

Основные понятия языка Паскаль

1. Алфавит языка.
2. Элементарные конструкции:
 - Идентификаторы.
 - Константы.
3. Типы данных.

Алфавит языка

$\langle \text{алфавит} \rangle ::= \langle \text{буквы} \rangle \mid \langle \text{цифры} \rangle \mid$
 $\langle \text{ограничители} \rangle$

$\langle \text{буквы} \rangle ::= \mathbf{A \mid B \mid \dots \mid Z \mid a \mid b \mid \dots \mid z \mid}$
 $\langle \text{знак подчеркивания} \rangle$

$\langle \text{цифры} \rangle ::= \mathbf{0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9}$

$\langle \text{знак подчеркивания} \rangle ::= \mathbf{_}$

Ограничители

<ограничители> ::= <знаки операций> | <скобки> |
<зарезервированные слова> | <разделители>

<знаки операций> ::= <арифметические> |
<отношения> | <логические> | <над строками> |
<над множествами>

<знаки арифметических операций> ::=

+ | - | * | / | div | mod

<знаки операций отношения> ::= **= | <> | > | < | >= | <=**

Ограничители

< знаки логических операций> ::= **not | and | or | xor**

<знаки операций над строками> ::= **+** |

<знаки операций отношения>

<знаки операций над множествами> ::=

*** | + | - | = | <> | <= | >= | in**

Скобки

<скобки> ::= <для индексов> |
<для выражений и функций> | <операторные> |
<для строк>

<скобки для индексов> ::= [] | (. .)

<скобки для выражений и функций> ::= ()

<скобки операторные> ::= **begin end**

<скобки для строк> ::= ' '

Зарезервированные слова

**<зарезервированные слова> ::= and | asm | array |
begin | case | const | constructor | destructor | div |
do | downto | else | end | exports | file | for | function
| goto | if | implementation | in | inherited | inline
| interface | label | library | mod | nil | not | object | of
| or | packed | procedure | program | record | repeat
| set | shl | shr | string | then | to | type | unit | until
| uses | var | while | with | xor**

Разделители

$\langle \text{разделители} \rangle ::=$

$:= | \cdot | , | : | ; | ^ | \$ | \#$

Комментарий

{ это комментарий }

(* и это тоже комментарий*)

Идентификатор

Идентификатор – последовательность букв и цифр, а также знака подчеркивания, начинающаяся с буквы.

Max, min_stroka, _sum, kol, vector, matr

Константы

Константы – это данные, значения которых не изменяются в процессе работы программы. Константы могут быть поименованными (с типом и без типа) и явно заданными.

12345

123.45

1.2345E+4

Целые константы

Shortint	-128..127	1 байт	Со знаком
Byte	0..255	1 байт	Без знака
Integer	-32768..32767	2 байта	Со знаком
Word	0..65535	2 байта	Без знака
Longint	-2 147 483 648 ..2 147 483 647	4 байта	Со знаком

Целые константы

567 -23456 \$A1F3

```
Const    a=45;    b=-2349;  
         c:integer= 0; d:word= 4;  
         f:longint= 1;
```

```
MaxInt= 32767    MaxLongInt=2147483647
```

Вещественные константы

Single	$1.5E-45..3.4E38$	4 байта	7–8 значащих цифр, со знаком
Real	$2.9E-39..1.7E38$	6 байт	11–12 значащих цифр, со знаком
Double	$5.0E-324 ..1.7E308$	8 байт	15–16 значащих цифр, со знаком
Extended	$3.4E-4932 ..1.1E4932$	10 байт	19–20 значащих цифр, со знаком
Comp	$-9.2E18..9.2E18$	8 байт	19–20 значащих цифр, со знаком

Вещественные константы

`<мантисса>E{±}<порядок>`

`123.7`

`1.237E+2`

`1237E-1`

`Const R:real=-13.888;`

Логические константы

True(истина) или False(ложь)

Boolean (1 байт)

Bytebool(1 байт)

Wordbool(2 байта)

Longbool(4 байта)

```
Const  z:boolean=true;  
        y=false;
```

Символьные константы

Char представляет собой 1 символ кодового набора ASCII (американский стандартный код обмена информацией)

'A' < 'B' < ... < 'Z'

