

Н. Н. СОКОЛОВСКАЯ, А. В. ЕРОШЕНКО

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

ЧАСТЬ 1

ПРЕДСТАВЛЕНИЕ ЧИСЛОВОЙ ИНФОРМАЦИИ В КОМПЬЮТЕРАХ

ОМСК 2010

Министерство транспорта Российской Федерации
Федеральное агентство железнодорожного транспорта
Омский государственный университет путей сообщения

Н. Н. Соколовская, А. В. Ерошенко

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

Часть 1

ПРЕДСТАВЛЕНИЕ ЧИСЛОВОЙ ИНФОРМАЦИИ В КОМПЬЮТЕРАХ

Утверждено редакционно-издательским советом университета
в качестве методических указаний для самостоятельной работы студентов
по дисциплине «Информатика»

Омск 2010

УДК 004. 432
ББК 32.988-5я7
С59

Теоретические основы информатики. Часть 1. Представление числовой информации в компьютерах: Методические указания для самостоятельной работы студентов по дисциплине «Информатика» / Н. Н. Соколовская, А. В. Ерошенко; Омский гос. ун-т путей сообщения. Омск, 2010. 17 с.

Содержат основы представления числовой информации в компьютерах. Рассмотрены различные типы данных, используемых при составлении программ. Приведены практические рекомендации и примеры кодирования целых чисел без знака, целых чисел со знаком и вещественных чисел; исходные данные к индивидуальным заданиям.

Предназначены для студентов первого курса всех специальностей очной и заочной форм обучения.

Библиогр.: 4 назв. Табл. 5. Рис. 2.

Рецензенты: доктор техн. наук, профессор Е. И. Сковородников;
канд. техн. наук, доцент О. П. Шафеева.

ОГЛАВЛЕНИЕ

Введение.....	5
1. Представление числовой информации.....	6
1.1. Представление целых чисел.....	6
1.1.1. Целые числа без знака.....	6
1.1.2. Целые числа со знаком.....	7
1.2. Представление вещественных чисел.....	9
2. Контрольные вопросы	12
3. Задания.....	13
Библиографический список.....	16

ВВЕДЕНИЕ

В информатике выделяются два направления – теоретическое и прикладное. Исследования в области теоретической информатики обеспечивают выявление и формулировку общих законов, касающихся информации и информационных процессов; определение принципов функционирования технических систем, связанных с информационными процессами и обработкой дискретной информации, а также построение методологии создания и применения информационных моделей. Прикладная информатика обеспечивает непосредственное создание информационных систем и программного обеспечения для них, а также их применение для решения практических задач.

В данных методических указаниях рассматриваются теоретические основы дисциплины «Информатика».

Теоретическая информатика – дисциплина, использующая методы математики. Поскольку любая информация может быть представлена в дискретном виде, для описания информационных процессов может быть применен аппарат дискретной математики, в частности, теория автоматов и алгоритмов, теория графов, теория формальных языков и грамматик, алгебра логики, теория информации и др.

Вся информация, которая попадает в компьютер, преобразуется в последовательность электрических импульсов. Наличие импульса принято условно обозначать «1», а его отсутствие – «0». Такой способ кодирования информации называется двоичным, или бинарным. Один двоичный символ получил название бит (*bit* – от английского словосочетания *binary digit* – двоичная цифра). Таким образом, двоичное кодирование – это представление информации при помощи минимально возможного числа элементарных символов. Двоичное кодирование автоматически дает способ кодирования информации любого вида (числовой, символьной, логической, графической, звуковой) в двоичной системе счисления.

1. ПРЕДСТАВЛЕНИЕ ЧИСЛОВОЙ ИНФОРМАЦИИ

Существует два основных формата представления чисел в памяти компьютера, один из них применяется для кодирования целых чисел (представление числа в формате с фиксированной точкой), второй – для задания некоторого подмножества действительных чисел (представление числа в формате с плавающей точкой). Рассмотрим каждый из форматов подробнее.

1.1. Представление целых чисел

Любое целое число можно рассматривать как вещественное, но с нулевой дробной частью, т. е. можно было бы ограничиться представлением в компьютере вещественных чисел и реализацией арифметических операций над ними, однако для эффективного использования памяти ЭВМ, повышения скорости выполнения вычислений и введения операции целочисленного деления целые числа представляются специально для них предназначенными способами.

Для компьютерного представления целых чисел обычно применяются несколько различных способов, отличающихся друг от друга количеством двоичных разрядов и наличием или отсутствием знакового разряда.

Целые числа в компьютере хранятся в памяти в формате с *фиксированной запятой*. В этом случае каждому разряду ячейки памяти соответствует всегда один и тот же разряд, а «запятая» «находится» справа после младшего разряда, т. е. вне разрядной сетки.

