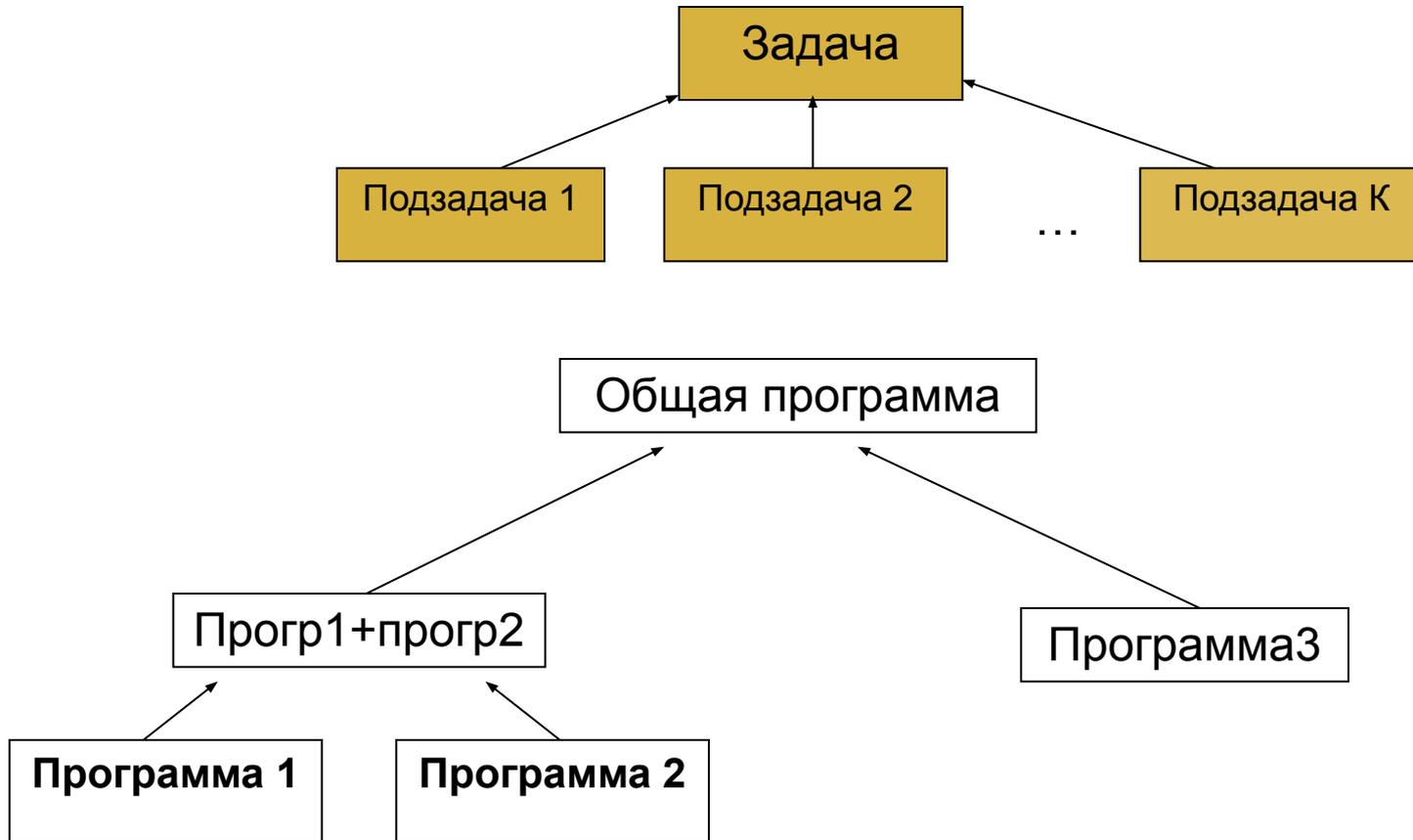


# Структуризация

(расчленение алгоритма на части)

