

Основные понятия информатики

Информатика = информация + автоматика

Информатика - это комплексная техническая наука, которая систематизирует приемы создания, сохранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими.

Предмет информатики как науки составляют:

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

Основные понятия информатики

Средства взаимодействия в информатике принято называть **интерфейсом**.

Программно-аппаратный интерфейс – это средства взаимодействия аппаратного и программного обеспечения.

Интерфейс пользователя - это средства взаимодействия человека с аппаратными и программными средствами.

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

Цель систематизации состоит в том, чтобы выделять, внедрять и развивать передовые, более эффективные технологии автоматизации этапов работы с данными, а также методически обеспечивать новые технологические исследования.

Основные понятия информатики

Основные направления информатики для практического применения :

- архитектура вычислительных систем (приемы и методы построения систем, предназначенных для автоматической обработки данных);
- интерфейсы вычислительных систем (приемы и методы управления аппаратным и программным обеспечением);
- программирование (приемы, методы и средства разработки комплексных задач);
- преобразование данных (приемы и методы преобразования структур данных);
- защита информации (обобщение приемов, разработка методов и средств защиты данных);
- автоматизация (функционирование программно-аппаратных средств без участия человека);
- стандартизация (обеспечение совместимости между аппаратными и программными средствами, между форматами представления данных, относящихся к разным типам вычислительных систем).

Понятие информации

В настоящее время не существует единого определения информации как научного термина. С точки зрения различных областей знания данное понятие описывается своим специфическим набором признаков.

Информация (от лат. *informatio*, разъяснение, изложение, осведомленность) — сведения о лицах, предметах, фактах, явлениях, процессах, событиях реального мира независимо от их представления.

Информация — это отображение знаний и фактов, используемых в различных областях человеческой деятельности

Знания — это осознанные и запомненные людьми свойства предметов, явлений и связей между ними, а также способы выполнения тех или иных действий для достижения нужных результатов.

Сообщение (сигнал) — это материальная форма информации.

Информация — это нематериальный смысл, извлекаемый человеком из сообщения.

Классификация информации

по **способу восприятия:**

- Визуальная — воспринимаемая органами зрения.
- Аудиальная — воспринимаемая органами слуха.
- Тактильная — воспринимаемая тактильными рецепторами.
- Обонятельная — воспринимаемая обонятельными рецепторами.
- Вкусовая — воспринимаемая вкусовыми рецепторами.

по **форме представления:**

- Текстовая — передаваемая в виде символов, предназначенных обозначать лексемы языка.
- Числовая — в виде цифр и знаков, обозначающих математические действия.
- Графическая — в виде изображений, предметов, графиков.
- Звуковая — устная или в виде записи и передачи лексем языка аудиальным путём.

Классификация информации

по назначению:

- Массовая — содержит тривиальные сведения и оперирует набором понятий, понятным большей части социума.
- Специальная — содержит специфический набор понятий, при использовании происходит передача сведений, которые могут быть не понятны основной массе социума, но необходимы и понятны в рамках узкой социальной группы, где используется данная информация.
- Секретная — передаваемая узкому кругу лиц и по закрытым (защищённым) каналам.
- Личная (приватная) — набор сведений о какой-либо личности, определяющий социальное положение и типы социальных взаимодействий внутри популяции.

Классификация информации

по значению:

- **Актуальная** — информация, ценная в данный момент времени.
- **Достоверная** — информация, полученная без искажений.
- **Понятная** — информация, выраженная на языке, понятном тому, кому она предназначена.
- **Полная** — информация, достаточная для принятия правильного решения или понимания.
- **Полезная** — полезность информации определяется субъектом, получившим информацию в зависимости от объёма возможностей её использования.

по истинности:

- **Истинная** - верно отражающая действительность в сознании человека.
- **Ложная** — не соответствующая действительности.

Информационные процессы

Данные — представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.

Информационный процесс — процесс получения, создания, сбора, обработки, накопления, хранения, поиска, распространения и использования информации.

Данные — это результат фиксации, отображения информации на каком-либо материальном носителе, то есть зарегистрированное на носителе представление сведений независимо от того, дошли ли эти сведения до какого-нибудь приёмника и интересуют ли они его (документы, книги, фильмы, картины и т.п.).

