

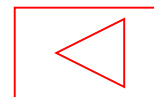
# Введение в программирование на языке Pascal

Язык программирования Pascal был создан швейцарским профессором Никлаусом Виртом в 1971 году.

- Структура программы
- Блок-схема
- Основные операторы (линейный алгоритм)
- Операции с целыми числами div, mod
- Ветвление (условный оператор)
- Циклы
- Массивы
- Строковые переменные. Операции со строками
- Модуль CRT
- Правильное оформление программы
- Программное и информационное обеспечение

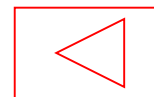
Автор: Михеева О. Ф., школа №359

[Просмотр презентации по щелчку](#)



# Структура программы

**Program** name;            {Заголовок программы}  
**Var**                         {Заголовок раздела описаний}  
x,y,z: **real**;                {имена ячеек памяти для дробных чисел }  
i,j,k: **integer**;            {имена ячеек памяти для целых чисел }  
**Begin**                        {Начало тела программы}  
\*\*\*\*\*  
\*\*\*\*\*                        {Операторы программы}  
\*\*\*\*\*  
**End.**                         {Конец тела программы}



# Основные операторы

(линейный алгоритм)

`x:=2.5;`

`i:=1;`      *{присвоение значений ячейкам памяти }*

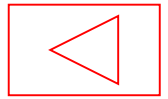
`c:=a+b;`

**WriteIn**('Здравствуйте'); *{вывод на экран текста}*

**WriteIn**('Значение x= ', x); *{вывод на экран значения ячейки **x** с комментарием}*

**ReadIn**(a);      *{считывание числа с клавиатуры и запись его в ячейку **a** }*

## Пример 1



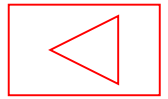
*Написать программу, которая складывает любых два целых числа и выводит результат на экран. Числа вводятся с клавиатуры.*

```
Program summa;  
Var a,b,c: integer;  
begin  
  writeln('Введите 1-ое число');  
  readln(a);  
  writeln('Введите 2-ое число');  
  readln(b);  
  c:=a+b;  
  Writeln('Сумма чисел=', c);  
end.
```

[Отладка программы](#)

*[Просмотр презентации по щелчку](#)*

## Пример 2

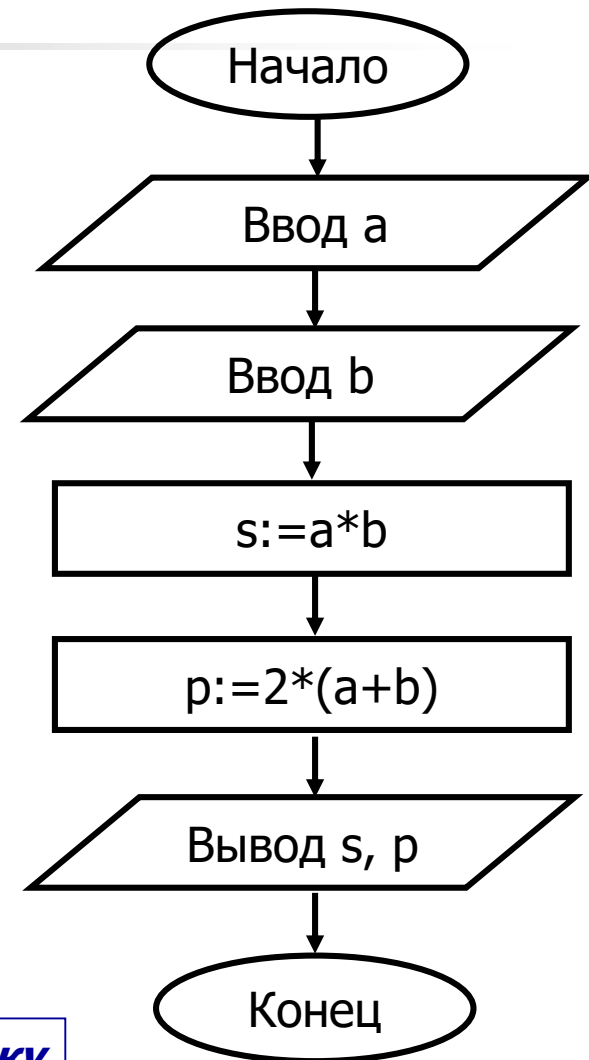


Написать программу для вычисления площади и периметра прямоугольника. Значения сторон прямоугольника вводятся с клавиатуры. Нарисовать блок-схему алгоритма программы.

