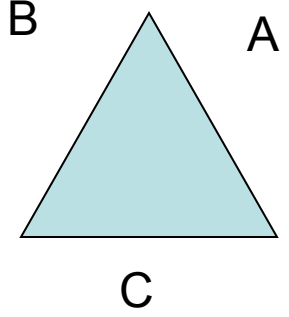


Программирование алгоритмов.  
Типы переменных. Операторы,  
процедуры и функции

ЭТАПЫ	ПОЯСНЕНИЯ	
<b>1-й этап ПОСТАНОВКА ЗАДАЧИ</b>	Даны длины сторон треугольника А, В, С. Найти площадь треугольника S. Составьте блок-схему алгоритма решения поставленной задачи.	
<b>2-й этап РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ</b>		<p>Дано А, В, С – стороны треугольника, чтобы найти площадь воспользуемся формулой Герона</p> $S = \sqrt{P(P-A)(P-B)(P-C)}$ <p>Для расчета нам нужно знать полупериметр <math>P = \frac{A+B+C}{2}</math></p>
<b>3-й этап КОНСТРУИРОВАНИЕ АЛГОРИТМА</b>	Все формулы, записанные на языке математики необходимо перевести в формулы понятные исполнителю. $S := \text{SQRT}(P*(P-A)*(P-B)*(P-C))$ $P := (A+B+C)/2$	
<b>4-й этап АЛГОРИТМ в виде блок-схемы</b> См. слайд 17	<b>5-й этап РЕШЕНИЕ ЗАДАЧИ НА КОМПЬЮТЕРЕ</b>	<b>6-й этап</b> См. слайд 18 <b>КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ</b>

# Структура программы

Этапы решения задачи на  
компьютере

Этап 1.

Составление программы

Начало

Название программы  
Блок описания переменных  
Начало

Ввод А, В, С

Блок ввода переменных

$P := (A + B + C) / 2$

Действия

$S := \text{SQRT}(P * (P - A) * (P - B) * (P - C))$

Вывод S

Блок вывода переменных

Конец

Конец

# Типы переменных

Раздел описания переменных

Для программной обработки в ЭВМ данные представляются в виде величин и их совокупностей.

**Величина — это элемент данных с точки зрения их смыслового содержания или обработки.**

Смысловое разбиение данных производится во время постановки задачи и разработки алгоритма ее решения.

Выделяют разбиения на: ***входные, выходные и промежуточные***

- Исходные (входные) — это данные, известные перед выполнением задачи, из условия.
- Выходные данные — результат решения задачи.
- Переменные, которые не являются ни аргументом, ни результатом алгоритма, а используются только для обозначения вычисляемого промежуточного значения, называются промежуточными.

**Пример из задачи:**

Исходные

Выходные данные

Промежуточные

# Характеристики (атрибуты) величины

## Имя

это ее обозначение  
и место в памяти

## Тип

множество допустимых значений и  
множество применимых  
операций к ней

**Значение** — динамическая характеристика, может меняться многократно в ходе исполнения алгоритма. Во время выполнения алгоритма в каждый конкретный момент величина имеет какое-то значение или не определена.

**Постоянной** называется величина, значение которой не изменяется в процессе исполнения алгоритма, а остается одним и тем же, указанным в тексте алгоритма.

**Переменной** называется величина, значение которой меняется в процессе исполнения алгоритма.



- Тип выражения определяется типами входящих в него величин, а также выполняемыми операциями.
- В языке Pascal тип величины задают заранее, т.к. все переменные, используемые в программе, должны быть объявлены в разделе описания с указанием их типа.

# Различают переменные следующих простых типов :

- ***Целые*** (Integer, Byte, ShortInt, Word, LongInt),
- ***Вещественные*** (Real, Comp, Double, Single, Extended),
- ***Логический*** (Boolean),
- ***Символьный*** (Char),
- ***Перечисляемый***,
- ***Диапазонный***.

# Иерархия типов в языке Pascal



## Некоторые типы данных в языке Паскаль

Название	Обозначение	Допустимые значения	Область памяти
Целочисленный	integer <sup>1</sup>	-32 768 .. 32 767	2 байта со знаком
Вещественный	real	$\pm(2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{+38})$	6 байтов
Символьный	char	Произвольный символ алфавита	1 байт
Строковый	string	Последовательность символов длиной меньше 255	1 байт на символ
Логический	boolean	true и false	1 байт

В вещественном числе целая часть от дробной отделяется точкой, при этом перед точкой и после неё должно быть, по крайней мере, по одной цифре. Пробелы внутри числа недопустимы.

