

Паскаль для начинающих

Основы грамматики в таблицах и примерах

Pascal (Паскаль) - язык программирования общего назначения. Создан в 1970 году специалистами Технического Университета в Цюрихе (ETH, Zurich) под руководством профессора Никлауса Вирта (Niklaus Wirth). После появления языка Pascal были созданы новые промышленные версии языка, а также языки-потомки, которые сохраняют основные синтаксические и семантические свойства языка Вирта: Modula, Turbo Pascal, Object Pascal (Delphi), Component Pascal (Oberon-2), TMT Pascal, Free Pascal, Zonnon, Ada. Идеи языка Pascal оказали большое влияние на современные языки программирования, которые активно используются в современной промышленности: Java, Javascript, C#. При этом Pascal значительно проще и хорошо приспособлен для начального изучения программирования. Описание языка дано в соответствии с диалектом языка фирмы Borland Software (США): Turbo Pascal 7.1. (1992).

ЧАСТЬ I.

1. Основные правила грамматики

1. Текст программы состоит из трех составных частей. Это **заголовок**, **описания**, **операторы**. Заголовок программы содержит имя программы. Для версии ETH необходимо дополнительно указать источник и приемник данных для работы программы - файлы для ввода и вывода информации. По умолчанию файл **input** связан с клавиатурой компьютера, а файл **output** - с экраном монитора. В Turbo Pascal заголовок имеет декоративное значение и использовать его не обязательно.
2. **Описания** - это символьные конструкции для описания свойств объектов программы. Все объекты программы надо обязательно указать в этой части программы. **Операторы** - это символьные конструкции для описания действий над объектами программы. Операторы должны находиться между словами **begin** и **end**. Эти слова считаются скобками.
3. В одной строке текста можно размещать несколько операторов или описаний. Поэтому между операторами надо обязательно ставить символ-разделитель. Для разделения операторов используется **символточка с запятой (;)**. После последнего оператора разделитель ставить не обязательно.
4. Малые и большие буквы считаются **эквивалентными** (равнозначными). Имя любого объекта в записи программы может содержать только **буквы** латинского алфавита, **цифры** и **символ подчеркивания (_)**. Имя не может начинаться с цифры. Никакие другие типы символов в записи имен не разрешаются.

2. Общее строение программы

ETH Pascal	Пример
program имя(ввод,вывод);	program echo(input,output);
описания;	var n: integer;
BEGIN	BEGIN

оператор;	write('число?');
оператор;	read(n);
оператор	write(n)
END.	END.

Turbo Pascal-Delphi Пример

program имя;	program echo;
модули;	uses crt;
описания;	var n: integer;
BEGIN	BEGIN
оператор;	clrscr; read(n);
оператор;	write(n);
оператор	readkey
END.	END.

3. Основные типы переменных

Тип	Описатель типа (имя)	Значение типа (пример)
Действительный	real	16.32 -1.6320000000E+01

Тип	Описатель типа (имя)	Значение типа (пример)
Целый	integer	30000
Длинный целый	longint	2000000000

Тип	Описатель типа (имя)	Значение типа (пример)
Символьный (литерный)	char	'a' #13
Строковый	string	'задача'
Логический (булевский)	boolean	false true
Интервальный	M1..M2	100

4. Конструкции для описания величин

Описание переменных величин Пример

var	var
имя-переменной: имя-типа;	a: real;
имя1, имя2, имя3: имя-типа;	x, y, z: integer;
	i, j, k: char;
	p, q, r: 0..100;
	s, t, u: 'a'..'z';

Описание постоянных величин Пример

const	const
имя1 = значение1;	X = 40000;
имя2 = значение2;	Y = 16.32;
имя3 = значение3;	Z = -1.632E+10;
имяN = значениеN;	A = 'a';
	R = #13;
	T = 'тема'#13#10'урока:';

5. Оператор присваивания

Форма оператора Пример

имя := выражение **a := 0;**
 x := x + 1;

6. Ввод и вывод

Процедура	Пример 1	Пример 2
read (список ввода)	program echo1(input,output);	program echo2(input,output);
readln (список ввода)	var m, n: integer ;	var m, n: integer ;
write (список вывода)	BEGIN	BEGIN
writeln (список вывода)	read(m,n);	readln(m,n);
	write('m=',m);	writeln('m=',m);
	write('n=',n)	writeln('n=',n)
	END.	END.