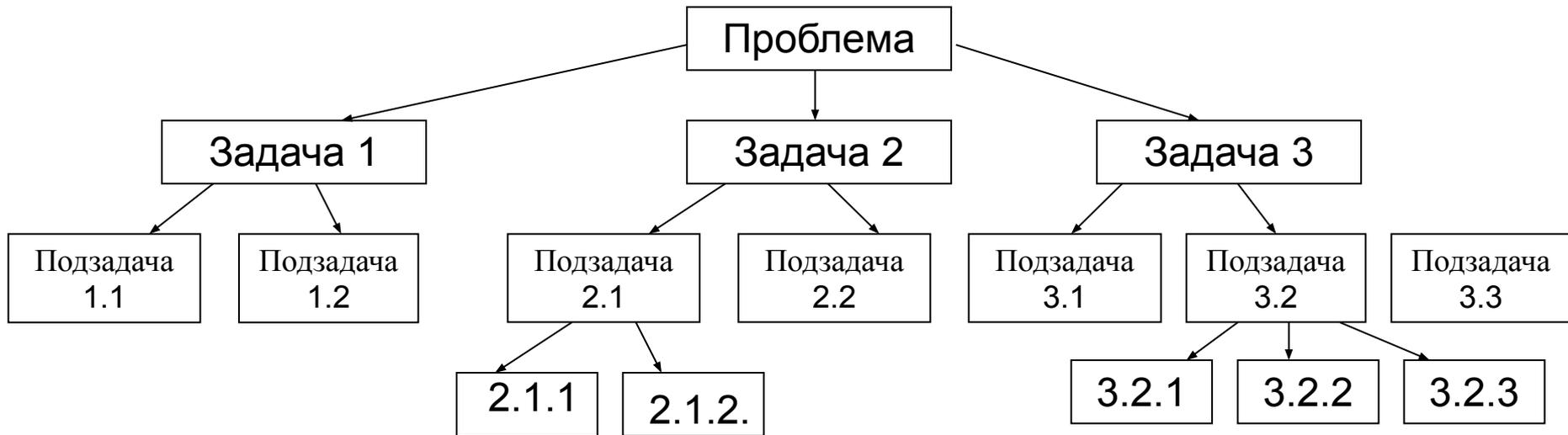
Восходящее проектирование алгоритмов и программ («снизу – вверх»)



