

# Основные элементы языка программирования Паскаль

```
program bkt;  
uses crt;  
var b,k,t:integer;  
begin  
  clrscr;  
  for b:=0 to 20 do  
    for k:=0 to 20 do  
      begin  
        t:=100*(b+k);  
        if t < 200 then  
          write('b=',b,' k=',k,' t=',t);  
        k:=k+1;  
      end;  
    end;  
  end.
```



# Алфавит ЯП

это конечный набор знаков, состоящий из букв, десятичных и шестнадцатеричных цифр, специальных символов.

- В качестве букв в языке Pascal используются прописные и строчные буквы латинского алфавита (A..Z, a..z) и знак подчеркивания (\_).

- Специальные символы:

+ - \* / < > = ; # ' , . : [ ] { } \$ ( ) ^ @.



# Составные символы

Комбинации специальных символов

`:=` присваивание

`<=` меньше или равно

`<>` не равно

`>=` больше или равно

`..` диапазон значений

`(* *)` `{ }` комментарии

**Задание 1.** Укажите буквы, символы, составные символы:

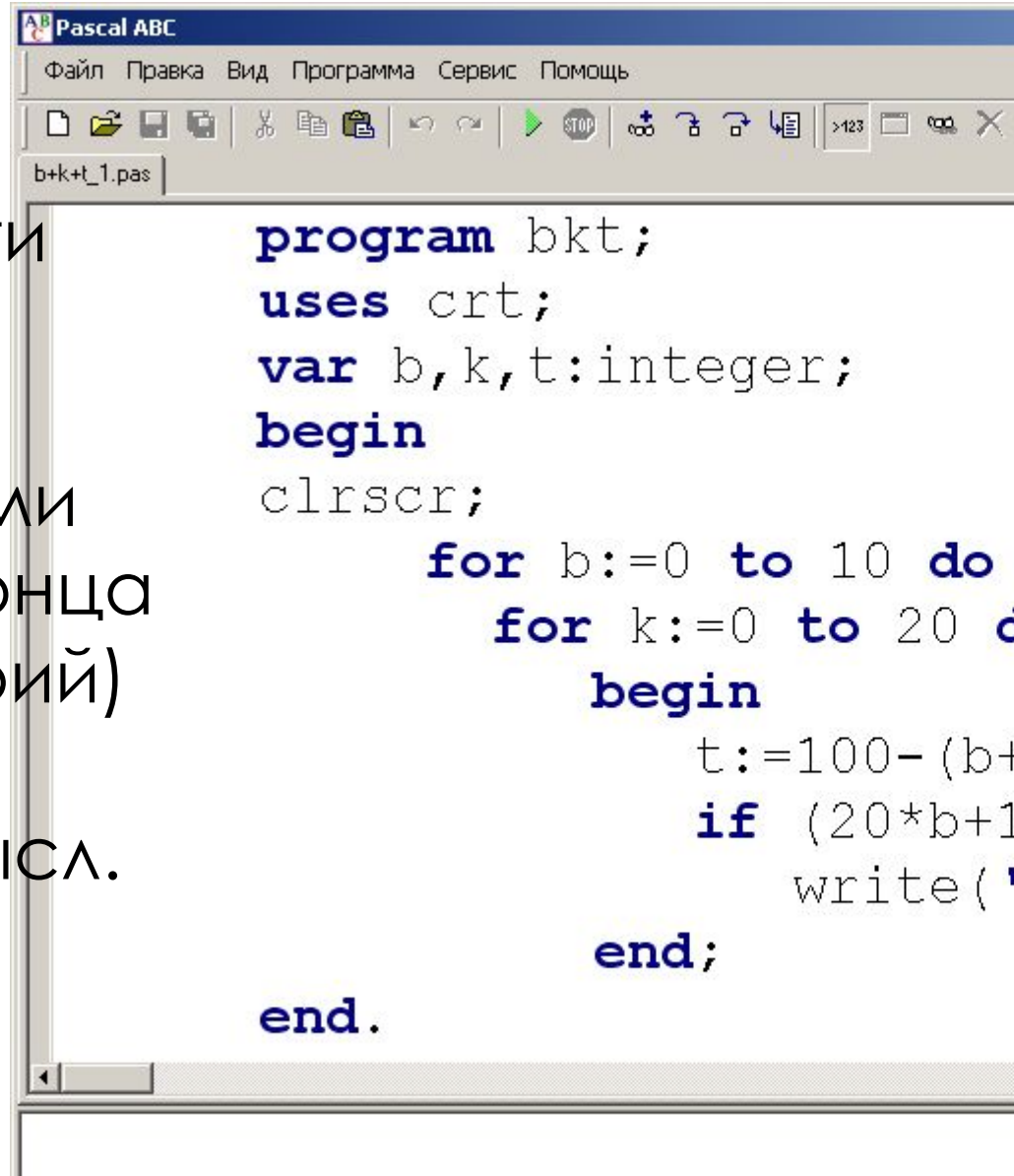
`*`, `X`, `)`, `!`, `<>`, `^`, `$`, `+`, `?`, `F`, `{}`, `t`, `:=`

