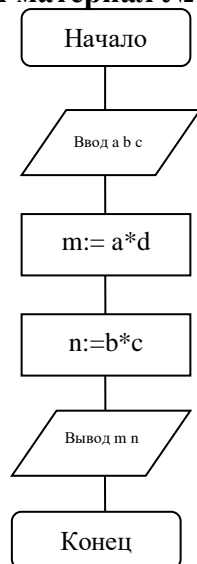
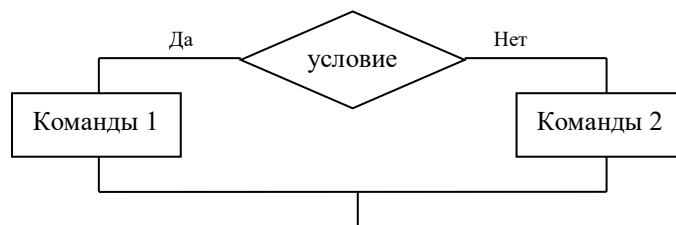


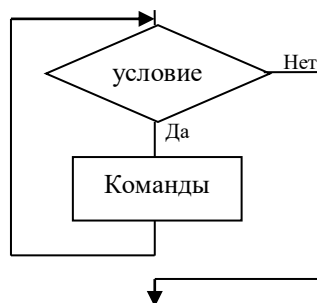
Раздаточный материал №1



Раздаточный материал № 2



Раздаточный материал № 3



Раздаточный материал № 4

Program <имя программы>;
Uses <Раздел подключения внешних модулей>
label <раздел меток>;
Const <раздел констант>;
Type <раздел типов>;
Var <раздел переменных>;
Procedure (Function) <раздел подпрограмм>;
Begin
<Раздел операторов>
End.

Раздаточный материал № 5

// комментарий - все до конца строки;
{ комментарий - все,
пока не встретится }
Фигурные скобки { ... } можно заменить на (* ... *).

Раздаточный материал № 6



Раздаточный материал № 7

Длина, байт	Без знака	Со знаком
1	byte 0 .. 255	shortint -128 .. 127
2	word 0 .. 65 535	smallint -32 768 .. 32 767
4	longword, cardinal 0 .. 4 294 967 295	integer, longint -2 147 483 648 .. 2 147 483 647
8	uint64 0 .. 18 446 744 073 709 551 615	int64 -9 223 372 036 854 775 808 .. 9 223 372 036 854 775 807
переменная		BigInteger Ограничено только памятью компьютера

Раздаточный материал № 8

Длина, байт	Тип и количество значащих цифр	Диапазон представляемых значений
4	single 7-8	$-1.8 \times 10^{308} \dots 1.8 \times 10^{308}$
8	real (double) 15-16	$-3.4 \times 10^{38} \dots 3.4 \times 10^{38}$
16	Decimal 28-29	-79228162514264337593543950335 .. 79228162514264337593543950335

Формы представления вещественного типа	1.47, -235.02, -0.0004, -5.0 – в форме с фиксированной точкой; 1e3, 5.4E-2 – в экспоненциальной форме
---	--

Раздаточный материал № 9

список имен переменных: тип;

Раздаточный материал № 10

var
a, b, c: integer;
d: real;

Раздаточный материал № 11

имя константы = значение;

Например,
const
Pi = 3.14;
Count = 10;
Name = 'Vova';

Раздаточный материал № 12

```
var a, b, c: integer;  
const pi = 3.14;
```

Раздаточный материал № 13

```
var n: word := 18; // описание типа, совмещенное с инициализацией  
var s := 0; // описание с автовыведением типа (integer)  
var MyBytes := $C7; // шестнадцатичное значение с типом integer
```

Раздаточный материал № 14

```
var s: integer := 35+457;
```

Раздаточный материал № 15

```
var (Имя1, Имя2, ... ИмяN) := (Значение1, Значение2, ... ЗначениеN);
```

Здесь переменная Имя1 получает Значение1 (и соответствующий тип), Имя2 получает Значение2 и т.д.

Конструкция, записанная в круглых скобках, называется кортежем.

```
var (a, b, c, i) := (132, -58, 0, 1);
```

Раздаточный материал № 16

```
var (a, b, c, i) := (132, -58, int64(0), byte(1));
```

Раздаточный материал № 17

Оператор	Описание
A += b	Переменную A увеличить на b
A*=b	Переменную A увеличить в b раз
B -= 5	Переменную B уменьшить на 5

Раздаточный материал № 18

Знак	Выражение	Типы операндов	Тип результата	Операция
+	A+B	R, R I, I I, R; R, I	R I R	Сложение
-	A-B	R, R I, I I, R; R, I	R I R	Вычитание
*	A* B	R, R I, I I, R; R, I	R I R	Умножение
/	A/B	R, R I, I I, R; R, I	R R R	Вещественное деление
div	A div B	I, I	I	Целое деление
mod	A mod B	I, I	I	Остаток от целого деления

где I — данные целого типа, а R — вещественного.

Раздаточный материал № 19

1. Все символы писать в строку, т.е. на одном уровне. Проставлять все знаки операций, не пропуская знак «*».
2. Не допускать записи двух знаков операций подряд, т. е. нельзя писать $A + -B$, следует писать $A + (-B)$.
3. Операции с более высоким приоритетом выполняют раньше операций с меньшим приоритетом. Порядок убывания приоритетов операций следующий:
 - вычисление функции;
 - унарная операция смены знака (-);
 - *, /, div, mod;
 - +, - .
4. Несколько записанных подряд операций с одинаковым приоритетом выполняют последовательно слева направо.
5. Часть выражения, заключенная в скобки, вычисляется в первую очередь. (Например, в выражении $(A + B) * (C - D)$ умножение производится после сложения и вычитания.)
6. Не следует записывать выражения, не имеющие математического смысла, например: деление на нуль, логарифм отрицательного числа и т.п.
7. Для вычисления x^y необходимо следующим образом:
 - если y — целое значение, следует использовать умножение, например: $x^y \rightarrow x * x * \dots * x$. Для больших степеней следует использовать умножение в цикле;
 - если y — вещественное значение, используется следующая математическая формула: $x^y = e^{y \ln(x)}$, запись которой на Паскале имеет вид $\exp(Y * \ln(x))$

Раздаточный материал № 20 (Справочный материал)

Обращение	Тип аргумента	Тип результата	Функция
Pi	—	R	Число $\pi = 3.1415926536E+00$
abs (x)	I, R	I, R	Модуль аргумента x
arctan (x)	I, R	R	Арктангенс x, рад
cos (x)	I, R	R	Косинус x, рад
exp (x)	I, R	R	e^x — экспонента x
frac (x)	I, R	R	Дробная часть x
int (x)	I, R	R	Целая часть x
ln(x)	I, R	R	Натуральный логарифм x
random		R	Псевдослучайное число в интервале [0, 1)
random (x)	I	I	Псевдослучайное число в интервале [0, x)
round (x)	R	I	Округление x до ближайшего целого
sin (x)	I, R	R	Синус x, рад
sqr (x)	I, R	I, R	Квадрат x
sqrt (x)	I, R	R	Корень квадратный из x
trunc (x)	R	I	Ближайшее целое, не превышающее x по модулю

Раздаточный материал № 21

read (список переменных)
или
readln (список переменных)

Раздаточный материал № 22

read (P,Q)

вводим

13 6 [Enter]

readln (P,Q)

вводим

13 [Enter]

6 [Enter]

Переменная P примет значение 13, а Q – значение 6.

