

G L O S S A R Y

На тему: Object Pascal

Object Pascal

Алфавит языка

Лексическая структура языка

Ячейка

Значение

Константа

Переменная

Операторы присваивания

Система типов

Арифметические выражения

Логические выражения

Строковые выражения

Object Pascal. Язык объектно-ориентированного

программирования Object Pascal применяется при работе в среде визуального программирования Delphi. Язык Object Pascal в основном включает «старый» язык Borland Pascal.

Программы на языке Object Pascal состоят из нескольких файлов: файла проекта (Delphi Project) с расширением *.dpr, одного или нескольких файлов модулей (Unit) с расширением *.pas и файлов дизайнера экранных форм с расширением *.dfm.

Файл проекта содержит текст основной программы Program, с которой начинается выполнение всей программы. Тексты вызываемых подпрограмм и используемых объектов находятся в файлах модулей.



PASCAL

Алфавит языка. Основными символами языка Object Pascal

являются:

СИМВОЛЫ _ + -

26 больших и 26 малых латинских букв A,B, ...Y,Z, a,b, ..., y,z

10 арабских цифр 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,

специальные символы * / = ^ < > () [] { } . , : ; ' # \$ @

Буквы русского алфавита не входят в состав алфавита языка. Их использование допустимо только в строковых и символьных значениях.

Нет различий при использовании больших и малых букв в записи имен переменных, процедур, функций и меток. Их максимальная длина ограничена 126 символами.



[Вернуться назад](#)

Лексическая структура языка. Строительным материалом для конструирования программного текста модуля являются лексемы – особые языковые конструкции, имеющие самостоятельный смысл. Лексемы строятся при помощи символов алфавита языка. В Object Pascal различают следующие основные классы лексем:

Зарезервированные (служебные) слова. Этот класс состоит из слов, построенных только с помощью букв алфавита. Служебные слова можно использовать только по прямому назначению, т. е. так, как их назначение определил разработчик языка. Ни в каком другом виде, например в качестве имен переменных, их использовать нельзя.

Идентификаторы (имена). Идентификаторы или имена предназначены для обозначения констант, переменных, типов, процедур, функций, меток. Они формируются из букв, цифр и символа "_" (подчеркивание). Длина имени может быть произвольной, однако компилятор учитывает имена по его первым 63 символам. Внутри имени не должно быть пробелов.

Object Pascal в именах не различает больших и малых букв. Так следующие имена будут идентичны:

SaveToFile, SAVETOFILE, savetofile, sAVeOfILE.

Среди программистов установилось хорошее правило, в соответствии с которым имена формируются таким образом, чтобы одновременно выполнять роль комментария, поясняющего назначение имени. Так, в приведенном примере имя переводится с английского как "сохранить в файле". Кроме того, с учетом невозможности вставки внутри такого имени пробелов, первые буквы слов обычно пишут заглавными, а прочие строчными. Из приведенного примера хорошо видно, что именно такой способ записи наиболее нагляден для визуального восприятия имени. Нередко в качестве заменителя пробела используют символ "_". Однако это удлиняет и без того длинные имена. Преимущества длинных имен совсем не означают, что нельзя применять короткие имена. Понятно, что проще набрать с клавиатуры и использовать оператор

`a := a + 1,`

чем идентичный ему оператор

`Disk_C_DirctoryCounter := Disk_C_DirctoryCounter + 1.`

Следует, однако, с большой осторожностью использовать короткие имена, т. к. это нередко приводит к путанице между глобальными и локальными переменными, обозначенными одинаковыми именами, и, как следствие, к ошибкам в работе программы. Наиболее удобным, безопасным и желательным можно считать локальное использование

коротких имен, когда они описаны и использованы внутри какой-нибудь сравнительно небольшой по объему текста процедуры или функции и их действие ограничено пределами только этой алгоритмической единицы. При подозрении на путаницу, действие такой переменной легко проконтролировать визуально.

Изображения. К их числу относятся константы, символьные строки и некоторые другие значения.

Знаки операций формируются из одного или нескольких символов по определению действий, связанных с преобразованием данных.

Разделители используются с целью большего структурирования модуля, с тем чтобы повысить визуальное восприятие длинных текстов. К их числу можно отнести ; := (.

Комментарии. Эти лексемы используют для пояснения отдельных фрагментов текста программы. Они представляют собой последовательность символов, заключенную в фигурные скобки { } или в разделители (* и *), а также последовательность символов, расположенных в строке справа от двух следующих друг за другом символов /.

Примеры комментариев:

```
{ Функция вычисления количества дней между двумя датами }  
(* Функция вычисления количества дней между двумя датами *)  
// Неправильный ответ
```

Пробел. Этот символ не имеет видимого изображения и служит для отделения лексем друг от друга в тех случаях, когда это необходимо. Обычно использование одного или нескольких рядом стоящих пробелов не искажает смысл программы.

Ячейка. Этот несколько устаревший, но весьма удобный термин обозначает фрагмент памяти, который можно представить как некий контейнер для хранения данных определенной структуры. Ячейка всегда имеет свое уникальное имя, которое служит адресом, по которому расположены находящиеся в ней данные. Примером ячейки могут служить любые разрешенные имена, например `a1`, `Imk12`, `Count` и т. д. Термин "ячейка" не является языковым термином Object Pascal и используется здесь только для большей наглядности при описании основ языка.

Значение. Значение – это постоянная величина или структурный комплекс постоянных величин, выраженных в явном виде. Значение не имеет имени.

Примеры значений:

-55.455051 { обыкновенное вещественное число},

'Расчет посадки с натягом' {строка символов}.