Кодирование — это представление сообщения в виде набора определенных знаков или сигналов.

Модель — это упрощенный образ объектов или явлений, содержащий только необходимые для решения задачи атрибуты этих объектов или явлений.

Операции с данными

- **ввод (сбор)** данных — накопление данных с целью обеспечения достаточной полноты для принятия решений;
- **формализация** данных — приведение данных, поступающих из разных источников, к одинаковой форме, для повышения их доступности;
- **фильтрация** данных — это отсеивание «лишних» данных, в которых нет необходимости для повышения достоверности и адекватности;
- **сортировка** данных — это упорядочивание данных по заданному признаку с целью удобства их использования;
- **архивация** — это организация хранения данных в удобной и легкодоступной форме;
- **защита** данных — включает меры, направленные на предотвращение утраты, воспроизведение и модификацию данных;
- **транспортировка** данных — приём и передача данных между участниками информационного процесса;
- **преобразование** данных — это перевод данных из одной формы в другую или из одной структуры в другую.

Информационная система

Информационная система - взаимосвязанная совокупность средств, методов и персонала, используемая для сохранения, обработки и выдачи информации с целью решения конкретной задачи.

В работе информационной системы можно выделить следующие этапы:

- **Зарождение данных** - формирование первичных сообщений, которые фиксируют результаты определенных операций, свойства объектов и субъектов управления, параметры процессов, содержание нормативных и юридических актов и т.п..
- **Накопление и систематизация данных** - организация такого их размещения, которое обеспечивало бы быстрый поиск и отбор нужных сведений, методическое обновление данных, защита их от искажений, потери, деформирования целостности и др.
- **Обработка данных** - процессы, вследствие которых на основании прежде накопленных данных формируются новые виды данных: обобщающие, аналитические, рекомендательные, прогнозные. Производные данные тоже можно обрабатывать, получая более обобщенные сведения.
- **Отображение данных** - представление их в форме, пригодной для восприятия человеком.

Измерение информации

● Формула Хартли для равновероятных событий (состояний):

$$I = \log_2 N \quad (\text{бит})$$

где I – количество информации (бит) в сообщении об одном событии

N – количество возможных состояний системы (событий)

Если $N=1$, то $I(N)=0$ – неопределенности нет, информации в сообщении о таком состоянии системы нет.

При $N=2$ $I=1$ бит – количество информации, уменьшающее неопределенность в 2 раза.

Формула Шеннона для событий (состояний) с разной вероятностью:

$$I = - \sum_{i=1}^n p_i \log_2 p_i$$

где n – число состояний системы,

p_i – вероятность перехода системы в i -е состояние, причем сумма всех p_i должна равняться 1

При $p_1 = p_2 = \dots = p_n = \frac{1}{n}$ $I = \log_2 n$ – формула Хартли

Измерение информации

Пример

Студент может получить на экзамене любую оценку. из четырех возможных (2,3,4,5). Сколько бит информации несет сообщение о том, что он получил двойку?

$$N=4$$

$I = \log_2 4 = 2$. Таким образом, сообщение о полученной оценке несет 2 бита информации.

Смысл полученного результата.

Допустим, нужно угадать, какую оценку студент получил за минимальное количество вопросов. Сколько вопросов гарантирует получение ответа?

Первый вопрос: «оценка выше 3? Ответ Нет. (Получен бит информации, неопределенность уменьшилась в 2 раза)

Второй вопрос – «оценка 3?» - НЕТ. (Получен еще бит информации, неопределенность уменьшилась в 2 раза).

Итого $1+1=2$ бита. Знание оценки снимает неопределенность (дает информацию) 2 бита.

8.417-2002)

1 байт(Б)=8 бит

1 Килобайт(КБ)=1024Б= 2^{10} байт

1 Мегабайт(МБ)=1024КБ= 2^{20} байт

1 Гигабайт(ГБ)=1024МБ= 2^{30} байт

1 Терабайт(ТБ)=1024ГБ= 2^{40} байт

1 Петабайт(ПБ)=1024ТБ= 2^{50} байт

1 Эксабайт(ЭБ)=1024ПБ= 2^{60} байт

1 Зеттабайт(ЗБ)=1024ЭБ= 2^{70} байт

1 Йоттабайт(ЙБ)=1024ЗБ= 2^{80} байт

Человечеству потребовалось 300 тысяч лет, чтобы создать первые 12 эксабайт информации, зато вторые 12 эксабайт были созданы всего за два года.