1.1.1. Целые числа без знака

Рассмотрим кодирование целых чисел без знака на примере данных типа *byte* в языке *Basic* и *unsigned char* в языке *C++*, занимающих в памяти один байт.

Для получения компьютерного (внутреннего) представления однобайтового целого неотрицательного числа достаточно перевести его в двоичную систему счисления и полученный результат, называемый прямым кодом числа, дополнить слева нулями до восьми битов.

Минимальное число представляется нулями во всех разрядах и равно нулю. Максимально представимому числу соответствуют единицы во всех разря-

дах ячейки (двоичное число, состоящее из восьми единиц), оно равно 255 ($2^8 - 1$). Примеры кодирования однобайтовых целых чисел без знака приведены в табл. 1.

Однобайтовые целые неотрицательные числа могут применяться, например, для организации различных счетчиков, записи адресов ячеек, даты и времени, размеров графических изображений в пикселях.

Для улучшения читаемости внутреннего представления числа его записывают в шестнадцатеричной системе счисления.

Таблица 1

Примеры кодирования целых чисел без знака

Целое число без знака	Внутреннее представление числа	
	двоичная система счисления	шестнадцатеричная си- стема счисления
0	0000 0000	00
1	0000 0001	01
17	0001 0001	11
255	1111 1111	FF

1.1.2. Целые числа со знаком

Рассмотрим кодирование целых чисел со знаком на примере данных типа *integer* в языке *Basic* и *int* в языке *C++*, занимающих в памяти два байта (16 битов).

Каждый из 16 битов имеет определенное назначение, форма представления целого числа со знаком показана на рис. 1. Под знак отводится старший разряд ячейки: 0 – для положительных чисел, 1 – для отрицательных.

Для представления в компьютере целых чисел со знаком применяют дополнительный код, который позволяет заменить арифметическую операцию вычитания операцией сложения, что существенно увеличивает скорость вычислений.

Для того чтобы понять, что такое дополнительный код, рассмотрим прямой и обратный коды.

Замечание 1. Для положительных чисел все три кода совпадают с двоичным представлением числа с помощью шестнадцати двоичных разрядов, при этом в пустые разряды записываются нули.



Рис. 1. Форма представления целого числа со знаком

Представим *алгоритм* получения дополнительного шестнадцатиразрядного двоичного кода отрицательного числа.

1) Записать прямой код отрицательного числа в 16 двоичных разрядах. Для этого модуль целого отрицательного числа надо перевести в двоичную систему счисления и дополнить полученный результат слева нулями до 16 битов.

2) Записать обратный код отрицательного числа в 16 двоичных разрядах. Для этого значения всех разрядов прямого кода инвертировать (все нули заменить на единицы, а все единицы – на нули).

3) Записать дополнительный код отрицательного числа в 16 двоичных разрядах. Для этого к обратному коду, рассматриваемому как шестнадцатиразрядное неотрицательное двоичное число, прибавить единицу.

Замечание 2. Обратный код отрицательного числа является дополнением модуля этого числа до числа $2^{16} - 1$, а дополнительный код – до числа 2^{16} .

Примеры представления двухбайтовых целых чисел со знаком приведены в табл. 2.

Минимальное отрицательное число, которое можно представить с помощью двух байтов, равно -32768 .

Максимально представимому положительному числу соответствуют единицы во всех разрядах ячейки (двоичное число, состоящее из нуля (в знаковом разряде) и пятнадцати единиц), оно равно $32767 (2^{15} - 1)$.

Примеры представления двухбайтовых целых чисел со знаком

Вид кода	Целое число со знаком		
	−32768	−23	18
Прямой	1000 0000 0000 0000	0000 0000 0001 0111	0000 0000 0001 0010
Обратный	0111 1111 1111 1111	1111 1111 1110 1000	0000 0000 0001 0010
Дополнитель- ный (внутреннее представление)	1000 0000 0000 0000	1111 1111 1110 1001	0000 0000 0001 0010
	80 00	FF E9	00 12

1.2. Представление вещественных чисел

Рассмотрим представление вещественных чисел на примере данных типа *single* в языке *Basic* и *float* в языке *C++*, занимающих в памяти четыре байта (32 бита). Каждый из 32 битов имеет определенное назначение (рис. 2).

Вещественные числа хранятся и обрабатываются в компьютере в формате *с плавающей запятой*. Формат чисел с плавающей запятой базируется на экспоненциальной форме записи, в которой может быть представлено любое число.



Рис. 2. Форма представления вещественного числа

В общем случае в формате с плавающей точкой число представляется в виде произведения двух сомножителей:

$$R = mP^n,$$

где m – мантисса числа;

P – основание системы счисления;

n – порядок, указывающий, на какое количество позиций и в каком направлении должна сместиться точка, отделяющая дробную часть в мантиссе.