Число (n)	Тип	Формат вывода	Результат вывода
16	integer	write(n)	16
16	integer	write(n:2)	16
16	integer	write(n:5)	16
16.32	real	write(n);	1.6320000000E+01
16.32	real	write(n:12)	1.63200E+01
16.32	real	write(n:5:2)	16.32
16.32	real	write(n:10:4)	16.3200

7. Модуль crt (cathode ray tube)

Процедуры и функции	Операция	Пример
clrscr	Очистить экран	clrscr
textcolor (цвет)	Установить цвет текста	textcolor(1)
		textcolor(blue)
textbackground (цвет)	Установить цвет фона	textbackground(1)
		textbackground(blue)
delay (время в миллисекундах)	Остановить выполнение на заданное время	delay(1000)
gotoxy (колонка, строка)	Установить курсор на заданную позицию	gotoxy(40,20)
keypressed	Если была нажата клавиша, то true (функция)	repeat write('o') until keypressed
readkey	Остановить выполнение до нажатия клавиши, прочитать код нажатой клавиши (функция)	readkey
		c:=readkey

Номер и имя цвета	Номер и имя цвета
0 black	8 darkgray
1 blue	9 lightblue
2 green	10 lightgreen
3 cyan	11 lightcyan
4 red	12 lightred
5 magenta	13 lightmagenta

6 brown
7 lightgray

14 yellow
15 white

8. Арифметические средства языка

Арифметическая операция	Описание операции	Пример
+	Сложение	
-	Вычитание	
*	Умножение	
/	Деление	5.0/2.0 = 2.5
div	Целочисленное деление (деление без остатка)	5 div 2 = 2
mod	Остаток от деления	5 mod 2 = 1
shl	Двоичный сдвиг влево	5 shl 1 = 10
shr	Двоичный сдвиг вправо	5 shr 1 = 2
Функция	Результат функции	Пример
abs(x)	Модуль числа (абсолютное значение)	abs(-2) = 2
sqr(x)	Квадрат числа	sqr(2) = 4
sqrt(x)	Квадратный корень числа	sqrt(4) = 2.0
pi	Число пи ~ 3.14...	
Функция	Результат функции	Пример
sin(x)	Синус числа	sin(pi) = 0
cos(x)	Косинус числа	cos(0) ~ 1.0
ln(x)	Логарифм натуральный	ln(1) = 0
exp(x)	Экспонента (число e в степени x)	exp(1) ~ 2.72
Функция	Результат функции	Пример
int(x)	Целая часть числа	int(3.6) = 3.0
trunc(x)	Целое число без округления	trunc(3.6) = 3
round(x)	Целое число с округлением	round(3.6) = 4

ЧАСТЬ II.

9. Логические средства языка

Оператор условия - полная форма		Пример
if условие then	if (x>0) and (y>0) then	
оператор 1	begin x:=x+y; y:=0 end	
else	else	
оператор 2	begin x:=x-y; y:=-y end	
Оператор условия - неполная форма		Пример
if условие then	if x>y then	
оператор	x:=x mod y	
Логическая операция	Описание операции	Приоритет (порядок выполнения операции)
not	Отрицание (НЕ)	1
and	Конъюнкция (И)	2
or	Дизъюнкция (ИЛИ)	3
xor	Сильная дизъюнкция (ИЛИ-ИЛИ)	3
Логическое отношение	Описание отношения	Приоритет (порядок выполнения)

(предикат)		операции)
=	Равно	4
>	Больше	4
<	Меньше	4
<>	Не равно	4
<=	Не больше (меньше или равно)	4
>=	Не меньше (больше или равно)	4

Оператор варианта Пример

case выражение **of case** с **of**

вариант-1: оператор; 0: x:=x+1;

вариант-2: оператор; 1,2: x:=x-1;

вариант-N: оператор 3..100: x:=-x

else else

оператор x:=0

end end

10. Операторы цикла

Оператор цикла с управляющей переменной - форма "to" Пример

for имя:=значение1 **to** значение2 **do**

оператор

for i:=1 **to** n **do**

begin read(a); s:=s+a**end**

Оператор цикла с управляющей переменной - форма "downto" Пример

for имя:=значение1 **downto** значение2 **do**

оператор

for i:=n **downto** 0**do**

write(i)