'a' < 'b' < ... < 'z'

```
Const    a='z' ;    b:char='5' ;
```

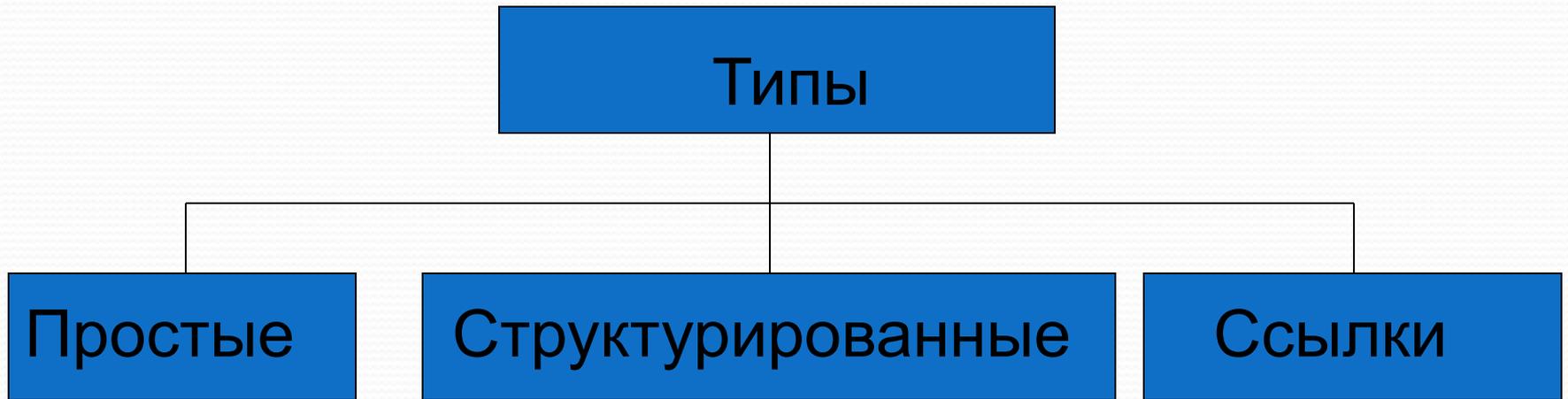
Строковые константы

```
Const    a='строка';  
         b='Журнал' 'Новый мир' ' ';  
         c:string[5]='ABC+D';
```