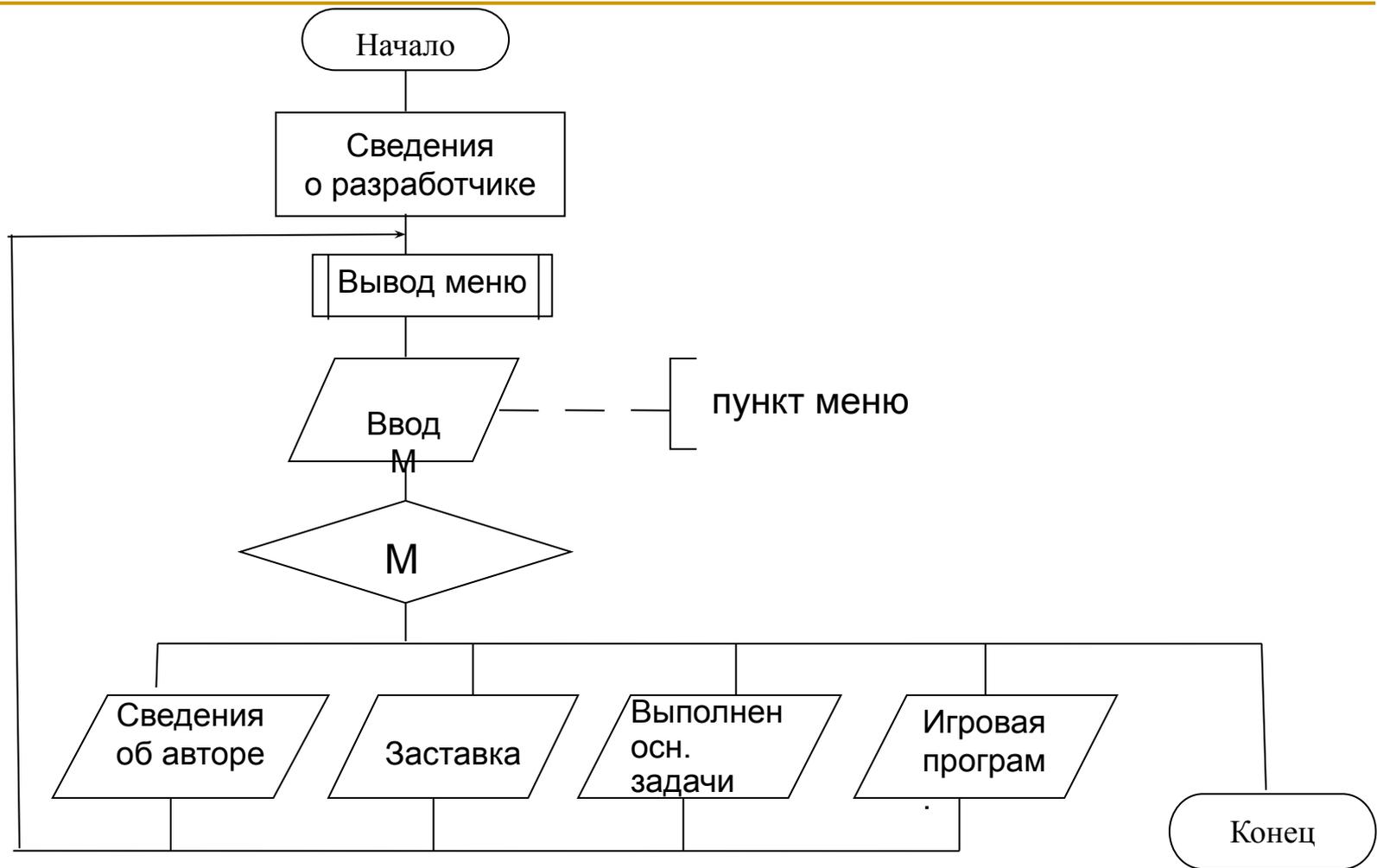
Примером проектирования методом «снизу-вверх» является проектирование программы для РГР