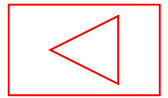
```
Program schet;  
var a,b,s,p:real;  
begin  
  writeln('Введите длину прямоугольника');  
  readln(a);  
  writeln('Введите ширину прямоугольника ');  
  readln(b);  
  s:=a*b;  
  p:=2*(a+b);  
  writeln('Площадь= ', s, ' периметр= ',p);  
end.
```

Отладка программы

[Просмотр презентации по щелчку](#)



## Задача 1



*Написать программу, которая считает стоимость покупки, если известно, что купили 2 кг груш и 3 кг картофеля. Цена 1 кг груш - 50 руб.; 1 кг картофеля - 20 руб. Числа вводятся с клавиатуры.*

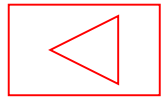
- (1) **end.**
- (2) **readln**(ng,nk);
- (3) **st:=cg\*ng+ck\*nk;**
- (4) **var** cg,ck,ng,nk,st: **integer**;
- (5) **writeln**('Стоимость покупки =', st,' рублей' );
- (6) **begin**
- (7) **writeln**('Введите цену груш и цену картофеля ');
- (8) **readln**(cg,ck);
- (9) **writeln**('Введите количество кг груш и картофеля');
- (10) **Program** рокуп;

Восстановить правильный  
порядок операторов

Отладка программы

***Просмотр презентации по щелчку***

## Вопрос:

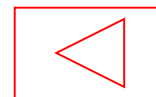


Какие действия будут выполняться при запуске следующей программы?

```
Program vopros;  
var n,m,k: integer;  
begin  
writeln('Сколько тебе лет?');  
readln(n);  
writeln('В каком классе ты учишься?');  
readln(m);  
writeln('В какой школе ты учишься?');  
readln(k);  
writeln('Тебе ', n, ' лет, ты учишься в ', m, ' классе ', k, ' школы');  
end.
```

[Отладка программы](#)

*[Просмотр презентации по щелчку](#)*



## Операции с целыми числами **div , mod**

**div** – операция, вычисляющая результат деления нацело;

**mod** – операция, результатом которой является остаток от деления нацело.

**Вычислить значения переменных a и b после выполнения фрагмента программы:**

**a:=2599;**

**b:=(a mod 10)\*1000 + 26;**

**a:=(b div 10) mod 100;**

**b:=9\*1000 + 26= 9026;**

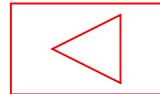
**a:=902 mod 100 =2;**

**Примеры:**

a	b	a div b	a mod b
10	5		
12	11		
17	3		
100	10		
111	100		
58	25		
324	10		



## Задачи с использованием операций `div` , `mod`



**1. Общее количество конфет и количество конфет в одной коробке вводится с клавиатуры. Определить, сколько полных коробок конфет получилось и сколько конфет осталось?**

**Program konf;**

**var** n, k, nkor, nkonf : integer;

**begin**

**writeln**('Введите общее количество конфет и количество конфет в 1 коробке');

**readln**(n, k);

nkor:= n **div** k;

nkonf:= n **mod** k;

**Writeln**('Количество коробок =', nkor, ', осталось ', nkonf, ' конф.');

**end.**

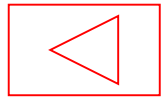
[Отладка программы 1](#)

