

**ИНФОРМАТИКА
И ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ
КОНСПЕКТ ЛЕКЦИЙ**

УДК 37.6
ББК 92
Ц27

Цветкова, А.В.

Ц27 Информатика и информационные технологии / А.В. Цветкова. — М. : Т8 Издательские технологии / RUGRAM, 2020. — 192 с. — (Полезные конспекты).

Лаконичное и четкое изложение материала, продуманный отбор необходимых тем позволяют быстро и качественно подготовиться к семинарам, зачетам и экзаменам по данному предмету.

BIC JN
BISAC EDU000000

ЛЕКЦИЯ № 1. Введение в информатику

1. Информатика. Информация. ***Представление и обработка информации***

Информатика занимается формализованным представлением объектов и структур их взаимосвязей в различных областях науки, техники, производства. Для моделирования объектов и явлений используются различные формальные средства, например логические формулы, структуры данных, языки программирования и др.

В информатике такое фундаментальное понятие, как **информация** имеет различные значения:

- 1) формальное представление внешних форм информации;
- 2) абстрактное значение информации, ее внутреннее содержание, семантика;
- 3) отношение информации к реальному миру.

Но, как правило, под информацией понимают ее абстрактное значение — *семантику*. Интерпретируя представления информации, мы получим ее смысл, семантику. Поэтому, если мы хотим обмениваться информацией, нам необходимы согласованные представления, чтобы не нарушалась правильность интерпретации. Для этого интерпретацию представления информации отождествляют с некоторыми математическими структурами. В этом случае обработка информации может быть выполнена строгими математическими методами.

Одно из математических описаний информации — это представление ее в виде функции $y = f(x, t)$, где t — время, x — точка некоторого поля, в которой измеряется значение y . В зависимости от параметров функции x и t информацию можно классифицировать.

Если параметры — скалярные величины, принимающие непрерывный ряд значений, то полученная таким образом информация называется непрерывной (или *аналоговой*). Если же параметрам придать некоторый шаг изменений, то информация называется

дискретной. Дискретная информация считается универсальной, так как для каждого конкретного параметра можно получить значение функции с заданной степенью точности.

Дискретную информацию обычно отождествляют с *цифровой* информацией, которая является частным случаем символьной информации алфавитного представления. **Алфавит** — конечный набор символов любой природы. Очень часто в информатике возникает ситуация, когда символы одного алфавита надо представить символами другого, т. е. провести операцию *кодирования*. Если количество символов кодируемого алфавита меньше количества символов кодирующего алфавита, то сама операция кодирования не является сложной, в противном случае необходимо использовать фиксированный набор символов кодирующего алфавита для однозначного правильного кодирования.

Как показала практика, наиболее простым алфавитом, позволяющим кодировать другие алфавиты, является двоичный, состоящий из двух символов, которые обозначаются, как правило, через 0 и 1. С помощью n символов двоичного алфавита можно закодировать 2^n символов, а этого достаточно, чтобы закодировать любой алфавит.

Величина, которая может быть представлена символом двоичного алфавита, называется минимальной единицей информации или **битом**. Последовательность из 8 бит — **байт**. Алфавит, содержащий 256 различных 8-битных последовательностей, называется *байтовым*.

В качестве стандартного сегодня в информатике принят код, в котором каждый символ кодируется 1 байтом. Существуют и другие алфавиты.

2. Системы счисления

Под **системой счисления** подразумевается набор правил наименования и записи чисел. Различают позиционные и непозиционные системы счисления.

Система счисления называется *позиционной*, если значение цифры числа зависит от местоположения цифры в числе. В противном случае она называется *непозиционной*. Значение числа определяется по положению этих цифр в числе.

3. Представление чисел в ЭВМ

32-разрядные процессоры могут работать с оперативной памятью емкостью до 232-1, а адреса могут записываться в диапазоне 00000000 — FFFFFFFF. Однако в реальном режиме процессор работает с памятью до 220-1, а адреса попадают в диапазон 00000 — FFFFF. Байты памяти могут объединяться в поля как фиксированной, так и переменной длины. **Словом** называется поле фиксированной длины, состоящее из 2 байтов, **двойным словом** — поле из 4 байтов. Адреса полей бывают *четные* и *нечетные*, при этом для четных адресов операции выполняются быстрее.

Числа с фиксированной точкой в ЭВМ представляются как целые двоичные числа, и занимаемый ими объем может составлять 1, 2 или 4 байта.

