разработкой стандартов кодирования и сжатия видео- и аудиоданных.



### Вопросы и задания

- 2.47. Опишите процесс кодирования видеоданных в самом упрощённом виде.
- 2.48. В каком из форматов — AVI или MPEG2 — одна и та же видеозапись будет занимать меньший объём?
- 2.49. Можно ли отделить звук от видеозаписи?
- 2.50. Как вы считаете, чем видеозапись отличается от мультипликации?



### Поисково-исследовательская работа

- 2.3. Задание выполняется в группах. Найдите в Интернете сведения об устройствах для двух видов видеозаписи: аналоговой и цифровой. Определите основные характеристики этих устройств. Проведите сравнительный анализ и дайте рекомендации по выбору видеокамеры для разных назначений.
- 2.4. Найдите в Интернете описание форматов файлов видеозаписи. Объясните, почему существуют разные форматы.

## 2.7

### Кодирование данных произвольного вида

*Изучив эту тему, вы узнаете:*

- как минимизировать длину двоичного кода;
- как кодировать данные в разных алфавитах.

В предыдущих темах вы познакомились со стандартными подходами к кодированию числовых, текстовых, графических и звуковых данных. Вы узнали, что в стандартной ситуации для кодирования этих типов данных выделяется определённое количество байт — 1, 2, 4, 8.

Однако иногда ставится задача закодировать данные кодом минимально возможной длины. Например, данные могут быть представлены в некоторой системе символов (алфавите), отличной от стандартных наборов символов, представленных таблицами ASCII или Unicode. Требуется сопоставить каждому символу двоичный код минимальной длины. Или, например, сопоставить некоторому набору данных двоичный код, опять-таки минимально возможный.

Такая задача возникает при хранении и обработке данных в условиях ограниченных вычислительных ресурсов, например при хранении и обработке данных на чипах магнитных карт, при решении вопросов о хранении информации в базах данных и в общем случае в задачах минимизации двоичного кода и минимизации времени обработки данных.

В общем виде задача кодирования ставится так: «Имеется некоторый набор значений (набор данных). Надо сопоставить каждому элементу набора двоичный код (кодовое слово), удовлетворяющий следующим требованиям:

- все коды должны быть одинаковой длины — состоять из одинакового количества бит. Это необходимо для вычисления объёма кодируемой информации и правильного распознавания кода;
- длина двоичного кода должна быть минимально необходимой для кодирования всех значений из набора».

Чтобы решить эту задачу, воспользуемся приведённым в параграфе 2.4 алгоритмом для определения глубины цвета. На основе этого алгоритма можно сделать вывод: минимальное количество бит, необходимое для кодирования  $N$  элементов набора, определяется из следующего неравенства:

$$2^{K-1} < N \leq 2^K,$$

где  $K$  — количество бит, необходимых для кодирования.

Из неравенства видно: чтобы определить число бит, надо найти степень 2, большую или равную  $N$ , но самую близкую к этому числу.

Другая (обратная) постановка задач, связанных с кодированием набора данных, звучит так: «Какое максимальное количество двоичных кодов можно составить из  $K$  бит?». Ответ выражается формулой

$$N = 2^K.$$



**Примеры 2.17, 2.18.** См. приложение 2.

Иногда для получения минимального объёма кодируемых данных используют код, в котором отдельным элементам сопоставлены двоичные коды разной длины. Однако при этом надо обеспечить однозначное распознавание при декодировании любой цепочки закодированных данных.



**Пример 2.19.** См. приложение 2.

Для кодирования может использоваться не только двоичный алфавит, но и алфавиты большей мощности, состоящие из трёх и более символов. Если для кодирования используется алфавит из трёх символов, то можно говорить о троичном кодировании (по аналогии с троичной системой счисления); если для кодирования используется 8 символов, то можно называть код восьмеричным, и т. д.

Чтобы определить, сколько существует различных кодовых слов длиной  $K$  символов, введём обозначения:  $P$  — мощность алфавита (количество символов в алфавите);  $K$  — фиксированная длина кодового слова, составленного из символов алфавита;  $N$  — количество различных кодов длиной  $K$ , которые можно составить из алфавита мощности  $P$ .