Раздаточный материал № 23

var ИмяПеременной := ReadInteger('Текст приглашения ко вводу'); // аналогично ReadReal

или

var ИмяПеременной := ReadInteger; // аналогично ReadReal

var n := ReadInteger('Введите размерность массива');

Раздаточный материал № 24

var (имя1, имя2) := ReadInteger2('Текст приглашения ко вводу'); // аналогично ReadReal

var (имя1, имя2, имя3) := ReadInteger3('Текст приглашения ко вводу'); // аналогично ReadReal

var (имя1, имя2, имя3, имя4) := ReadInteger4('Текст приглашения ко вводу'); // аналогично ReadReal

Раздаточный материал № 25

write (список выражений)

или

writeln (список выражений)

Раздаточный материал № 26

Program Summa;

Var

P,Q: integer;

Begin

read (P,Q);

write ('Первое число=',P,', Второе число=',Q,', Сумма чисел=', P+Q)

End.

На экране

Первое число=13, Второе число=6, Сумма чисел=19

Program Summa1;

Var

P,Q: integer;

Begin

read (P,Q);

writeln ('Первое число=',P,', Второе число=',Q);

writeln ('Сумма чисел=',P+Q)

End.

На экране

Первое число=13, Второе число=6

Сумма чисел=19

Раздаточный материал № 27

Println ('Выводит текст на экран и осуществляет переход к следующей строке');

Print ('Выводит текст на экран и не переходит к следующей строке');

Раздаточный материал № 28

Write (I:6) – в поле шириной 6 символов выводится значение I в крайние правые позиции (I=186)

На экране _ _ _ 186

Write (R:8:4) – в крайние правые позиции поля шириной 8 символов выводится R в формате с фиксированной точкой (R=511,04)

На экране 511.0400

Раздаточный материал № 29

Вариант 1	Вариант 2
<pre>//программа деления двух обыкновенных дробей Program Division; Var a,b,c,d,m,n : Integer; Begin ReadLn (a, b, c, d); m := a * d; n := b * c; WriteLn(m, n) End.</pre>	<pre>//программа деления двух обыкновенных дробей Program Деление_Дробей; Var (a,b,c,d):= ReadInteger4('Введите четыре целых числа'); m,n : Integer; Begin m := a * d; n := b * c; Write('Результат = '); WriteLn(m, '/', n) End.</pre>
<pre>//программа деления двух обыкновенных дробей Program Деление_Дробей; Var (a,b,c,d):= ReadInteger4('Введите четыре целых числа'); m,n : Integer; Begin m := a * d; n := b * c; Print('Результат = '); PrintLn(m, '/', n) End.</pre>	<pre>//программа деления двух обыкновенных дробей Program Деление_Дробей; Var (a,b,c,d):= ReadInteger4('Введите четыре целых числа'); Begin Write('Результат = ', a * d, '/', b * c); End.</pre>

Для анализа ожидаемых результатов подготовим контрольный пример (тестовый набор).

Пусть a = 2, b = 5, c = 6, d = 4.

Тогда программа должна выдать результат 8 / 30.

Раздаточный материал № 30

Шаг	a	b	c	d	m	n
1.	2	5	6	4		
2.					$m := 2 * 4$	
3.						$n := 5 * 6$
4.					Вывод m, n (8 / 30)	

Задание 2. Составить программу нахождения суммы первой и третьей цифр трехзначного числа, провести трассировку программы и выполнить анализ полученных результатов.

Раздаточный материал № 31

```
{программа нахождения суммы первой и
третьей цифр трехзначного числа}
Program Сумма_Чисел;
Var
a:= ReadInteger('Введите целое трехзначное число');
Begin
Print('Результат = ', a div 100 + a mod 10);
End.
```

Для анализа ожидаемых результатов подготовим контрольный пример (тестовый набор).

Пусть исходное число = 567

Программа должна выдать результат: $5 + 7 = 12$.

Раздаточный материал № 32

Шаг	a					
1.	567					
2.		$a \text{ div } 100 = 5$				
3.			$a \text{ mod } 10 = 7$			
4.				$5 + 7 = 12$		
5.					Вывод = 12	

Раздаточный материал № 33

1. Дано трехзначное число. Вывести число, полученное при прочтении исходного числа справа налево.
2. Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее справа. Вывести полученное число.
3. Дано трехзначное число. В нем зачеркнули первую справа цифру и приписали ее слева. Вывести полученное число.
4. Дано целое число, большее 999. Используя одну операцию деления нацело и одну операцию взятия остатка от деления, найти цифру, соответствующую разряду сотен в записи этого числа.
5. Даны целые положительные числа A, B, C. На прямоугольнике размера A x B размещено максимально возможное количество квадратов со стороной C (без наложений). Найти количество квадратов, размещенных на прямоугольнике, а также площадь незанятой части прямоугольника.

Раздаточный материал № 34

Символ	Операция
=	Равно
<>	Не равно
<	Меньше
>	Больше
>=	Больше или равно
<=	Меньше или равно

Например,
 $x < y$; $a + b \geq c/d$; $\text{abs}(m - n) \leq 1$.

Примеры вычисления значений отношений:

Отношение Результат

$12 \geq 12$ True

$56 > 10$ True

$11 \leq 6$ False

Раздаточный материал № 35

and («и») - логическое умножение,
or («или») - логическое сложение,
not («не») - логическое отрицание.