Целые двоичные числа представляются в дополнительном коде, соответственно числа с фиксированной точкой представляются в дополнительном коде. При этом если число занимает 2 байта, то структура числа записывается по следующему правилу: старший разряд отводится под знак числа, а остальные — под двоичные цифры числа. Дополнительный код положительного числа равен самому числу, а дополнительный код отрицательного числа может быть получен по такой формуле: $x = 10n - |x|$, где n — разрядность числа.

В двоичной системе счисления дополнительный код получается путем инверсии разрядов, т. е., заменой единиц нулями и наоборот, и прибавлением единицы к младшему разряду.

Количество битов мантиссы определяет точность представления чисел, количество битов машинного порядка определяет диапазон представления чисел с плавающей точкой.

4. Формализованное понятие алгоритма

Алгоритм может существовать только тогда, когда в то же самое время существует некоторый математический объект. Формализованное понятие алгоритма связано с понятием рекурсивных функций, нормальных алгоритмов Маркова, машин Тьюринга.

В математике функция называется однозначной, если для любого набора аргументов существует закон, по которому опреде-

ляется единственное значение функции. В качестве такого закона может выступать алгоритм; в этом случае функция называется **вычислимой**.

Рекурсивные функции — это подкласс вычислимых функций, а алгоритмы, определяющие вычисления, называются *сопутствующими алгоритмами* рекурсивных функций. Сначала фиксируются базовые рекурсивные функции, для которых сопутствующий алгоритм тривиален, однозначен; затем вводятся три правила — операторы *подстановки*, *рекурсии* и *минимизации*, при помощи которых на основе базовых функций получаются более сложные рекурсивные функции.

Базовыми функциями и их сопутствующими алгоритмами могут выступать:

- 1) функция n независимых переменных, тождественно равная нулю. Тогда, если знаком функции является φn , то независимо от количества аргументов значение функции следует положить равным нулю;
- 2) тождественная функция n независимых переменных вида Ψn_i . Тогда, если знаком функции является Ψn_i , то значением функции следует взять значение i -го аргумента, считая слева направо;
- 3) Λ -функция одного независимого аргумента. Тогда, если знаком функции является λ , то значением функции следует взять значение, следующее за значением аргумента.

Разные ученые предлагали свои подходы к формализованному представлению алгоритма. Например, американский ученый Черч предположил, что класс вычислимых функций исчерпывается рекурсивными функциями и, как следствие, каким бы ни был алгоритм, перерабатывающий один набор целых неотрицательных чисел в другой, найдется алгоритм, сопутствующий рекурсивной функции, эквивалентный данному. Следовательно, если для решения некоторой поставленной задачи нельзя построить рекурсивную функцию, то и не существует алгоритма для ее решения. Другой ученый, Тьюринг, разработал виртуальную ЭВМ, которая перерабатывала входную последовательность символов в выходную. В связи с этим им был выдвинут тезис, что любая вычислимая функция вычислима по Тьюрингу.

ЛЕКЦИЯ № 2. Язык Pascal

1. Введение в язык Pascal

Основные символы языка — буквы, цифры и специальные символы — составляют его алфавит. Язык Pascal включает следующий набор основных символов:

1) 26 латинских строчных и 26 латинских прописных букв:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z;

2) _ (знак подчеркивания);

3) 10 цифр: 0 1 2 3 4 5 6 7 8 9;

4) знаки операций:

+ — × / = <> < > <= >= := @;

5) ограничители:

. , ' () [] (.) { } (* *) .. : ;

6) спецификаторы: ^ # \$;

7) служебные (зарезервированные) слова:

ABSOLUTE, ASSEMBLER, AND, ARRAY, ASM, BEGIN, CASE, CONST, CONSTRUCTOR, DESTRUCTOR, DIV, DO, DOWNT0, ELSE, END, EXPORT, EXTERNAL, FAR, FILE, FOR, FORWARD, FUNCTION, GOTO, IF, IMPLEMENTATION, IN, INDEX, INHERITED, INLINE, INTERFACE, INTERRUPT, LABEL, LIBRARY, MOD, NAME, NIL, NEAR, NOT, OBJECT, OF, OR, PACKED, PRIVATE, PROCEDURE, PROGRAM, PUBLIC, RECORD, REPEAT, RESIDENT, SET, SHL, SHR, STRING, THEN, TO, TYPE, UNIT, UNTIL, USES, VAR, VIRTUAL, WHILE, WITH, XOR.

Кроме перечисленных, в набор основных символов входит пробел. Пробелы нельзя использовать внутри двоянных символов и зарезервированных слов.

Концепция типа для данных

В математике принято классифицировать переменные в соответствии с некоторыми важными характеристиками. Произво-

дится строгое разграничение между вещественными, комплексными и логическими переменными, между переменными, представляющими отдельные значения и множество значений, и т. д. При обработке данных на ЭВМ такая классификация еще более важна. В любом алгоритмическом языке каждая константа, переменная, выражение или функция бывают определенного типа.