В 2012 году среднемесячный объём глобального IP-трафика достиг уровня 44 эксабайта.

Единицы информации (МЭК и IEEE 1541—2002)

1 Киббайт(КиБ)=1024Б= 2^{10} байт

1 Мебибайт(МиБ)=1024КБ= 2^{20} байт

1 Гиббайт(ГиБ)=1024МБ= 2^{30} байт

1 Тебибайт(ТиБ)=1024ГБ= 2^{40} байт

1 Пебибайт(ПиБ)=1024ТБ= 2^{50} байт

1 Эксбибайт(ЭиБ)=1024ПБ= 2^{60} байт

Скорость передачи информации

Бит в секунду (англ. *bits per second*, **bps**) — базовая единица измерения **скорости** передачи информации.

Битами в секунду измеряется эффективный объём информации, без учёта служебных битов.

Для обозначения больших скоростей передачи применяют более крупные единицы, образованные с помощью приставок системы СИ *кило-*, *мега-*, *гига-* и т. п. получая:

- *Килобиты в секунду* — Кбит/с (Kbps, Kbit/s или Kb/s) = 1000 бит/с
 - *Мегабиты в секунду* — Мбит/с (Mbps, Mbit/s или Mb/s) = 10^6 бит/с
 - *Гигабиты в секунду* — Гбит/с (Gbps, Gbit/s или Gb/s) = 10^9 бит/с
- ... и т. д.

Часто путают Mb/s и MB/s ($1 \text{ MB/s} = 8 \text{ Mb/s}$), поэтому рекомендуется использовать сокращение Мбит/с (Mbit/s).

Скорость передачи информации

●Пример.

Определить размер файла в МБ если передача его в вычислительной сети с пропускной способностью 100Мбит/с заняла 16с.

Решение:

Количество переданной информации :

$10^8 \times 16$ (бит)

Переводим биты в МБ:

$$\text{Размер} = \frac{16 \times 10^8}{8 \times 1024 \times 1024} \approx 191 \text{ МБ}$$

КОМПЬЮТЕРЕ

Кодирование информации — это процесс преобразования её из формы, удобной для непосредственного использования, в форму, удобную для передачи, хранения или обработки.

Недостатки **аналогового кодирования** информации:

- ограниченная точность
- необходимость масштабирования данных и результатов
- восприимчивость к помехам
- сложность точного воспроизведения
- искажения сохраненной информации со временем

и т.п.

При **двоичном цифровом кодировании** любая информация представляется сериями сигналов, каждый из которых принимает одно из двух возможных значений.

Эти возможные значения сигнала обозначаются цифрами 0 и 1.

Системы счисления

Система счисления – это совокупность знаков, называемых цифрами, и правил их использования, применяемых для представления чисел.

В **непозиционной** системе счисления цифра всегда соответствует одному числу.

- **Унарная** – для записи любого числа используется только одна цифра:

||| - 3

||||||| - 12

Знак |, являющийся в данном случае цифрой (а также чёрточка, зарубка, камушек и т.п.) всегда означает 1. Число в унарной системе – количество цифр.

- **Римская** – в качестве цифр используются буквы латинского алфавита:

I = 1 C = 100

V = 5 D = 500

X = 10 M = 1000

L = 50

Непозиционные системы счисления в основном используются в быту

Системы счисления

В **позиционной** системе счисления значение, передаваемое цифрой, зависит от её места (позиции) в записи числа.

Любое число в позиционной системе счисления можно записать в виде выражения:

$$N_q = a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \dots + a_2 \cdot q^2 + a_1 \cdot q + a_0,$$

где $a_{n-1}, a_{n-2}, \dots, a_2, a_1, a_0$ – цифры,

q – основание системы счисления,

причём все $a < q$.

n – количество цифр в записи числа.

