**Материалы к занятию онлайн по теме:**

**«Строковые переменные», 10 класс**

*Умение решать задачи – такое же практическое искусство, как умение плавать или бегать.  
Ему можно научиться только путем подражания или упражнения.  
Д. Пойа*

Строковые переменные используются в задачах различных типов:

* подсчет частоты появления символа в строке;
* поиск подстроки в строке, замена найденной подстроки на другую строку;
* выделение слов из строки, нахождение числа слов в строке, нахождение минимальной (максимальной) длины слова в строке;
* сравнение и сортировка строк и др.

Все приведенные программы написаны на языке программирования PascalABC.

Когда все задачи по теме будут прорешены и станут понятны, буду знать, что эта тема, строковые переменные, вами усвоена!

По всем вопросам пишите в группу или в личку.

А теперь немного теории, и за задачи!

**Определение**

* ***Символ* —** это буква, цифра или какой-либо знак.
* ***Строка* –** упорядоченная последовательность символов, заключенная в апострофы.
* Количество символов в строке называется **ее *длиной*.**
* Длина строки может лежать в диапазоне **от 0 до 255.**
* **Каждый символ строковой величины занимает 1 байт памяти и имеет числовой код в соответствии с таблицей кодов ASCII**

**Таблица кодов ASCII**

* Символ может быть записан, ограниченный апострофами (‘a’) или записан в коде (#97)
* **сhr(<код символа>) –** функция, возвращающая символ по его коду в кодовой таблице ASCII.
* **ord(<символ>) –** функция, возвращающая код по соответствующему символу.

**Вывод таблицы кодов**

**Описание строк**

* **Var**

**<имя\_строки>:string;**

**<имя\_строки>:string[n];**

n - максимально возможная длина строки. Если параметр опущен, то по умолчанию он считается равным 255.

* **Var**

**name : string[15];**

**Элемент строки**

* **К строке можно обратиться посимвольно.**
* **i-й символ строки s записывается как s[i].**
* **Например, если s = ‘Мир',**

то s[1] = это ‘М',

s[2] = это ‘и',

s[3] = это ‘р'

**Конкатенация (слияние)**

* Над строковыми данными определена операция слияния, обозначаемая знаком +.
* Например:

**a := 'Pascal'; b := 'ABC' c := a + b;** В этом примере переменная **c** приобретет значение **'PascalABC'.**

* Аналогичную операцию выполняет функция **concat(s1,s2,...,sn)**

**c:=concat(a,b);**

**результат с='PascalABC'**

**Сравнение строк**

* Над строками определены операции сравнения **<,>,=,<>,<=,>=,**  результат выполнения операции **логический.**
* Две строки сравниваются посимвольно, слева направо, по кодам символов. Если одна строка меньше другой по длине, при сравнении недостающие символы короткой строки заменяются символом с кодом 0.
* В результате сравнения двух строк получается логическое значение (**true** или **false**).

Примеры:

‘строка’**<>**’строки’ (true);

‘Abc’**<**’abc’ (true);

‘год’**>**’век’; (true);

**Длина строки**

* **length(s);** определяет текущую длину строки s.

**Пример:**

**s=‘Программирование’; length(s) = 16**

**Копирование в строке**

* **copy(s,p,k);** возвращает подстроку, выделенную из строки s, длиной k символов, начиная с символа под номером p.

**Пример:**

**s := ‘порт сервера DNS';**

**s1 := copy(s, 9, 4); {s2 будет ‘вера'}**

**Позиция символа в строке**

* **pos(s1,s);** возвращает номер первой позиции подстроки s1 в исходной строке s. Если подстроки нет, то выдает 0.

**Пример:**

**s := способ хранения';**

**x1 := pos(‘ранения', s); {x1 будет 9}**

**x2 := pos(‘рана', s); {x2 будет 0}**

**Вставка строки в строку**

* **insert(s1,s,p);** вставляет строку s1 в строку s, начиная с символа под номером p строки s.

**Пример:**

**s := 'Система Pascal';**

**insert('Turbo ',s,9); {s будет равно 'Система Turbo Pascal'}**

**Удаление в строке**

* **delete(s,p,k);** удаляет из строки s подстроку длиной k символов, начиная с символа под номером p.

