

ОГРАНИЧЕННЫЕ ТИПЫ (ДИАПАЗОНЫ)

TYPE

<имя_диапазона> = <начальн.граница> .. <конечная_граница>;

Пример: TYPE NEDEL=(PN, VT, SR, HT, PT,SB, VS);

TEST=(test1,test2,test3,test4,test5);

Byte20=1..20; {поддиапазон для базового типа Byte}

LETTERS='A'..'Я'; {прописные буквы из базового тип Char }

TESTOD=test2..test5; {часть перечислимого типа TEST}

RABNED=PN..PT; {ограничение на перечислимый тип NED}

Пример:

TYPE date=1..31; di=2*(2*3-2).. (6+23);

VAR L:di; L1:date;

BEGIN

L:=2*10+5; ...

L1:=2*10+60; {неверно, т.к. L1 выходит за пределы}

END. {диапазона}

TYPE DIAP= 1..10;

vektor = array[diap] of real;

VARdate: 1..31; {без введения имени типа}

month: 1..12;

lchr: 'A'..'Z'; {-}

ПОЛНАЯ СТРУКТУРА ПРОГРАММЫ

PROGRAM <имя программы>;

Блок подключаемых модулей;

Блок описания меток перехода;

Блок определения констант;

Блок определения типов;

Блок описания переменных;

Блок *определения процедур и функций*;

BEGIN

Блок исполняемых операторов

END.

-

СТРУКТУРА ПРОГРАММЫ

PROGRAM <имя программы>;
USES <список библиотечных модулей>;
LABEL <список меток в основном блоке>;
CONST <определение констант>;
TYPE <определение типов>;
VAR <определение глобальных переменных>;
 <определение процедур>
 <определение функций>;
BEGIN
 <основной блок программы>
END.

```
USES CRT, DOS, GRAPH;
```

Модуль **SYSTEM**

-

Блок описания меток

Метка - числовое значение от 1 до 9999 или идентификатор

```
LABEL <метка1>,<метка2>,...<меткаN>;
```

```
LABEL 1, 40, M1, EPS;
```

3, 03,003 считаются одинаковыми

Блок описания констант

CONST

```
<имя1>=<константа1>;
```

```
<имя2>=<константа2>;
```

```
CONST <имяK>:<тип>=<константаK>;
```

```
<имяR>:<тип>=<константаR>;
```

Например: CONST

```
NIK=100;
```

```
AN= -NIL;
```

```
A='XYZ';
```

```
d1:integer=24;  
d2:boolean=false;  
d= '----';  
-
```

Блок определения типов

TYPE <идентификатор1>=(список_компонентов);

...
<идентификаторN>=(компоненты, входящие в данный тип);

Пример: TYPE **NEDEL**=(PN,VT,SP,4T,PT,SB,VS);
FAMILY=(FATHER, MATHER, SON);

Блок описания переменных

```
VAR      A1, A2, ..., An: <тип1>;  
...  
          B1, B2, ..., Bn: <типK>;
```

```
VAR A, B, C: INTEGER;      R:NEDEL;  
    D, M: REAL;           Mas:array[1..13] of byte;
```

```
CONST K=10; L=5;  
VAR A: ARRAY [1..K] OF INTEGER;  
    D, B:ARRAY[1..K,1..L] OF REAL;
```

Определение процедур и функций

Заголовок Тело подпрограммы

Структура описания заголовка **процедуры**:

PROCEDURE <имя>(<список формальных параметров>);

PROCEDURE <имя>;

VAR

PROCEDURE SUMMA(A: REAL; B:INTEGER; VAR S,P:REAL)

Структура описания заголовка **функции** имеет вид

FUNCTION <имя>(<список формальных параметров>):<тип>;

Пример.

```
FUNCTION SUMMA (x,y: REAL): REAL;
```

```
VAR a,b:real; {описания локальных переменных}
```

```
BEGIN
```

```
    read(a,b); ... ..
```

```
        SUMMA:=X+Y+A*B
```

```
END;
```

Функция возвращает единственное значение в основную программу через имя функции

<имя функции>:=<выражение>

$$Z = A1^{x1} + A2^{x2}$$

$$A^x$$

```
PROGRAM AstepX;  
VAR A1,A2,X1,X2,Z:real;  
Function ST(A, x: real): real;  
Begin  
    ST:=Exp (X*Ln(A))  
End;  
BEGIN  
    READ (X1,X2,a1,a2);  
    Z:=ST(A1,X1)+ST(A2,X2);  
    WRITELN(` Z= ` ,Z:10:4)  
END.
```

Длина строки 126 символов. Объем текста файла – 64К

{текст комментария} либо (*комментарий*)

(*внешний охватывает{внутренний} комментарий*)

Структура сложной программы:

```
PROGRAM сложная_программа;  
USES {подключаемые библиотеки(модули)};  
CONST {константы и переменные для }  
VAR {выполнения математических расчетов}  
    {Определение процедур и функций математических расчетов}  
CONST{константы, типы и переменные,}  
TYPE {необходимые для графического представления}  
VAR {результатов расчетов}  
    { Определение процедур и функций построения графиков}  
LABEL {Метки, константы и переменные, }  
CONST {используемые только в основном}  
VAR {блоке программы}  
BEGIN  
    {Основной блок программы}  
END.
```

Все что используется в программе или подпрограмме должно быть определено выше.