Раздаточный материал № 36

Таблица истинности операций and и or

A	B	A and B	A or B
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Таблица истинности операции not

A	not A
True	False
False	True

Логическое выражение	Описание
$(x \geq 5) \text{ and } (x \leq 10)$	Истинно когда x принимает значения от 5 до 10
$(x = 3) \text{ or } (x = 5) \text{ or } (x = 7)$	Истинно когда x принимает одно из трех значений: 3, 5 или 7
$\text{Not } (x = 5)$	Истинно когда $x \neq 5$
$(x = 3) \text{ and } (x = 5)$	Всегда ложно

Раздаточный материал № 37

Таблица приоритетов операций

1.	not
2.	*, /, div, mod, and
3.	+, -, or
4.	=, <>, <, >, <=, >=

Раздаточный материал № 38

$(x \geq 10) \text{ and } (x < 100) \text{ and Odd}(x)$

Раздаточный материал № 39

Вычислить значения следующих логических выражений:

а) $K \bmod 7 = K \text{ div } 5 - 1$ при $K = 15$;

б) $\text{Odd}(\text{Trunc}(10 * P))$ при $P = 0.182$;

Написать оператор присваивания, в результате выполнения которого логическая переменная t получит значение True, если следующее утверждение истинно, и значение False — в противном случае:

а) из чисел x, y, z только два равны между собой;

б) x — положительное число;

Раздаточный материал № 40

1. Вычислить значения следующих логических выражений:

- а) $K \bmod 7 = K \operatorname{div} 5 - 1$ при $K = 15$;
- б) $\text{Odd}(\text{Trunc}(10 * P))$ при $P = 0.182$;
- в) $\text{Not Odd}(n)$ при $n = 0$;
- г) $t \text{ And } (P \bmod 3 = 0)$ при $t = \text{True}$, $P = 10101$;
- д) $(x * y <> 0) \text{ And } (y > x)$ при $x = 2$, $y = 1$;
- е) $a \text{ Or Not } b$ при $a = \text{False}$, $b = \text{True}$.

2. Написать оператор присваивания, в результате выполнения которого логическая переменная t получит значение True , если следующее утверждение истинно, и значение False — в противном случае:

- а) из чисел x , y , z только два равны между собой;
- б) x — положительное число;
- в) каждое из чисел x , y , z положительное;
- г) только одно из чисел x , y , z положительное;
- д) p делится без остатка на q ;
- е) цифра 5 входит в десятичную запись трехзначного целого числа k .

Раздаточный материал № 41

If <Условие> Then <Оператор 1> Else <Оператор 2>;

Кроме того, возможно использование неполной формы условного оператора:

If <Условие> Then <Оператор>;

Раздаточный материал № 42

```
Program Proizvedenie;
var
p, q: integer;
Pr: real;
begin
  writeln('Введите первое число:');
  readln(p);
  writeln('Введите второе число:');
  readln(q);
  Pr:=p*q;
  if (Pr<0) then Pr:=Pr*8 else Pr:=Pr*1.5;
  write('Полученное значение: ', Pr);
end.
```

Раздаточный материал № 43

Правила оформления текстов программ

№ 1. На одной строке размещается один оператор.

№ 2. Все операторы, входящие во вложенный оператор, должны быть сдвинуты вправо (клавиша TAB).

№ 3. Операторные скобки, относящиеся к одному блоку, должны располагаться следующим образом. Открывающая скобка должна находиться на той же строчке, что и следующий до блока оператор, а закрывающая должна находиться строго на одной линии по вертикали с началом оператора, предшествующему данному составному.

№ 4. В конце строки пробелы не ставятся. Пробелы полезны вокруг знака какой-либо операции.

№ 5. Обязательны пустые строки перед и после определения функций. Логические части блоков тоже нужно отделять друг от друга пустой строчкой.

№ 6. Тип переменной следует писать для каждой из них. На одной строчке — одна переменная.

№ 7. Идентификаторы (программ, функций, переменных, массивов и т.п.) должны нести смысловую нагрузку.

Раздаточный материал № 44

1. Ввести 2 числа. Если их произведение отрицательно, умножить его на 8, в противном случае увеличить его в 1.5 раза.
2. Ввести число. Если оно четное, разделить его на 4, если нечетное - умножить на 5.
3. Расчет площади прямоугольника. Если площадь треугольника чётная, разделить её на 2, в противном случае вывести сообщение «Не могу делить на 2!»
4. Ввести двухзначное число. Если сумма цифр числа четная, то увеличить число на 2, в противном случае уменьшить на 2.
5. Дано целое число. Если оно является положительным, то прибавить к нему 20, в противном случае вычесть из него 5.
6. Дано два числа. Если их сумма кратна 5, то прибавить 1, иначе вычесть 2.

Раздаточный материал № 45

Составить программы на языке PascalABC.NET с учетом правил составления программ.

1. Даны три целых числа A, B, C. Проверить истинность высказывания: «Каждое из чисел A, B, C положительное».
2. Даны три целых числа A, B, C. Проверить истинность высказывания: «Хотя бы одно из чисел A, B, C положительное».
3. Даны три целых числа A, B, C. Проверить истинность высказывания: «Ровно одно из чисел A, B, C положительное».
4. Даны три целых числа A, B, C. Проверить истинность высказывания: «Ровно два из чисел A, B, C являются положительными».
5. Дано целое положительное число. Проверить истинность высказывания: «Данное число является четным двузначным».
6. Дано целое положительное число. Проверить истинность высказывания: «Данное число является нечетным трехзначным».
7. Дано трехзначное число. Проверить истинность высказывания: «Все цифры данного числа различны».
8. Дано трехзначное число. Проверить истинность высказывания: «Цифры данного числа образуют возрастающую последовательность».
9. Даны числа x, y. Проверить истинность высказывания: «Точка с координатами (x, y) лежит во второй координатной четверти».
10. Даны числа x, y. Проверить истинность высказывания: «Точка с координатами (x, y) лежит в четвертой координатной четверти».
11. Даны координаты поля шахматной доски x, y (целые числа, лежащие в диапазоне 1-8). Учитывая, что левое нижнее поле доски (1,1) является черным, проверить истинность высказывания: «Данное поле является белым».

Раздаточный материал № 46

P ? A : B

Раздаточный материал № 47

Вариант 1

{Дано целое число. Если оно является положительным, то прибавить к нему 20, в противном случае вычесть из него 5. Результат сохраняется в переменной c}

```
program УсловныйОператор;  
var  
c := ReadInteger('Введите число');  
begin  
if c > 0 then c+=20 else c-=5;  
print('Результат = ', c);  
end.
```

Вариант 2

{Дано целое число. Если оно является положительным, то прибавить к нему 20, в противном случае вычесть из него 5. Результат сохраняется в переменной f}

```
program УсловнаяОперация;  
var  
c := ReadInteger('Введите число');  
f := c > 0 ? c+20 : c-5;  
begin  
print('Результат = ', f);  
end.
```

Вариант 3

{Дано целое число. Если оно является положительным, то прибавить к нему 20, в противном случае вычесть из него 5/
Результат не сохраняется}

```
program УсловнаяОперация;  
var  
c := ReadInteger('Введите число');  
begin  
print('Результат = ', c > 0 ? c+20 : c-5);  
end.
```

Раздаточный материал № 48

Нахождение максимума из трех чисел

```
var max := a > b ? a : b;  
max := c > max ? c : max;
```

Раздаточный материал № 49

type

ДвухзначноеЦелое = 10..99;

ДниНедели = (Пн, Вт, Ср, Чт, Пт, Сб, Вс);