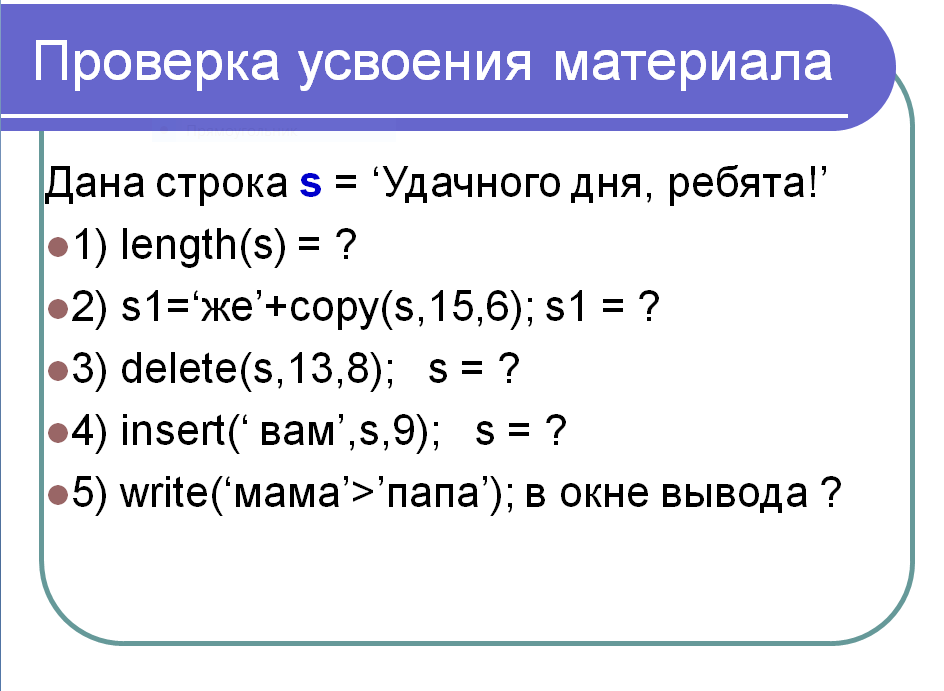
**Пример:**

**s := 'Система Turbo Pascal'; delete(s,8,6); {s будет равно 'Система Pascal'}**

**Преобразования**

**Str (x,s);** - преобразует числовое значение **x** в строковое **s**.

**Val (s,v,error); -** преобразует строковое значение **s** в числовое **v**, где значение переменной **error** равно позиции первого по порядку неправильного символа в **s**, если преобразование не произошло, или **0** в случае успешного преобразования.



Молодцы! Основные понятия усвоены, теперь приступим к задачам!

**Задача 1.**

С помощью функции копирования и операции конкатенации составить из частей слова “информатика” слово “торт”.

program novslovo;

uses crt;

var s, ns: string;

begin

     s:='информатика';

     ns:=copy(s,8,1)+copy(s,4,2)+copy(s,8,1);

     writeln(ns);

end.

**Задача 2.**

Подсчитать, сколько раз среди символов заданной строки встречается буква “а”.

program kolbukv\_a;

uses crt;

var s: string;

      i, k: integer;

begin

     write('Введите строку: ');

     readln(s);

     k:=0;

     for i:=1 to length(s) do

          if s[i]='а' then k:=k+1;

     writeln('Буква “а” встречается ', k, ' раз(а)');

end.

Буква “а” встречается 5 раз(а)

**Задача 3.**

Найти долю пробелов в строке *A*.

program dolya;

uses crt;

var a: string;

     i, k: integer;

     d: real;

begin

     write('Введите строку: ');

     readln(a);

     k:=0;

     for i:=1 to length(a) do

          if a[i]=' ' then k:=k+1;

     d:=k/length(a);

     writeln('Доля пробелов в строке ', d);

end.

**Задача 4.**

Заменить все буквы “а” на буквы “б” в заданной строке.

program zamena\_ab;

uses crt;

var s: string;

      i: integer;

begin

     write('Введите строку: ');

     readln(s);

     for i:=1 to length(s) do

          if s[i]='а' then s[i]:='б';

     writeln(s);

end.

**Задача 5.**

Из заданной строки получить новую, повторив каждый символ дважды.

program udvsimv;

uses crt;

var s, ns: string;

      i, j, l: integer;

begin

     write('Введите строку: ');

     readln(s);

     l:=length(s);

     j:=1;

     for i:=1 to l do begin

          ns [5]:=s[3];

          ns[6]:=s[3];

          j:=j+2;

     end;

     for i:=1 to 2\*l do write(ns[i]);

end.

**s-Авраллр**

**i-12345678**

Ns-ААвврр

J-123456

**Задача 6.**

Дано слово. Вывести слово, содержащее те же символы, но расположенные в обратном порядке.

program slovonaoborot;

uses crt;

var s: string;

      i: integer;

begin

     write('Введите слово: ');

     readln(s);

     for i:=length(s) downto 1 do write(s[i]);

end.

