

Справочник по Паскалю

Структура программы на Паскале

uses crt;	{список подключаемых библиотек}
var переменных }	{описатель используемых
c:string;	{с строковая переменная}
begin;	{начало программы}
writeln ('всем привет'); 'Всем привет'}	{напечатать на экране сообщение
readln;	
end.	

Операторы вывода (writeln, write)

uses crt;	{Список подключаемых библиотек}
var переменных }	{Описатель используемых
begin;	{Начало программы}
writeln ('Здравствуйе Вас приветствует Паскаль '); {напечатать на экране сообщение 'Здравствуйе Вас приветствует Паскаль '};	
writeln(100+100);	{Выведет сумму 2 чисел 200 }
readln;	
end.	{конец программы}

Оператор ввода (readln, read)

uses crt;	{Список подключаемых библиотек}
var переменных }	{Описатель используемых
c:string;	{с строковая переменная}
begin;	{Начало программы}
writeln ('Здравствуйе как ваше имя? '); {напечатать на экране сообщение 'Здравствуйе как ваше имя ? '};	

```

    readln (c);                {Ввести в символьную С свое имя}
    writeln ('Добро пожаловать ', c); {Выведет добро пожаловать
введенное имя}
    readln;
end.                            {конец программы}

```

Типы переменных (числовые: real, integer символные)

```

uses crt;                      {Список подключаемых библиотек}
var                             {Описатель используемых переменных}
c, m:string;                   {c, m строковые символьные переменная}
a:integer;                     {a числовая переменная целого типа}
p:real;                         {P числовая переменная Действительного
типа}
begin;                          {Начало программы}
c:='Школе 461'                 {Записать в ячейку С строку 'Школе 461' }
m:='Лет'                       {Записать в ячейку М строку 'Лет'}
a:=29;                         {Записать в ячейку А целое число '29'}
p:=3.14;                       {Записать в ячейку Р дробное число '3.14'}
write(c, ' ');                 {напечатать на экране сообщение ' Школе
461');
write(a+1, ' ');               {напечатать на экране сообщение '30');
write(m, ' ');                 {напечатать на экране сообщение ' Лет ');
write('Число Pi= ',p);        {напечатать на экране сообщение ' Число
Pi=3.14'};
readln;
end.                            {конец программы}

```

Случайные числа (random)

```

uses crt;                      {Список подключаемых библиотек}
var                             {Описатель используемых переменных}
a,b,c:integer                   {a,b,c Целые числа}
begin;                          {Начало программы}
a:=random(10)                   {Записать в переменную a случайное число в диапазоне
от 0-10}

```

b:=random(10)+5 {Записать в переменную b случайное число в диапазоне от 5-15}

c:=10-random(20) {Записать в переменную c случайное число в диапазоне от -10..10}

writeln(' a=',a,' b=',b,' c=',c); {напечатать на экране значения переменных a,b,c}

readln;

end. {конец программы}

Оператор перехода по метке (goto m)

Создание бесконечного цикла

uses crt; {Список подключаемых библиотек}

label M; {Описание метки перехода}

var {Описатель используемых переменных}

c:string; {с строковая переменная}

begin; {Начало программы}

writeln('Здравствуйте как ваше имя ? '); {напечатать на экране сообщение 'Здравствуйте как ваше имя ? '};

readln (c); {Ввести в символьную С свое имя}

M: writeln('Добро пожаловать ', c); {Выведет добро пожаловать введенное имя}

goto M; {Переход на строку M}

end. {конец программы}

Оператор цикла (for do)

Создание ограниченного цикла

uses crt; {Список подключаемых библиотек}

var {Описатель используемых переменных}

c:string; {с строковая переменная}

begin; {Начало программы}

writeln('Здравствуйте как ваше имя ? '); {напечатать на экране сообщение 'Здравствуйте как ваше имя ? '};

```

readln (c);           {Ввести в символьную С свое имя}
for n:=1 to 1000     {Создать цикл от 1 до 1000}
begin; {Начало тела цикла}
writeln('Добро пожаловать ', c); {Выведет добро пожаловать введенное имя}
end;   {Конец тела цикла}
end.   {конец программы}

```

Операторы условия (if then)

```

uses crt;           {Список подключаемых библиотек}
var                {Описатель используемых переменных}
c,d:string;       {c,d строковая переменная}
begin;           {Начало программы}
writeln('Здравствуйте как ваше? '); {напечатать на экране сообщение
'Здравствуйте как ваше имя ? '};
readln (c);       {ввод имени в переменную С}
writeln('Введи пароль');
readln (d);       {ввод пароля в переменную С}
if d='хрюша' then writeln('Приветствую вас ', c) {Если d='хрюша' тогда
приветствие}
else              {Иначе 'Вон отсюда негодяй '}
writeln('Вон отсюда негодяй ', c);
readln;
end.              {конец программы}

```

Массивы

Создание, заполнение числового одномерного массива на 10 элементов.
Вывод значений элементов массива его на экран.

