Язык программирования Object Pascal

Общий обзор.

§1 Историческая справка

В 1970г. - Паскаль (швейцарский ученый Никлаус Вирт) Первая версия была создана для машины CDC 6000.

В 1983г - Турбо-Паскаль фирмы Borland для ОС CP\M.

В 1984г. - Турбо-Паскаль для MS DOS.

В 1991г. - Turbo Pascal for Windows

В 1992 - Borland Pascal with Objects 7.0.

В 1995г. - Borland выпустила первую версию Delphi, фундаментом которой стал новый ЯП Object Pascal. Через год появилась Delphi 2 с новой версией Object Pascal 2.0. Затем, с интервалом в 1 год, выходят еще 3 версии Delphi: 2, 3, 4 и 5. Наконец, в середине 2001 г. выпускается версия 6, в 2002 – 7, 2003-8.

§2 Алфавит языка Object Pascal

Алфавит языка.

- Алфавит языка состоит из множества символов, включающих в себя буквы, цифры и специальные символы.
- Латинские буквы: от A до Z (заглавные) и от a до z (строчные).
- Цифры: 0,1,2,3,4,5,6,7,8,9.
- Шестнадцатиричные цифры: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

- Специальные символы: + * - / < > и т.д.
- Следующие комбинации специальных символов являются едиными символами (их нельзя разделять пробелами):

:= знак присваивания >=

<> не равно <=

(* *) ограничители комментария

(наряду с { })

Комментарий не может начинаться с $, т.к.

{$ воспринимается как начало директивы компилятору.

// - закомментировать одну строку

- К спецсимволам относят служебные слова, смысл которых определен однозначно. Служебные слова не могут быть использованы для других целей. С точки зрения языка - это единые символы.
- Например:

and type program as class

while if var uses unit

Идентификатор.

- Идентификатором называется символическое имя определенного программного объекта
- Идентификатор- это любая последовательность латинских букв, цифр и знака подчеркивания.

- Длина идентификатора не ограничена, (учитываются первые 256)
- Идентификатор <u>не может</u> содержать пробелов.
- Прописные и строчные буквы идентичны(LABEL1, Label1, label1)
- Не содержит спец. символов
- Не начинается с цифры

Примеры правильных идентификаторов:

a; MyProgramIsBestProgram; external

ALPHA; date_27_sep_39; _beta

Примеры неправильных идентификаторов:

1Program // *начинается цифрой*

block#l // *содержит специальный символ*

My Prog // *содержит пробел*

mod // *зарезервированное слово*

§3 Структура основного файла программы

- Любую программу на языке Object Pascal можно условно разделить на три основные части:
  - заголовок программы
  - раздел описаний
  - раздел операторов.

    - Каждое описание и определение заканчивается точкой с запятой ;

Основной файл программы имеет следующую структуру:

- **Program** <имя программы>;
- **Uses** <имена подключаемых модулей>;
- **const** <раздел описаний констант>;
- **type** <раздел описаний типов>;
- **var** <раздел описаний переменных>;
- **procedure (function)** <описание подпрограмм>;
- **begin**
  <раздел операторов>
- **end.**

Заголовок программы

Program <Имя программы>;

Имя программы – идентификатор.

Например:

Program smotr1;

Program Zadacha_1;

Раздел подключения модулей

Uses <имена модулей через запятую> **;**

В этом предложении перечисляются модули, загружаемые программой: системные модули и модули приложения.

Например:

uses SysUtils;

Раздел описания констант

В разделе описания констант перечисляются <u>именованные</u> константы и их значения.

Const <имя константы1> **=** <значение1>;

<имя константы2> **=** <значение2>;

Например:

Const n=10;

p=3.14159265358979323846264338327 95;

str =‘Маша’;

Целые числа записываются со знаком или без него по обычным правилам и могут иметь значение в диапазоне от -2^{63} до $+2^{63}-1^{1}$.

Вещественные числа записываются со знаком или без него с использованием десятичной точки и/или экспоненциальной части.

3.14E5 - 3,14 умножить на 10 в степени 5;

314000

-17e-2 -минус 17 умножить на 10 в степени минус 2 0,17

Логическая константа - false (ложь), либо true (истина).

Символьная константа - это любой символ ПК, заключенный в апострофы:

'z' - символ “z”; 'Ф' - символ “Ф”.

Если необходимо записать собственно символ апострофа, он удваивается: ''''

Строковая константа - любая последовательность символов (кроме символа *CR* - возврат каретки), заключенная в апострофы. Если в строке нужно указать сам символ апострофа, он удваивается, например:

'Это - строка символов';

'That''s all'.