ВыходныеДни = Сб..Вс;

МалыеЛатинскиеБуквы = 'a'..'z';

Numbers = 1 .. 31;

Alf = 'A' .. 'Z';

Var

Data: Numbers;

Bukva: Alf;

Раздаточный материал № 50

```
case селектор of
    список выбора 1: оператор 1;
    ...
    список выбора N: оператор N;
else оператор 0;
end
```

Например, программа перевода пятибалльной оценки в ее наименование будет выглядеть следующим образом:

```
Program Ozenka;
var
k: integer;
begin
writeln ('Введите оценку');
readln (k);
case k of
    1: writeln ('Это единица! Ужас!');
    2: writeln ('Это неудовлетворительно');
    3: writeln ('Это удовлетворительно');
    4: writeln ('Это хорошо');
    5: writeln ('Это отлично');
else writeln ('Нет такой оценки');
end;
end.
```

Раздаточный материал № 51

{По номеру дня недели определить, является день будним или выходным}

```
var
a := ReadInteger('Ввести целое число');
begin
    case a of
        1..5: Print ('Будний день');
        6,7: Print ('Выходной день');
        else Print ('Введено некорректное число');
    end;
end.
```

Раздаточный материал № 52

```
Type    Gaz = (C, O, N, F);
         Metal = (Fe, Co, Na, Cu, Zn);
Var      G1, G2, G3: Gaz;
         Met1, Met2: Metal;
         Day: (Sun, Mon, Tue, Wed, Thu, Fri, Sat);
```

Здесь Gaz и Metal – имена перечисляемых типов данных, которые ставятся в соответствие переменным G1, G2, G3 и Met1, Met2. Переменной Day назначается перечисляемый тип данных, которому не присвоено имени. Возможно следующее применение переменной Day:

```
If Day = Sun Then Writeln ('Сегодня выходной');
```

Раздаточный материал № 53

Даны три целых числа. Найти количество положительных чисел в исходном наборе.

```
Program KolPolChisel;  
var  
(a,b,c):= ReadInteger3('Введите три целых числа');  
k: integer;  
begin  
k:=0;  
if a>=0 then k+=1;  
if b>=0 then k+=1;  
if c>=0 then k+=1;  
Print ('Количество положительных чисел = ',k);  
end.
```

Раздаточный материал № 54.

Разработать программу на Паскале.

1. Вывести словесное описание роста взрослого человека: низкий, средний, высокий, необычный (подсказка: использовать в списке выбора диапазонный тип).
2. Даны четыре целых числа. Найти количество положительных и отрицательных чисел.
3. Даны три целых числа. Найти количество четных чисел меньше 20.
4. Перевести пятибалльную оценку в ее наименование.

Задачи из раздаточного материала № 44 №№ 1, 2, 4, 6 реализовать с использованием условной операции.

Раздаточный материал № 55

loop ЦелочисленноеВыражение do
ТелоЦикла;

где

ЦелочисленноеВыражение вычисляется и его значение определяет количество повторений тела цикла. ТелоЦикла - любой оператор языка PascalABC.NET, либо блок

Например, выведем в строке 40 звездочек.

```
begin  
loop 40 do  
  write('*');  
end.
```

Раздаточный материал № 56

№ 1

```
for var ИмяПараметра : тип := Выражение1 to Выражение2 do  
  ТелоЦикла;
```

№ 2

```
for var ИмяПараметра := Выражение1 to Выражение2 do  
  ТелоЦикла;
```

№ 3

```
for var ИмяПараметра : тип := Выражение1 downto Выражение2 do  
  ТелоЦикла;
```

№ 4

```
for var ИмяПараметра := Выражение1 downto Выражение2 do  
    ТелоЦикла;
```

Раздаточный материал № 57

<pre>//программа расчета n! Program Factorial; var n:=ReadInteger('Введите аргумент'); i,f:integer; begin if n<0 then writeln ('Введен неверный аргумент') else begin f:=1; for var i:=2 to n do f:=f*i; Print ('Значение факториала = ', f) end; end.</pre>	<pre>//вычислений произведения n чисел, введенных с клавиатуры Program Сумма; begin var p := int64(1); // для вычисления произведения for var i := 1 to ReadInteger('n=') do p *= ReadInteger('Введите число:'); Println('Результат = ', p) end.</pre>
--	---

Раздаточный материал № 58

while логическое выражение **do**
оператор (или группа операторов)

Примеры

<pre>//вычисление суммы всех нечетных двузначных чисел. s:=0; d:=11; while d<100 do begin s+=d; d+=2 end;</pre>	<pre>//найти максимальное значение среди заранее //неизвестного количества вводимых натуральных чисел Program ПоискМаксимальногоЧисла; begin var (max, n) := (0, 1); Println ('Введи натуральные числа. Признаком окончания ввода - число 0'); while n > 0 do begin n := ReadInteger; if n > max then max := n; end; Println('Результат: max = ', max) end.</pre>
---	---

Раздаточный материал № 59

repeat
ТелоЦикла
until условие

Раздаточный материал № 60

repeat
read(x);
until x=0;

Раздаточный материал № 61

repeat
write(1);

until 2=1;

Пример.

Раздаточный материал № 62

Program ПоискМаксимальногоЧисла;

begin

var max := 0;

var n: integer;

Println ('Введи натуральные числа. Признак окончания ввода - число 0');

repeat

n := ReadInteger;

if n > max **then** max := n;

until n <= 0;

Println('Результат: max = ', max)

end.

Раздаточный материал № 63

//программа, находящая сумму последних цифр

//введенного натурального числа, не превышающую 21.

begin

var (s, n) := (0, ReadInteger);

while n > 0 **do**

begin

var d := n mod 10; // очередная цифра числа

if s + d <= 21 **then** s += d **else break;**

n := n div 10;

end;

Println(s)

end.

Раздаточный материал № 64

// Программа вывода натуральных чисел, не превышающих 20, которые не кратны 3.

begin //самый простой и понятный вариант

Println('натуральные числа, не превышающие 20, которые не кратны 3');

for var i := 1 to 20 **do**

if i mod 3 <> 0 **then** Print(i);

Println

end.

// Программа вывода натуральных чисел, не превышающих 20, которые не кратны 3.

begin //надуманный вариант с continue

Println('натуральные числа, не превышающие 20, которые не кратны 3');

for var i := 1 to 20 **do**

begin

if i mod 3 = 0 **then continue;**

Print(i)

end;

Println

end.

Раздаточный материал № 65

//программа построения таблицы умножения

begin

Write(' ');

for var j :=1 to 9 **do**

Write(' ', j);

Println;

for var i :=1 to 9 **do**

begin

Write(' ', i);

```

for var j:= 1 to 9 do
Write(i*j:3);
Println;
end
end.