Spis(массив)	-3.2	24	11	-5	1.1	99	1015	3.3	17.2	44
К(номер яч)	1	2	3	4	5	6	7	8	9	10

```
uses crt;           {Список подключаемых библиотек}
var                 {Описание используемых переменных}
k,:integer;        {К целое число}
spis:array[1..10] of real; {Создать массив с именем 'spis' на 10 мест, для
записи действительных чисел}
n:real;            {N переменная для записи чисел введенных с клавиатуры}

begin;             {Начало программы}
for k:=1 to 10 do   {Создать цикл от 1 до 10}
    begin; {Начало тела цикла}

write('Введите число '); {Печать сообщения Введите число }
readln(n);          {Ввод числа с клавиатуры в переменную n}
spis[k]:=n;        {Запись числа N в ячейку K массива 'spis'}
end; {Конец тела цикла}
{-----Вывод содержимого массива 'spis' на экран}
for k:=1 to 10 do   {Создать цикл от 1 до 10}
    begin; {Начало тела цикла}
write(' spis [' ,k,']=',spis[n]); {Печать содержимого массива 'spis' ячейки K }
end; {Конец тела цикла}
end;                {конец программы}
```

Логическая конструкция "Счетчик"

```
uses crt;           {Список подключаемых библиотек}
label M;           {Описание метки перехода}
var               {Описатель используемых переменных}
s:integer;        {s целое число}

begin;            {Начало программы}
s:=0;             {Установка счетчика на 0}
M: s:=s+1;
writeln('S='s);   {Напечатать значение S};
Goto M:           {Переход на строку M}
end;              {конец программы}
```

Логическая конструкция "Сумматор"

Программа вычисляет сумму всех введенных чисел

```
uses crt;           {Список подключаемых библиотек}
label M;           {Описание метки перехода}
var               {Описатель используемых переменных}
s,n:integer;      {s целое число}

begin;            {Начало программы}
s:=0;             {Установка счетчика на 0}
write('Введите число '); {Печать сообщения Введите число }
readln(n);        {Ввод числа с клавиатуры в переменную n}
M: s:=s+n;        {Сумматор накапливает сумму введенных чисел}
writeln('Сумма чисел равна ',s); {Напечатать значение S};
Goto M:           {Переход на строку M}
end;              {конец программы}
```

Программа определяет все числа кратные трем

Использует ограниченный цикл For do

```
uses crt;           {Список подключаемых библиотек}
var                 {Описатель используемых переменных}
k,:integer;        {К целое число}
n:real;

begin;             {Начало программы}
for k:=1 to 10 do  {Создать цикл от 1 до 10}
    begin; {Начало тела цикла}

write('Введите число '); {Печать сообщения Введите число }
readln(n);           {Ввод числа с клавиатуры в переменную n}

if (n /3 )=int(n/3) then writeln('Число ',n, ' кратно трем'); {Если условие верно
напечатается сообщение 'Число ',n, ' кратно трем'}
    end; {Конец тела цикла}
end;                {конец программы}
```

Логическая конструкция "Определение максимума и минимума"

Определение минимального и максимального из 10 введенных чисел

```
uses crt;           {Список подключаемых библиотек}
var                 {Описатель используемых переменных}
k,:integer;        {К целое число}
n:real;
min,max:real;      {Действительные числа. Могут иметь дробную
составляющую min-Минимальное число Max-максимальное}

begin;             {Начало программы}
for k:=1 to 10 do  {Создать цикл от 1 до 10}
    begin; {Начало тела цикла}

write('Введите число '); {Печать сообщения Введите число }
readln(n);           {Ввод числа с клавиатуры в переменную n}
if k=1 then begin;  {Приравниваем Min и Max первому введенному
числу. Текущее число вводимое с клавиатуры N}
min:=n;
max:=n;
    end;
if min>n then min:=n; {Если значение Min> числа введенного с клавиатуры,
Min принимает значение Введенного числа}
if max<n then max:=n; { Если значение Max< числа введенного с клавиатуры,
Max принимает значение Введенного числа }
    end; {Конец тела цикла}
writeln('Минимальное число = ',Min); {Вывод на экран Минимального из
введенных чисел}
writeln('Максимально число = ',Max); {Вывод на экран максимального из
введенных чисел}
readln;
end;                {конец программы}
```