## Нисходящее программирование («сверху – вниз»)



Методы проектирования программ // Шафеева О.П. 2004. -32с.  
(Метод. указания к курсовой работе и РГР)



Укрупненная схема алгоритма для курсовой работы

## Типы в ТурбоПаскале

### TYPE

<НовыйТип1>=<МассивЦелыхЧисел>; {array[1..5] of byte;}

<НовыйТип2>=<Множество символов>; {set of 0..9;}

<НовыйТип3>=<запись>; {record}

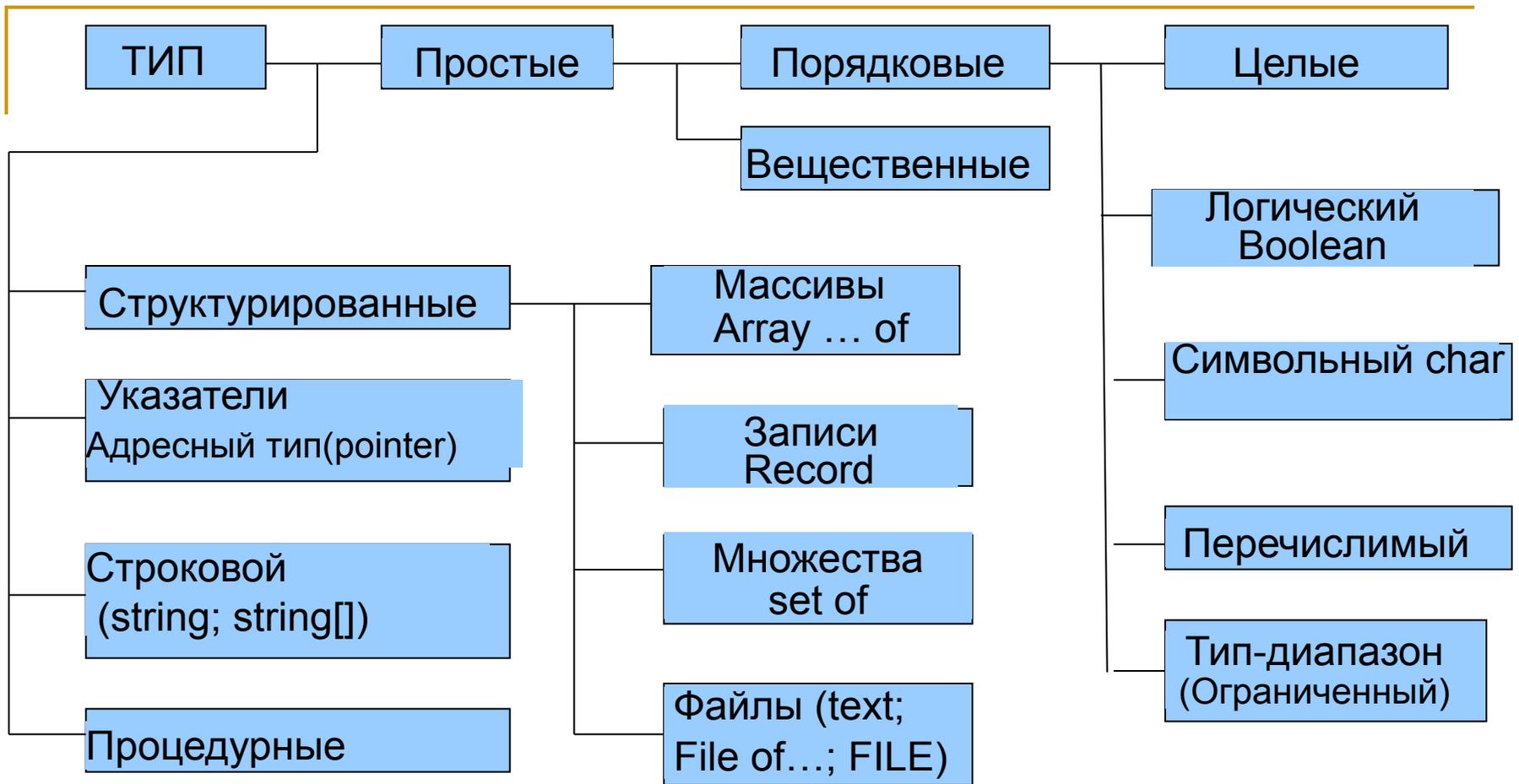
...

<НовыйТип8>=<другой тип>;

<НовыйТип9>=(<перечисленные здесь значения>); -перечислимый тип.

<идентификатор типа>=<определяющие тип зарезервированные слова или значения элементов>

-



## Классификация типов

-

## Порядковые типы

**ORD(X)** - возвращает порядковый номер значения выражения x;

Для целых типов  $ORD(X)=X$  (возвращает само значение X).

Для логического типа возвращает целое число 0 или 1.

Для символьного типа возвращает число в диапазоне 0..255(код символа)

Для перечислимого типа возвращает число в диапазоне 0..65353.

Тип-диапазон сохраняет свойства базового типа и  $ORD(X)$  зависит от него

**PRED(X)**             $ORD(PRED(X))=ORD(X)-1$ ;

**SUCC(X)**             $ORD(SUCC(X))=ORD(X)+1$ ;

## ЦЕЛЫЕ ТИПЫ

Длина (байт)	Название	Диапазон	Примечание
1	BYTE (B)	от 0 до 255	Короткое целое без знака
1	SHORTINT (S)	от -128 до 127	Короткое целое со знаком
2	WORD (W)	от 0 до 65535	Целое без знака
2	INTEGER (I)	-32768 .. 32767	Целое со знаком
4	LONGINT (L)	от -2147483648 до 2147483647	Длинное целое со знаком

*Пример:* VAR X:INTEGER;  
I,J:BYTE;

-MAXINT<= X<= MAXINT.

PROGRAM MAX;

Begin

writeln (MAXINT); {32 767}

readln; {Задержка до нажатия клавиши <ENTER>}

END.

