



# Первый этап – постановка задачи.

- На этом этапе участвует человек, хорошо представляющий предметную область задачи. Он должен четко определить цель задачи, дать словесное описание содержания задачи и предложить общий подход к её решению.

## Второй этап – математическое или информационное моделирование

- Цель этого этапа – создать такую математическую модель решаемой задачи, которая может быть реализована в компьютере.

# Третий этап - алгоритмизация задачи.

- На основе математического описания необходимо разработать алгоритм решения.

# Четвертый этап – программирование.

- **Программой** называется план действий, подлежащих выполнению некоторым исполнителем, в качестве которого может выступать компьютер.
- Составление программы обеспечивает возможность выполнения алгоритма и соответственно поставленной задачи исполнителем – компьютером.

## Пятый этап - ввод программы и исходных данных в ЭВМ.

- Программа и исходные данные вводятся в ЭВМ с клавиатуры с помощью редактора текстов, и для постоянного хранения осуществляется их запись на гибкий или жесткий магнитный диск.

## Шестой этап - тестирование и отладка программы.

- На этом этапе происходят выполнение алгоритма с помощью ЭВМ, поиск и исключение ошибок. При этом программисту приходится выполнять рутинную работу по проверке работы программы, поиску и исключению ошибок, и поэтому для сложных программ этот часто требует гораздо больше времени и сил, чем написание первоначального текста программы..

# Отладка программы

- сложный и нестандартный процесс. Исходный план отладки заключается в том, чтобы протестировать программу на контрольных примерах.
- **исполнение отлаженной программы и анализ результатов**

# Язык программирования Паскаль

Язык Паскаль является языком программирования высокого уровня. Начиная с момента своего создания Н. Виртом в 1970 г., играет особую роль и в практическом программировании. Текст Паскаль-программы представляет собой последовательность строк, состоящих из символов, образующих алфавит языка.

## Основные средства языка

Символы языка – это элементарные знаки, используемые при составлении текстов.

Алфавит языка – набор таких символов.

Алфавит языка Turbo Pascal 7.0 включает:

- все латинские прописные и строчные буквы
- арабские цифры (0 – 9)
- символы + - \* / = < > , . ; : ‘ \_ ( ) { } и др.
- служебные (зарезервированные) слова

# Арифметические операции

операция	
*	умножение
/	деление
div	Целочисленное деление (15 div 2=7)
mod	Остаток от деления (15 mod 2=1)
-	вычитание
+	сложение

# функции, определенные в Паскале:

Математика	PASCAL
$X^2$	SQR(X)
$\sqrt{X}$	SQRT(X)
$ X $	ABS(X)
SIN X	SIN(X)
COSX	COS(X)
$e^x$	EXP(X)
LN X	LN(X)
$X^Y$	EXP(Y*LN(X))

Для записи команд, имен функций, поясняющих терминов *Turbo Pascal 7.0* существует набор строго определенных слов, которые называются **служебными** или **зарезервированными** (это английские мнемонические сокращения).

### Служебные слова делятся на три группы:

- операторы (*READ, WRITELN* и др.)
- имена функций (*SIN, COS* и др.)
- ключевые слова (*VAR, BEGIN, END* и др.)

# Структура программы

Любая программа на Паскале имеет определенную структуру:

1. Заголовок программы
2. Раздел описаний
3. Тело программы

# Структура программы

```
PROGRAM имя;  
Const; - раздел констант  
Var; - раздел переменных;  
BEGIN  
оператор 1;  
оператор 2;  
.....  
оператор n;  
Readln  
END.
```

# Операторы ввода-вывода

Любая программа  
взаимодействует с пользователем  
с помощью операторов ввода-  
вывода.

# Оператор ввода *READ*

Ввод информации с клавиатуры осуществляется с помощью оператора *READ*.

Когда в программе встречается оператор *READ*, то её действие приостанавливается до тех пор, пока не будут введены исходные данные.

Имеет вид:

- *READ(a,b,...)*
- *READLN(a,b,...).*

где *a,b,...* – имена вводимых переменных

- *READLN* – применяется, когда исполнение программы желательно задержать до нажатия клавиши *ENTER*.

# Оператор вывода *WRITE*

Для вывода информации на экран используется оператор *WRITE*.