Ввод и вывод данных

Процедура (функция)	Назначение	Пример вызова
read (x)	ввод данных (клавиатура)	read(a, b, c)
Readln(x)	ввод данных, пропуск маркера конца строки	readln(a, b, c)
Write (x)	вывод данных (экран)	write('n = ',n:10) write('n = ',n:p:q)
Writeln (x)	вывод данных, вывод маркера конца строки	writeln('?':s+2) writeln('':spaces)
Clrscr	очистить весь экран (окрасить в цвет фона)	clrscr
Textcolor (цвет)	выбрать цвет символа	textcolor(magenta)
Textbackground (цвет)	выбрать цвет фона	textbackground(0)
Window (x1, y1, x2, y2)	окно (x1, y1) - (x2, y2)	window(1, 1, 40, 10)
Wherex	позиция курсора x - номер колонки	x := wherex
Wherey	позиция курсора y - номер строки	y := wherey
Gotoxy (x, y)	установить курсор в положение (x, y); x - колонка [1 .. 80]; y - строка [1 .. 25];	gotoxy(40, 10) gotoxy(50, wherey)

Определения констант - названий цветов палитры в текстовых режимах (модуль crt)

Номер	Цвет (имя константы)	Номер	Цвет (имя константы)
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	14	Yellow
7	LightGray	15	White

Процедуры и функции для управления программой

halt (результат)	выход из программы, <i>результат</i> - код завершения (целое число)
exit	выход из процедуры или функции пользователя
break	выход из цикла
readkey	остановка программы до нажатия клавиши; значение функции - код нажатой клавиши (символа); символ клавиши на экране монитора не отображается.
delay(время)	остановка программы; <i>время</i> - время задержки (миллисекунды (мс)) 1 мс. = 1 / 1000 с.

Операции над числами. Функции

Функция	Назначение	Пример вызова	Результат
abs(число)	Абсолютное значение числа	abs(-3.5)	+3.5
arctan(тангенс угла)	Арктангенс числа	arctan(0)	0
cos(угол)	Косинус угла (радианы)	cos(pi)	-1
exp(число)	Экспонента	exp(1)	2.718281828...
frac(число)	Дробная часть числа	frac(3.5)	0.5
int(число)	Целая часть числа	int(3.5)	3.0
ln(число)	Натуральный логарифм	ln(2.718281828)	~1.0
odd(число)	Проверка нечетности	odd(3)	True
pi	Число пи	pi	3.141592...
random(число)	«случайное» число	random(10)	Число в [0;10]
sin(угол)	Синус угла (радианы)	sin(pi)	0
sqr(число)	Квадрат числа	sqr(2.0)	4.0
sqrt(число)	Квадратный корень	sqrt(25.0)	5.0

ПРОЦЕДУРЫ

Процедура	Назначение	Пример вызова	Результат
inc(число)	Увеличить на 1	inc(n)	$n := n + 1$
dec(число)	Уменьшить на 1	dec(n)	$n := n - 1$

СТРУКТУРЫ ДАННЫХ

СТРОКИ

Модель организации данных строки ($s[0]$ = длина строки, $0 \leq s[0] \leq 255$)

$s[0]$	$s[1]$	$s[2]$	$s[3]$	$s[4]$	$s[n]$
--------	--------	--------	--------	--------	------	--------

`var <идентификатор> : string; {длина строки ≤ 255 символов}`

`var <идентификатор> : string [максимальное количество символов];`

ОПЕРАЦИЯ СОЕДИНЕНИЯ СТРОК (КОНКАТЕНАЦИЯ)

строка := строка1 + строка2 + ... + строка l ;

МАССИВЫ

Модель организации данных одномерного массива: $a[1..n]$

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$...	$a[n]$
--------	--------	--------	--------	--------	-----	--------