**Задача 7.**

Проверить, является ли заданное слово палиндромом.

program palindrom;

uses crt;

var s: string;

      i, l, m, k: integer;

begin

     write('Введите слово: ');

     readln(s);

     l:=length(s);

     m:=l div 2;

     k:=0;

     for i:=1 to m do

          if s[i]<>s[l+1-i] then k:=1;

     if k=1 then writeln('Слово “', s, '” не палиндром')

                else writeln('Слово “', s, '” палиндром');

end.

**Задача 8.**

Строка *X* состоит из нескольких предложений, каждое из которых кончается точкой, восклицательным или вопросительным знаком. Определить количество предложений в строке *X*.

program kolpredl;

uses crt;

var x: string;

      i, k: integer;

begin

     write('Введите строку: ');

     readln(x);

     k:=0;

     for i:=1 to length(x) do

          if (x[i]='.') or (x[i]='!') or (x[i]='?') then k:=k+1;

     writeln('В строке ', k, ' предложений');

end.

**Задача 9.**

Проверить правильность расстановки скобок в формуле. Расстановку считать правильной, если число открывающих скобок равно числу закрывающих скобок.

program skobki;

uses crt;

var f: string;

      i, ko, kz: integer;

begin

     write('Введите формулу: ');

     readln(f);

     ko:=0; kz:=0;

     for i:=1 to length(f) do begin

          if f[i]='(' then ko:=ko+1;

          if f[i]=')' then kz:=kz+1;

     end;

     if ko=kz then writeln('Правильная расстановка скобок')

                    else writeln('Неправильная расстановка скобок');

end.

**Задача 10.**

Проверить правильность расстановки скобок в формуле. Учитывать порядок скобок.

|  |  |
| --- | --- |
| ***Способ 1***  program skobki\_1;  uses crt;  var f: string;        i, k: integer;  begin       write('Введите формулу: ');       readln(f);       i:=1; k:=0;       repeat            if f[i]='(' then k:=k+1;            if f[i]=')' then k:=k-1;            i:=i+1;       until (k<0) or (i>length(f));       if k=0 then writeln('Правильная расстановка скобок')                  else writeln('Неправильная расстановка скобок');  end. | ***Способ 2***  program skobki\_2;  uses crt;  var f: string;        i, k, z: integer;  begin       write('Введите формулу: ');       readln(f);       k:=0; z:=0;       for i:=1 to length(f) do begin            if f[i]='(' then k:=k+1;            if f[i]=')' then k:=k-1;            if k<0 then z:=1;       end;       if (k=0) and (z=0) then            writeln('Правильная расстановка скобок')            else            writeln('Неправильная расстановка скобок');  end. |

**Задача 11.**

В заданной строке подсчитать количество букв латинского алфавита.

program kolbukvlat;

uses crt;

var s: string;

      i, k: integer;

begin

     write('Введите строку символов: ');

     readln(s);

     k:=0;

     for i:=1 to length(s) do

          if (s[i]>='A') and (s[i]<='Z') or (s[i]>='a') and (s[i]<='z') then

               k:=k+1;

     writeln('В строке “', s, '” ', k, ' латинских букв');

end.

**Задача 12.**

Подсчитать количество цифр в заданной строке.

program kolcifr;

uses crt;

var s, c: string;

      i, j, k: integer;

begin

     write('Введите строку символов: ');

     readln(s);

     c:='0123456789';

     k:=0;

     for i:=1 to length(s) do

          for j:=1 to 10 do

               if s[i]=c[j] then k:=k+1;

     writeln('В строке “', s, '” ', k, ' цифр');

end.

**Задача 13.**

Из заданной строки получить новую, удалив из нее все пробелы.

program udalprobel;

uses crt;

var s, ns: string;

      i, k: integer;

begin

     write('Введите строку: ');

     readln(s);

     k:=0;

     for i:=1 to length(s) do

          if s[i]<>' ' then begin

               k:=k+1;

               ns[k]:=s[i];

          end;

     for i:=1 to k do write(ns[i]);

end.

**Задача 14.**

Из заданной строки получить новую, удалив все буквы латинского алфавита.

program udalbukv;

uses crt;

var s, ns: string;

      i, k: integer;

begin

     write('Введите строку: ');

     readln(s);

     k:=0;

     for i:=1 to length(s) do

          if (s[i]<'A') or (s[i]>'Z') and (s[i]<'a') or (s[i]>'z') then begin

               k:=k+1;

               ns[k]:=s[i];

          end;

     for i:=1 to k do write(ns[i]);

end.