Примеры:

$$125_{10} = 1 \cdot 10^2 + 2 \cdot 10 + 5 \quad 512_{10} = 5 \cdot 10^2 + 1 \cdot 10 + 2$$

$q=10; n=3; 1, 2, 5$ – цифры $q=10; n=3; 1, 2, 5$ – цифры

СЧИСЛЕНИЯ

Перевод можно выполнить по формуле

$$N_q = a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \dots + a_2 \cdot q^2 + a_1 \cdot q + a_0$$

Примеры:

1). Перевести 512_8 в десятичную систему счисления

$$n=3, q=8$$

$$512_8 = 5 \cdot 8^2 + 1 \cdot 8 + 2 = 320 + 8 + 2 = 330_{10}$$

2). Какому десятичному числу соответствует 11010110_2 ?

$$q=2; n=8$$

$$11010110_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 = 128 + 64 + 0 + 16 + 0 + 4 + 2 + 0 = 214_{10}$$

2-й способ:

128 64 32 16 8 4 2 1

$$11010110_2 = 128 + 64 + 16 + 4 + 2 = 214_{10}$$

Перевод из десятичной системы в двоичную

1-й способ.

Исходное десятичное число представляем как сумму степеней двойки. Затем составляем двоичную запись числа, заменяя отсутствующие степени двойки нулями.

Например, дано число 117_{10} . Следует найти его двоичное представление.

$$117_{10} = 64 + 53 = 2^6 + 32 + 21 = 2^6 + 2^5 + 16 + 5 = 2^6 + 2^5 + 2^4 + 0 + 2^2 + 0 + 1 = 1110101_2$$

Для проверки достаточно выполнить обратный перевод полученного двоичного числа в десятичную систему любым способом.

64 32 16 8 4 2 1

$$1110101_2 = 64 + 32 + 16 + 4 + 1 = 117_{10}$$

Перевод из десятичной системы в двоичную

2-й способ.

Исходное десятичное число последовательно делим на новое основание системы счисления (2) до достижения нулевого результата, записывая все остатки от деления, в том числе и нулевые. После этого выписываем все полученные остатки от деления в порядке их появления справа налево. Это и будет двоичное представление нашего числа.

Например, дано число 117_{10} . Найдём его двоичное представление описанным способом. Полученные остатки от деления будем записывать рядом с частным в скобках.

$$117 : 2 = 58 (1)$$

$$58 : 2 = 29 (0)$$

$$29 : 2 = 14 (1)$$

$$14 : 2 = 7 (0)$$

$$7 : 2 = 3 (1)$$

$$3 : 2 = 1 (1)$$

$$1 : 2 = 0 (1)$$

1 1 1 0 1 0 1

$$117_{10} = 1110101_2$$

счисления

Системы счисления, у которых основание одной является степенью основания другой, называются **родственными**.

Для двоичной системы (основание 2) родственными будут четверичная ($4=2^2$), восьмеричная ($8=2^3$), шестнадцатеричная ($16=2^4$) и т.п. системы счисления.

Восьмеричная и шестнадцатеричная системы счисления обычно используются для **более компактной записи** двоичной информации. При этом одна восьмеричная цифра заменяет три двоичных разряда, а одна шестнадцатеричная – четыре.

В шестнадцатеричной системе кроме цифр 0..9 используются цифры A, B, C, D, E, F, соответствующие числам от 10 до 15

Шестнадцатеричная и восьмеричная системы

СЧИСЛЕНИЯ

Двоичное	Восьмерич.	Десятич.
000	0	0
001	1	1
010	2	2
011	3	3
100	4	4
101	5	5
110	6	6
111	7	7

Двоичное	Шестнадцатерич.	Десятич.
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

СЧИСЛЕНИЯ

Пример.

Имеется число 10110011_2 . Требуется записать его в восьмеричной и шестнадцатеричной системах счисления и выполнить проверку.

Решение.

Для получения восьмеричного представления следует разбить исходное число на группы из трёх двоичных цифр, начиная с младших разрядов, т.е. справа. Поскольку в числе всего восемь цифр, то в последней, левой группе, допишем 0 слева, который на значение числа не повлияет. В результате получим: 010 110 011.