Модель организации данных двумерного массива: $a[1..m, 1..n]$

$a[1,1]$	$a[1,2]$	$a[1,3]$	$a[1,4]$	$a[1,5]$	$a[1,n]$
$a[2,1]$	$a[2,2]$	$a[2,3]$	$a[2,4]$	$a[2,5]$	$a[2,n]$
$a[3,1]$	$a[3,2]$	$a[3,3]$	$a[3,4]$	$a[3,5]$	$a[3,n]$
...	
$a[m,1]$	$a[m,2]$	$a[m,3]$	$a[m,4]$	$a[m,5]$	$a[m,n]$

`type <идентификатор> = array [1 .. m, 1 .. n] of <тип массива>;`

МНОЖЕСТВА

Модель множества

(количество элементов ≤ 256 , литеры или целые числа 0..255)

0	1	2	3	4	...	255
1	1	0	1	0	...	0

type <идентификатор> = set of <тип элементов>;

ОПЕРАЦИИ НАД МНОЖЕСТВАМИ

Принадлежность			Объединение			Разность			Пересечение		
X	A	X in A	A	B	A+B	A	B	A-B	A	B	A*B
0	[1,2]	false	[1]	[1,2]	[1,2]	[1,2]	[2]	[1]	[1,2]	[2,3]	[2]
1	[1,2]	true	[2]	[3]	[2,3]	[1,2]	[3]	[1,2]	[1,2]	[3,4]	[]

ЗАПИСИ

Модель записи

<i>R1</i>	<i>R2</i>	<i>R3</i>	...	<i>Rn</i>
-----------	-----------	-----------	-----	-----------

type <идентификатор> = record

<идентификатор поля> : <тип 1>;

<идентификатор поля>: <тип 2>

...

end;

ОПЕРАЦИИ НАД СИМВОЛАМИ И СТРОКАМИ

ОПЕРАЦИИ НАД СИМВОЛАМИ И ДРУГИМИ РЕГУЛЯРНЫМИ ТИПАМИ

Функция	Значение	Пример вызова	Результат
chr(номер символа <i>n</i>)	Символ номер <i>n</i> (<i>#n</i>)	chr(33)	'!'
ord(величина)	номер величины (код)	ord('!')	33
succ(величина)	Следующее значение в последовательности	succ('y')	'z'
pred(величина)	Предыдущее значение в последовательности	pred('y')	'x'

ОПЕРАЦИИ НАД СТРОКАМИ

ФУНКЦИИ

Функция	Назначение	Пример вызова	Результат
copy(<i>s</i>, <i>p</i>, <i>n</i>)	Выделить <i>n</i> символов из строки <i>s</i> начиная с позиции <i>p</i>	copy('роза', 3, 2)	'за'
concat(<i>s1</i>, <i>s2</i>, <i>sn</i>)	Соединить строки (литеры) в одну строку (конкатенация)	concat('р', 'роза')	'гроза'
length(строка)	Длина строки [символ.]	length('роза')	4
pos(<i>s1</i>, <i>s2</i>)	номер позиции строки <i>s1</i> внутри строки <i>s2</i> (если не найдена, 0)	pos('за', 'роза')	3

ОПЕРАЦИИ НАД СТРОКАМИ

ПРОЦЕДУРЫ

Процедура	Назначение	Пример вызова	Результат
delete(s, p, n)	Удалить n символов из строки s с позиции p	delete('роза', 1, 2)	'за'
insert(s1, s2, p)	Вставить строку (литеру) s1 в строку s2 с позиции p	insert('г', 'роза', 1)	'гроза'

ПРЕОБРАЗОВАНИЯ ДАННЫХ

Процедура (функция)	Назначение	Пример вызова	Результат
round(число)	округлить число	n := round(3.5)	4
trunc(число)	отсечь дробную часть	n := trunc(3.5)	3
str(n: p: q, s)	преобразовать число n в строку s	str(3.5:3, s)	s = '3.5'
val(s, n, p)	преобразовать строку s (литеру) в число n	val('+3.5', n, p)	n = 3.5; p = 0 p = место ошибки

ФАЙЛЫ. ОПЕРАЦИИ НАД ФАЙЛАМИ

ТИПЫ ФАЙЛОВ (ФАЙЛОВЫХ ПЕРЕМЕННЫХ)

1. текстовый `var f: text;`
2. типизированный `var f: file of <тип компонента>;`
3. нетипизированный `var f: file;`

Данные в типизированных и нетипизированных файлах хранятся в формате внутримашинной обработки.