Константа. Константа – это ячейка, в которой всегда хранится одно значение. Константы не могут быть изменены в ходе выполнения программы. В этом смысле константа отвечает общепринятому определению постоянной (неизменяемой) величины. Всякая константа должна быть описана, т. е. должно быть явно указано ее значение. Значение константы неявно определяет ее тип.

Необходимо отметить, что в языке существуют так называемые типизованные константы, которые в ходе прохождения программы могут быть изменены. Тип константы указывается в специальной языковой конструкции, начинающейся словом `Type` (тип).

Переменная. Переменная – это ячейка, в которой в каждый момент времени хранится одно значение или не хранится ничего. Переменная в любой момент времени может быть изменена программой. Всякая переменная должна быть описана. т.е. должен быть явно указан ее тип. Тип переменной указывается в специальной языковой конструкции, начинающейся словом Var (от английского variable – постоянная).

Операторы присваивания. Оператор присваивания – это команда, предназначенная для изменения содержимого ячейки. С его помощью происходит изменение значения переменной (или типизованной константы).

Синтаксис оператора присваивания:

$x := y; \{ \text{читается "x присвоить y"} \}$

Здесь x – переменная, y – выражение. Выражением могут быть, в частности, переменная, константа или значение. Последовательность символов ":= " обозначает операцию присваивания, в соответствии с которой сначала вычисляется выражение y , затем получившийся результат в виде значения записывается в переменную x (см. подробнее гл. 9).

Примеры:

$d := 5; \{ \text{значение 5 записывается в переменную D} \},$

$h := d + 12.5; \{ \text{выч. } 5+12.5, \text{ рез. } 17.5 \text{ записывается в переменную h} \}.$

Система типов. В языке Object Pascal все переменные, т. е. ячейки памяти, предназначенные для записи, чтения и хранения значений, должны быть предварительно описаны. Это означает, что всякая переменная должна быть явно отнесена к какому-либо типу.

Тип – это одновременно структура и описание множества значений, которые могут быть присвоены такой переменной.

Язык Object Pascal имеет множество разнообразных типов. Более того он позволяет самому пользователю конструировать самые разнообразные типы, которые могут быть ему необходимы. Конструирование таких типов производится из сравнительно ограниченного количества стандартных типов.

Типы имеют свою иерархию. На верхнем этаже иерархии расположены следующие типы: простые, составные, ссылочные и процедурные.

Основными типами языка являются стандартные простые типы и стандартные структурные типы.

Простые типы делятся на скалярные и ограниченные типы. Скалярные типы делятся на стандартные и перечислимые. Стандартные скалярные типы делятся на пять видов:

целые [Integer],

вещественные [Real],

логический (булевский) [Boolean],

символьные [Char],

строковые [String].

К ним примыкает особый вариантный тип [Variant].

Арифметические выражения.

Вычислительная система выполняет вычислительные и управляющие операции по командам, которые представлены в программе с помощью операторов. Большинство таких операторов строится с использованием выражений, которые в практике программирования играют большую роль, определяя способ и порядок преобразования данных. Выражения состоят из операндов (значений, констант, переменных, функций), соединенных с помощью операций. Для изменения порядка выполнения операций могут быть использованы круглые скобки. Наиболее важную роль играют арифметические, логические и строковые выражения.

При описании арифметических выражений для простоты типом Integer обозначен любой целый тип, а типом Real – любой вещественный тип. Выражение строится с помощью арифметических значений, констант, переменных, функций, арифметических операций. В выражениях можно применять круглые открывающие и закрывающие скобки. При этом количество открывающих скобок должно быть равно количеству закрывающих скобок.

При вычислении выражения операции выполняются в строго определенной последовательности в соответствии с их приоритетом. Порядок выполнения операций можно изменить применением блоков, включающих подвыражения, заключенные в круглые скобки.

В языке Object Pascal существует шесть арифметических операций. Учитывая, что арифметические операции образуют подмножество множества всех операций языка, в табл. 7 показано абсолютное значение приоритета каждой операции.

Операция	Наименование	Приоритет
+	Сложение	2
-	Вычитание	2
*	Умножение	1
/	Деление	1
div	Деление нацело	1
mod	Остаток от деления	1

При вычислении выражения его тип определяется типами операндов. Операндом называется любая компонента, к которой применяется операция. Операндом может быть, например, значение, константа, переменная или выражение, заключенное в скобки. Типы элементарных выражений показаны в таблице ниже.

Операция	Тип операндов	Тип результата	Пример
+	Integer, real	Integer, real	X+Y
-	Integer, real	Integer, real	Result -1
*	Integer, real	Integer, real	P * InterestRate
/	Integer, real	Real	X/2
Div	Integer	Integer	Total div UnitSize
Mod	Integer	Integer	Y mod 6

Порядок выполнения операций определяется приоритетом операций и расположением внутренних выражений, заключенных в круглые скобки. Все операции в арифметическом выражении выполняются слева направо.

Логические выражения. Логическое выражение строится с помощью других выражений, (арифметических, строковых и др.), значений, констант, переменных, функций, логических операций и логических отношений.

В языке существует четыре логических операций. Приоритет операций показан в таблице ниже.

Операция	Наименование	Приоритет
Not	Отрицание	3
And	Конъюнкция	4
Or	Дизъюнкция	5

Значения элементарных логических выражений, поясняющих назначение этих операций, приведены в следующей таблице.

A	B	Not AB	And BA	Or BA
True	True	false	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	false

Строковые выражения. Строковые выражения, частными случаями которых могут быть пустой символ "" или одиночный символ (например 'A'), строятся из строковых или символьных значений, констант, переменных и строковых функций при помощи строковой операции конкатенации (присоединения). Эта операция обозначена символом + (плюс). Скобки в строковых выражениях не применяются.