### Операции и функции

1) сложение, вычитание (имеют более низкий приоритет);

DIV, MOD, умножение;

2) ABS(X), SQR(X); TRUNC(X), ROUND(X) дают целый результат;

3) деление и функция извлечения квадратного корня дает вещественный результат;

## Функции и процедуры для целых типов

Обращение	Тип результата	Действие
CHR(B) HI(I),HI(W) LO(I),LO(W) ODD(L)	CHAR BYTE BYTE BOOLEAN	Возвращает символ по его коду Возвращает старший байт аргумента. Возвращает младший байт аргумента. Возвращает значение (TRUE), если аргумент нечетное число; (FALSE) - если четное.
SWAP(I) SWAP(W) RANDOM(W)	INTEGER WORD Как у параметра	} <b>меняет местами байты в</b> } <b>слове</b> Возвращает псевдослучайные числа равномерно распределенные на интервале $0 \leq X \leq W$
<b>DEC(Vx[,i])</b> <b>INC(Vx[,i])</b>	Процедура Процедура	Уменьшает Увеличивает <span style="float: right; border-left: 1px solid black; padding-left: 10px;">                         значение Vx на i,                          при отсутствии i -                          на 1                     </span>

---

WORD>BYTE

LONGINT>INTEGER>SHORTINT

Trunc(x)      Round(x)

\$FF \$7 \$FFFFAB0C -\$80(со знаком) – ЦЕЛЫЕ в 16-ричном формате

Word, Integer, LongInt - в перевернутом виде

W=\$0102 ⇒ \$02 и \$01

-

## Упорядочение массивов

а) сортировка выбором:

```
for k:=n downto 2 do
```

```
  begin {поиск m - номера Max X[1...k];}
```

```
    m:=1;
```

```
    for i:=2 to k do if x [i]>x [m] then m:=i;
```

```
      {перестановка элементов x [m] и x [k];}
```

```
      r:= x[k];    x[k]:= x[m];    x[m]:=r
```

```
  end;
```

б) сортировка обменом

$x_k$  и  $x_{k+1}$  ( $k=1,2,\dots,n-1$ )

$x_k > x_{k+1}$

Ефимов С.С. Сортировка и поиск. м.у. 2001г.

## ЛОГИЧЕСКИЙ ТИП (BOOLEAN)

FALSE(ложь), TRUE(истина)

FALSE<TRUE,

Ord(false)=0; Ord(true)=1;

succ(false)=true, pred(true)=false.

NOT, AND, OR, XOR (<>,<=,>=,<,>=),

### BOOLEAN

```
VAR X, Y, A, B: BOOLEAN;
```

```
    A1, B1: byte;
```

```
BEGIN
```

```
    ...
```

```
    X:=A OR B;
```

```
    Y:=(A1>0) AND (A1<B1);
```

```
END.
```

```
FOR x:=FALSE TO TRUE do <оператор>;
```

```
FOR x:=TRUE DOWNTO false do <оператор>;
```

ODD(x) логическая функция проверки нечетности

TRUE FALSE

EOLN(f) - функция проверки конца строки

END OF LINE

~~EOF(f) - определяет конец файла~~

~~END OF FILE~~

## ПЕРЕЧИСЛИМЫЙ ТИП

Boolean (2 значения с номерами 0 и 1),  
char (256 символов с кодами 0..255),  
все целые типы: Byte (256 значений от 0 до 255),  
Word (65536 значений),  
а также другие целые типы: ShortInt, Integer, LongInt (нумерация значений которых начинается с отрицательного целого числа).

*Пример:* TYPE Char=(Симв0,симв1,...,симв64,'A','B','C',...,симв255);  
Boolean=(False,True);  
Byte=(0,1,2,...,255);

```
TYPE   <имя типа1>=(<элемент1>,<элемент2>...,<элементN>);  
       <имя типаK>=(<элемент1>,...,<элементM>);  
       <имя типаJ>=<имя типаK>; ...
```

*Пример:* TYPE TEST=(test1,test2,test3,test4,test5);  
Day = (PN, VT, SR, HT, PT, SB, VS);  
BOOLEAN3=(False\_,Noinfo,True\_) ;{значения новых типов}

Максимальное количество элементов в одном вводимом перечислимом типе равно **65536** (0..65535).

*Пример:* var  
T1, T2: TEST;  
B1, B2: BOOLEAN3;

for T1:=test2 to test4 do <оператор>;  
B1:=false\_; B2:=B1;  
T1:=test1; T2:=test(0);

*Пример:* True\_ > Noinfo  
test2 > test1.