Текстовый файл на диске – это полная модель клавиатуры и экрана (*input, output*).

Текстовый файл печати – *Ist* (модуль *printer*).

ФУНКЦИИ

Функция	Назначение	Пример вызова	Значения
eof(file)	условие «конец файла»	if eof(f) then ...	[true; false]
eoln(file)	условие «конец строки»	if eoln(f) then ...	[true; false]
filepos(file)	получить номер позиции для обмена	if filepos(f) > 0 then ...	[0;n]
filesize(file)	получить размер файла [количество записей]	seek(f, filesize(f))	[0;n]
ioresult	номер ошибки ввода-вывода ({ <i>\$I</i> -})	if iorresult = 0 then ...	[0;n]

ПРОЦЕДУРЫ

Процедура	Назначение	Пример вызова	Тип файла
assign(file, имя)	назначить файл для файловой переменной	assign(f, 'a: text') assign(f, path)	Все файлы
close(file)	закрывать файл	close(f)	Все файлы
reset(file [,длина-з.])	открыть файл	reset(f)	Все файлы
rewrite(file [,длина-з.])	создать файл	rewrite(f, 1)	Все файлы
append(file)	открыть для добавления	append(f)	Text
read(file, данные)	ввод данных	read(f, mydata)	Все файлы
readln(file, данные)	ввод данных, пропуск конца строки	readln(f, a, b, c)	Text
write(file, данные)	вывод данных	write(f, mydata)	Все файлы
writeln(file, данные)	вывод данных, вывод конца строки	writeln(f, a:10:2)	Text
blockread(f, данные, n, результат)	ввод блока из n записей, результат = сколько передано	blockread(f, a, 50,p)	File
blockwrite(f, данные, n, результат)	вывод блока из n записей результат = сколько передано	blockwrite(f, a,50,p)	File
seek(file, номер позиции)	установить номер позиции для обмена [0;n], n = число записей	seek(f, 0)	Кроме text

СТАНДАРТНЫЙ МОДУЛЬ (GRAPH) ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ

Процедура (функция)	Назначение	Пример вызова	Примечания
D := detect	Определить тип графического режима (номер драйвера)	d := detect	d = драйвер экрана (<i>integer</i>)
Initgraph(d, m, 'путь bgi')	Установить графический режим экрана	initgraph(d, m, 'd:\bp\bgi')	m = режим экрана (<i>integer</i>)
Cleardevice	Очистить экран и отменить установки цвета	cleardevice	
SetColor(цвет)	Установить цвет линии (рисунка)	setcolor(magenta)	
Setbkcolor(цвет)	Установить цвет фона (очистки)	setbkcolor(0)	
Putpixel(x, y, цвет)	Точка (x, y)	putpixel(5, 5, red)	
Line(x1, y1, x2, y2)	Линия (x1, y1) - (x2, y2)	line(10, 10, 20, 200)	
Lineto(x, y)	Чертить линию в (x, y)	lineto(100, 200)	
moveto(x, y)	Переместить перо в (x, y)	moveto(nx, ny)	
Circle(x, y, радиус)	Окружность (x, y, r)	circle(x, y, 20)	

СТАНДАРТНЫЙ МОДУЛЬ (GRAPH) ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ

arc(x, y, угл1, угл2, радиус)	Дуга окружности (x, y, r) от угла1 до угла2 (радиан)	arc(10, 10, 0, pi,5)	
setfillstyle(s, цвет заливки)	Установить стиль и цвет заливки	setfillstyle(1, green)	
floodfill(x, y, цвет границы)	Залить область с границей (цвет) цветом заливки	floodfill(p, q, 10)	
rectangle(x1, y1, x2, y2)	Прямоугольник (x1, y1) - (x2, y2)	rectangle(2, 2, 5, 10)	
bar(x1, y1, x2, y2)	Прямоугольник (x1, y1) - (x2, y2) с заливкой цвета	bar(2, 2, m, n)	
bar3d(x1, y1, x2, y2, z, n)	3-мерный прямоугольник (колонка) (x1, y1) - (x2, y2)	bar3d(2, 2, 20, 20, 10,true)	z = глубина n = <i>boolean</i> <i>true</i> = крыша колонки
x := getmaxx	максимальная граница для x	mx := getmaxx	0..639 (vga)
y := getmaxy	максимальная граница для y	my := getmaxy	0..479 (vga)
closegraph	Закрывает графический режим экрана	closegraph	