

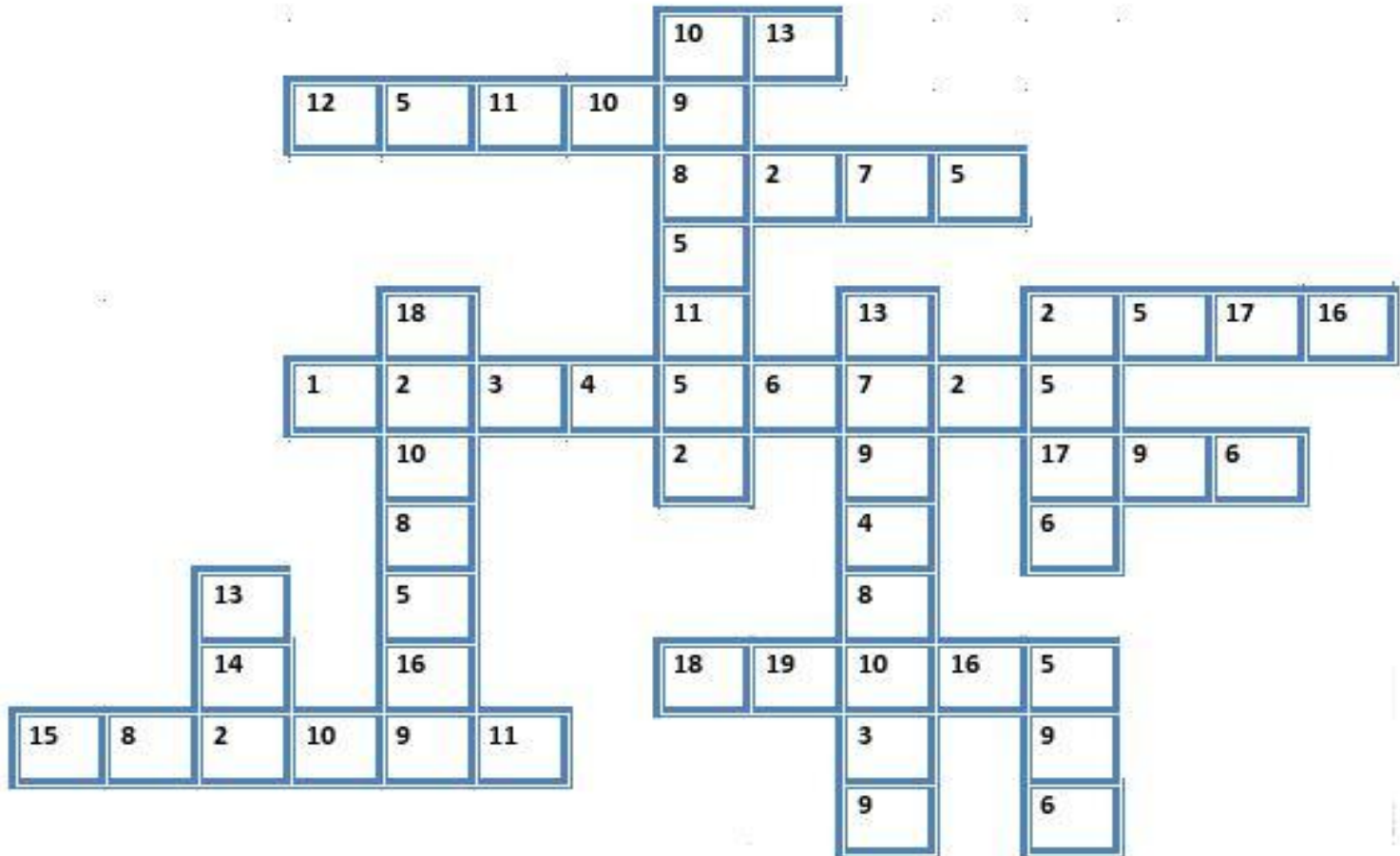
**Программы-процедуры.  
Правила описания и  
использования программ-  
процедур.**

# Повторение материала

1. Какова структура программы?
2. Обязателен ли заголовок программы?
3. Перечислите разделы описаний
4. Какие служебные слова служат для описания переменных?
5. Какое служебное слово является началом основной части программы?
6. Как заканчивается программа?

