

Теория для Занятия №12.

Основные понятия языка Паскаль.

Программа на языке Паскаль формируется с помощью конечного набора знаков, образующих *алфавит* языка, и состоит из букв, цифр, специальных символов.

В качестве *букв* используются прописные и строчные буквы латинского алфавита и знак подчёркивания; в качестве *цифр*: арабские цифры от 0 до 9.

При написании программ применяются *специальные символы*: +, -, *, /, =, <, >, [], (), @, {, :, ;, ', # (номер), \$ (знак денежной единицы), ^ (тильда), пробел, точка и запятая.

Неделимые последовательности знаков алфавита образуют слова, отделённые друг от друга разделителями и несущими определённый смысл в программе. Разделителем может служить пробел, символ конца строки, комментарий. Слова подразделяются на зарезервированные, стандартные идентификаторы и идентификаторы пользователя.

Зарезервированные слова являются составной частью языка и их нельзя использовать в качестве идентификаторов. В языке Паскаль зарезервированными являются следующие слова: **and, array, begin, case, const, div, do, downto, else, end, file, for, forward, function, goto, if, in, lable, mod, nil, not, of, or, packed, procedure, program, record, repeat, set, shl, shr, string, then, to, type, unit, until, uses, var, while, with, xor.**

Стандартные идентификаторы служат для обозначения заранее определённых разработчиками языка типов данных, констант, процедур и функций.

Идентификаторы пользователя используются для обозначения меток, констант, типов, переменных, процедур и функций, определённых самим программистом.

Общая структура программы.

Описание меток, определение констант, определение типов, описание переменных

Структура программы.

Program <имя программы>;

Uses {подключаемые модули}

Label {объявление глобальных меток}

Const {объявление констант}

Type {объявление типов}

Var {объявление переменных}

Procedure {описание процедур}

Function {описание функций}

Begin

{операторы}

End.

Раздел описаний

Раздел операторов

Все данные, в зависимости от способа их хранения и обработки можно разделить на две группы константы и переменные.

Константами называются элементы данных, значения которых установлены в описательной части программы и в процессе выполнения программы не изменяются.

Стандартные виды констант:

1. *Целочисленные* - определяются посредством чисел, записанных в десятичном или шестнадцатеричном формате, не содержащих десятичной точки.

2. *Вещественные* - определяются посредством чисел, записанных в десятичном формате данных.

3. *Символьные* - это любой символ персонального компьютера, заключённый в апострофы.

Строковые - определяются последовательностью произвольных символов, заключённых в апострофы.

Типизированные - переменные с начальным значением. Каждой типизированной константе ставится в соответствие имя, тип и начальное значение.

Зарезервированные константы.

Формат описания констант:

Const

Идентификатор=значение;

Например:

Const

A=5; b=3.51; s1=0.25

Переменные в отличие от констант могут менять свои значения в процессе выполнения программы. Тип констант автоматически распознаётся компилятором без предварительного описания. Тип переменной должен быть описан перед тем, как с переменными будут выполняться какие-либо действия.

Формат описания переменных:

Var

Идентификатор: тип;

Например:

Var

A,b,c: Real;

X: integer;

Арифметические выражения

Арифметические выражения строятся из имен переменных, констант, знаков операций и круглых скобок так, как это принято в математике. При вычислении их значений операции выполняются в порядке приоритета: *, /, DIV, MOD, а затем + и -. Операции одинакового старшинства выполняются слева направо.

Арифметические операторы

| Оператор | Назначение |
|----------|-----------------------------------|
| + | Сложение |
| - | Вычитание |
| * | Умножение |
| / | Деление |
| div | Целочисленное деление |
| mod | Остаток от целочисленного деления |