**2. Даны числитель и знаменатель неправильной дроби ( $m > n$ ). Вывести на экран смешанную дробь. [Программа 2.](#)**

**3. Дано трехзначное число, где  $a, b, c$  – цифры числа. Получить число с обратным порядком цифр. [Программа 3.](#)**

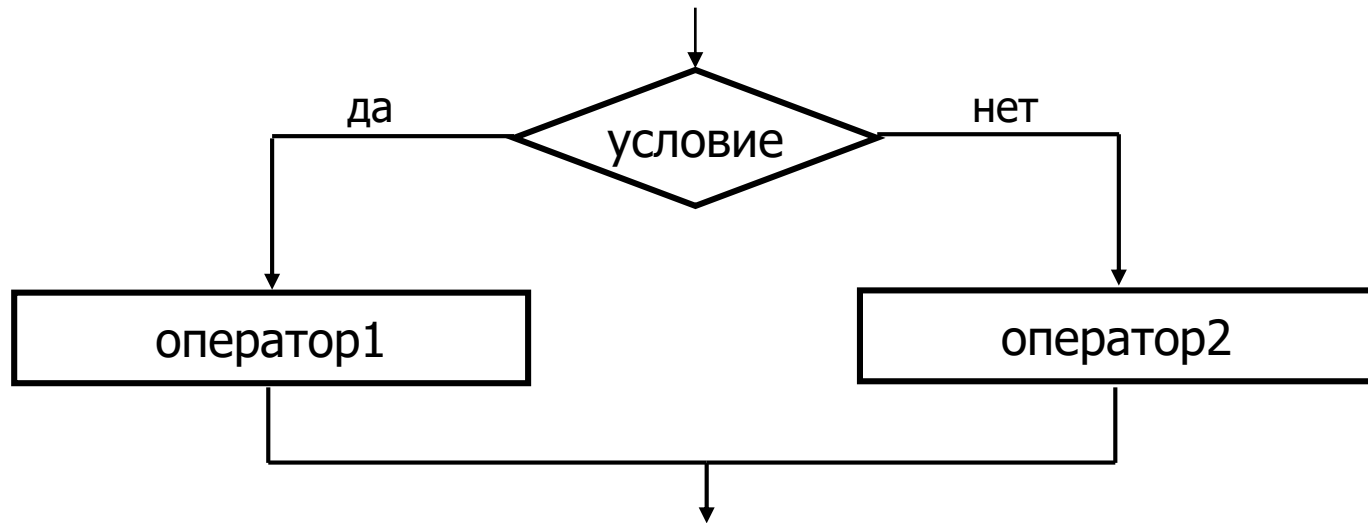
# Ветвление

(условный оператор)

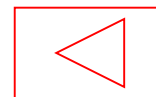


## А) Полная форма

**if** <условие > **then** <оператор1> **else** <оператор2>;

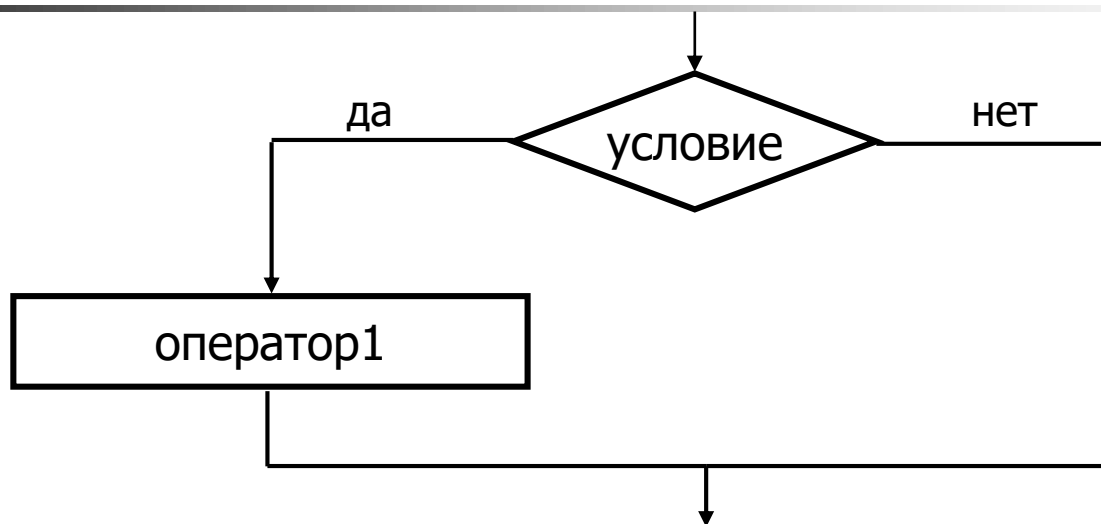


**Пример:** if  $x < 0$  then writeln('число  $x$  - отрицательное')  
else writeln('число  $x$  – больше или равно нулю');



## Б) Неполная форма

**if** <условие > **then** <оператор1> ;



**Пример:** **if**  $x \neq 0$  **then**  $y := 1/x$ ;

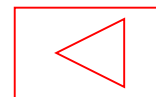
Если при соблюдении условия должны выполняться несколько операторов, их заключают в **операторные скобки**.