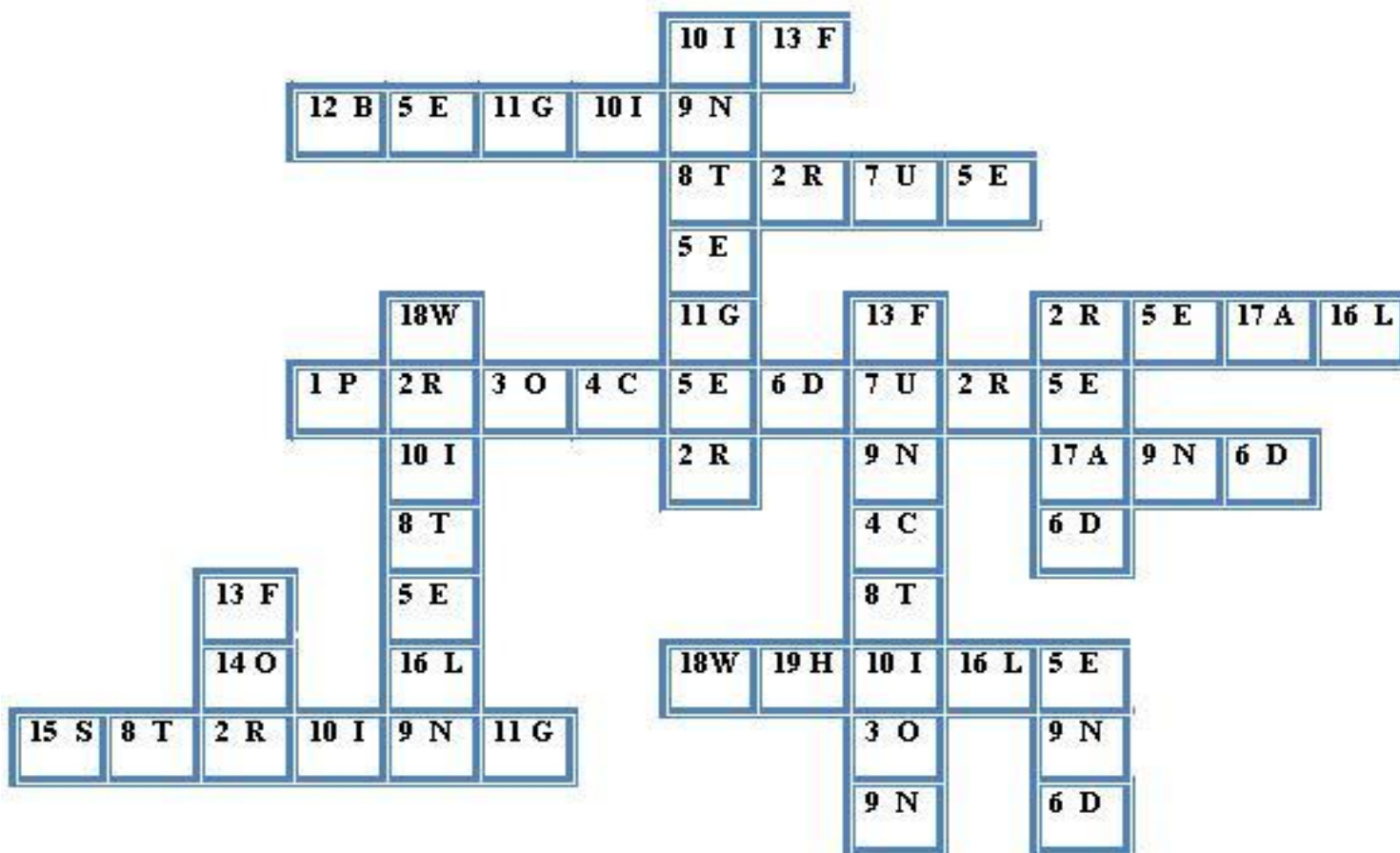
# Ключворд

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
P	R	O	C	E	D	U												



# Ключворд - ответ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
P	R	O	C	E	D	U	T	N	I	G	B	F	O	S	L	A	W	H



**Подпрограмма** – повторяющаяся группа операторов, оформленная в виде самостоятельной программной единицы.

В языке Pascal механизм подпрограмм реализуется в виде **ПРОЦЕДУР (PROCEDURE)** и **ФУНКЦИЙ (FUNCTION)**