По таблице находим восьмеричные цифры, соответствующие каждой группе, и записываем их в том же порядке: 2 6 3 .

То есть $10110011_2 = 263_8$.

Проверка:

128 64 32 16 8 4 2 1

$$1\ 0\ 1\ 1\ 0\ 0\ 1\ 1_2 = 128 + 32 + 16 + 2 + 1 = 179_{10}$$

$$263_8 = 2 \cdot 8^2 + 6 \cdot 8 + 3 = 128 + 48 + 3 = 179_{10}$$

Счисления

Для получения шестнадцатеричного представления следует разбить исходное двоичное число на группы из четырёх двоичных цифр, начиная с младших разрядов, т.е. справа: 1011 0011.

По таблице находим шестнадцатеричные цифры, соответствующие каждой группе, и записываем их в том же порядке: В 3 . То есть $10110011_2 = В3_{16}$.

Для проверки переведём полученное шестнадцатеричное число в десятичную систему счисления, при этом десятичное значение цифры В = 11 найдём по таблице:

$$В3_{16} = 11 \cdot 16 + 3 = 176 + 3 = 179_{10} .$$

Такое же значение было получено ранее для двоичного и восьмеричного представлений.

$$\text{Итак, } 10110011_2 = 263_8 = В3_{16} .$$

знаком

За знак числа в компьютерном представлении отвечает самый **левый** разряд в записи числа. Если в этом разряде записан **0**, то **число положительное**, если **1**, то **отрицательное**. При этом в компьютере любое число занимает целое количество байт.

Таким образом, если в памяти записано число со знаком в виде **00000001**, то это **+1**, т.к. в самом левом разряде **0**, соответствующий знаку **+**.

Однако, **10000001** – это не **-1**.

Если сложить **+1** и **-1**, должен получиться **0**. Сложим наши двоичные числа.

$$\begin{array}{r} 00000001 \\ + \\ \underline{10000001} \\ 10000010 \end{array}$$

Двоичное представление числа **0** в пределах байта **00000000**, а полученный результат не равен **0**, то есть **10000001 \neq -1**.

знаком

Для представления целых отрицательных чисел в компьютере обычно используется запись в так называемом **дополнительном** коде.

Двоичное представление отрицательного числа получается по следующему алгоритму:

- 1). Находится двоичное представление такого же по абсолютной величине положительного числа
- 2). Полученное число при необходимости слева дополняется нулями до нужного количества разрядов: 8, 16, 24, 32 и т.д. – целое количество байт (8-разрядных ячеек)
- 3). Полученное в п.2 число инвертируется, т.е. нули в нём заменяются на единицы, а единицы на нули (получаем так называемый обратный код)
- 4). К полученному в п.3 числу прибавляется (не приписывается!).

знаком

Пример 1.

Записать однобайтовое двоичное представление числа -123_{10} .

1). Находим двоичное представление числа $+123$:

$$123_{10} = 64+32+16+8+2+1 = 1111011_2.$$

2). Поскольку требуется получить однобайтовое представление числа, дополняем полученное число слева одним нулём, чтобы получить 8-разрядов – 01111011 . Обратите внимание, что в левом разряде 0 , что соответствует знаку $+$.

3). Инвертируем все разряды:

до инвертирования 01111011

после инвертирования 10000100

4). Прибавляем 1 к полученному результату:


$$\begin{array}{r} 10000100 \\ + \quad \underline{1} \\ \hline 10000101 \end{array}$$

Итак, однобайтовое представление числа $-123 = 10000101$. Обратите внимание, что в левом разряде 1 , что соответствует знаку $-$.

знаком

Проверка.

Чтобы проверить правильность результата, сложим двоичные однобайтовые представления чисел +123 и -123. Должно получиться однобайтовое представление нуля - 00000000 .

$$\begin{array}{r} 01111011 \\ +10000101 \\ \hline 00000000 \end{array}$$


Строго говоря, в результате получается число 1 00000000, но единица оказывается за пределами однобайтовой ячейки памяти и просто теряется, то есть в пределах байта получится 00000000.

Итак, однобайтовое представление числа -123 = 10000101.