Наряду с переменными и константами в арифметические выражения можно включать функции. При определении значения выражения, прежде всего, вычисляются значения входящих в него функций. В Паскале имеются следующие стандартные функции:

| Функция | Назначение |
|---------------------|---|
| ABS(X) | Модуль числа x |
| ARCTAN(X) | Arctgx |
| COS(X) | cosx |
| EXP(X) | e^x |
| FRAC(X) | Дробная часть x |
| INT(X) | Целая часть x |
| LN(X) | lnx |
| PI | Значение PI=3.1415926535897932385 |
| ROUND(X) | Округление до ближайшего целого |
| SIN(X) | Sin x |
| SQR(X) | x^2 |
| SQRT(X) | Квадратный корень X |
| TRUNC(X) | Отбрасывание дробной части |
| POWER(X,Y) | X^Y возведение X в степень Y (PascalABC.NET) |
| EXP(B*LN(A)) | A^B – В Паскале нет функции возведения в степень, поэтому степень определяется через уже определенные функции экспоненты(Exp) и натурального логарифма(Ln). |

Аргумент функции обязательно заключается в скобки.

Выражение на Паскале, как впрочем, и на других языках программирования, записывается в одну строчку, а для сохранения порядка действий используются скобки. Все действия должны быть указаны. Например, $XY+X^2$ надо записать как $X*Y+SQR(X)$.

Типы данных в языке Паскаль

Каждый элемент данных относится к одному из конечного множества типов, допустимых для конкретного языка программирования. Тип - это множество значений, которые могут принимать объекты программы, и совокупность операций, допустимых над этими значениями.

Все типы данных разделяются на две группы: скалярные (простые) и структурированные (составные).

Простые (скалярные) типы данных:

К скалярным (scalar - простые) типам данных относят типы данных таких величин, значения которых не содержат составных частей.

Все простые данные имеют два характерных свойства: неделимость и упорядоченность их значений.

Целочисленные типы данных паскаль:

| Тип | Диапазон значений | Размер, байт |
|-----------------|--------------------------------|--------------|
| shortint | -128..127 | 1 |
| byte | 0..255 | 1 |
| integer | -32768..32767 | 2 |
| word | 0..65535 | 2 |
| longint | -2147483648..2147483647 | 4 |

Целочисленные типы данных PascalABC.NET:

| Тип | Диапазон значений | Размер, байт |
|---------------------------|--|--------------|
| shortint | -128..127 | 1 |
| smallint | -32768..32767 | 2 |
| integer, longint | -2147483648..2147483647 | 4 |
| int64 | -9223372036854775808..9223372036854775807 | 8 |
| byte | 0..255 | 1 |
| word | 0..65535 | 2 |
| longword, cardinal | 0..4294967295 | 4 |
| uint64 | 0..18446744073709551615 | 8 |

Целочисленные типы данных представляют собой значения, которые могут использоваться в арифметических выражениях и занимать в памяти от 1 до 4 байт. Над ними определены следующие операции: +, -, *, **div**, **mod**, а так же функции **Orc**, **Succ**, **Pred**.

Практическая часть

Перед разбором программы рекомендуется рассмотреть примеры выполнения операций *Div* и *Mod* над различными (положительными и отрицательными) числами:

| | |
|---------------|---------------|
| 19 Div 4=4; | 19 Mod 4=3; |
| -19Div4=-4; | -19Mod4=-3; |
| 19 Div -4=-4; | 19 Mod -4=3; |
| -19 Div-4=4; | -19 Mod-4=-3. |

Программа выделения цифр из десятичного числа

В данной программе определяются цифры трехзначного числа. Можно ее использовать и для определения цифр двузначного числа, просто цифра сотен в этом случае равна нулю.

```
Program My2_1;  
Var    a, one, dec, hun, rez:Integer;  
Begin  
WriteLn('Введите число'); ReadLn(a);  
one:=a Mod 10;  
WriteLn('Цифра единиц числа — ',one);  
dec:=(a Div 10) Mod 10;  
WriteLn('Цифра десятков числа — ',dec);  
hun:=a Div 100;  
WriteLn('Цифра сотен числа — ',hun);  
Rez:=hun*100+dec*10+one;  
WriteLn('А это тоже число — ',rez);  
Write('Enter ');  
ReadLn  
End.
```

Например, если вы введете число 137, то значение переменной *one* будет равно 7,

dec — 3 и *hun* — 1. Вспомните деление чисел столбиком.