```

Раздаточный материал № 66

Random(m, n) - возвращает целое случайное число из диапазона m .. n

Random2(n) - возвращает кортеж из двух целых случайных чисел в диапазоне 0 .. n;

Random2 (m, n) - возвращает кортеж из двух целых случайных чисел в диапазоне m .. n;

Random3(n) - возвращает кортеж из трех целых случайных чисел в диапазоне 0 .. n;

Random3(m, n) - возвращает кортеж из трех целых случайных чисел в диапазоне m .. n;

Раздаточный материал № 67

1. **begin**
2. **var** (Длина, Ширина) := Random2(100, 400);
3. **var** (ПериметрПрямоугольника, ПлощадьПрямоугольника) := (2 * (Длина + Ширина), Длина * Ширина);
4. **var** (Катет1, Катет2) := Random2(10, 90);
5. **var** КвадратГипотенузы := Sqr(Катет1) + Sqr(Катет2);
6. **var** МаксимумТрех := Max(Random(30), Max(Random(25), Random(21)));
7. Println(Длина, Ширина, ПериметрПрямоугольника, ПлощадьПрямоугольника);
8. Println(Катет1, Катет2, КвадратГипотенузы, МаксимумТрех)
9. **End.**

Раздаточный материал № 68

Решает преподаватель совместно со студентами	Решает студент
1. Найти и вывести на экран сумму N чисел. 2. Ввести 3 числа. Найти и вывести на экран сумму положительных чисел.	1. Ввести 4 числа. Найти и вывести на экран сумму и количество отрицательных чисел. 2. Ввести 4 числа. Найти и вывести на экран количество четных чисел. 3. Найти и вывести на экран квадраты и кубы чисел от 2 до 5. 4. Найти и вывести на экран $S=1!+2!+3!+4!+\dots+n!$ ($n>1$). 5. Ввести N чисел. Найти и вывести их среднее арифметическое. 6. Ввести N чисел. Посчитать и вывести количество чисел равных нулю.
3. Даны два целых числа A и B ($A < B$). Вывести в порядке возрастания все целые числа, расположенные между A и B (включая сами числа A и B), а также количество этих чисел (использовать оператор цикла).	7. Даны два целых числа A и B ($A < B$). Вывести в порядке убывания все целые числа, расположенные между A и B (включая сами числа A и B), а также количество этих чисел (использовать оператор цикла). 8. Даны два целых числа A и B ($A < B$). Найти сумму всех целых чисел от A до B включительно (использовать оператор цикла).
4. Получить и вывести следующую арифметическую прогрессию: $a_1=1$, $a_2=4$, $a_3=7$, $a_4=10$, $a_5=13$, ... 5.	9. Посчитать и вывести количество элементов арифметической прогрессии, удовлетворяющих условию $10 < a_i < 30$. 10. Вывести первые N ($N \geq 3$) чисел Фибоначчи и посчитать количество четных чисел. 11. Дана арифметическая прогрессия $a_1=1$, $a_2=4$, $a_3=7$, $a_4=10$, $a_5=13$, ... Составить программу, которая каждый элемент прогрессии разделит

	на 2 и результат округлит до ближайшего целого.
--	---

Раздаточный материал № 69**procedure** *имя*;*раздел описаний процедуры***begin***операторы процедуры***end**;**Раздаточный материал № 70**

Задача: в азбуке Морзе буквы кодируются с помощью точек и тире. Буква S кодируется как «...», а буква O как «---». Вывести сообщение «SOS».

```
Program SOS;
Procedure S; // буква S
begin
    Print ('...');
End;
Procedure O; // буква O
begin
    Print ('---');
End;
Procedure Pause; // символ «пробел»
begin
    Print (' ');
End;
Begin //основной текст программы
S; Pause; O; Pause; S; Pause;
End.
```

Раздаточный материал № 71**procedure** *имя* (*x, y:integer; b:boolean; var k:real*);

где *x, y, b* – это параметры-значения,
k - это параметр-переменная.

Раздаточный материал № 72*Polol (g,w,t,formal);***Раздаточный материал № 73**

Составить программу расчета вывода длины окружности и площади круга ($L=2\pi r$, $S=\pi r^2$). Расчеты оформить процедурой с параметрами (*a* – входной параметр, *K* и *F* – выходные параметры, все параметры вещественного типа). Программу составить для трех окружностей.

```
Program Krug;
procedure Raschet (a: real; var K,F:real);
begin
    K:=2*a*3.14;
    F:=3.14*exp(2*ln(a));
end;

var
```

```

x,dlina, ploch: real;
begin
for var i:=1 to 3 do
begin
Println ('Введите радиус ',i,' окружности');
readln(x);
Raschet (x,dlina,ploch);
Println ('Длина окружности ',i,' составляет: ',dlina);
Println ('Площадь круга ',i,' составляет: ',ploch);
end;
end.

```

Раздаточный материал № 74

procedure ИмяПроцедуры(Параметр1; ... ПараметрN) : = Оператор;

procedure Privet; // процедура без параметров begin Println('PascalABC.NET приветствует Вас!') end;	procedure Privet := Println('PascalABC.NET приветствует Вас!');
procedure ПлощадьПрямоугольника(Длина, Ширина : real; var Площадь : real); begin Площадь := Длина * Ширина end;	procedure ПлощадьПрямоугольника(Длина, Ширина : real; var Площадь : real) := Площадь := Длина * Ширина;
procedure ReadInt64(var a : int64); begin Readln(a) end;	procedure ReadInt64(var a : int64) := Readln(a);

Раздаточный материал № 75

1. Даны три целых числа. Определить у какого числа больше сумма цифр. Вывод результата предусмотреть в основной программе. Расчет суммы цифр оформить в процедуре.
2. Рассчитать и вывести периметр и площадь прямоугольника. Расчеты оформить в процедуре.
3. Написать программу, подсчитывающую количество цифр числа, используя для этого процедуру.

Раздаточный материал №76

function имя (список параметров) : тип возвращаемого значения;

раздел описаний

begin

операторы

end;

Раздаточный материал №77

Result := возвращаемое значение;

(Для удобства переменную Result желательно использовать в любой функции)

Раздаточный материал № 78

Оформить в виде функции операцию возведения целого числа в третью степень.