Например, число 5,14 может быть записано в виде $0,514 \cdot 10^1$ или $51,4 \cdot 10^{-1}$ и т. д. Запятая (десятичная точка) перемещается («плавает») вправо и влево в зависимости от порядка числа.

Представим *алгоритм* получения тридцатидвухразрядного двоичного кода вещественного числа.

1) Перевести число в двоичную систему счисления.

2) Выполнить нормализацию числа.

Нормализованная запись отличного от нуля вещественного числа – это запись вида $a = \pm m \times 2^p$, где p – порядок, целое число (положительное, отрицательное или ноль); m – мантисса, причем справедливо соотношение: $1 \leq m < 2$.

Нормализация состоит в смещении дробной запятой влево или вправо таким образом, чтобы в целой части числа всегда стояла единица, при этом изменяется значение числа. Компенсация этого изменения достигается умножением мантиссы на 2^p , где p – количество разрядов, на которое была смещена запятая. При смещении запятой вправо (число увеличивается) значение p отрицательное, при смещении влево (число уменьшается) оно положительное.

В нормализованной форме принято мантиссу записывать в двоичной системе счисления, а порядок и само число p – в десятичной.

В нормализованной форме все числа записываются одинаково (в том смысле, что запятая у них ставится в одном и том же месте – после первой и единственной цифры целой части, которая всегда равна единице). Значение числа (т. е. ее порядок) указывается отдельно с помощью соответствующей степени основания системы счисления, т. е. степени числа два. Количество цифр в мантиссе может оказаться меньше, чем число значащих цифр в исходном числе. Это связано с тем, что под мантиссу отводится строго определенное число двоичных разрядов.

3) Выполнить приведение мантиссы.

Поскольку у нормализованной мантиссы целая часть всегда равна единице, то эту единицу подразумевают, но не записывают в разряды формы (см. рис. 2).

Следует иметь в виду, что все нули после запятой в нормализованной мантиссе, предшествующие первой единице, необходимо учитывать. Напри-

мер, если бы нормализованная мантисса была равна 1,00101, то приведенная мантисса была бы равна 00101.

4) Вычислить характеристику числа.

Характеристика числа равна порядку этого числа, увеличенному на 127. Полученное число перевести в двоичную систему счисления.

5) Заполнить знаковый разряд числа.

В знаковый разряд для положительного числа записать ноль, для отрицательного числа – единицу.

Пример 1. Задано десятичное представление вещественного числа $-13,375_{10}$. Определить внутреннее представление этого числа.

Переводим число $-13,375_{10}$ в двоичную систему счисления, в результате получим: $1101,011_2$.

В нормализованном виде экспоненциальная запись числа $-13,375_{10}$ имеет вид: $1,101011 \times 2^3$.

Приведенная мантисса равна 101011.

Характеристика заданного числа $3 + 127 = 130$. В двоичной системе счисления характеристика имеет вид: 10000010.

Таким образом, внутреннее двоичное представление вещественного числа $-13,375_{10}$ имеет вид: 1 10000010 1010110000000000000000, а шестнадцатеричное – C1 56 00 00. Пробелы в двоичном коде поставлены только для удобства чтения и отделяют знаковый бит от характеристики, а характеристику – от приведенной мантиссы.

Замечание 3. Способ хранения вещественных чисел в формате с плавающей запятой принят исходя из удобства реализации в компьютерах арифметики вещественных чисел.

Замечание 4. Диапазон вещественных чисел: $|\tilde{\alpha}| \leq 3,402823 \times 10^{38}$.

Пример 2. Задано внутреннее представление вещественного числа в шестнадцатеричной системе счисления: 41 C4 00 00. Определить внешнее (десятичное) представление этого числа.

В двоичном коде внутреннее представление числа имеет вид:

0100 0001 1100 0100 0000 0000 0000 0000.

Расставим пробелы так, чтобы отделить знаковый бит от характеристики, а характеристику – от приведенной мантиссы:

0 10000011 100010000000000000000000.

Далее выполним в обратном порядке шаги приведенного выше алгоритма.

Число положительное, так как в знаковом разряде стоит ноль.

Характеристика числа равна 10000011, т. е 131 в десятичной системе счисления, отсюда порядок числа равен 4 ($131 - 127 = 4$).

Приведенная мантисса равна 1000100...0, отсюда нормализованная мантисса равна 1, 1000100...0.

Искомое число в нормализованной форме равно 1, 10001 $\times 2^4$. Выполним умножение. Умножение мантиссы на 2^4 эквивалентно переносу запятой на четыре разряда вправо. Получаем число 11000,1. Переведем результат в десятичную систему счисления и получаем: $24,5_{10}$.