Описание переменной:

**Var <имя переменной > : <тип>;**  
**(идентификатор)**

Пример описания:

```
Var    D, C, N : Integer;  
      LogPer : Boolean;  
      A, B : Real;  
      K : Char;
```

- Обмен информацией с ЭВМ предполагает использование определенных средств ввода-вывода. В ЭВМ основным средством ввода является клавиатура, вывода — дисплея.
- **Процедура, которая в режиме диалога с клавиатуры присваивает значение для переменной величины, называется процедурой ввода.**

# Процедура ввода

**Read(список переменных);**

**или**

**ReadLn(список переменных);**

*с переходом на следующую строку*

Примеры ввода данных с помощью процедуры ReadLn:

ReadLn(A, B, C);

ReadLn(X);

ReadLn(LogPer);

# Структура записи раздела операторов

**Begin**

{раздел операторов}

**End**

Операторные скобки.

Они необходимы для сложного оператора, состоящего из нескольких простых

■ = Оператор  
■ = присваивания

*Замечание.*

*Если после выполнения сложного оператора следует конец программы, то после скобки End нужно поставить точку, иначе точку с запетой.*

Пример

**P:=1**

**S:=0**



# Структура программы на ПАСКАЛЕ

```
program <имя программы>;  
  const <список постоянных значений>;  
  var <описание используемых переменных>;  
begin <начало программного блока>  
  <оператор 1>;  
  <оператор 2>;  
  ...  
  <оператор n>  
end.
```

**Операторы** — языковые конструкции, с помощью которых в программах записываются действия, выполняемые над данными в процессе решения задачи.

Точка с запятой служит разделителем между операторами, а не является окончанием соответствующего оператора.

Процедура, которая выводит содержимое переменных на экран, называется процедурой вывода на экран.

**Write** (список констант и/или переменных, разделенных запятой);

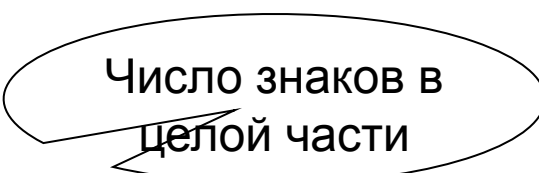
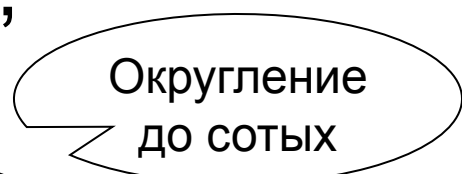
**WriteIn**(список констант и/или переменных, разделенных запятой);