**begin** .... **end**;

**Пример:** **if**  $x \geq 0$  **then** **begin**  $y := x$ ;  $x := x - 1$ ; **end**

**else** **begin**  $y := -x$ ;  $x := x + 1$ ; **end**;

*[Просмотр презентации по щелчку](#)*



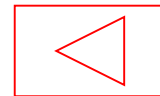
## Задача1

*Проверить правильность ответа на вопрос: «Назовите год основания С–Петербурга» («Вы правы» или «Вы ошиблись, город основан в 1703 году».*

```
Program test;  
var n: integer;  
begin  
  writeln('Назовите год основания С–Петербурга ');  
  readln(n);  
  if n=1703 then  
    Writeln('Вы правы') else  
    Writeln('Вы ошиблись, город основан в 1703 году' );  
end.
```

[Отладка программы](#)

*[Просмотр презентации по щелчку](#)*



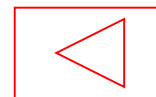
## Задача2

Два целых числа вводятся с клавиатуры. Вывести на экран сообщение: «1-ое число больше 2-го», «1-ое число меньше 2-го» или «Числа равны»

```
Program chisla;  
var a,b: integer;  
begin  
  writeln('Введите 1-ое число');  
  readln(a);  
  writeln('Введите 2-ое число');  
  readln(b);  
  if a>b then  
    writeln('Первое число больше второго');  
  if a<b then  
    writeln('Первое число меньше второго');  
  if a=b then  
    writeln('Числа равны');  
end.
```

[Отладка программы](#)

*[Просмотр презентации по щелчку](#)*



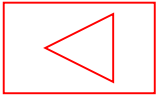
## Задача 3

Получить случайное целое число в диапазоне от 1 до 1000 с помощью генератора случайных чисел. Определить, кратно ли число 3.

```
Program chislo;  
var n: integer;  
Begin  
Randomize;  
n:=Random(1000) +1;  
if n mod 3 = 0 then  
Writeln(`Число ', n, ` кратно 3') else  
Writeln(`Число ', n, ` делится на 3 с остатком');  
end.
```

[Отладка программы](#)

*[Просмотр презентации по щелчку](#)*



## Сложные условия

В логических условиях используются следующие операции отношения:

больше	больше или равно	равно	не равно	меньше	меньше или равно
>	>=	=	<>	<	<=

Сложные логические условия строятся из простых с помощью логических операций (связок):

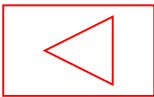
**and** – логическое «и»

**or** – логическое «или»

### Примеры:

```
if (x>3) and (x<5) then writeln ('число x в диапазоне от 3 до 5' );  
    { одновременное выполнение двух условий }  
if (x<3) or (x>5) then writeln ('число x вне диапазона от 3 до 5' );  
    { выполнение хотя бы одного из условий }
```

## Задача 4



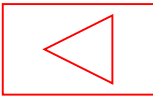
По введенному № дня недели вывести сообщение: «Это рабочий день» ( $n=1-5$ ) или «Это выходной день» ( $n=6,7$ ).

```
Program den;  
var n: integer;  
begin  
  writeln('введите № дня недели');  
  readln(n);  
  if (n>=1) and (n<=5) then  
    writeln('Это рабочий день');  
  if (n=6) or (n=7) then  
    writeln('Это выходной день');  
end.
```

[Отладка программы](#)



# Ветвление на несколько направлений



Оператор **CASE** предназначен для организации выбора одного из нескольких вариантов дальнейшего хода программы.