знаком

Пример 2.

Записать число -215 в двухбайтовом двоичном представлении и выполнить проверку.

Решение.

Два байта – это 16 разрядов, то есть число должно получиться 16-разрядное, причём в левом разряде должна стоять единица.

1). Находим двоичное представление числа $+215_{10}$, используя последовательное деление:

$$215 : 2 = 107 (1)$$

$$107 : 2 = 53 (1)$$

$$53 : 2 = 26 (1)$$

$$26 : 2 = 13 (0)$$

$$13 : 2 = 6 (1)$$

$$6 : 2 = 3 (0)$$

$$3 : 2 = 1 (1)$$

$$1 : 2 = 0 (1)$$

$$215_{10} = 11010111_2$$

знаком

2). Дополняем полученное число слева нулями, чтобы получить 16 разрядов: 0000000011010111.

3). Инвертируем все разряды:

до инвертирования 0000000011010111

после инвертирования 111111100101000

4). Прибавляем 1 к полученному результату:

$$\begin{array}{r} 111111100101000 \\ + \quad \underline{\quad\quad\quad 1} \\ 111111100101001 \end{array}$$

Итак, двухбайтовое представление числа $-215 = 111111100101001$.
Обратите внимание, что в левом разряде 1, что соответствует знаку -.

Проверка.

$$\begin{array}{r} 0000000011010111 \\ + \underline{111111100101001} \\ 0000000000000000 \end{array}$$



Двоичное представление целых чисел со знаком

Одно и то же содержимое байта может соответствовать разным числам.

Например, в байте хранится 11101110.

Если это число без знака, то получим:

$$11101110_2 = 128 + 64 + 32 + 8 + 4 + 2 = 238_{10} .$$

Если это число со знаком, то оно равно -18 (в левом разряде 1).

Выполним в обратном порядке действия 1)..4). Сначала нужно вычесть из числа 1 или, что то же самое сложить число с -1 (1111111₂):

$$\begin{array}{r} 11101110 \\ + \underline{11111111} \\ \hline 11101101 \end{array}$$

Сложение выполняем только в пределах байта.

Результат инвертируем: 00010010 – это положительное число, равное по абсолютной величине исходному отрицательному. Нули слева можно не учитывать. Таким образом,

$$10010_2 = 16 + 2 = 18_{10} .$$

$$11101110_2 = -18_{10}$$

Пределные значения целых чисел в памяти компьютера

При представлении целых чисел в компьютере всегда указывается, со знаком они или без и сколько байт отводится в памяти для хранения числа.

Размер числа в памяти, байт	Пределные значения чисел	
	без знака	со знаком
1	0...255	-128...+127
2	0...65 535	-32 768...+32 767
3	0... 16 777 215	-8 388 608... +8 388 607
4	0... 4 294 967 295	-2 147 483 648... +2 147 483 647

Наименьшее отрицательное однобайтовое число $10000000 = -128_{10}$.
Здесь левый разряд выполняет и роль знака, и роль значащей цифры.

Представление действительных чисел

Действительное число может иметь дробную часть. В компьютере такие числа могут представляться двумя способами:

- с фиксированной запятой (точкой)
- с плавающей запятой (точкой).

Для представления числа в форме с фиксированной запятой используются только степени числа 2. При этом дробная часть числа представляется как сумма отрицательных степеней двойки ($2^{-1}=1/2$; $2^{-2}=1/4$ и т.д.).

Для перевода числа из двоичного представления с фиксированной запятой в десятичное представление можно использовать формулу :

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

Здесь N_q – само число, q – основание системы счисления, a – цифры данной системы счисления, n – число разрядов целой части числа, m – число разрядов дробной части числа.

Представление действительных чисел

Пример.

Требуется перевести $100111,0111_2$ в десятичную систему счисления.

Решение.

Для перевода воспользуемся формулой

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

$$q=2; n=6; m=4$$

$$100111,0111_2$$

$$= 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} = \\ 32 + 4 + 2 + 1 + 1/4 + 1/8 + 1/16 = 39,4375_{10}$$

Формат с фиксированной запятой иногда называют форматом с фиксированной точностью, так как в памяти компьютера для хранения таких чисел выделяется фиксированное количество разрядов для хранения дробной части.