Примечание

1. Не забудьте сохранить программу под именем *Program2_1.pas*.
2. Измените программу *My2_1* для нахождения цифр двузначного числа.

Сохраните ее под именем *Program2_2.pas*.

3. Измените программу *My2_1* для нахождения цифр четырехзначного числа. Сохраните ее под именем *Program2_3.pas*.

4. Деление на 10 и нахождение остатков от деления мы рассмотрели выше. Рассмотрите пример деления столбиком числа 137 на 2.

5. Наберите следующую программу, отладьте ее и попробуйте дать объяснение полученному результату. Измените программу так, чтобы она правильно работала, например, с числом 115.

```
Program My2_2;  
Var rez:Integer;  
Begin  
WriteLn('137');  
WriteLn('10001001');  
Rez:=1*128+0*64+0*32+0*16+1*8+0*4+0*2+1*1;  
WriteLn(rez);  
End.
```

6. Наберите следующую программу:

```
Program My2_3;  
Var a:Integer; b:Word; r1: Smallint; r2:Integer;  
Begin  
a:=32000; b:=64000;  
r1:=a+b; WriteLn(r1);  
r2:=a+b; WriteLn(r2);  
End.
```

Фигурные скобки } и текст в них можно не набирать, это пояснения к строкам программы!!!

После запуска вы увидите, что значение переменной **r1** равно 30464, а значение переменной **r2** — 96000. Если изменить тип переменной **r1** на *Word*, то результат не изменится. Используя информацию из таблицы, приведенной в начале занятия, измените программу так, чтобы проделать аналогичные эксперименты с данными других целых типов. Попробуйте понять, как получаются эти результаты.

7. Добавьте в программу *My2_3* перед оператором *ReadLn* следующие два оператора:

```
WriteLn(LongInt(100*a));
```

```
WriteLn(100*LongInt(a));
```

Функция *LongInt* преобразует переменную типа *Integer* в тип *LongInt*. В первом случае преобразование осуществляется после умножения, а во втором — перед умножением. В первом случае получается результат, далекий от истины, — отрицательное число -11264, во втором правильный, 3200000.

Задания для самостоятельной работы

1. Чему равны значения переменных *a* и *b* после выполнения приведенной последовательности действий?

```
a := 15 Div (16 Mod 7) ; b := 34 Mod a * 5 - 29 Mod 5 * 2; a := 4 * 5 Div 3 Mod 2; b := 4 * 5 Div (3 Mod 2);
```

2. Дано двузначное число. Определите:

- сумму и произведение цифр числа;
- число, образованное перестановкой цифр исходного числа.

3. Дано трехзначное число. Определите:

- сумму и произведение цифр числа;
- число, образованное перестановкой цифр исходного числа;
- число, полученное перестановкой цифр десятков и единиц;
- число, полученное перестановкой цифр сотен и десятков;
- четырехзначное число, полученное приписыванием цифры единиц в качестве цифры тысяч (например, из числа 137 необходимо получить число 7137).

Примечание

Сколько различных чисел можно получить из трехзначного числа путем перестановки цифр?

4. Решите задачу 3 (кроме последнего пункта) для четырехзначных чисел.**(программу прислать по почте!!!)**

Примечание

Предложить максимальное количество разумных модификаций рассматриваемой задачи.

5. Арифметическая прогрессия — это последовательность чисел, в которой разность между последующим и предыдущим элементами остается неизменной. Последовательность 12, 15, 18, 21, 24, ... является арифметической прогрессией, 12 — первый член прогрессии (a_1), разность прогрессии равна 3. Любой член прогрессии вычисляется по формуле $a_n = a_1 + d \cdot (n - 1)$, где d — разность прогрессии, n — номер взятого члена. Даны a_1 и d . Найдите (экспериментальным путем) n , при котором значение a_n выходит за диапазон типа *Integer*.**(программу прислать по почте!!!)**

6. Сумма первых n членов арифметической прогрессии вычисляется по формуле $S_n = (a_1 + a_n) \cdot n / 2$. Даны a_1 и d . Найдите (экспериментальным путем) n , при котором значение S_n выходит за диапазон типа *Integer*.**(программу прислать по почте!!!)**