**Case** <переменная> **of**

Перечень 1 значений переменной : оператор 1;

Перечень 2 значений переменной : оператор 2;

-----

Перечень n значений переменной : оператор n

**Else** оператор n+1;

**End;**

Пример:

**Case n of**

**1, 2, 12 : writeln('Зима');**

**3..5 : writeln('весна');**

**6..8 : writeln('лето');**

**9..11: writeln('осень')**

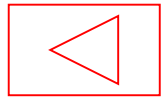
**Else writeln('номер месяца д. б. от 1 до 12');**

**End;**

[Отладка программы](#)

*[Просмотр презентации по щелчку](#)*

# Циклический алгоритм



## А) Цикл с постусловием

**Repeat**

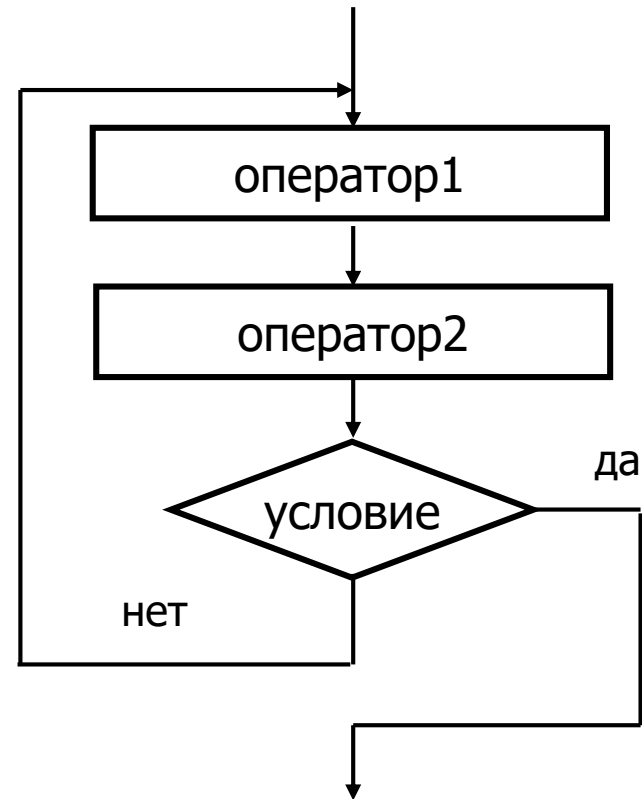
----- { тело  
----- цикла }

**until** <условие выхода>;

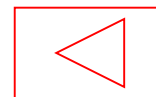
Пример: Вводить числа с клавиатуры  
до тех пор, пока их сумма  
не превысит 100.

```
S:=0;  
Repeat  
  Writeln('введите число');  
  Readln(x);  
  S:=S+x;  
Until S>100;  
Writeln('сумма чисел=',S);
```

[Отладка программы](#)



***[Просмотр презентации по щелчку](#)***



## Б) Цикл с предусловием

```
While <условие входа> do  
Begin  
----- { тело  
-----   цикла}  
end;
```

Пример: Угадать «задуманное» число  
в интервале от 1 до 5.

N:=4; {задуманно число 4}

M:=0;