2. КОНТРОЛЬНЫЕ ВОПРОСЫ

1) Какое количество различных чисел можно представить данными типа *byte* в языке *Basic* и *unsigned char* в языке *C++*?

2) Какое количество различных чисел можно представить данными типа *integer* в языке *Basic* и *int* в языке *C++*?

3) Является ли количество вещественных чисел типа *single* в языке *Basic* и *float* в языке *C++* бесконечным?

4) Чему равно десятичное представление числа, если шестнадцатеричная форма внутреннего представления имеет вид:

а) FF (тип *byte* в языке *Basic* и *unsigned char* в языке *C++*);

б) FF FF (тип *integer* в языке *Basic* и *int* в языке *C++*);

в) FF 7F FF FF (тип *single* в языке *Basic* и *float* в языке *C++*)?

3. ЗАДАНИЯ

Задание 1. Получить шестнадцатеричную форму внутреннего представления значения переменной типа *byte*, а из заданного внутреннего представления значения переменной типа *byte* извлечь десятичное значение. Варианты исходных данных для задания приведены в табл. 3.

Таблица 3

Исходные данные (тип *byte* в языке *Basic* и *unsigned char* в языке *C++*)

Номер варианта	Десятичное целое число без знака	Внутреннее представление целого числа без знака
1	123	F5
2	211	E0
3	222	BD
4	223	85
5	166	7F
6	132	9A
7	102	CD
8	188	A9
9	176	D1
10	221	C5
11	199	D2
12	221	CE
13	219	EB
14	232	C4
15	207	CD
16	218	C3

Задание 2. Получить шестнадцатеричную форму внутреннего представления значения переменной типа *integer*, а из заданного внутреннего представления значения переменной типа *integer* извлечь десятичное значение. Варианты исходных данных для задания приведены в табл. 4.

Таблица 4

Исходные данные (тип *integer* в языке *Basic* и *int* в языке *C++*)

Номер варианта	Десятичное целое число со знаком	Внутреннее представление целого числа со знаком
1	–123	00 F5
2	211	FF 2A
3	–222	00 BD
4	223	FE B3
5	–277	00 80
6	354	FE BD
7	–256	02 00
8	231	FF 50
9	–311	01 41
10	287	FF 39
11	–300	00 FB
12	214	FE CA
13	–273	00 FA
14	170	FF 60
15	–190	00 E3
16	–255	FF 82

Задание 3. Получить шестнадцатеричную форму внутреннего представления значения переменной типа *single*, а из заданного внутреннего представления значения переменной типа *single* извлечь десятичное значение. Варианты исходных данных для задания приведены в табл. 5.

Таблица 5

Исходные данные (тип *single* в языке *Basic* и *float* в языке *C++*)

Номер варианта	Десятичное число	Внутреннее представление вещественного числа
1	−3,5625	41 BA 80 00
2	−7,1250	40 A6 00 00
3	−12,0625	41 56 00 00
4	55,7500	40 92 00 00
5	−1,6250	42 24 80 00
6	−21,7500	3F B0 00 00
7	−12,1250	3F F0 00 00
8	2,3125	C2 91 00 00
9	54,2500	C2 1C 40 00
10	−117,5000	42 59 00 00
11	−39,0625	C2 EB 00 00
12	31,3125	41 39 00 00
13	−63,2500	42 81 40 00
14	−65,3125	41 71 00 00
15	−16,6250	3F 10 00 00
16	−4,5625	42 5F 00 00

Библиографический список

1. Угринович Н. Д. Информатика и информационные технологии / Н. Д. Угринович. М.: Бином. Лаборатория знаний, 2004. 164 с.
2. Информатика. Базовый курс / С. В. Симонович, Г. А. Евсеев и др. СПб: Питер, 2005. 640 с.
3. Информатика / Б. В. Соболев, А. Б. Галин и др. Ростов-на-Дону: Феникс, 2005. 448 с.
4. Соколовская Н. Н. Системы счисления / Н. Н. Соколовская / Омский гос. ун-т путей сообщения. Омск, 2008. 32 с.

Учебное издание

СОКОЛОВСКАЯ Нина Николаевна,
ЕРОШЕНКО Александра Викторовна

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

Часть 1

ПРЕДСТАВЛЕНИЕ ЧИСЛОВОЙ ИНФОРМАЦИИ В КОМПЬЮТЕРАХ

Редактор Т. С. Паршикова

Подписано в печать .09.2010. Формат $60 \times 84 \frac{1}{16}$.
Плоская печать. Бумага офсетная. Усл. печ. л. 1,1. Уч.-изд. л. 1,3.
Тираж 250 экз. Заказ .

**

Редакционно-издательский отдел ОмГУПСа
Типография ОмГУПСа

*

644046, г. Омск, пр. Маркса, 35