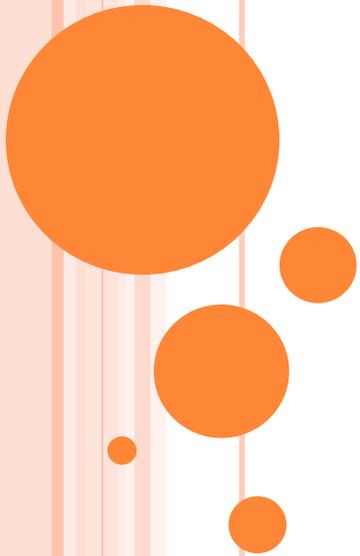


# *Программирование*

*9 класс*



Назначение  
*программирования* – разработка  
программ управления  
компьютером с целью решения  
различных информационных  
задач.



# Программирование делиться на

- **Системное программирование** — разработка системного программного обеспечения: операционных систем, утилит и т.д.
- **Прикладное программирование** создание прикладных программ: редакторы, табличные процессоры, игры, обучающие программы и т.д.



Для составления программ существуют разнообразные **языки программирования** - это фиксированная система обозначений для описания алгоритмов и структур данных.



Для создания и использования на компьютере программы, написанной на языке программирования, используется **Система программирования** — это программное обеспечение компьютера, предназначенное для разработки, отладки и исполнения программ, записанных на определенном языке программирования (Паскаль, Бейсик и т.д.)



Разработка любой программы начинается с построения алгоритма решения задач. Такие алгоритмы называют **алгоритмами работы с величинами**.

В качестве исполнителя рассматривается — компьютер, оснащенный системой программирования на определенном языке.

Компьютер-исполнитель работает с определенными данными по определенной программе. Данные — это множество величин.



Существует три основных типа величин, с которыми работает компьютер:

*числовой, символьный и логический.*

## *Числовые величины*

Переменные

Константы



***Константы*** записываются в алгоритмах своими десятичными значениями и остаются неизменными в течение работы программы.

***Переменные*** обозначаются символическими именами





***Язык  
программирования  
Паскаль***

Язык Паскаль разработан в 1971 году и назван в честь Блеза Паскаля – французского ученого, изобретателя механической вычислительной машины.

Автор Языка Паскаль – швейцарский профессор *Никлаус Вирт*.

*Паскаль* – это универсальный язык программирования, позволяющий решать самые разнообразные задачи обработки информации



# *ПРОЦЕДУРЫ ВЫВОДА WRITE И WRITELN*

(переводится – «пиши» и «пиши строку»)

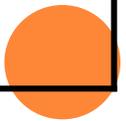
С помощью данных операторов изображают на экране ту или иную информацию, состоящую из СИМВОЛОВ.

Выводить на экран можно не только числа, но и результаты вычисления арифметических выражений, а также тексты, которые, в отличие от чисел и выражений, нужно брать в одинарные кавычки.



# Примеры:

<i>Как пишем</i>	<i>Что видим</i>
Write(-500)	-500
Write(2*2-1)	3
Write('Хорошо!')	Хорошо!



Один оператор Write может выводить сразу несколько элементов. Элементы нужно отделять друг от друга запятыми.

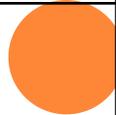
Все элементы выводятся в одну строку вплотную друг к другу.

На экране отображаются только те пробелы, которые встречаются внутри кавычек.



## Примеры:

<i>Как пишем</i>	<i>Что видим</i>
<code>Write('Это',4+4,'Кошек')</code>	Это8Кошек
<code>Write('Это ',4+4,' кошек')</code>	Это 8 кошек
<code>Write('16+17=',16+17)</code>	16+17=33
<code>Write(3+2,' ',4)</code>	5 4
<code>Write(3+2,4)</code>	54
<code>Write('125+1',5+1,'=',120+21)</code>	125+16=141



Правила записи и выполнения оператора `WriteLn` те же, что и у `Write`, с одним исключением – после его выполнения следующий оператор `Write` или `WriteLn` печатает свою информацию с начала следующей строки, а после выполнения оператора `Write` продолжает печатать в той же.

Оператор `WriteLn` можно использовать просто для перевода курсора в начало следующей строки.



Программы на Паскале содержат следующие «знаки препинания»:

- Служебные слова BEGIN и END;
- Точка с запятой;
- Точка.



**BEGIN** (переводится – «начало») – ставят в начале программы, чтобы было видно, откуда она начинается.

**END** (переводится – «конец») – с точкой ставится в конце программы, чтобы было видно, где она заканчивается.

**Точкой с запятой** отделяют операторы друг от друга.

Служебные слова **BEGIN** и **END** от операторов точкой с запятой не отделяются.



Программу можно записывать и в строку, и в столбец.

Служебные слова и операторы могут быть записаны любыми буквами (заглавными или строчными, а также любым шрифтом).

Программа на Паскале может содержать комментарии, взятые в фигурные скобки, которые не влияют на выполнение программы.



## Примеры:

*Программа:* Begin Write('АМа'); Write('ЗОНКа'); End.

*Результат:* АМаЗОНКа

*Программа:* Begin Write('АМа'); WriteLn('ЗОНКа'); End.

*Результат:* АМаЗОНКа

*Программа:* Begin WriteLn('Ама'); Write('Зонка'); End.

*Результат:*  
Ама  
Зонка

*Программа:* Begin WriteLn('Ама'); Write('Зонка'); End.

*Результат:*  
Ама  
Зонка



## ОПЕРАТОР ПРИСВАИВАНИЯ.

При выполнении оператора присваивания компьютер «в уме» вычисляет правую часть и присваивает вычисленное значение переменной, стоящей в левой части.

Обозначение оператора присваивания

:=



## Пример:

Begin

a:=2\*3+4;

b:=a;

y:=a+b+1;

Write('y=',y)

End.

Ответ: y=21

Замечание. Если переменная принимает новое значение, то старое значение автоматически стирается

## ОПИСАНИЕ ПЕРЕМЕННЫХ

Описание переменных начинается со служебного слова VAR (переводится – «переменная»), которое записывается выше Begin.

После VAR записываются имена всех переменных, встречающихся в программе с указанием через двоеточие типа значений, которые каждая переменная имеет право принимать.



# ТИПЫ ЗНАЧЕНИЙ ПЕРЕМЕННЫХ

Тип	Перевод	Диапазон принимаемых значений
Integer	целый	целые числа от - 32 768 до 32 767
LongInt	длинное целое	целые числа от - 2 147 483 648 до 2 147 483 647
Byte		целые числа от 0 до 255
Real	Вещест- венный	целые и дробные числа

Для того, чтобы Паскаль выводил вещественные числа в понятном виде, нужно в оператор вывода `WriteLn` дописывать формат численного значения переменной:

`WriteLn(x:n:m),`

где  $n$  – натуральное число, показывающее сколько символов, включая целую часть, дробную часть, знак и десятичную точку, должно занимать все изображение числа;  $m$  – натуральное число, показывающее количество символов после десятичной точки.



## Пример:

```
Var a,b:Integer;
```

```
    c:Real;
```

```
Begin
```

```
    a:=6;
```

```
    b:=7;
```

```
    c:=b/a;
```

```
    WriteLn('c=',c:5:2);
```

```
End.
```

ОТВЕТ: c=1,17