Раздел описания типов

- Раздел описания типов позволяет определить новый тип в программе. (могут быть использованы ранее определенные константы.)

Type <имя тип>= <описание типа>;

Например:

Type MyType1= integer;

VyType2=1..n;

Раздел описания переменных

В разделе описания переменных содержится список переменных, используемых в программе, и определяется их тип.

Var V1,V2,..,Vn **:** T ;

где V1,V2,.., Vn -имена переменных данного типа T

Например:

```
Var   i, j : integer ;
            L : MyType2;
      M, k : char ;
```

Раздел операторов

Раздел операторов состоит из операторов языка ОР, отделенных друг от друга точкой с запятой. Он заключен в операторные скобки BEGIN END.

При этом после end ставится точка.

После слова begin и перед словом end точка с запятой обычно не ставится.

Операторы действия

1) **присваивание**

<перем>:=<выр-е>

Обязательно: **тип выражения и тип переменной должны быть совместимы по присваиванию.**

Пример. Присвоить значения переменным N, X, Y.

N:=-5; X:=5678.9; Y:=5.6789e3

2) Ввод

- **- READ (<список ввода>);**
- **- READLN (<список ввода>);**
- **Пример. Ввести значения переменных N, X,Y**
- readln(N); readln(X,Y)
- или readln (N,X,Y)
- или read(N); read (X,Y)
- Или …

Вывод

WRITE (<список ввода>);

WRITELN (<список ввода>) ***(после вывода переход на новую строку)***

элемент вывода имеет вид:

<выр-е> [:<мин. поле> [:<кол. дес. знаков после точки>]]

a b

- Readln(a,b);
- Readln(c,d);

a b

- 1 2 3
- c d
- 4 5

12

- writeln(a,b);
- write(c);
- write(d);

12

45

- a:=3;
- Write(a:5);
- Write(a/3);
- Write(a/3:6:4);

```
    3
1.66666666666667E+0000
1.6667
```

§4 Классификация типов данных

- Тип определяет
- 1) **формат внутреннего представления** объекта данного типа в оперативной памяти,
- 2) **множество допустимых значений**, которые может иметь объект данного типа,
- 3) **множество допустимых операций**, которые применимы к объекту данного типа.

- В ОР группы целых, символьных и строковых типов подразделяются на две категории:
- 1. **Фундаментальные (fundamental) типы,** формат представления которых в памяти (число битов и наличие знака) строго фиксируются и *будут выдерживаться неизменными* во всех последующих версиях ОР для любых операционных систем и компьютерных платформ.

- 2) **Родовые (generic) типы,** формат представления которых в памяти не фиксируется и *будет устанавливаться наиболее оптимальным способом*, в зависимости от реализации для конкретной операционной системы и компьютерной платформы.

Простые типы данных

Простые типы данных делятся на

- порядковые типы
- вещественные типы и
- типы работающие с датой-временем.

Порядковые типы данных

- **Порядковыми (ordinal) типами** называются те, в которых значения упорядочены и для каждого из них можно указать предшествующее и следующее значение.
- Для порядковых типов определен ряд стандартных функций:

функция	Параметр	Возвращаемое значение
Ord(x)	Выражение порядкового типа	Порядковый номер значения выражения в типе
Pred(x)	Выражение порядкового типа	Величина, предшествующая значению данного выражения
Succ(x)	Выражение порядкового типа	Величина, следующая для значения данного выражения
High(x)	Переменная порядкового типа	Максимально возможное значение
Low(x)	Переменная порядкового типа	Минимально возможное значение

x:=Pred(c); // при c=10 x=9

- Для порядковых типов определены процедуры инкремента **INC(X)** и декремента **Dec(X).** Эти процедуры соответственно увеличивают или уменьшают на единицу порядковый номер своего аргумента.

```
…
X:=5;
Inc(x);          //x=6
…
```

- Целый тип данных

Фундаментальные типы:

Идентификатор типа	Диапазон значений	Представление в компьютере
Shortint	–128..127	8-bit со знаком
Smallint	–32768..32767	16-bit со знаком
Longint	–2147483648..2147483647	32-bit со знаком
Int64	–2^63..2^63–1	64-bit со знаком
Byte	0..255	8-bit без знака
Word	0..65535	16-bit без знака
Longword	0..4294967295	32-bit без знака

Родовые типы:

Идентификатор типа	Диапазон значений	Представление в компьютере
Integer	–2147483648..2147483647	32-bit со знаком
Cardinal	0..4294967295	32-bit без знак