Тогда справедлива формула

$$N = P^K.$$

Если требуется с помощью алфавита мощности  $P$  закодировать  $N$  сигналов, то минимальная длина кода  $K$ , необходимая для кодирования, определяется неравенством

$$P^{K-1} \leq N \leq P^K.$$

Из неравенства видно, что число  $K$  является степенью числа  $P$ , равной или большей, чем  $N$ , но ближайшей к нему.

**Пример 2.20.** См. приложение 2.



## Равномерные и неравномерные коды

Особенностью рассмотренных кодов является то, что они имеют одинаковую длину кодовой комбинации. Например, при двоичном кодировании текста с помощью ASCII длина каждого кода равна 8, при двоичном кодировании 256-цветного изображения длина кода цвета равна 8 и т. д.

---

**Равномерный код** — это код, в котором все кодовые комбинации имеют одинаковую длину.

---



Использование равномерного кода позволяет однозначно декодировать сообщение принимающим устройством, что является его безусловным достоинством.

Пусть некоторый русский текст закодирован с помощью ASCII, т. е. кодовые комбинации всех букв имеют одинаковую длину. Анализ частоты встречаемости букв в русском тексте позволяет сделать вывод, что символ «о» встречается в русском языке наиболее часто, а символ «ф» — наиболее редко. Однако каждый из этих символов по таблице ASCII будет закодирован кодовой комбинацией одной длины. Анализ взятого в качестве примера русского текста позволяет утверждать, что недостатком равномерного кода является невозможность уменьшения объёма данных при хранении и передаче.

Уменьшить объём данных при хранении закодированного текста возможно, если букву, которая встречается очень часто, закодировать кодовой комбинацией меньшей длины. Таким образом, можно говорить о неравномерном коде.

---

**Неравномерный код** — это код, в котором кодовые комбинации имеют различную длину.

---



Примером неравномерного кода является телеграфный код американского изобретателя С. Морзе, который учёл частоту появления

А ··	Б ····	В ---	Г ---·	Д ---·
Е ··	Ж ····	З ---·	И ··	К ---·
Л ····	М --	Н ··	О ---	П ····
Р ···	С ···	Т -	У ···	Ф ····
Х ····	Ц ····	Ч ····	Ш ····	Щ ····
Ъ ······	Ы ····	Ь ····	Э ······	
	Ю ····	Я ····		
1 ·····	2 ·····	3 ·····	4 ·····	
5 ·····	6 ·····	7 ·····	8 ·····	
9 ·····	0 ·····			

Рис. 2.5. Код Морзе

букв в тексте. Алфавит Морзе состоит из двух символов: точки и тире длиной от двух до шести символов (рис. 2.5).

При этом возникла проблема неоднозначности при декодировании. Например, код из четырёх тире можно было бы декодировать по-разному: или как код одной буквы Ш, или как код сочетаний из двух букв — ММ, ОТ или ТО. Чтобы обеспечить однозначность декодирования, Морзе ввёл ещё один символ «пауза». Таким

образом, код Морзе не является двоичным.

Для того чтобы код при различных длинах кодовых комбинаций обеспечивал однозначность декодирования, необходимо соблюдение условия **Фано** при кодировании, названного в честь итальяно-американского ученого Р. Фано:

- никакое кодовое слово не является началом другого кодового слова;
- закодированное сообщение можно однозначно декодировать с конца, если выполняется обратное условие Фано: никакое кодовое слово не является окончанием другого кодового слова.

Например, пусть по каналу связи передаются сообщения, содержащие только 4 буквы «А», «Б», «В», «Г», и для передачи используется двоичный неравномерный код: А: 100, Б: 0, Г: 111. Выберем код для буквы «С», чтобы обеспечить однозначность декодирования (если таких кодов несколько, то следует выбрать код с минимальным значением).

- 1) код «В» не может начинаться с нуля, так как код «Б» состоит из одного знака — 0;
- 2) код «В» может начинаться с 1. Рассмотрим коды длиной 2: а) код 10 использовать нельзя, так начинается код буквы «А»; б) код 11 использовать нельзя, так начинается код буквы «Г»;
- 3) рассмотрим коды длиной 3: а) коды 100 и 111 уже заняты, б) коды 101 и 110 — свободны, условие Фано выполняется в обоих случаях;
- 4) меньшее значение имеет код 101.

Таким образом, для буквы «С» следует выбрать код 101.

Задача, связанная с минимизацией объёма данных при хранении и передаче в условиях ограниченных вычислительных

ресурсов, может решаться за счёт применения неравномерных кодов.