# Подпрограмма - процедура

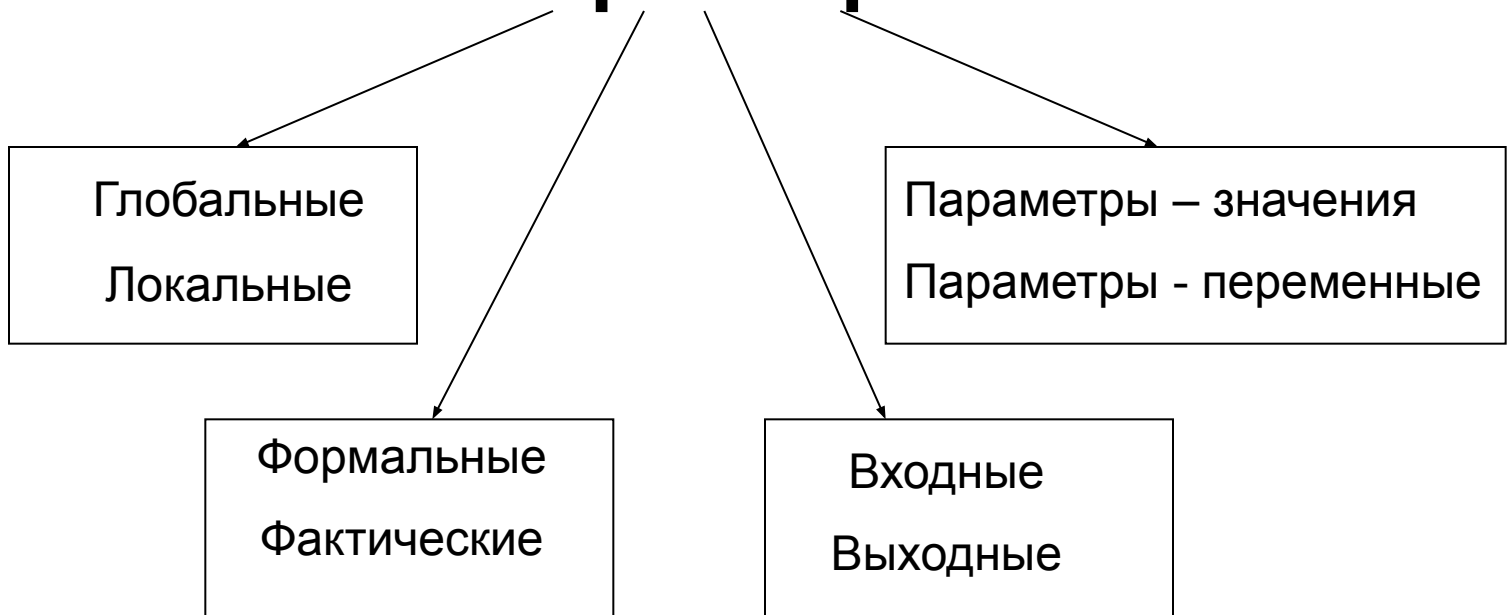
## Описание:

- выбрать для неё имя.
- определить параметры.
- составить последовательность действий, которые надо выполнять, для получения результата.

## Обращение:

- вызов выполнения указанных действий, для конкретных значений параметров.

# Параметры



# Параметры

- *Глобальные* – описываются в головном модуле, доступны любой подпрограмме.
- *Локальные* – используются только в процедуре, они могут быть или не быть, описываются после слова VAR, с указанием типа.



# Параметры

- *Формальные* – описываются в заголовке процедуры, к ним относятся входные и выходные параметры.
- *Входные* – это и параметры значений, описываются через запятую с указанием типа. При выходе из процедуры – не сохраняются.
- *Выходные* – это и параметры переменные, описываются после VAR через запятую, с указанием типа. При выходе из процедуры – сохраняются.
- *Фактические* – располагаются в головном модуле при вызове процедуры.

*Формальные и фактические*  
параметры должны совпадать  
по 3 признакам:

- по количеству
- по типу
- по порядку следования

# Общий вид структуры подпрограммы

***Procedure*** **<имя>** (формальные параметры);

**VAR** (описание локальных параметров, они могут быть или не быть)

**begin**

тело процедуры

**end;** (конец процедуры)

**BEGIN** (головного модуля)

Фактические параметры

**END.**

A spiral-bound notebook with a white page and a brown cover. The spiral binding is on the left side. The text "Спасибо за внимание" is centered on the page.

Спасибо за внимание

# Пример

1. Составить программу сложения 2-х чисел, значения вводятся с клавиатуры.

```
uses crt ;  
var a,b,s:real;  
procedure summa (x,y:real; var s:real);  
begin  
s:=x+y;  
end;  
begin  
  writeln('vved 2 chisla ');read(a,b);  
  summa(a,b,s);  
  writeln('s=',s:3:1);  
  readkey;  
end.
```

(глобальные) (формальные)

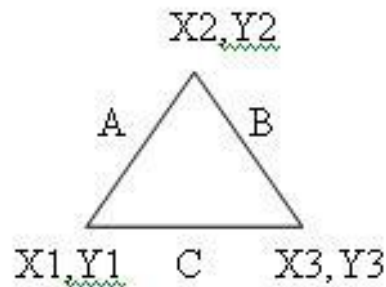
(параметры – значения)  
(параметры – переменные, выходные, при выходе сохраняются)  
входные при выходе не сохраняются)

# Домашнее задание

- Треугольник задан координатами вершин, вычислить периметр и площадь. Вычисление длин сторон оформить с помощью процедуры.

Объяснение:

- чтобы подсчитать длину одной стороны необходимо 4 числа



$X1, Y1, X2, Y2, X3, Y3$  – входные параметры

$A, B, C$  – промежуточные параметры

$P, S$  – выходные периметр, площадь

$P1$  – полупериметр

- формула для подсчёта длины:

$$A = \sqrt{(x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)}$$

Какая функция в Паскале означает корень квадратный? – SQRT  
означает возведение в квадрат – SQR

- периметр находим по формуле  $P = A + B + C$
- формула для нахождения площади:  
 $S = \sqrt{P1 * (P1 - A) * (P1 - B) * (P1 - C)}$
- сколько раз будем обращаться к процедуре? – 3 раза