Допустимые операции, функции

1) Изменение знака -;

2) *,/,**Div,mod**

3) +,-

Div-деление нацело, 13 **div** 3 =4

mod-остаток от деления. 13 **mod** 3 =1

-1000 **mod** 3 = -1

$$
\begin{array}{r|l}
-\ 13 & 3 \\
\hline
12 & 4 \\
\hline
1 &
\end{array}
$$

Функция	Тип результата	Пример
Abs(X)	Целый	Abs(-9)=9
Sqr(x)	Как у параметра	sqr(9)=81
Chr(x)	Символьный	Chr(65)=‘a’
Odd(x)	Логический	Odd(9)=true
Dec(x,[i])	Процедура	Dec(9,2)=7
Inc(x,[i])	Процедура	Dec(9,2)=11
Random(x)	Как у параметра	Random(10);

Randomize; генератор случайных чисел

Пример программы, в которой значение выходит из допустимого диапазона

```
Var k:Word;
begin
  k := 65535; // Максимальное значение типа Word
  k := k+1; // По правилам математики k=65536
  writeln(k); // На самом деле k=0!
  end;
```

Логические типы

Идентификатор типа	Длина, байт
Boolean	**1**
ByteBool	1
Bool	2
wordBool	2
LongBool	4

Значения логического типа **False (ложь) или True (истина).**

Для них справедливы правила:

- `Ord(False) = 0`
- `Ord(True) = 1`
- `Succ(False)= True`
- `Pred(True) = False`
- `False < True`

Логические операции

- Or
- And
- Not

B or C

A and B

Not(C)

Символьные типы

AnsiChar	1байт	Символ кода ANSI
WideChar	2байта	Символ кода Unicode
Char	**1 байт**	**Родовой тип** ~AnsiChar

Код ANSI

(American National Standard Institute)

коды **0... 127** (Код 43 ‘+’ код 91 ‘[’ код 126 ‘~’)

коды**128...255** Стандартные Windows-шрифты:

Arial Cyr

Courier New Cyr

Times New Roman

Коды **192..223 - “А”... “Я”,**

Коды **224…255 - “а”... “я”**

1. Операции отношения (<,>,=,>=, <=, <>)

2. Chr(b)

Chr(48)=‘0’

Chr(58)=‘:’

3. Upcase(ch)

Upcase(‘a’)=‘A’

Upcase(‘5’)=‘5’

Перечисляемый тип

- Задается перечислением тех значений, которые он может получать.
- Каждое значение - идентификатор и располагается в списке в круглых скобках

```
Type
  colors=(red, white, blue);
```

Пример

```
type
colors=(black, red, white);
ordenal=(one, two, three);
days=(Monday, Tuesday,
  Wednesday);
...
```

- `Ord(black)=0, ... , Ord(white)=2,`
- `Ord(one)=0, ... , Ord(three)=2,`

Var col : colors;

num : ordenal;

day : days;

допустимы операторы

col := black;

num := Succ(two);

day:=Pred(tuesday);

недопустимы

col := one; day:=black;

эквивалентны следующие присваивания:

```
col := black;
col := colors (0) ;
```

? col:=0 ?

Переменные перечисляемого типа можно объявлять без предварительного описания этого типа, например:

```
var
col: (black, white, green);
```

Тип-диапазон
Интервальный тип

Подмножество своего базового типа, в качестве которого может выступать любой порядковый тип, кроме типа-диапазона.

Type

T = N1..N2;

```
type
  digit = '0'..'9';
  dig2 = 48 .. 57;
```

Тип-диапазон можно указывать непосредственно при объявлении переменной, например:

```
var
  date : 1..31;
  month: 1..12;
  Ichr : 'A'..'Z';
```

High(X)

Low (X)

Вещественные типы

Real Double	8	$\pm 5.0*10^{-324}$ до $\pm 1.7*10^{308}$
Real48	6	$\pm 2.9*10^{-39}$ до $\pm 1.7*10^{38}$
Single	4	$\pm 1.5*10^{-45}$ до $\pm 3.4*10^{38}$
Extended	10	$\pm 3.6*10^{-4951}$ до $\pm 1.1*10^{4932}$
Comp	8	-2^{63} до 2^{62}
Currency		-922337203685477.5808 до 922337203685477.5807

