

Организация работы со строками

Строки, как частный случай массивов

Типы строк

- **ShortString**

ANSI, $L \leq 255$

L	C	C	C	C	C	...	C	C
0	1	2	3	...				L

- **AnsiString**

L	L	L	L	C	C	...	C	C
0	1	2	3	4	...			L+3

- **WideString**

Unicode

L	L	L	L	C	C	...	C	C
0	1	2	3					

- **PChar**

L	L	L	L	C	C	...	C	#0
0	1	2	3	4	...			

- **PWideChar**

C	C	C	C	C	...	C	#0	
0	1	2	3			...	2L	

Пример работы с ANSIStrng

```
var
  string1, string2 : AnsiString;
begin
  // Присвоение предложения первой строке
  string1 := 'Hello World';
  // Присвоение второй строке значения путем указания string2 на
  string1
  // Число ссылок на строку 'Hello World' равно 2
  string2 := string1;
  //Разъединение string1 и string2
  string2 := string2 + ', how is everyone?';
  //Длина первой строки 5
  SetLength(string1, 5);
writeln('String1 = '+string1);
writeln ('String2 = '+string2);
end;
String1 = Hello
String2 = Hello World, how is everyone?
```

В модулях System, SysUtils, Strings определены методы для обработки строк

Функциональное назначение методов:

- Создание и удаление строк.
- Копирование и объединение.
- Длина и позиционирование.
- Представление строки.
- Преобразования строки в иной тип.
- Преобразования иного типа в строку.
- Сравнение строк.

Средства модуля System

Вызов	Описание
Length(S)	Длина строки S
UpCase(S)	Преобразование строки S в верхний регистр
Copy(S, I, Count)	Копирование строки S с позиции I на длину Count
Pos(S1,S2)	Позиция первого вхождения строки S1 в строку S2
Insert(S1,S2, I)	Вставка строки S1 в строку S2 на позицию I
Delete(S, I,C)	Удаление строки S с позиции I на длину C
Concat(S1,S2)	Соединение строк S1 и S2
LowerCase(S)	Преобразование строки S в нижний регистр
Str(X; S)	Преобразование значения X в строку S
Val(S, V, Code)	Преобразование строки S в значение V с кодом ошибки Code
StringOfChar(C, Count)	Создание строки из Count символов C

```

VAR  Words: STRING;
BEGIN
  Write ('Введите, пожалуйста, слово: ');
  Readln  Words := 'Фирма Microsoft';
  Writeln Words := 'картографирование';
  Words := 'Электрификация';
  Search_Word := 'кац';
  Position := Pos(Search_Word, Words);
  IF Position <= Length(Words) THEN
    Words := 'компьютеризация';
  ELSE
    Word1 := 'Г';
    Delete(Words,1,7);
    Delete(Words,3,2);
    Insert(Word1,Words,1);
    Word1 := 'не';
    Insert(Word1,Words,3); { Words=Генерация }
  END IF;
  Writeln Words;
END;

```

в слове "

я "

дробных позиции) 'А',4);

E+02

'AAAA'

Средства модуля SysUtils

Вызов	Действие
IntToStr(V).	Превращает целое число V в строку.
StrToInt(S)	Превращает строку S в целое число.
StrToFloat(S)	Превращает строку S в число.
FloatToStr(V)	Превращают число V в строку.
FloatToStrF(V,Format) Format : ffcurrency – денежный, ffExponent – научный, ffFixed – фиксированный, ffGeneral – обобщенный, с самым коротким результатом, ffNumber – фиксированный с разделителями тысяч	Превращают число V в строку, используя формат вывода Format S:=FloatToStr(V,ffFixed,6,3); // Число => строка с форматом writeln(S); // Ответ 123.450

Строка , как частный случай массива

```
Var s: array [1..10] of char;
```

```
Str1: string[10];
```

```
Str2:string[5];
```

```
Begin
```

```
Str1:='Это строка';
```

```
Str2:='Пример';
```

```
S:=str1; // s[1]='Э', s[2]='т', s[3]='о',... ,s[10]='а'
```

```
S:=str2;// s[1]='П', s[2]='р', s[3]='и',... ,s[10]='#0'
```


Определить, является ли строка палиндромом.
Палиндром – это число, слово или фраза,
одинаково читающиеся в обоих направлениях.