Оператор цикла с логическим пред-условием Пример

while условие-повторения **do**

оператор

while i<=n **do**

begin read(a); s:=s+a; i:=i+1 **end**

Оператор цикла с логическим пост-условием Пример

repeat

операторы

until условие-выхода

repeat

write(i); i:=i-1

until i=0

11. Массивы

Тип массива

Описание массива

Пример

Одномерный массив (вектор) **array** [тип-индекса] **of** тип-элемента **var** a: **array** [1..100] **of** integer;

Тип массива

Описание массива

Пример

2-мерный массив (матрица)

array [тип-индекса-1, тип-индекса-2] **of** тип-элемента;

var a: **array** [1..M, 1..M] **of** real;

3-мерный массив

array [тип-индекса-1, тип-индекса-2, тип-индекса-3] **of** тип-элемента;

var a: **array** [0..1, 0..M, 0..N] **of** char;

4-мерный массив

array [тип-индекса-1, тип-индекса-2, тип-индекса-3, тип-индекса-4] **of** тип-элемента;

var a: **array** [0..1, 0..1, 0..1, 'a'..'z'] **of** boolean;

12. Процедуры и функции пользователя

Процедура	Пример процедуры для обмена значений между целыми переменными m и n
program имя-программы;	program primer; var x,y: integer ;
procedure имя(параметры);	procedure swap(var m,n: integer);
описания локальных величин	var s: integer ;
begin	begin
операторы процедуры	s:=m; m:=n; n:=s
end ;	end ;
BEGIN	BEGIN
операторы программы	read(x,y); swap(x,y); writeln(x:10,y:10)
END.	END.
Функция	Пример функции для вычисления натуральной степени (n) целого числа (m)
program имя-программы;	program primer; var x,y: integer ;
function имя(параметры): тип-результата;	function power(m,n: integer): integer ;
описания локальных величин	var s,i: integer ;
begin	begin
операторы функции	s:=1; for i:=1 to n do s:=s*m; power:=s
end ;	end ;
BEGIN	BEGIN
операторы программы	read(x,y); writeln(power(x,y))
END.	END.

Часть III.

13. Символьные типы данных

Тип	Описатель типа (имя)	Значение типа (пример)	
Символьный (литерный)	char	'a' #13	
Строковый	string	'задача' 'тема'#13#10'урока:'	
Функция	Результат функции	Пример	
length(s)	Длина строки	n:=length(s)	
pos(s1,s2)	Начало первого вхождения строки s1 внутри строки s2 (номер позиции)	n:=pos('+',s)	
copy(s,p,n)	Часть строки s длиной n, начиная с позиции p	c:=copy(s,4,2)	
concat(s1,s2,s3,...sn)	Соединить строки (функция конкатенации)	c:=concat(s,',',t,',')	
Функция	Результат функции	Пример	
upcase(c)	Преобразовать малую букву латинского алфавита в большую (символ верхнего регистра)	c:=upcase(c)	
chr(n)	Символ с номером n	c:=chr(n)	
ord(c)	Номер символа со значением c	n:=ord(c)	
succ(c)	Следующий символ таблицы ASCII	c:=succ(c)	
pred(c)	Предыдущий символ таблицы ASCII	c:=pred(c)	

Процедура	Операция	Пример
delete (s,p,n)	Удалить часть строки s длиной n, начиная с позиции p	delete(s,n,1)
insert (s1,s2,p)	Вставить строку s1 в строку s2, начиная с позиции p	insert('+',s,1)
val (s,x,r)	Преобразовать строку s в число x с признаком результата r	val(s,x,nan)
str (x:f,s)	Преобразовать число x в строку s с форматом числа f	str(x:10,s)
		str(x:10:2,s)
s1 + s2	Соединить строки (операция конкатенации)	s:=s+';'+t+'.'