Внутреннее представление вещественного числа в памяти ПК

s	e	m

S – знаковый разряд числа

E – экспоненциальная часть

M – мантисса числа

Встроенные математические функции

Функция	Тип результата	Пример
abs (x)	Как у параметра	
Int (x)	Вещественный	Int(123.456)=123.0
frac(x)	Вещественный	frac(123.456)=0.456

arctan(x) cos (x) sin (x)

exp(x) ln(x) sqr(x)

sqrt(x) pi

Арифметические выражения

<u>Правила записи арифметических выражений</u>:

1. Все символы пишутся в одну строку

$$\frac{2a + \sqrt{0,5 \sin x}}{0,2 - \ln y}$$

(2*a+sqrt(0,5+sin(x)))/(0.2-ln(y))

2. Нельзя ставить два знака подряд

+(-3)

3. Операции с более высоким приоритетом выполняются раньше операций с меньшим приоритетом

4. Операции одного приоритета выполняются слева направо

Приоритет операций

Приоритет	Операции
1	Not
2	* / div mod and
3	+ - or
4	= <> < > <= >=

4. На каждую арифметическую операцию и вычисление стандартной функции тратится 2 единицы времени

$$\frac{x+y}{y+1}-\frac{xy-12}{34+x}$$

$$x*\ln(x)+\frac{y}{\cos(x)-\frac{x}{3}}$$

Тип дата-время TDateTime

Назначение: одновременное хранение и даты, и времени.

Внутреннее представление: вещественное число с фиксированной дробной частью ~Double (8 байт).

Целая часть числа - дата,

Дробная часть - время

0 12/30/1899 12:00 am

Дата - количество суток, прошедших с 30 декабря 1899 года,

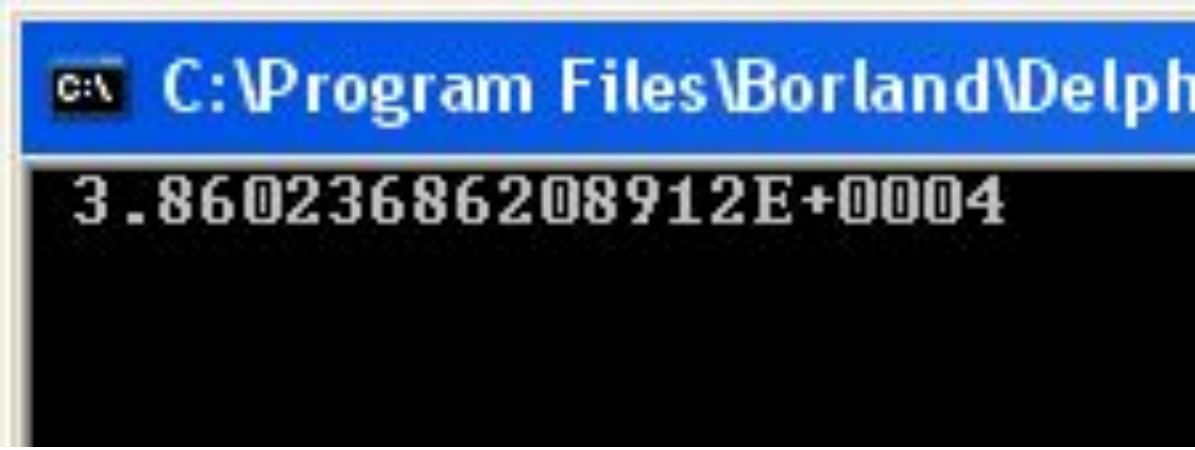
Время - часть суток, прошедших с 0 часов.

- 2.75 1/1/1900 6:00 pm
- -1.25 12/29/1899 6:00 am

- TDate используется для получения сегодняшей даты
- TDate represents a special type of TDateTime value that has no decimal part. Значение TDate – количество дней прошедших с 12/30/1899.

function Now: TDateTime;

- Возвращает текущую дату и время
- Writeln(Now);



- Writeln(DateTimeToStr(Now));

функция Time или GetTime возвращает TDateTime;

- **Var**
- DateTime : TDateTime;
- **begin**
- DateTime := Time;
- Writeln(TimeToStr(DateTime));

- TimeToStr – переводит время в строку

Процедура DecodeTime

- Выделяет из переменной TDateTime часы минуты, секунды и миллисекунды

- DecodeTime(Time,Hour, Min, Sec, MSec);

Hour, Min, Sec, MSec типа Word

Time типа TDateTime

```
var
 Present: TDateTime;
 Year, Month, Day, Hour, Min, Sec, MSec:
   Word;
Begin
  Present:= Now;
  DecodeDate(Present, Year, Month, Day);
end.
```