Радар, Топот, ПатоП

```
var
  s: string;
  i: byte; flag: boolean;
begin
  write('String: ');
  readln(s);
  flag := true;
  for i := 1 to length(s) div 2 do
    if s[i] <> s[length(s)-i+1] then begin
      writeln('No palindrome');
      flag := false;
      break;
    end;
  if flag then
    write('Palindrome');
  readln
end.
```

Найти в строке определенную последовательность символов и заменить ее другой.

```
var
  s,s_old,s_new: string;
  i,l_old: byte;
begin
  write('Исходная строка: '); // я изучаю Turbo Pascal
  readln(s);
  write('Что заменить?'); //Turbo
  readln(s_old);
  l_old := length(s_old);
  write('Чем заменить? '); //Free
  readln(s_new);
  i := pos(s_old,s);
  delete(s,i,l_old);
  insert(s_new,s,i);
  writeln(s); // я изучаю Free Pascal
end.
```


Выделить из строки целые числа и поместить их в массив.

var

arr: array[1..100] of string;

i, j: integer; strinput, str: string;

a: char; n: byte;

begin

i := 1; j := 1; str := "";

readln(strinput); //abc 56 de7 ff15

n := length(strinput); //15

while i <= n **do begin**

a := strinput[i];

while (a >= '0') and (a <= '9') **do**

begin

str := str + a; i := i + 1; a := strinput[i]

end;

i := i + 1;

if str <> " **then begin** arr[j] := str; j := j + 1; str := ""; **end;**

end;

for i := 1 **to** j - 1 **do**

write(arr[i]:6); // 56 7 15

end.

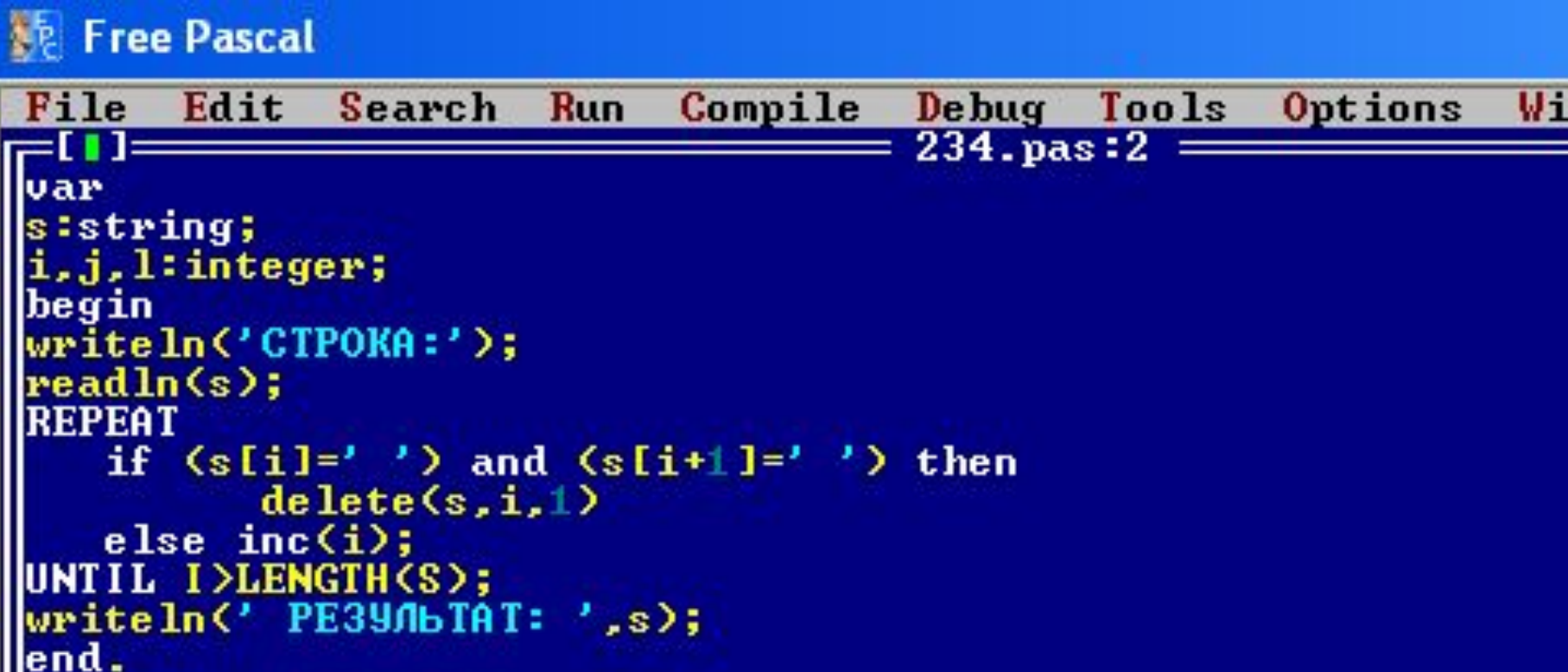
Оставить в строке только один экземпляр
каждого встречающегося символа.

```
var
  s1,s2,s3: string;
  i: byte;
begin
  write('String: '); // adc hghj fgc yugl aa
  readln(s1);
  s2 := "";
  for i:=1 to length(s1) do
  begin
    s3 := copy(s1,i,1);
    if pos(s3,s2) = 0 then
      s2 := s2 + s3
    end;
  writeln('String: ',s2);
  readln
end.
```