**Задача 15.**

Подсчитать, сколько раз встречается в тексте заданный фрагмент.

program kolfragm;

uses crt;

var t, f: string;

      i, k: integer;

begin

     write('Введите текст: ');

     readln(t);

     write('Введите фрагмент: ');

     readln(f);

     k:=0;

     for i:=1 to length(t) do

          if copy(t, i, length(f))=f then k:=k+1;

     writeln('Фрагмент “', f, '” встречается в тексте ', k, ' раз(а)');

end.

**Задача 16.**

Проверить, является ли частью данного слова слово “сок”. Ответ должен быть “да” или “нет”.

program chastslova;

uses crt;

var s: string;

     i, f: integer;

begin

     write('Введите слово: ');

     readln(s);

     f:=0;

     for i:=1 to length(s) do

          if copy(s, i, 3)='сок' then f:=1;

     if f=1 then writeln('да')

               else writeln('нет');

end.

**Задача 17.**

В заданной строке всюду “1999” заменить на “2000”.

program zamena\_1999;

uses crt;

var a: string;

      i: integer;

begin

     write('Введите строку: ');

     readln(a);

     for i:=1 to length(a) do

          if copy(a,i,4)='1999' then begin

               delete(a, i, 4);

               insert('2000', a, i);

          end;

     writeln(a);

end.

**Задача 18.**

Ввести с клавиатуры строку, содержащую фамилию, имя и отчество. Слова в строке разделять одним пробелом. Получить новую строку, содержащую фамилию и инициалы.

program fio;

uses crt;

var s, f, i, o: string;

      p: integer;

begin

     write('Введите фамилию, имя и отчество: ');

     readln(s);

     p:=pos(' ', s);

     f:=copy(s, 1, p-1);

     delete(s, 1, p);

     p:=pos(' ', s);

     i:=copy(s, 1, 1);

     delete(s, 1, p);

     o:=copy(s, 1, 1);

     s:=f+' '+i+'.'+o+'.';

     writeln(s);

end.

**Задача 19.**

Подсчитать количество слов в исходной строке. Слова считать отделенными друг от друга одним пробелом.

program kolslov;

uses crt;

var s: string;

      i, k: integer;

begin

     write('Введите строку: ');

     readln(s);

     s:=s+' ';

     k:=0;

     for i:=1 to length(s) do

          if s[i]=' ' then k:=k+1;

     writeln('Количество слов в строке ', k);

end.

**Задача 20.**

Подсчитать количество слов в исходной строке. Слова считать отделенными друг от друга одним или несколькими пробелами.

program kolslov;

uses crt;

var s: string;

      i, k, f: integer;

begin

     write('Введите строку: ');

     readln(s);

     s:=s+' ';

     k:=0;

     f:=0;

     for i:=1 to length(s) do begin

          if (s[i]<>' ') and (f=0) then f:=1;

          if (s[i]=' ') and (f=1) then begin

               f:=0;

               k:=k+1;

          end;

     end;

     writeln('Количество слов в строке ', k);

end.

**Задача 21.**

Проверить цепочку слов на совпадение первой буквы следующего слова с последней буквой предыдущего слова.

program igra\_v\_slova;

uses crt;

var s: string;

      i, f: integer;

begin

     write('Введите слова через пробел: ');

     readln(s);

     f:=0;

     for i:=2 to length(s)-1 do

          if s[i]=' ' then

               if s[i-1]<>s[i+1] then f:=1;

     if f=0 then writeln('Совпадают')

               else writeln('Не совпадают');

end.

**Задача 22.**

Определить наименьшую длину слова в заданной строке. Слова считать отделенными друг от друга одним или несколькими пробелами.

program mindlina;

uses crt;

var s: string;

      i, lw, min: integer;

begin

     write('Введите строку: ');

     readln(s);

     s:=s+' ';

     min:=length(s);

     lw:=0;

     for i:=1 to length(s) do begin

          if s[i]<>' ' then

               lw:=lw+1

               else if (min>lw) and (lw>0) then begin

                    min:=lw;

                    lw:=0;

                    end

                    else

                    lw:=0;

     end;

     writeln('Наименьшая длина слова ', min);

end.

**Задача 23.**

Ввести с клавиатуры 10 слов и вывести их на экран в алфавитном порядке.

program sort\_slov;

uses crt;

const n=10;

var s: array[1..n] of string;

      pp: string;

      i, j: integer;

begin

     writeln('Введите слова:');

     for i:=1 to n do

          readln(s[i]);

     for i:=1 to n-1 do

          for j:=i+1 to n do

               if s[i]>s[j] then begin

                    pp:=s[i];

                    s[i]:=s[j];

                    s[j]:=pp;

               end;

     writeln('Список слов по алфавиту:');

     for i:=1 to n do

          writeln(s[i]);

end.