14. Множества

Тип множества	Описание множества	Пример
Постоянное множество (константа)	[тип-элемента]	const a = [0..100];
Переменное множество (переменная)	set of [тип-элемента]	var a: set of ['0'..'9', '+', '-'];

Пример описания постоянного множества Элементы множества

const A = ['A'..'Z', 'a'..'z'];	Буквы латинского алфавита
const B = ['a'..'z'];	Малые буквы латинского алфавита
const C = ['A'..'Я', 'a'..'я', 'ё', 'ё'];	Буквы алфавита-кириллицы
const D = ['A'..'Я'];	Большие буквы алфавита-кириллицы
const M = ['0'..'9'];	Цифровые символы от 0 до 9
const N = [0..100];	Целые числа от 0 до 100

Пример описания переменного множества Элементы множества

var a set of ['A'..'Z', 'a'..'z'];	Буквы латинского алфавита
var b set of ['a'..'z'];	Малые буквы латинского алфавита
var c set of ['A'..'Я', 'a'..'я', 'ё', 'ё'];	Буквы алфавита-кириллицы
var d set of ['A'..'Я'];	Большие буквы алфавита-кириллицы
var m set of ['0'..'9'];	Цифровые символы от 0 до 9
var n set of [0..100];	Целые числа от 0 до 100

Операция	Результат операции	Пример
M1 + M2	Объединение множеств M1 и M2	['0'..'9'] + ['+']
M1 - M2	Разность множеств M1 и M2	['a'..'z'] - ['x'..'z']
M1 * M2	Пересечение множеств M1 и M2	['a'..'m'] * ['c', 'i', 'n']

Операция	Когда результат операции равен true	Пример
значение in множество	Значение является элементом множества	not (s[i] in ['a'..'z'])
M1 = M2	Множества M1 и M2 равны между собой	
M1 <> M2	Множества M1 и M2 не равны	
M1 < M2	Множество M1 является подмножеством множества M2	
M1 <= M2	Множество M1 является подмножеством множества M2 или равно M2	
M1 > M2	Множество M1 является надмножеством множества M2	
M1 >= M2	Множество M1 является надмножеством множества M2 или равно M2	

Часть IV.

15. Файлы

Тип файла	Описание файла	Пример
-----------	----------------	--------

текстовый файл (текст)	text	var t: text;
типизированный файл	file of тип-элемента	var a: file of integer;
Процедуры и функции	Операция	Пример
read (файл, список ввода)	Ввод данных из файла	read (t,a,b,c);
readln (файл, список ввода)	Ввод данных из файла	readln (t,a,b,c);
write (файл, список вывода)	Вывод данных в файл	write (t,x,y,z);
writeln (файл, список вывода)	Вывод данных в файл	writeln (t,x,y,z);
assign (файл, имя файла)	Определить имя файла	assign (t,'c:\text\info.txt');
reset (файл)	Открыть файл для ввода	reset (t);
rewrite (файл)	Открыть файл для вывода	rewrite (t);
close (файл)	Закрыть файл	close (t);
eof (файл)	Если конец файла, то true (функция)	while not eof(t) do read(t,a);
eoln (файл)	Если конец строки, то true (функция)	while not eoln(t) do read(t,a);

16. Типы и записи

Описание типа записи	Пример
имя типа = record	type data = record
имя-1: тип-1;	x:integer;
имя-2: тип-2;	y:integer;
имя-N: тип-N;	t:string;
end;	end;
	var a1, a2, a3: data;

17. Ссылки и указатели

Описание указателя	Пример	
имя-указателя: ^имя-типа;	var pdata: ^data;	
Ссылка с помощью указателя	Пример	
имя-указателя^	pdata^ := 0;	
Процедура	Операция	Пример
new (имя-указателя)	Выделить память для переменной (создать переменную для указателя)	new (pdata);
dispose (имя-указателя)	Освободить память переменной (указатель:=nil)	dispose (pdata);

18. Объекты и классы

Описание типа объекта (класс)	Пример
имя класса = object (класс-родитель)	type donkey = object (animals)
свойство-1: тип-1;	x:integer;
свойство-2: тип-2;	y:integer;
свойство-N: тип-N;	t:string;
метод-1(параметры);	function run(m:string):boolean;
метод-2(параметры);	procedure show;
метод-N(параметры);	procedure hide;
end;	end;
	var a1, a2, a3: donkey;