Необходимо сформировать из встречающихся в предложении слов их массив. При этом повторяющиеся слова не должны входить в массив дважды.

```
var u,w: string; i, j, q: integer;
ws: array[1..100] of string; : string; len: integer;
begin
  writeln('Введите текст: ');
  read(u);
  len := length(u);
  i := 1; q:=0;
  while i <= len do
    if (lowercase(u[i]) >= 'a') and (lowercase(u[i]) <= 'z') then
      begin w := lowercase(u[i]);
        i := i + 1;
        while (i <= len) and ((lowercase(u[i]) >= 'a') and (lowercase(u[i]) <= 'z')) do
          begin w := w + lowercase(u[i]); i := i + 1; end;
        j := 1;
        while (j <= q) and (w<>ws[j]) do
          j := j + 1;
        if j > q then
          begin q := q + 1; ws[q] := w; end;
        end
      else i := i + 1;
    for i := 1 to q do
      writeln(ws[i]); end.
```


Удалить в строке все лишние пробелы. Лишними считаются пробелы, следующие непосредственно за пробелами. Т.е. между словами всегда должен находиться один пробел



```
Free Pascal
File Edit Search Run Compile Debug Tools Options Wi
234.pas:2
var
s:string;
i,j,l:integer;
begin
writeln('СТРОКА:');
readln(s);
REPEAT
  if (s[i]=' ') and (s[i+1]=' ') then
    delete(s,i,1)
  else inc(i);
UNTIL I>LENGTH(S);
writeln(' РЕЗУЛЬТАТ: ',s);
end.
```


Средства модуля String

ВЫЗОВ	Действие
StrNew(S)	Создать строку S.
StrDispose(S);	Удалить строку S.
StrCopy(Dest, Source)	Копирует в строку Dest строку Source.
StrLCopy(Dest, Source, MaxLen)	Копирует в строку Dest строку Source с ограничением в MaxLen числа символов в Dest.
StrMove(Dest, Source, Count)	Копирует в строку Dest из строки Source Count символов с заменой.
StrCat(Dest, Source)	Присоединяет к строке Dest копию строки Source.
StrLCat(Dest, Source, MaxLen)	Присоединяет к строке Dest копию строки Source. Число символов в результирующей строке не больше MaxLen.
StrCopy(Dest, Source)	Копирует в строку Dest строку Source.
StrLCopy(Dest, Source, MaxLen)	Копирует в строку Dest строку Source с ограничением в MaxLen числа символов в Dest.

StrLen(Str:pChar)	Возвращает длину строки без учета символа с кодом #0.
StrEnd(Str:pChar)	Возвращает указатель на конец строки.
StrPos(Str1, Str2)	Возвращает указатель на начало первого появления строки Str2 в Str1
StrScan(Str, Chr)	Возвращает указатель на первое появление символа Chr в строке Str.
StrRScan(Str, Chr)	Возвращает указатель на последнее появление символа Chr в строке Str.
StrLower(Str)	Перевод строчных букв в прописные (только латинские).
StrUpper(Str)	Перевод прописных букв в строчные (только латинские).
StrPas(Str)	pChar -> string.
StrPCopy(Dest, Source)	string -> pChar.
StrComp(Str1, Str2)	Сравнивает строки. Возвращается: число < 0, если Str1<Str2, 0, если Str1=Str2, число > 0, если Str1>Str2.
StrIComp(Str1, Str2)	То же, но игнорируется разница в высоте латинских букв.
StrLComp(Str1, Str2, MaxLen)	Подобна StrComp, но сравниваются первые MaxLen символов.


```

uses SysUtils;
var
p1,p2,p3,p:pChar; s1,s2:string; MaxLen:word;
begin
p:=StrNew('1234'); // Создать строку (аргумент не пустой)
StrDispose(p); // Удалить строку
p1:=StrNew('AAAAAAAAAAAAAAAAAAAA');
p2:=StrNew('123456');
p3:=StrNew('abcdef');
StrCopy(p1,p2); // Копировать p2 в p1
s1:=StrPas(p1); // Преобразовать p1 в s1 для вывода
writeln(s1); // s1 = 123456
StrLCopy(p1,p2,4); // Копировать p2 в p1 с ограничением длины
writeln(StrPas(p1)); // 1234
p1:=StrMove(p1,p3,2); // Копирование из p3 2 элементов в p1 с заменой
writeln(StrPas(p1)); // ab34
p1:=StrLCat(p1,p2,8); // Дописать p2 в p1 с ограничением длины
writeln(StrPas(p1)); // ab34abcd
End;

```



```
var
p1,p2,p4:pChar;
x:integer;
begin
p1:=StrNew('ABC123');
p2:=StrNew('abc123');
p4:=StrNew('ABC456');
x:=StrComp(p1,p3);
writeln(x);
x:=StrComp(p1,p2);           // p1 < p2
writeln(x);                 // x = -32
x:=StrComp(p4,p1);         // p4 > p1
writeln(x);                 // x = 3
x:=StrIComp(p2,p1);        // p2 = p3
writeln(x);                 // x = 0
x:=StrLComp(p1,p4,3);      // p1 = p4
writeln(x);                 // x = 0
x:=StrLIComp(p1,p4,4);     // Сравнить p1 и p4
writeln(x);                 // x = -3
End;
```