*с переходом на следующую строку*

Например **Текст заключается в апострофы**

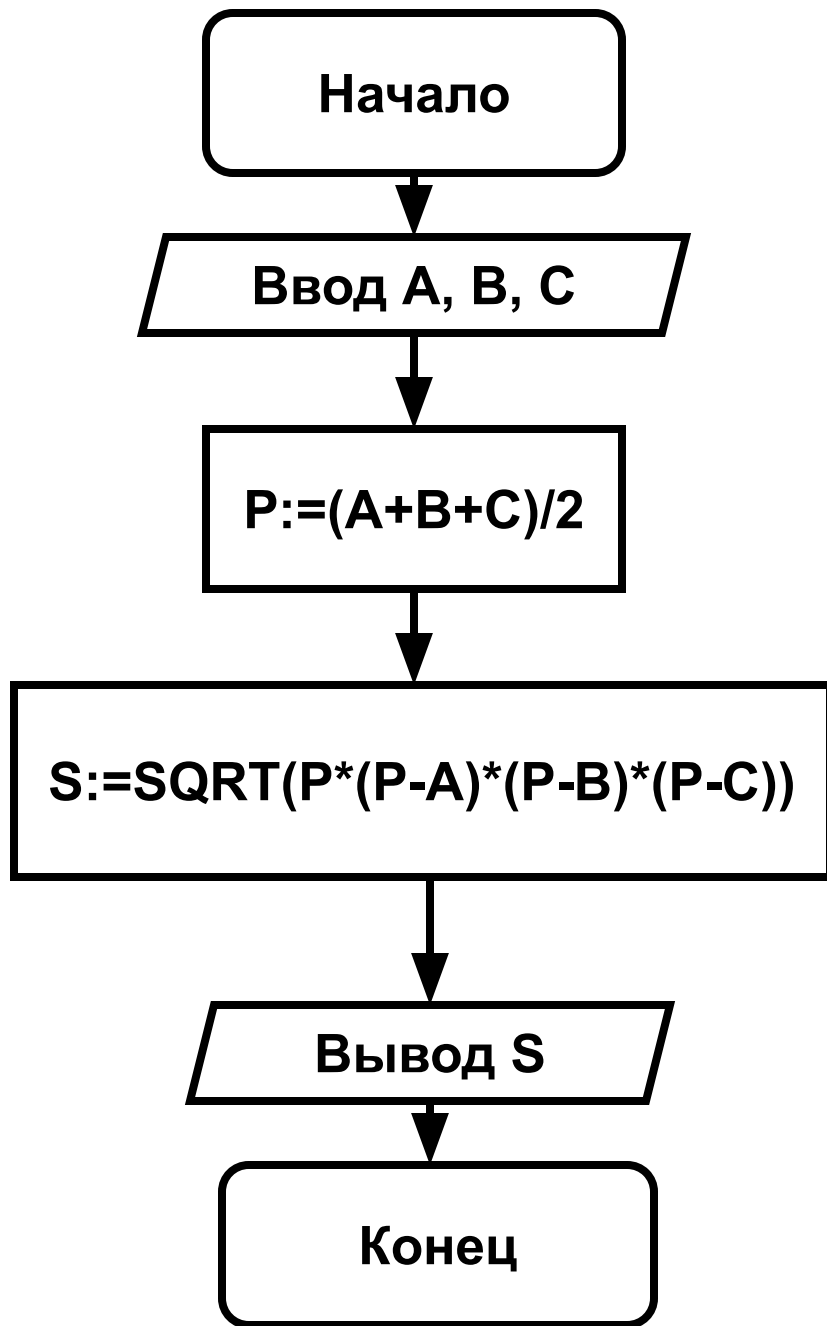
WriteIn ('**введите числа А, В, С**');

WriteIn ('S=', S:4:**2**);



## Служебные слова языка Паскаль

Служебное слово языка Паскаль	Значение служебного слова
<b>and</b>	и
<b>array</b>	массив
<b>begin</b>	начало
<b>do</b>	выполнить
<b>else</b>	иначе
<b>for</b>	для
<b>if</b>	если
<b>of</b>	из
<b>or</b>	или
<b>procedure</b>	процедура
<b>program</b>	программа
<b>repeat</b>	повторять
<b>then</b>	то
<b>to</b>	до (увеличивая до)
<b>until</b>	до (до тех пор, пока)
<b>var</b>	переменная
<b>while</b>	пока



```
PROGRAM treygolnik1;
```

```
VAR A,B,C: integer; P,S: real;
```

```
BEGIN
```

```
  WRITELN ('введите стороны  
треугольника A,B,C');
```

```
  READLN (A,B,C);
```

```
  P:=(A+B+C)/2;
```

```
  S:=SQRT(P*(P-A)*(P-B)*(P-C));
```

```
  WRITELN ('площадь  
треугольника  
равна_',S:4:2);
```

```
END.
```

## Этап 2. КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

1. Запуск выбранной среды.  
программирования.
2. Набор текста программы.
3. Запуск программы на выполнение.
4. Анализ результатов (тестирование)

№	Набор входящих данных	Результат	Подтверждение
1	3,5,4	6	истина
2	7,5,7	16,35	истина
3	4,4,4	6,93	истина

5. Сохранение.

# Домашняя работа

П. 2.1 конспект, задача 1 в тетради оформить по всем этапам.

Задача 1. Даны координаты вершин треугольника. Найти его площадь.  
Составьте блок-схему алгоритма решения поставленной задачи.  
Напишите программу.  
Составьте тест для проверки правильности работы программы.

В электронной версии приложения к учебнику, есть ссылка на скачивание программы