While M<>N do

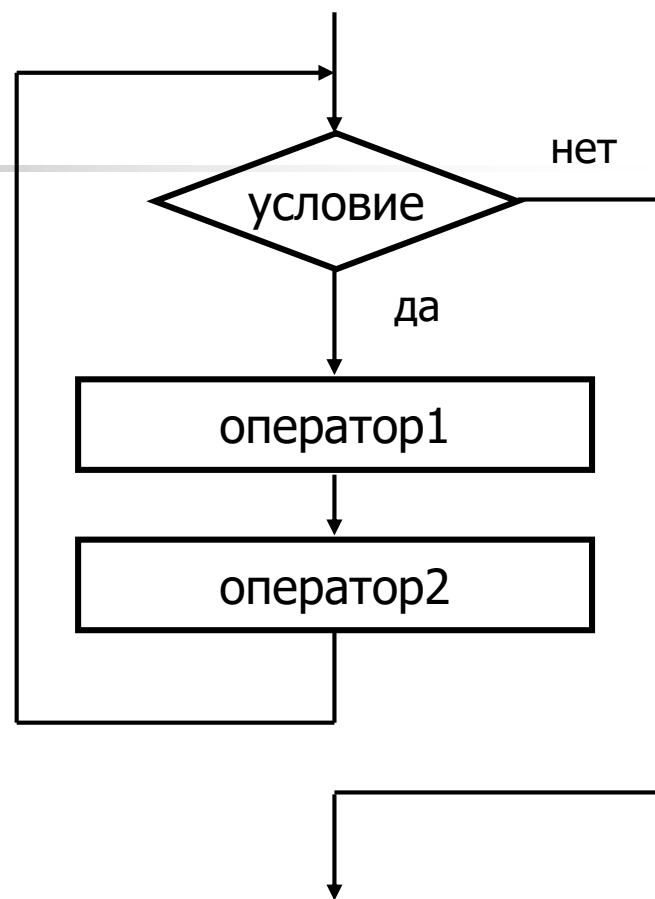
Begin

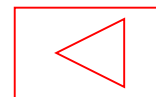
Writeln(' Отгадайте число от 1 до 5');

Readln(M);

End;

Writeln('вы угадали, число=', M);





## В) Цикл с параметром

```
For <параметр>:= <граница1> to <граница2> do  
Begin  
----- { тело  
-----   цикла}  
end;
```

Пример: Найти сумму натуральных чисел от 1 до 10.

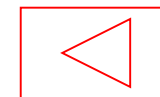
```
S:=0;
```

```
For i:=1 to 10 do
```

```
Begin S:=S+i; end;
```

```
Writeln('сумма чисел=',S);
```

# Массив -



это набор переменных одного и того же типа, в котором каждый элемент имеет номер, а все элементы – общее имя.

## А) Одномерный массив (столбец или строка)

$a_1, a_2, a_3, \dots, a_{20}$

Описание: **var a: array[1..20] of real;**

### Задача 1

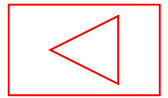
Ежедневные значения температур (в течение месяца) вводятся с клавиатуры. *Определить среднее значение температуры за месяц.*

```
Program pogoda;
Var t: array[1..31] of integer;
i, st: integer; sr: real;
Begin
Writeln('Введите температуру');
For i:=1 to 31 do
begin
Writeln(i, '-го числа');
Readln(t[i]);
st:=0;
For i:=1 to 31 do
st:=st+t[i];
sr:=st/31;
Writeln( 'Средняя температура месяца=', sr:6:2);
end.
end;
```

*Дополнить задачу вводом количества дней в месяце*

[Просмотр презентации по щелчку](#)

## Задача 2



**Определить максимальный элемент массива из 15 вещественных чисел, введенных с клавиатуры.**

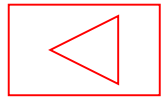
```
Program maxel;  
Const n=15;  
Var a: array[1..n] of real;  
i: integer; max: real;  
Begin  
Writeln('Введите ',n,' элементов массива');  
For i:=1 to n do  
  Readln(a[i]);  
  max:=a[1];  
For i:=1 to n do  
  if a[i]>max then max:=a[i];  
Writeln('Максимальный элемент массива', max);  
end.
```

Изменить программу следующим образом:  
найти минимальный элемент массива из 10 целых чисел, полученных генератором случайных чисел в диапазоне от 0 до 50

[Отладка программы](#)

*[Просмотр презентации по щелчку](#)*

## Б) двумерный массив (таблица, матрица)



$a_{1,1}$     $a_{1,2}$     $a_{1,3}$    ...   ... $a_{1,m}$

$a_{2,1}$     $a_{2,2}$     $a_{2,3}$    ...   ... $a_{2,m}$

-----

$a_{n-1,1}$     $a_{n-1,2}$    ...   ...    $a_{n-1,m}$

$a_{n,1}$     $a_{n,2}$  ...   ...    $a_{n,m}$

### Описание

a: **array**[1.. 10, 1..5] **of integer**;

**Ввод с клавиатуры** (по строкам)

**For** i:=1 to n **do**

**For** j:=1 to m **do readln**(a[i,j]);

**Задача:** Заполнить двумерный массив (5 x 10) единицами и напечатать.

**Program** chisla;

**Var** i,j : integer; a: **array**[1.. 5, 1..10] **of integer**;

**Begin**

**For** i:=1 to 5 **do**

**For** j:=1 to 10 **do** a[i,j]:=1;

**For** i:=1 to 5 **do begin**

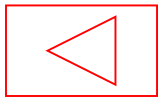
**For** j:=1 to 10 **do** write(a[i,j], ' '); writeln; **end**;