В языке Pascal существует правило: тип явно задается в описании переменной или функции, которое предшествует их использованию. Концепция типа языка Pascal имеет следующие основные свойства:

- 1) любой тип данных определяет множество значений, к которому принадлежит константа, которые может принимать переменная или выражение либо вырабатывать операция или функция;
- 2) тип значения, задаваемого константой, переменной или выражением, можно определить по их виду или описанию;
- 3) каждая операция или функция требуют аргументов фиксированного типа и выдают результат фиксированного типа.

Отсюда следует, что транслятор может использовать информацию о типах для проверки вычислимости и правильности различных конструкций.

Тип определяет:

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

Следует заметить, что обязательное описание типа приводит к избыточности в тексте программ, но такая избыточность является важным вспомогательным средством разработки программ и рассматривается как необходимое свойство современных алгоритмических языков высокого уровня.

В языке Pascal существуют скалярные и структурированные типы данных. К скалярным типам относятся стандартные типы и типы, определяемые пользователем. Стандартные типы включают целые, действительные, символьный, логические и адресный типы.

Целые типы определяют константы, переменные и функции, значения которых реализуются множеством целых чисел, допустимых в данной ЭВМ.

Тип	Диапазон значений	Требуемая память (байт)
Byte	0...255	1
Word	0...65535	2
Shortint	-128...127	1
Integer	-32768...32767	2
Longint	- 2147483648...2147483647	4

Действительные типы определяет те данные, которые реализуются подмножеством действительных чисел, допустимых в данной ЭВМ.

Тип	Диапазон значений	Количество цифр мантиссы	Требуемая память (байт)
Real	$2.9e - 39 \dots 1.7e + 38$	11–12	6
Single	$1.5e - 45 \dots 3.4e + 38$	7–8	4
Double	$5.0e - 324 \dots 1.7e + 308$	15–16	8
Extended	$3.4e - 4932 \dots 1.1e + 4932$	19–20	10
Comp	$-9.2e + 18 \dots 9.2e + 18$	19–20	2

Типы, определяемые пользователем, — перечисляемый и интервальный. Структурированные типы имеют четыре разновидности: массивы, множества, записи и файлы.

Кроме перечисленных, Pascal включает еще два типа — процедурный и объектный.

Выражение языка состоит из констант, переменных, указателей функций, знаков операций и скобок. Выражение задает правило вычисления некоторого значения. Порядок вычисления определяется старшинством (приоритетом) содержащихся в нем операций. В языке Pascal принят следующий приоритет операций:

- 1) вычисления в круглых скобках;
- 2) вычисления значений функций;
- 3) унарные операции;
- 4) операции *, /, div, mod, and;
- 5) операции +, —, or, xor;
- 6) операции отношения =, <>, <, >, <=, >=.

Выражения входят в состав многих операторов языка Pascal, — также могут быть аргументами встроенных функций.

2. Стандартные процедуры и функции

Арифметические функции

1. Function Abs(X);

Возвращает абсолютное значение параметра.

X — выражение вещественного или целочисленного типа.

2. Function ArcTan(X: Extended): Extended;

Возвращает арктангенс аргумента.

X — выражение вещественного или целочисленного типа.

3. Function Exp(X: Real): Real;

Возвращает экспоненту.

X — выражение вещественного или целочисленного типа.

4. Function Frac(X: Real): Real;

Возвращает дробную часть аргумента.

X — выражение вещественного типа. Результат — дробная часть X, т. е.

$$\text{Frac}(X) = X - \text{Int}(X).$$

5. Function Int(X: Real): Real;

Возвращает целочисленную часть аргумента.

X — выражение вещественного типа. Результат — целочисленная часть X, т. е. X, округленный к нулю.

6. Function Ln(X: Real): Real;

Возвращает натуральный логарифм ($\text{Ln } e = 1$) выражения X вещественного типа.

7. Function Pi: Extended;

Возвращает значение Pi, которое определено как 3.1415926535.

8. Function Sin(X: Extended): Extended;

Возвращает синус аргумента.

X — выражение вещественного типа. Sin возвращает синус угла X в радианах.

9. Function Sqr(X: Extended): Extended;

Возвращает квадрат аргумента.

X — выражение с плавающей запятой. Результат того же самого типа, что и X.

10. Function Sqrt(X: Extended): Extended;

Возвращает квадратный корень аргумента.

X — выражение с плавающей запятой. Результат — квадратный корень X.