Вариант 1	Вариант 2 упрощенный синтаксис
<pre>{Возведение числа в третью степень} Program ТретьяСтепень; function CubeInt (x : integer) : integer; begin Result := x*x*x; end; var x:integer := 0; begin x := ReadInteger('Введите целое число'); Println('Результат = ',CubeInt(x)) end.</pre>	<pre>{Возведение числа в третью степень} Program ТретьяСтепень; function CubeInt (a: integer) := a*a*a; // упрощенный синтаксис var x:integer := 0; begin x := ReadInteger('Введите целое число'); Println('Результат = ',CubeInt(x)) end.</pre>
Вариант 3	
<pre>{Возведение числа в третью степень} Program ТретьяСтепень; function CubeInt (a: integer) := a*a*a; begin Println('Результат = ',CubeInt(ReadInteger('Введите целое число'))) end.</pre>	

Раздаточный материал № 79

Составить программу, содержащую процедуру расчета $y := (x+1)^3 + (x-1)^3$ и оформить ее вызов (x – вещественное число). Составить программу, содержащую функцию расчета $y := (x+1)^3 + (x-1)^3$ и оформить ее вызов (x – вещественное число). Произвести сравнительный анализ полученных кодов. Сделать соответствующие выводы.

Раздаточный материал № 80

Var идентификатор **array** [тип индекса] **of** тип данных;

Где *идентификатор* – имя массива, в качестве *типа индекса* удобно использовать интервальный тип, *тип данных* – может быть любой тип.

Пусть имеется одномерный массив «Числа» размерностью 10, состоящий из целых чисел. Тогда его объявление будет иметь вид:

Var Числа : array [1..10] of integer;

Чтобы обратиться к отдельному элементу необходимо указать его имя и в квадратных скобках индекс элемента:

Числа[5], Числа[8], Числа[2] и т.п. Или обобщенное обращение - Числа[i]

Пусть имеется двумерный массив «Числа2» размерностью 3*3, состоящий из 9 целых чисел. Тогда его объявление будет иметь вид:

Var Числа2 : array [1..3,1..3] of integer;

Для обращения к отдельному элементу необходимо указать:

Числа2[1,1], Числа2[3,2], Числа2[2,1] и т.п. Или обобщенное обращение - Числа2[i,j], где i – номер строки, j – номер столбца

Раздаточный материал № 81

Операция	Одномерный массив	Двумерный массив
Ввод элементов массива	<pre>for var i:=1 to 10 do Число[i]:=ReadInteger('Введите элемент массива')</pre>	<pre>for var i:=1 to 3 do for var j:=1 to 3 do Число2[i,j]:=ReadInteger('Введите элемент массива')</pre> <p style="text-align: center;">или</p> <pre>for var i:=1 to 3 do for var j:=1 to 3 do Число2[i,j]:=Random(10,20);</pre>
Вывод элементов массива	<pre>for i:=1 to 10 do Writeln(Число[i])</pre>	<pre>for var i:=1 to 3 do for var j:=1 to 3 do Println(Число2[i,j])</pre> <p style="text-align: center;">или</p> <pre>Println(Число2)</pre>
Поиск наибольшего элемента массива	<pre>RMax := Число[1]// 12, 10, 45, 14, 66 for i:=2 to 10 do if Число[i] > RMax then RMax:= Число[i];</pre>	<pre>MaxЧисло2:= Число2[1,1]; for i:=2 to 7 do for j:=1 to 2 do if MaxЧисло2 < Число2[i,j] then MaxЧисло2 := Число2[i,j];</pre>

Раздаточный материал № 82

Решает преподаватель совместно со студентами	Решает студент
<ol style="list-style-type: none"> В одномерном массиве на 5 элементов найти сумму всех элементов. В одномерном массиве на 5 элементов найти сумму положительных элементов. В одномерном массиве на 5 элементов найти сумму и количество четных элементов. В одномерном массиве на 5 элементов вывести элементы с нечетными индексами. В одномерном массиве на 5 элементов найти элементы, кратные 7. В одномерном массиве на 5 элементов найти минимальный элемент. В одномерном массиве на 5 элементов отрицательные элементы заменить на 0. В одномерном массиве на 5 элементов вывести на экран первые два элемента. 	<ol style="list-style-type: none"> В одномерном массиве на 5 элементов найти произведение всех элементов. В одномерном массиве на 5 элементов найти сумму отрицательных элементов. В одномерном массиве на 5 элементов найти сумму и количество нечетных элементов. В одномерном массиве на 5 элементов вывести элементы с четными индексами. В одномерном массиве на 5 элементов найти элементы, кратные 6, и их количество. В одномерном массиве на 5 элементов найти минимальный элемент. В одномерном массиве на 5 элементов положительные элементы заменить на 12. В одномерном массиве на 5 элементов вывести элементы в обратном порядке. В одномерном массиве на 5 элементов найти среднее арифметическое элементов. В одномерном массиве на 5 элементов вывести на экран последние три элемента.
<ol style="list-style-type: none"> В двумерном массиве 4*4 найти сумму всех элементов. В двумерном массиве 4*4 найти сумму элементов главной диагонали. В двумерном массиве 4*4 найти сумму элементов первого столбца. 	<ol style="list-style-type: none"> В двумерном массиве 4*4 найти сумму всех четных элементов. В двумерном массиве 4*4 сумму элементов главной диагонали умножить на все элементы массива. В двумерном массиве 4*4 найти сумму элементов первой строки. В двумерном массиве 4*4 найти сумму элементов первого и третьего столбцов. В двумерном массиве 4*4 найти количество четных элементов.

Раздаточный материал № 83

```
var c1, p135, rz: char; // три переменные
var Символ1: char; // одна переменная
var ar: array of char; // динамический массив символов
```

Описание можно соединить с инициализацией:

```
var a: char := 'a'; // тип указан явно
var b:= 'b'; // автовыведение типа
```

Раздаточный материал № 84

```
begin
  PrintLn((2 * ('м' + 'а') + ' ') * 5)
end.
```

Раздаточный материал № 85

```
begin
  ('$' + 132.5).PrintLn
end.
```

Раздаточный материал № 86

- Ord(c) - код символа c в Unicode (тип word);
- char.Code - то же самое, точечная нотация;
- OrdAnsi(c) - десятичный код символа в однобайтной кодировке Windows (тип byte).

Раздаточный материал № 87

```
{Программа выводит коды символов кириллицы в кодировке Unicode, без букв Ё и ё}
begin
  for var c := 'А' to 'я' do // параметр цикла for может быть любого порядкового типа,
                             // следовательно, и типа char
    Print(Ord(c))
  end.
```

Раздаточный материал № 88

- Chr(word) - символ с указанным кодом Unicode;
- #word - символ с указанным кодом Unicode; принимает только литерал;
- ChrAnsi(byte) - символ с кодом в однобайтной кодировке Windows.

Раздаточный материал № 89

```
{Программа выводит символ кириллицы в кодировке Unicode}
begin
  Print(Chr(ReadInteger))
end.
```

Раздаточный материал № 90

```
ReadLnChar('ТекстПриглашения'); // ввод одного символа
ReadLnChar2('ТекстПриглашения'); // ввод двух символов
ReadLnChar3('ТекстПриглашения'); // ввод трех символов
```

Раздаточный материал № 91

char.IsLetter(c)

char - это имя класса, а c означает переменную или непосредственно символ.