**End.**

[Отладка программы](#)

*[Просмотр презентации по щелчку](#)*

# Строковые переменные



Описание: `var s: string; {255 символов}`  
`ss : string[15]; {15 символов}`

## Операции со строками

### 1. Сложение (конкатенация)

**+**

**Пример:** `s1:='школа'; s2:='№'; s3:='359'; ss:=s1+' '+s2+ s3; writeln(ss);`

### 2. Получение длины строки

**Length**

**Пример:** `n:= Length('длинношеее'); writeln(n); {напечатано число 10}`

### 3. Копирование части строки

**Copy**

**Пример:** `s1:= Copy('победа', 3,4); writeln(s1); {напечатано слово «беда»}`

### 4. Удаление части строки

**Delete**

**Пример:** `s:='победа'; Delete(s,1,3); writeln(s); {напечатано слово «еда»}`

### 5. Вставка подстроки

**Insert**

**Пример:** `s:='Иванов'; Insert(' - победитель', s, 7); writeln(s);`  
`{напечатано «Иванов - победитель»}`

### 6. Поиск

**Pos**

**Пример:** `s1:='оловянный'; s2:='серебряный'; n1:= Pos('нн', s1);`  
`n2:= Pos('нн', s2); writeln(n1, ' ', n2); {будет напечатано 6, 0}`



# Модуль CRT –

работа с окнами, управление цветом в текстовом режиме



**Uses CRT;** {после заголовка программы — подключение модуля}

**Window(x1,y1,x2,y2);** { процедура открытия окна, где  $x1,y1$  – координаты левого верхнего угла окна, а  $x2,y2$  – координаты правого нижнего угла окна. Курсор устанавливается в левый верхний угол окна ( $x1,y1,x2,y2$  – целые числа)}

**TextBackGround(k);** { задание цвета фона}

**ClrScr;** {очистка экрана или текущего окна цветом фона}

**TextColor(n);** {задание цвета шрифта}

**GotoXY(x,y);** {перемещение курсора в позицию с координатами (x,y)относительно полного экрана или текущего окна}

**Delay(m);** {Задержка выполнения программы на m миллисекунд}

k	0	1	2	3	4	14	15
Цвет	Черный	Синий	Зеленый	Голубой	Красный	Желтый	Белый

Например,

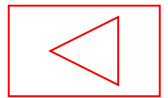
**Window(1,1,80,25);**

**TextBackGround(2);**

**ClrScr;**

[Просмотр презентации по щелчку](#)

## Задача

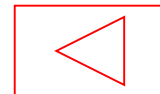


*На стандартном цветном экране расположить в центре окно другого цвета, в котором организовать диалог с вопросом о фамилии и имени пользователя, используя цветной шрифт.*

```
Program test;  
Uses CRT;  
Var s:string[40]; {Строковая переменная}  
Begin  
Window(1,1,80,25);  
TextBackGround(7);  
ClrScr;  
Window(20,6,60,16);  
TextBackGround(1);  
ClrScr;  
TextColor(14);  
writeln('Введите фамилию и имя');  
readln(s);  
Delay(1000);  
ClrScr;  
end.
```

[Отладка программы](#)

*[Просмотр презентации по щелчку](#)*

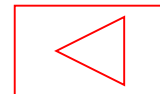


## Правильное оформление программы

предполагает использование:

---

- осмысленных имен переменных;
- отступов при записи блоков программы (использовать сдвиг вправо нажатием клавиши **TAB**);
- комментариев (использовать **{....}**, **//** )



# Программное и информационное обеспечение

---

Учебная система программирования **Pascal ABC** и электронный задачник, авторы: С. С. Михалкович, М. Э. Абрамян, 2002-2007

## Литература

1. Зубов В. С. Программирование на языке Turbo Pascal (версии 6.0 и 7.0). — М.: Информационно-издательский дом «Филинь», 1997.
2. Культин Н. Б. Turbo Pascal в задачах и примерах. — СПб. : БХВ-Петербург, 2000.