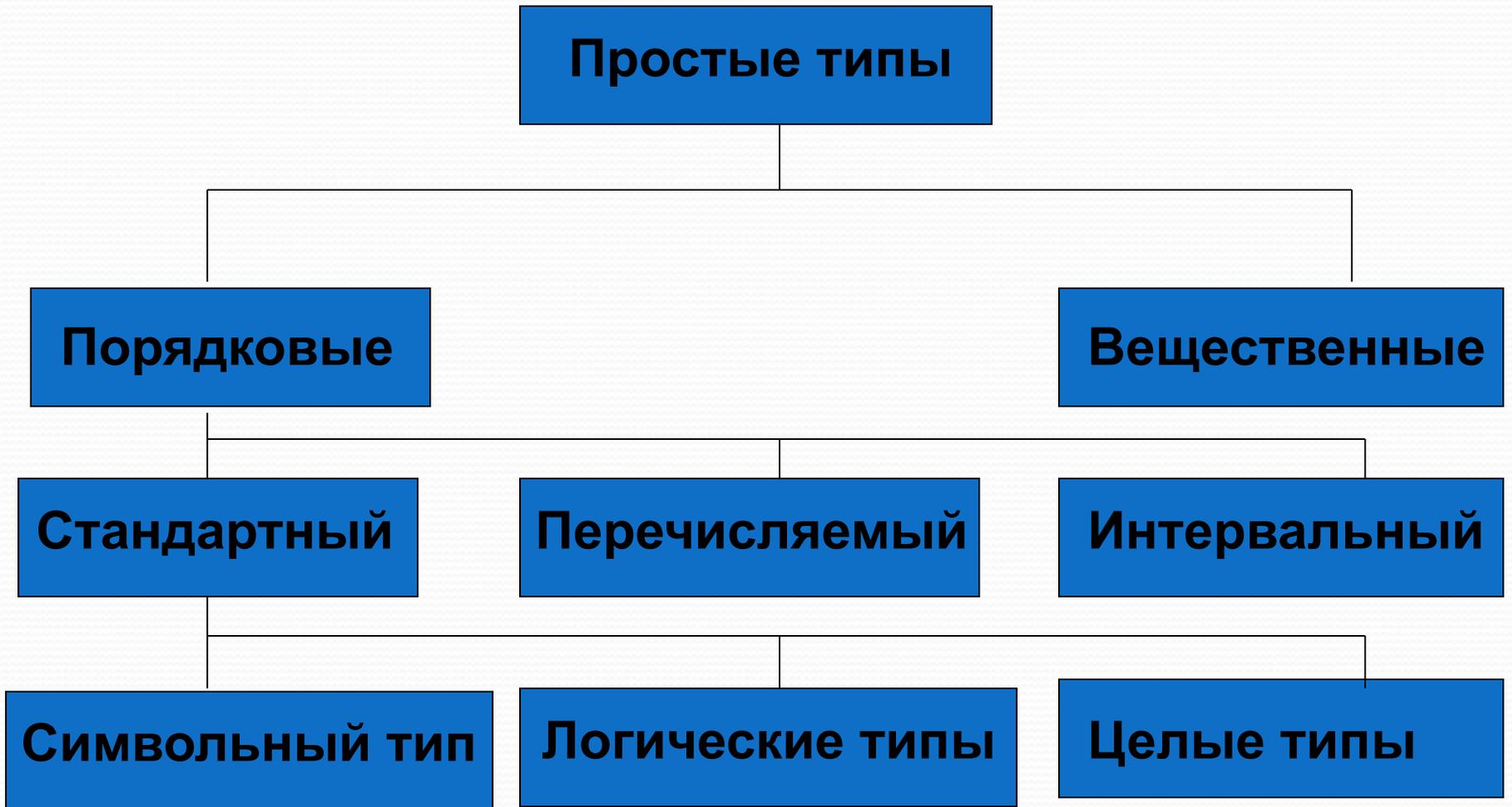
Переменная

Переменная – это наименование, данное некоторому значению. Обозначается с помощью идентификатора переменной.

Основные типы данных



Простые типы



Структурированные типы



Раздел описания переменных

VAR <идентификатор> [, <идентификатор>, ...]: <тип>;
 [<идентификатор> [, <идентификатор>, ...]:
 <тип>; ...]

```
Var A, K, M: integer;  
    B: real;  
    F: boolean;
```

Раздел описания типов

TYPE <идентификатор типа> = <тип> ;

```
Type    I = integer; L = Boolean;
```

```
        R = real;
```

```
Var    A, K, M: I; F: L;
```

```
        B: R;
```

Перечисляемый тип

```
TYPE <идентификатор типа> =  
    (<идентификатор> [,<идентификатор>,...]);
```

```
Type GR= (DK101 ,DK102 ,DK103 ,DK104) ;
```

```
Var GRUP:GR;
```

Интервальный тип

TYPE <идентификатор типа> =
 <константа>..<константа>;

```
TYPE   GR = (D101, D102, D201, D202,  
             D301, D302);
```

```
    SPEC = D201..D302;
```

```
    DIGIT = 0..9;
```

```
VAR   A:DIGIT; B:SPEC; D:1..999;
```

Строчный тип

```
TYPE <идентификатор типа> = String  
      [<максимальная длина>];
```

```
TYPE   TString = String[50];
```

```
      TS = String;
```

```
      VAR   S, S1 : TString;
```

```
          S2 : String[20];
```

```
          SS : TS;
```

Строчный тип

К каждому символу строки возможен доступ по его номеру.

```
Var s:string;
```

```
Begin
```

```
    readln(s); writeln(s[5])
```

```
End.
```

Массив

Массив – это упорядоченная совокупность однотипных переменных, обладающих одинаковыми свойствами.

TYPE <идентификатор типа> = **ARRAY**
[<список типов индексов>] **OF** <тип>;

<тип индексов> ::= <простой тип>

<простой тип> ::= <идентификатор типа> |
<идентификатор> [,<идентификатор>] |
<константа> . . <константа>

Описание массивов

```
TYPE T1 = ARRAY [-10..20, 1..30] OF BYTE;  
      T2 = ARRAY [0..50] OF BOOLEAN;  
      T3 = ARRAY [BYTE] OF INTEGER;  
VAR A, B: T1;  
      C: T2;  
      Z: ARRAY [1..100] OF REAL;  
      MAS: T3;
```

Обращение к элементам массивов

...

```
S := S + Z[I];
```

```
P := P * A[I][J];
```

```
C[6] := TRUE;
```

```
P := P * A[I, J];
```

```
R := B [I+5, J];
```

```
MAS[I] := MAS[I-1] * MAS[I];
```

...

Описание матрицы

```
1. CONST  N =10; M =50;
```

```
TYPE TMATR = ARRAY [1..N,1..M] OF  
                                                    INTEGER;
```

```
VAR A:TMATR;
```

```
2. TYPE TSTR = ARRAY [1..50] OF INTEGER;
```

```
    TMATR = ARRAY [1..10] OF TSTR;
```

```
VAR A:TMATR;
```

Описание матрицы

```
3. VAR A: ARRAY [1..10, 1..50] OF  
                                           INTEGER;
```

```
4. VAR A: ARRAY [1..10] OF  
                ARRAY [1..50] OF INTEGER;
```