Раздаточный материал № 92

{Дана матрица A(5x3), состоящая из символов.

- Определить количество элементов матрицы, равных символу 'f'.
- Определить, является ли очередной введенный символ буквой}

Program Символы1;

var МассивСимволов: **array** [1..5,1..3] **of** char;

k: **integer** := 0;

begin

for var i := 1 **to** 5 **do**

for var j := 1 **to** 3 **do**

begin

 МассивСимволов[i,j] := readlnChar('Введите символ');

 k := (МассивСимволов[i,j] = 'f' ? k+1 : k);

if МассивСимволов[i,j].IsLetter **then** Println('Это буква') **else** Println('Это не буква');

 Println(', код',Ord(МассивСимволов[i,j]))

 // или **if** char.IsLetter(МассивСимволов[i,j]) **then** Println('Это буква') **else** Println('Это не буква')

end;

 Println('Количество элементов матрицы, равных символу 'f' = ',k)

end.

Раздаточный материал № 93

№	Расширение / статический метод	Описание
1.	c.IsLetter	расширение возвращает True, если символ c принадлежит к группе букв и False в противном случае
2.	char.IsLetter(c)	статический метод класса, делающий то же самое.
3.	c.IsDigit	расширение возвращает True, если символ c принадлежит к группе цифр (0, 1, ... 9) и False в противном случае.
4.	char.IsDigit(c)	статический метод класса, делающий то же самое.
5.	char.IsLetterOrDigit(c)	статический метод класса возвращает True, если символ c принадлежит к группе букв или цифр и False в противном случае.
6.	char.IsWhiteSpace(c)	статический метод класса возвращает True, если символ c принадлежит к группе пробельных символов и False в противном случае.
7.	char.IsPunctuation(c)	статический метод класса возвращает True, если символ c принадлежит к группе знаков пунктуации (разделителям) и False в противном случае.
8.	c.IsUpper	расширение возвращает True, если символ c принадлежит к буквенным символам верхнего регистра (прописным) и False в противном случае.
9.	char.IsUpper(c)	статический метод класса, делающий то же самое.
10.	c.IsLower	расширение возвращает True, если символ c принадлежит к буквенным символам нижнего регистра (строчным) и False в противном случае.
11.	char. IsLower(c)	статический метод класса, делающий то же самое

Раздаточный материал № 94 (Справочный материал)

№	Расширение / статический метод/ функция/ процедура	Описание
1.	c.ToUpper	расширение возвращает буквенный символ с, приведенный к верхнему регистру, если он принадлежит к нижнему регистру. В противном случае символ возвращается без изменения.
2.	UpCase(c)	функция, делающая то же самое
3.	UpperCase(c)	функция, синоним UpCase.
4.	char.ToUpper(c)	статический метод класса, переводящий символ с на верхний регистр, если он принадлежит к нижнему регистру. В противном случае никаких действий не выполняется.
5.	c.ToLower	расширение возвращает буквенный символ с, приведенный к нижнему регистру, если он принадлежит к верхнему регистру. В противном случае символ возвращается без изменения
6.	LowCase(c)	функция, делающая то же самое
7.	char.ToLower(c)	статический метод класса, переводящий символ с на нижний регистр, если он принадлежит к верхнему регистру. В противном случае никаких действий не выполняется
8.	c.Pred	расширение возвращает буквенный символ, код которого предшествует коду символа с
9.	Pred(c)	функция, делающая то же самое
10.	c.Succ	расширение возвращает буквенный символ, код которого следует за кодом символа с
11.	Succ(c)	функция, делающая то же самое.
12.	Dec(c)	процедура, заменяющая значение переменной, содержащей символ с на символ, предшествующий ему. Декремент - операция Dec - это она же, только для символов
13.	Dec(c, n)	процедура, заменяющая значение переменной, содержащей символ с символом, находящимся в кодовой таблице на n позиций раньше. Тоже декремент
14.	Inc(c)	процедура, заменяющая значение переменной, содержащей символ с на символ, следующий за ним. Это инкремент
15.	Inc(c, n)	процедура, заменяющая значение переменной, содержащей символ с символом, находящимся в кодовой таблице на n позиций дальше. Тоже инкремент
16.	c.InRange(c1, c2)	расширение возвращает True, если код символа с принадлежит интервалу, в котором находятся коды символов, начиная от c1 и заканчивая c2. В противном случае возвращается False. Например, буква L находится в интервале от 0 до R, но не находится в интервале от Д до Я

Раздаточный материал № 95

{Дана матрица A(5x3), состоящая из символов.

- привести символы из нижнего регистра в верхний.
- вывести символ, следующий за указанным, в таблице Unicode}

Program Символы1;

var МассивСимволов: **array** [1..5,1..3] **of** **char**;

k: **integer** := 0;

begin

for var i :=1 to 5 **do**

for var j :=1 to 3 **do**

begin

 МассивСимволов[i,j] := readlnChar('Введите символ');

 Println('Буквенный символ переведен в верхний регистр',UpCase(МассивСимволов[i,j]));

 Println('Следующий символ в таблице Unicode',Succ(МассивСимволов[i,j]))

end;
end.

Раздаточный материал № 96

```
var st, МояСтрока, p18: string; // три строки  
var Строчка: string; // одна строка  
var sh1: string[27]; // короткая строка, максимум 27 символов
```

Раздаточный материал № 97

```
ReadlnString  
ReadlnString(строка приглашения ко вводу)  
ReadlnString2  
ReadlnString3  
ReadSrting  
ReadSrting2  
ReadSrting3
```

Раздаточный материал № 98

```
begin  
  var (s1, s2, s3) := ReadlnString3('Введите три строки через Enter:');  
  ('(' + s1 + ')').Println;  
  ('(' + s2 + ')').Println;  
  ('(' + s3 + ')').Println;  
end.
```

Раздаточный материал № 99

```
begin  
  Println(' --- 25 + 3 * 8 = '+(25 + 3 * 8) + ' ---')  
end.
```

Раздаточный материал № 100

'cosm1' < 'cosm2'	True
'pascal' > 'PASCAL'	True
'Ключ_' <> 'Ключ'	True
'OC Windows' = 'OC Windows'	True

Раздаточный материал № 101

```
'ABCDEFGH' → Copy(S,2,3) → 'BCD'  
'ABCDEFGH' → Copy(S,4,4) → 'DEFGH'
```

Раздаточный материал № 102

```
Concat ('AA', 'bb', 'CC') → 'AAAbbCC'
```

Раздаточный материал № 103

```
'test-5' → Length (S) → 6
```

Раздаточный материал № 104

```
'abcdefg' → Pos ('cd', S2) → 3  
'abcdcdefg' → Pos ('cd', S2) → 3  
'abcdefg' → Pos ('k', S2) → 0
```