Если, например, на дробную часть выделено 8 разрядов, то точность представления чисел будет на уровне $2^{-8} = 1/256 \approx 0,0039$.

Если на дробную часть выделено 16 разрядов, то точность представления чисел будет на уровне $2^{-16} = 1/65536 \approx 0,000015$.

Представление действительных чисел

Если нужно перевести смешанное число (имеющего как целую так и дробную части) из десятичной в двоичную систему,

- целую часть переводим любым ранее описанным способом.
- для перевода дробной части числа необходимо умножить ее на 2, затем, отбрасывая у результата целую часть, продолжать процесс умножения до тех пор, пока дробная часть произведения не окажется равной нулю или не будет достигнута нужная точность дроби.
- Целые части произведений, записанные после запятой в прямой последовательности (начиная с первого), образуют дробную часть числа в системе счисления с основанием 2.

Пример 1. Перевести число 12,75 в двоичную систему с фиксированной запятой.

Решение. $12_{10} = 8+4 = 1100_2$

$$0,75 \times 2 = 1,5 \quad | \quad 0,75_{10} = 0,11_2$$

$$0,5 \times 2 = 1 \quad |$$

Ответ: $12,75_{10} = 1100,11_2$

Представление действительных чисел

Не любая дробная часть числа может быть точно записана в двоичном виде.

Пример 2. Перевести число $0,3_{10}$ в двоичную систему с фиксированной запятой.

Решение.

$$0,3 \times 2 = 0,6$$

$$0,6 \times 2 = 1,2$$

$$0,2 \times 2 = 0,4$$

$$0,4 \times 2 = 0,8$$

$$0,8 \times 2 = 1,6$$

$$0,6 \times 2 = 1,2$$

....

Дальше будут циклично появляться уже имевшиеся результаты, и процесс будет повторяться бесконечно. То есть в результате получим:

$$0,3_{10} = 0,0100110011001..._2$$

Чтобы указать, что дробь периодическая, её можно записать $0,0(1001)_2$

Представление действительных чисел

Форма с плавающей запятой использует представление действительного числа N в виде произведения **мантиссы** m на **основание** системы счисления 10 в некоторой целой степени p , которую называют **порядком**:

$$N = m \times 10^p$$

Например, число $139,76$ можно записать в виде: $0,13976 \times 10^3$. Здесь $m = 0,13976$ – мантисса, $p = 3$ – порядок. Порядок указывает, на какое количество позиций и в каком направлении должна «переплыть», т.е. сместиться десятичная запятая в мантиссе. Отсюда название «плавающая запятая». Однако справедливы и следующие равенства:

$$139,76 = 13,976 \times 10^1 = 1,3976 \times 10^2 = 0,013976 \times 10^4 = 13976 \times 10^{-2}$$

Чтобы не было неоднозначности, в компьютере используют нормализованное представление числа в форме с плавающей точкой. Мантисса в нормализованном представлении должна удовлетворять условию:

$$0,1 \leq m < 1,$$

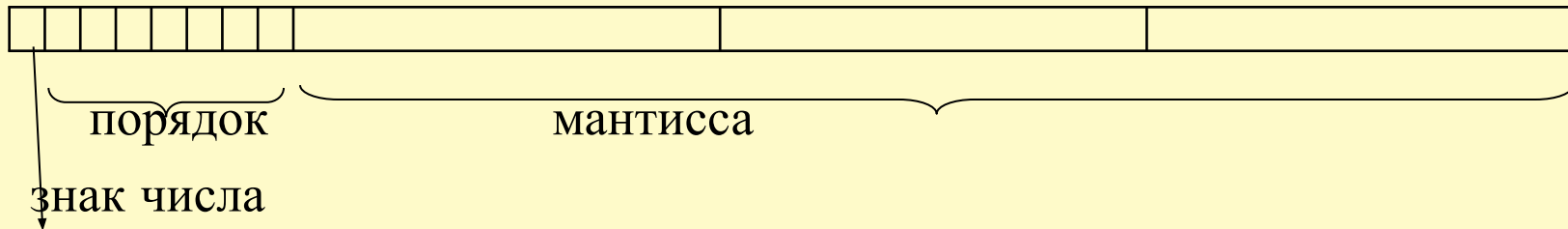
то есть мантисса меньше единицы и первая значащая цифра – не ноль. Следовательно, для рассмотренного числа нормализованным представлением будет: $0,13976 \times 10^3$.