**Задание 2.** Какие из комментариев неправильны:

- а) {Программа вычисляет логарифм введенного числа}
- б) (\*Это тоже комментарий\*)
- в) {{Комментарий в комментарии}}
- г) {(\*И это комментарий в комментарии\*)}
- д) (\*{Еще один вариант}\*)
- е)>(\*(\*Самый последний вариант\*)\*)

# Слова

- неделимые последовательности знаков алфавита, отделенные друг от друга разделителями (пробел, символ конца строки, комментарий) и несущие определенный смысл.



```
program bkt;  
uses crt;  
var b,k,t:integer;  
begin  
  clrscr;  
  for b:=0 to 10 do  
    for k:=0 to 20 do  
      begin  
        t:=100-(b+k);  
        if (20*b+1) < t then  
          write(' ');  
      end;  
end.
```



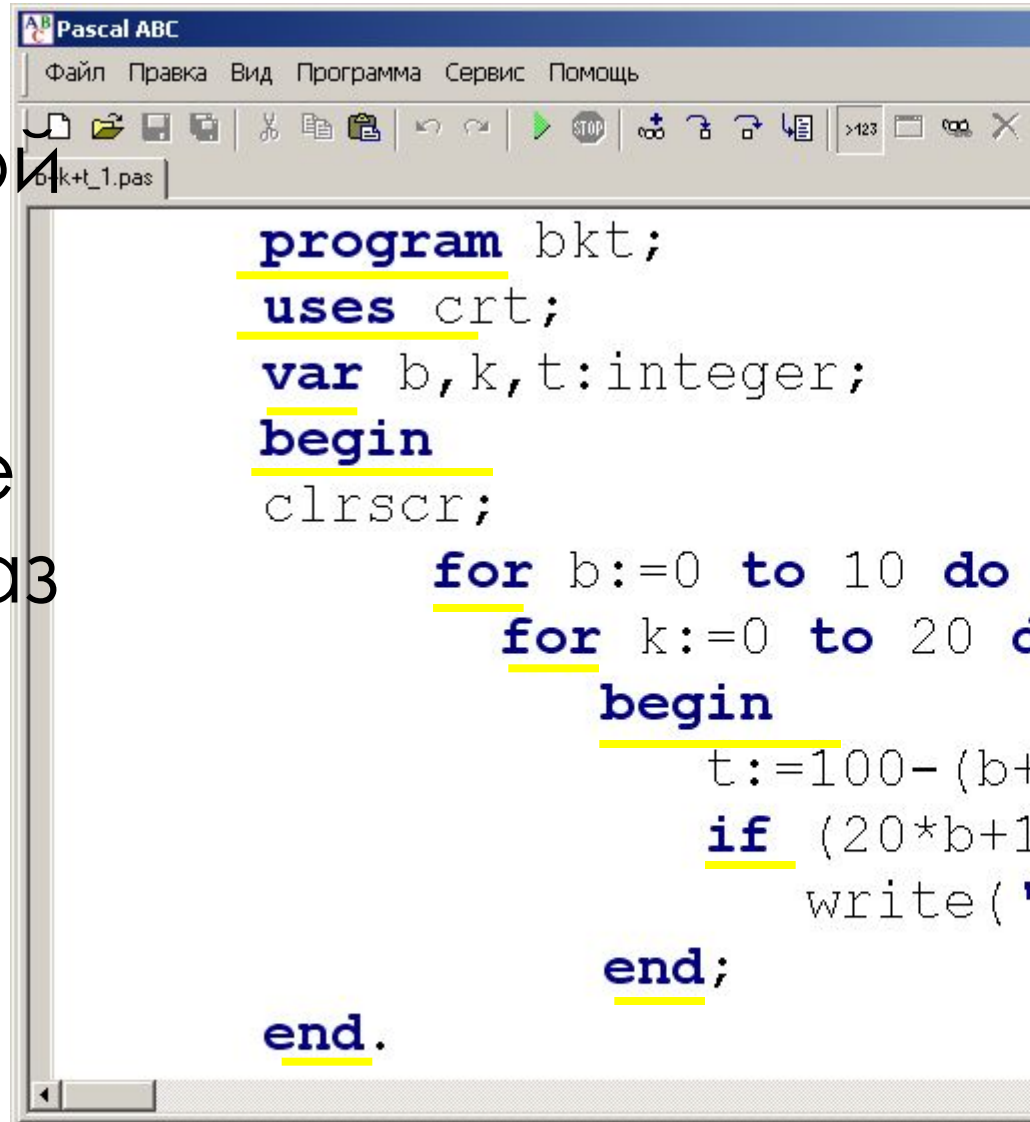
# Слова

Набор слов, используемый в Паскале, можно разделить на три группы:

- зарезервированные слова;
- стандартные идентификаторы;
- идентификаторы пользователя.

# Зарезервированные слова

являются составной частью языка, имеют фиксированное начертание и раз и навсегда определенный смысл.



```
program bkt;  
uses crt;  
var b,k,t:integer;  
begin  
  clrscr;  
  for b:=0 to 10 do  
    for k:=0 to 20 do  
      begin  
        t:=100-(b+  
          if (20*b+1  
            write(''  
      end;  
end.
```

# Словосочетание

– группа слов, имеющая некоторый смысл.

Пример: Program Example 1.

**Предложение** – минимальная конструкция языка, представляющая собой законченную мысль.

Пример: `read(a,b);`



# Выражение

– словосочетание, состоящее из слов и символов и задающее правило вычисления некоторого значения.

Пример: `if X>5 then K:=K*2;`

# Предложение

– минимальная конструкция языка,  
представляющая собой  
законченную мысль.

Пример: `read(a,b);`



# Оператор

– предложение языка программирования, задающее полное описание некоторого действия, которое необходимо выполнить.

Пример: линейные, разветвляющиеся, циклические.



# Идентификатор

– имя, свободно выбираемое программистом для элементов программы.

## **Идентификаторы подразделяются на:**

- **стандартные** (для обозначения заранее определенных разработчиком языка типов данных, констант, процедур, функций. Например: `integer`, `sin`, `read`, `write`).
- **пользователя** (для обозначения меток, констант, переменных, процедур, функций, определенных самим программистом)



# Общие правила написания идентификаторов

- Начинается только с буквы или знака подчеркивания.
- Может состоять из букв, цифр, знака подчеркивания (пробелы, точки, спецсимволы при написании идентификатора недопустимы).
- Максимальная длина идентификатора 127 символов, но значимы только первые 63 символа.
- Два разных элемента не могут иметь одинаковые имена, то есть имя должно быть уникальным в пределах программы.
- При написании идентификаторов можно использовать как прописные, так и строчные буквы. Компилятор не делает различий между ними. На практике рекомендуется применять эту особенность для более простого чтения и понимания значений идентификаторов.

**Задание 3:** Что в списке можно рассматривать как идентификаторы? Почему?

FIО, ФИО, F.I.O., X, WWW, >=, \$, Result, 2\_Result, primer1, 3Primer, DomZ12, S

**Задание 4:** Укажите идентификаторы, которые проще воспринимаются при чтении? Объясните причину.

Kl1, klass1, Nomerdoma, Nomer\_Doma, DomZ3, DomZadanie3, Example5, E5, HW, Homework, HomeWork, R, Result, Rez.



# Константы

– элементы данных, значения которых установлены в описательной части программы и в процессе выполнения программы не изменяются.

**Формат:**

**Const <идентификатор>=<значение>;**

## Пример:

Const

MyName = 'Sidorov';

MyBirthday = '27 июня';

Max = 1000;

Min = 0;

Center = (Max-Min)/2;

Num\_School = 28;





# Переменные

– величины, которые могут менять свои значения в процессе выполнения программы.

**Формат:**

**Var <идентификатор>:<тип>;**



# Структура программы

**Program** Заголовок программы

**Uses** Раздел подключения модулей-библиотек  
дополнительных процедур и функций

**Label** Раздел описания меток безусловного перехода

**Const** Раздел описания констант

**Type** Раздел описания типов данных

**Var** Раздел описания переменных

**Procedure**

**Function** Раздел описания подпрограмм – процедур и функций, содержащих некоторые элементы логики программы.

**Begin** Начало раздела описания логики программы

**End.** Конец раздела описания логики программы.

## Задание 5:

Найдите ошибки в следующей программе:

**Program** Ошибки1;

**Begin**

Summa: =6+8;

Writeln(Summaa)

**End**;

**Program** My Error;

Uses ;

**Begin**

Writeln('введите значение x');

Writeln('x в квадрате', x\*x);

**End**.

Var x: integer;