Раздаточный материал № 105

‘abcdefg’ → Delete (S, 3, 2) → ‘abefg’
‘abcdefg’ → Delete (S, 2, 6) → ‘a’

Раздаточный материал № 106

‘Рис. 2’ → Insert (‘№’, S2, 5) → ‘Рис. №2’

Раздаточный материал № 107

```
//программа получения из слова "ВЕЛИЧИНА" слова "НАЛИЧИЕ"
Program Slovo_1;
var S11, S12 : String[10];
begin
S11:= 'ВЕЛИЧИНА';
S12:= Copy (S11,7,2) + copy(S11,3,4)+ S11[2];
writeln(S12);
end.

//программа подсчета в символьной строке количество цифр, предшествующих
первому символу "!"
Program Slovo_2;
var S : String;
k, i : byte;
begin
S := ReadString('Введите строку');
k:=0;
i:=1;
while (i <= Length(S)) and (S[i] <> '!') do
begin
if (S[i] >= '0') and (S[i] <= '9') then inc(k);
inc (i);
end;
writeln('Количество цифр, предшествующих первому символу "!" равно ', k);
end.
```

Раздаточный материал № 108

[3, 4, 7, 15, 20] - множество из пяти целых чисел;
[1 .. 100] - множество целых чисел от 0 до 100;
[‘a’, ‘b’, ‘c’] - множество, содержащее три литеры;
[‘a’ .. ‘z’, ‘?’, ‘!'] - множество, содержащее все строчные латинские буквы и знаки ?, !.

Раздаточный материал № 109

[1, 2, 3] и [3, 2, 1] - это эквивалентные множества.
[1, 2, 3, 4, 2, 3, 4, 5] и [1 .. 5] - это эквивалентные множества.

Раздаточный материал № 110

Var <идентификатор> : Set Of <базовый тип>

Var

A, D : Set Of Byte;

B : Set Of ‘a’ .. ‘z’;

C : Set Of Boolean;

Раздаточный материал № 111

< множественная переменная > := <множественное выражение>

A := [50, 100, 150, 200];

B := ['n', 'm', 'k'];

D := A;

Раздаточный материал № 112

[1, 2, 3, 4] + [3, 4, 5, 6] → [1, 2, 3, 4, 5, 6]

[1, 2, 3, 4] * [3, 4, 5, 6] → [3, 4]

[1, 2, 3, 4] - [3, 4, 5, 6] → [1, 2]

[3, 4, 5, 6] - [1, 2, 3, 4] → [5, 6]

Раздаточный материал № 113

var s := [5, 8, 11, 6, -3, 18, 40, 26];

Println(s);

s += [7];

s += [12];

Include(s, 0);

Include(s, 13);

Println(s)

End.

{ 11,-3,26,8,40,5,6,18} // !!! Порядок следования элементов во множестве меняется хаотично.

{40,13,12,11,26,8,7,-3,5,6,18,0}

Раздаточный материал № 114

begin

var s := [5, 8, 11, 6, -3, 18, 40, 26];

Println(s);

s -= [11];

Exclude(s, 18);

s -= [9]; // не ошибка

Exclude(s, 10); // не ошибка

Println(s)

End.

{ 11,-3,26,8,40,5,6,18}

{40,-3,26,8,6,5}

Раздаточный материал № 115

Отношение	Результат	
	True	False
A = B	Множества A и B совпадают	В противном случае
A < > B	Множества A и B не совпадают	То же
A <= B	Все элементы A принадлежат B	«
A >= B	Все элементы B принадлежат A	«

Раздаточный материал № 116

Пусть переменная M описана в программе следующим образом:

Var M : Set Of Byte;

В разделе операторов ей присваивается значение

M := [3, 4, 7, 9]

Тогда операции отношения дадут следующие результаты:

```
M = [4, 7, 3, 3, 9] - True
M <> [7, 4, 3, 9] - False
[3, 4] <= M - True
[] <= M - True
M >= [1 .. 10] - False
M <= [3 .. 9] - True
```

Раздаточный материал № 117

x In M

4 In M - True

5 In M - False

Результатом здесь будет логическая величина True, если значение x входит во множество M, и False – в противном случае.

Раздаточный материал № 118

begin

var n := ReadInteger;

if n in [1..9] then Println('Введено натуральное однозначное число')

end.

// Здесь множество [1..9] используется непосредственно, без объявления, что делает множество удобным в данной задаче.

Раздаточный материал № 119

begin

Println([3, 1, 2] < [1, 2, 3, 4], [1, 2, 3] < [2, 3, 1])

end.

True False

Раздаточный материал № 120

begin

Println([1, 2, 3, 4] > [3, 1, 2], [1, 2, 3] > [2, 3, 1])

end.

True False

Раздаточный материал № 121

```
// Дана символьная строка. Необходимо
определить в ней количество знаков препинания
(. - , * : ! ?)
Program Znaki;
var
s: string;
i, k: byte;
begin
readln (s);
k :=0;
for i:=1 to length (s) do
if S[i] In ['.', '-', ',', '*', ':', '!', '?']
then inc (k);
writeln ('Число знаков препинания равно ', k)
end.
```

Задачу можно решить без множества, записав в условном операторе длинное логическое выражение (S[i] = \'.') Or (S[i] = \'-') и т.д., что не является оптимальным решением.

Раздаточный материал № 122

{Существует набор продуктов, продаваемых в нескольких магазинах города. Определить: какие продукты есть во всех магазинах города; полный набор продуктов в городе}

Program Множества;

type Food = (хлеб, молоко, мясо, масло, соль, сыр, сахар, рыба);

shop=**set of** Food;

var a,b,c: shop;

begin

a:=[хлеб,молоко];

b:=[хлеб,сыр,масло];

c:=[хлеб,сыр];

Println('продукты есть во всех магазинах города: ', a*b*c);

Print('полный набор продуктов в городе: ', a+b+c);

end.

Раздаточный материал № 123

1. Вычислить значения следующих отношений.

- a. $[2] \supset [2, 2, 2]$
- b. $['a', 'b'] = ['b', 'a']$
- c. $[4, 5, 6] = [4, 5, 6]$
- d. $['c', 'b'] = ['c' .. 'b']$
- e. $[2, 3, 5, 7] \leq [1..9]$
- f. $[3, 6..8] \leq [2..7, 9]$
- g. $\text{Trunc}(3.9) \text{ In } [1, 3, 5]$
- h. $\text{Odd}(4) \text{ In } []$

2. Вычислить значения следующих выражений:

- a. $[1,3,5] + [2,4]$
- b. $[1,3,5] * [2,4]$
- c. $[1,3,5] - [2,4]$
- d. $[1..6] + [3..8]$
- e. $[1..6] * [3..8]$
- f. $[1..6] - [3..8]$
- g. $[] + [4]$
- h. $[] * [4]$
- i. $[] - [4]$