### Вопросы и задания

- 2.51. Какие условия должны выполняться при кодировании произвольного набора данных?
- 2.52. Какое минимальное количество бит требуется для двоичного кодирования набора данных из  $N$  элементов?
- 2.53. Сколько элементов можно закодировать с помощью  $K$  бит?
- 2.54. Какая минимальная длина кода необходима при кодировании набора данных из  $N$  элементов с помощью алфавита мощностью  $P$ ?
- 2.55. Для кодирования используется система, состоящая из  $P$  знаков. Сколько различных кодовых слов длиной  $K$  можно составить из этих символов?

### Задания для самостоятельной работы

- 2.66. Будем считать, что в магазине продаётся 114 наименований товаров. Какое наименьшее количество бит достаточно для двоичного кодирования наименований товаров в магазине?
- 2.67. Информационный объём сообщения, содержащего 128 символов, записанных в двоичном коде, составляет 64 байта. Найдите мощность алфавита, с помощью которого записано сообщение.
- 2.68. Определите информационный объём сообщения в битах, если сообщение состоит из 98 символов, мощность алфавита (число символов в алфавите) равна 28 и каждый символ закодирован минимальным двоичным кодом одинаковой длины.
- 2.69. В школе 30 классов. В каждом классе не более 30 учеников. Для кодирования информации о каждом ученике используется код класса и код порядкового номера ученика в электронном журнале. Какое минимальное количество двоичных разрядов необходимо выделить для кодирования информации о каждом ученике?
- 2.70. Для 5 букв русского алфавита заданы их двоичные коды (для некоторых букв — из двух бит, для некоторых — из трёх). Эти коды представлены в таблице:

К	М	Н	О	П
001	01	000	10	011

Определите, какой набор букв закодирован двоичной строкой 0010101000001.

- 2.71. В английском алфавите 26 букв. Слова кодируются только строчными буквами. Какое минимальное количество бит потребуется для кодирования одной буквы?
- 2.72. С учётом того что некоторые страны не имеют собственной валюты, будем считать, что в мире существует около 200 валют. Какое наименьшее количество бит достаточно для двоичного кодирования валют государств?





- 2.73. Для передачи по каналу связи сообщения, состоящего только из символов «А», «Б», «В» и «Г», используется посимвольное кодирование: «А»—00, «Б»—11, «В»—010, «Г»—011. Через канал связи передаётся сообщение: БВГББАБ. Закодируйте сообщение данным кодом. Полученную двоичную последовательность переведите в шестнадцатеричный вид.
- 2.74. Мальчик, чтобы безошибочно определять, кто звонит в дверь, предложил своим друзьям использовать сочетания из длинных и коротких звонков по 3 звонка. Он раздал всем друзьям индивидуальные комбинации, и у него осталось ещё 2 комбинации для родителей. Сколько друзей у мальчика?
- 2.75. Для передачи по каналу связи сообщения используют цифры 0, 1, 2, 3. Для кодирования каждой цифры используют секретный код: сначала каждой цифре сопоставляют соответствующее двоичное число из двух разрядов (при этом дописывают слева недостающие нули), а затем в этом числе 0 заменяют на 1, а 1 на 0. Закодируйте сообщение 101032 данным кодом. Полученную двоичную последовательность переведите в шестнадцатеричный вид.
- 2.76. Игорь составляет таблицу кодовых слов. Он использует 5-буквенные слова, в которых есть только буквы «П», «И», «Р», причём буква П появляется ровно 2 раза. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?
- 2.77. Игорь составляет таблицу кодовых слов. Он использует 5-буквенные слова, в которых есть только буквы «П», «И», «Р», «Т», «С», причём буква «П» появляется ровно 2 раза. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?
- 
- 2.78. По каналу связи передаются сообщения, содержащие только буквы «А», «Б», «В», «Г» и «Д». Для передачи информации используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность: «А» — 1; «Б» — 0100; «В» — 000; «Г» — 011.  
Укажите кратчайшее кодовое слово для буквы «Д», при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите код с наименьшим числовым значением.
- 2.79. По каналу связи передаются сообщения, содержащие только буквы «А», «Б», «В», «Г» и «Д». Для передачи информации используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность: «А» — 10; «Б» — 11; «В» — 000; «Г» — 001.  
Укажите кратчайшее кодовое слово для буквы «Д», при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите код с наименьшим числовым значением.