**ORD(x):** Ord(test1)=0; Ord(test2)=1;... Ord(test5)=4;

t1:=Test(0) эквив. t1:=Test1

<имя Перечисл.типа> (номер элемента)

Для типа BOOLEAN3 Pred(True\_)=Succ(False\_)=Noinfo ;  
(предыдущее) (следующее в типе значение)

Если x:=test2, то Succ(x)=test3; Pred(x)=test1.

X:=<имяПеречислимогоТипа>(ПорядковыйНомер)

~~Пример: T1:=TEST(0) и T1:=TEST1 - нулевой элемент типа TEST~~

## ВЕЩЕСТВЕННЫЕ ТИПЫ



Знак  $m \cdot E_p$ ,

где  $m$  - мантисса числа, причем  $0.5 \leq |m| < 1$  (нормализованное);  
 $p$  - порядок (экспоненциальная часть: степень  $E$ ), целое двоичное число со знаком

-

длина (байт )	название типа	мантисса (значащие цифры)	диапазон десятичного порядка
4	SINGL	7..8	-45..+38
6	REAL	11..12	-39..+38
8	DOUBLE	15..16	-324..+308
10	EXTENDED	19..20	-4951..+4932
8	COMP	19..20	-2E63+1..+2E63-1

{\$N+} - подключить сопроцессор,

{\$N- } - отключить сопроцессор. REAL

Операции +,-,\* дадут вещественный результат, если один из аргументов вещественен. Для деления результат получается вещественный, даже если оба операнда целые.

{\$N+, E+} - подключение эмулятора сопроцессора

**Встроенные стандартные функции:**  $\sin(x)$ ,  $\cos(x)$ ,  $\arctg(x)$ ,  $\ln(x)$ ,  $\text{Sqrt}(x)$ ,  $\text{Exp}(x)$  дают вещественный результат при вещ. или целом аргументе;

$\text{Abs}(x)$ ,  $\text{Sqr}(x)$  при вещ.  $x$  дают вещ. результат.

Дополнительно в ТП определены функции:

Обращение	тип параметра	тип результата	примечание
$Pi$	-	вещественный	$Pi = 3.141592653$
$\text{Frac}(x)$	вещественный	вещественный	Дробная часть числа $x$
$\text{Int}(x)$	вещественный	вещественный	<b>Целая часть</b> числа $x$
$\text{Random};$	-	Вещественный	$0 < x < 1$ случайное число
$\text{Random}(x);$	-	Целый	<b>Случайное целое число</b>
$\text{Randomize};$	-		<b>Инициализация датчика</b> псевдослучайных чисел. Гарантирует несовпадение последовательностей случ. чисел, вызываемых функцией $\text{Random}$

**Пример.** Задать массив из 16 элементов через генератор случайных чисел.  
Напечатать их по 4 элемента в строке

```
Program Mass2;
```

```
Uses CRT;
```

```
Const N=16;
```

```
VAR  A:array[1..N] of real;
```

```
    i :byte;
```

```
    c: char;
```

```
{СИМВОЛЬНЫЙ ТИП}
```

```
BEGIN
```

```
  Randomize;
```

```
  For i:=1 to N do
```

```
    begin
```

```
      a[i]:= random;
```

```
      Write(a[i]:7:3);
```

```
      if (i mod 4) = 0 then writeln;
```

```
    end;
```

```
  repeat
```

```
    c:=readkey {читает код нажатой клавиши}
```

```
  until ord(c) = 27 { пока не нажата клавиша ESC окно вывода не закроется}
```

```
END.
```

## 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 NEDEL}

Пример: **TYPE** date=1..31;

**VAR** L,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;

**VAR**date: 1..31;

    month: 1..12;

**letchr**: 'A'..'Z';

## ИЗМЕНЕНИЕ ТИПА ВЫРАЖЕНИЯ

**<имяТипа> (<выражение>);** - операция **приведения** типа

Пример. TYPE TyType=array[1..5] of char;

Var I,J, D:byte;

B:boolean;

R:real;

A:integer;

C:char; Arr:TyType;

Begin ...

A:=Integer('A'); {65}

D:=byte (R);

C:=Char(I+J-D);

R:=real (D);

B:=Boolean(100); {True}

Arr:= TyType('TP\_60'); где TyType=array[1..5] of char;

b:=3.5

A:=integer(b+4.8); {8}

A:=integer (b) + integer (4.8) {7}