Представление действительных чисел

В компьютере для хранения действительных чисел применяются различные варианты представления чисел в форме с плавающей запятой.

Пример для 4-байтового числа.

байт байт байт байт



В первом байте хранится знак всего числа (левый бит) и порядок (как целое число со знаком).

Остальные 3 байта (24 бита) используются для хранения значащих цифр мантиссы. При этом мантисса хранится как целое число без знака, то есть вместо числа 0,13976 записывается число 13976.

Представление действительных чисел

Предельные значения хранимых таким образом чисел.

Наименьшим по абсолютной величине числом будет 0.

Наибольшим по абсолютной величине числом, представляющим мантиссу будет $2^{24}-1=16\ 777\ 215$

Наибольшее значение порядка для двоичного семибитного числа со знаком равно $2^6-1=63$.

Следовательно, наибольшее число $0,16\ 777\ 215 \cdot 10^{63} \approx 1,7 \cdot 10^{62}$.

При этом разряды мантиссы позволяют точно сохранить любые 7 значащих цифр и 8 значащих цифр не более, чем 16 777 215. Поэтому говорят, что такое представление числа обеспечивает **точность в 7-8 значащих цифр**.

Наименьшее значение порядка для двоичного семибитного числа со знаком равно -64, поэтому наименьшим по абсолютной величине дробным числом, которое можно представить таким образом, является $0,1 \cdot 10^{-64} = 1 \cdot 10^{-65}$.

Поскольку знак числа хранится отдельно от мантиссы, для отрицательных чисел будут такие же по абсолютной величине предельные значения, только в самом левом бите числа будет 1, а не 0.

Представление действительных чисел

Итоги по числовым форматам.

Достоинством формата с **фиксированной запятой** является простота выполнения основных математических операций. С точки зрения процессора нет разницы между целой и дробной частями: операции выполняются по одному алгоритму.

Формат с **фиксированной запятой** в настоящее время в основном используется для ускорения вычислений в случаях, когда не требуется высокая точность, а также для обеспечения минимальной поддержки дробных чисел на целочисленном процессоре.

Недостаток **фиксированной запятой** — очень узкий диапазон чисел, с угрозой переполнения на одном конце диапазона и потерей точности вычислений на другом.

Из-за сложности выполнения математических операций с числами с **плавающей запятой**, раньше в компьютере использовался второй процессор (сопроцессор). В современных процессорах имеется блок для вычислений с плавающей запятой

Представление текстовой информации

Каждому символу соответствует свой код, который представляет собой целое число без знака.

Таблица, содержащая символы и соответствующие им коды, называется кодировочной таблицей.

В таблице **ASCII** (American Standard Code for Information Interchange) используются 8-битные коды, что позволяет закодировать 256 различных символов.

Другие 8-битные таблицы: ANSI, КОИ-8 , ISO, Mac

Unicode - стандарт кодирования символов, позволяющий представить знаки почти всех письменных языков.

В первой версии Unicode используются 16-битные коды.

UTF (Unicode Transformation Format - формат преобразования Юникода) – способ представления кода.

Примеры: UTF-8, UTF-16, UTF-32.

Представление графической информации

Растровая графика – изображение хранится как набор точек, каждая из которых может иметь свой цвет.

Каждому цвету соответствует свой код, который хранится как целое число без знака.

Цветовая модель — математическая модель описания представления цветов в виде последовательностей чисел, называемых цветовыми составляющими или координатами.

Все возможные значения цветов, задаваемые моделью, определяют **цветовое пространство**.

Примеры цветовых моделей: RGB, CMYK, HSV (HSB), Lab.

Векторная графика — способ представления изображений, основанный на использовании элементарных геометрических объектов (точки, линии, сплайны, многоугольники). Объекты векторной графики являются графическими изображениями математических функций.