Имеет вид:

- *WRITE(a,b,...)*
- *WRITELN(a,b,...)*, где *a,b,...* – СПИСОК ВЫВОДИМЫХ КОНСТАНТ, ПЕРЕМЕННЫХ, ВЫРАЖЕНИЙ.

# Оператор вывода *WRITE*

1. *WRITE(2\*2)* – выводит на экран: 4
2. *WRITE('2\*2=')* – выводит на экран: 2\*2=
3. *WRITE('2\*2='; 2\*2)* – выводит на экран: 2\*2=4
4. *write (x : 7 : 2, y : 7 : 2);*  
где **7** – количество позиций под все  
число,  
а **2** – количество позиций после  
запятой

## 2. Оператор вывода

`WRITELN` (список переменных) – тоже что и `write`, но с переводом курсора **на новую строку** после вывода последней переменной;

`WRITELN` – вывод пустой строки;

# Структура программы

**PROGRAM** *имя*;

*Const*; - раздел констант;

*Var*; - раздел переменных;

**BEGIN**

*оператор 1*;

*оператор 2*;

.....

*оператор n-1*;

*оператор n*;

**Readln**

**END.**

**В** языке Turbo Pascal имеются следующие группы простых типов данных:

- 1) Целые типы**
- 2) Вещественный тип**
- 3) Перечисляемый тип**
- 4) Тип-диапазон**
- 5) Символьный тип**
- 6) Логический тип**



# Линейный алгоритм

Program tr;

var a, b, c, p, s : real;

Begin

writeln('Введите длины сторон треугольника');

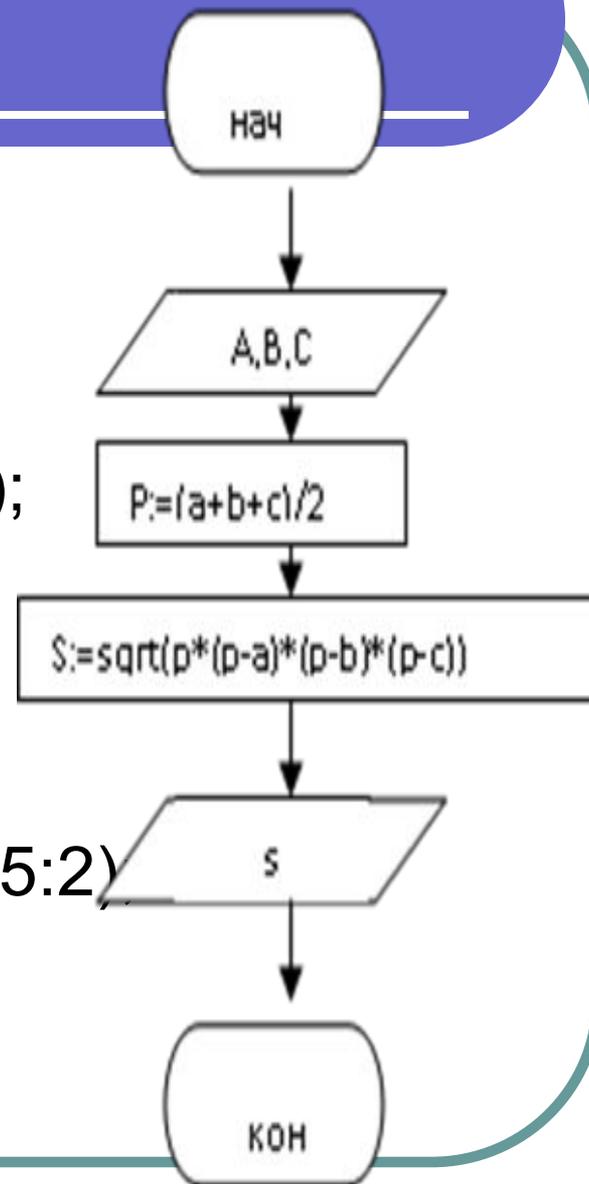
readln(a,b,c);

p := (a + b + c)/2;

s := sqrt(p\*(p - a)\*(p - b)\*(p - c));

writeln('Площадь треугольника =', s:5:2);

end.



# Вычислить произведение двух целых

чисел.

```
Program p1;  
Var a,b,c: integer; {Числа в языке Pascal различаются как целые и вещественные}  
Begin  
WriteLn('Введите два целых числа через пробел');  
ReadLn(a,b);  
c:=a*b;  
WriteLn('Их произведение равно: ', c);  
ReadLn  
End.
```