

Лекция №8
Строки

План лекции

1. Символьный тип данных
2. Таблица символов ASCII
3. Строковый тип данных
4. Строковые константы
5. Длина строки
6. Обращение к символам строки
7. Внутренняя структура представления строк
8. Операции над строковыми выражениями
9. Функции работы со строками
10. Процедуры работы со строками
11. Решение задач

Символьный тип данных

В символьный тип `char` входит 256 символов расширенной таблицы ASCII

Например,

`'a', 'b', 'я', '7', '&'`

Номер символа, возвращаемый функцией `ord()`, совпадает с номером этого символа в таблице ASCII.

СИМВОЛЬНЫЙ ТИП ДАННЫХ

Пример описания символьной переменной:

```
Var
```

```
  simb1, simb2 : char;
```

```
Begin
```

```
  simb1:='R'; simb2:=#65; { С помощью # производится  
                           перевод целого числа в          соответствующий  
СИМВОЛ                      данного ASCII-кода }
```

```
  write (simb1,simb2);
```

```
End.
```

Результат работы программы

RA

Символьный тип данных

Все символы упорядочены, т.к. имеют свой личный номер. Важно, что соблюдаются следующие отношения:

'A' < 'B' < 'C' < ... < 'X' < 'Y' < 'Z'

'0' < '1' < '2' < ... < '7' < '8' < '9'

Для проверки равенства или неравенства переменных типа `char` могут использоваться операторы логического сравнения.

СИМВОЛЬНЫЙ ТИП ДАННЫХ

Задача 1.

Написать программу, которая считывает два символа и печатает больше, равны или меньше первый символ второго.

```
Program Z1;  
Var  
    First, Second : char;  
Begin  
    Write ('Введите два символа через пробел ');  
    Readln (First, Second);  
    Write ('Первый символ ');  
    If First > Second then  
        Write ('больше второго. ');  
    else if First = Second then  
        Write ('равен второму. ');  
    else  
        Write ('меньше второго. ');  
End.
```

СИМВОЛЬНЫЙ ТИП ДАННЫХ

Так как `char` – порядковый тип, то к его значениям применимы следующие функции.

`Succ` – возвращает следующий символ;

`Pred` – возвращает предыдущий символ;

`Ord` – возвращает значение ASCII-кода символа;

`Chr` – возвращает значение символа, является обратной по отношению к функции `Ord`.

Примеры:

`Succ('0')='1' // символ, следующий за символом 0, равен символу 1.`

`Pred('3')='2' // символ, предшествующий символу 3, равен 2;`

`Chr(65)='A' // символ, соответствующий ASCII-коду 65, равен A;`

`Ord('A')=65 // ASCII-код символа A равен 65`

СИМВОЛЬНЫЙ ТИП ДАННЫХ

Задача 2.

Запросите у пользователя символ и выведите на экран 5 символов, следующих за ним в таблице американских стандартных кодов обмена информацией. Проверьте, есть ли среди них знаки препинания.

```
Program Z2;
```

```
Var
```

```
  A: char;
```

```
  i : integer;
```

```
  X : boolean;
```

```
Begin
```

```
Write ('Введите символ ');
```

```
Readln (A);
```

```
X:=FALSE;
```

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

```
  begin
```

```
    inc(A);
```

```
    Write (A);
```

```
    If (A=',' ) or (A='-') or (A=':') or (A='!')
```

```
      or (A='?') or (A=':') then
```

```
      X:=TRUE;
```

```
  end;
```

```
if X then
```

```
  Writeln ('Знаки препинания есть.');
```

```
Else
```

```
  Writeln ('Знаки препинания отсутствуют');
```

```
End.
```

СИМВОЛЬНЫЙ ТИП ДАННЫХ

Задача 3.

Вывести в одну строку символы ABBCCCDDDD...ZZ...Z.

```
Program Z3;  
Var  
  a : char;  
  j,i: integer;  
Begin  
  a:='A';  
  for i:=1 to 26 do  
    begin  
      for j:=1 to i do  
        write(a);  
      inc(a);  
    end;  
End.
```

СИМВОЛЬНЫЙ ТИП ДАННЫХ

Задача 4.

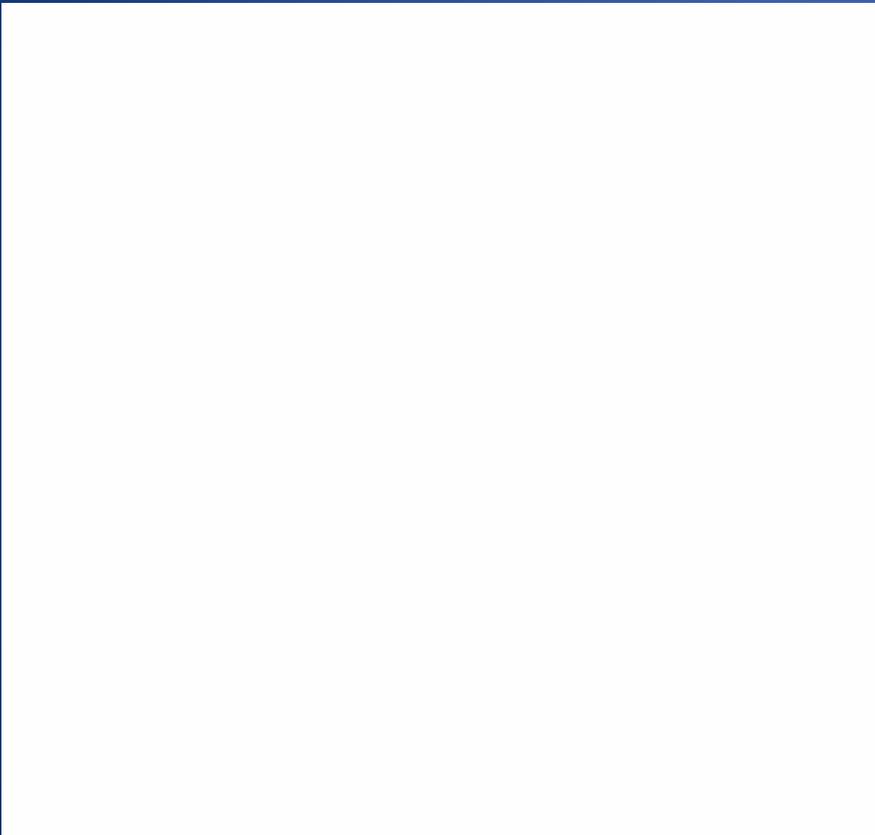
Вывести треугольник из
символов:

```
Аяяяяяя...я
Бюююю...ю
Вээээээ...э
...
Эввв
Юбб
Яа
```

```
Program Z4;
Var
i,j : char;
n,k : integer;
Begin
j:='я';
n:=32;
for i:='А' to 'Я' do
begin
write(i);
for k:=1 to n do
write(j);
dec(j);
dec(n);
if ord(j)=223 then
j:=chr(175);
```

```
writeln;
end;
End.
```

Таблица символов ASCII



Код	Символ	Код	Символ	Код	Символ	Код	Символ
128	À	160	à	192	Ĺ	224	р
129	Á	161	á	193	Ł	225	с
130	Â	162	â	194	Ť	226	т
131	Ã	163	ã	195	Ŧ	227	у
132	Ä	164	ä	196	—	228	ф
133	Å	165	å	197	†	229	х
134	Æ	166	æ	198	‡	230	ц
135	Ç	167	ç	199	£	231	ч
136	È	168	è	200	ℓ	232	ш
137	É	169	é	201	ℓ	233	щ
138	Ê	170	ê	202	ℓ	234	ъ
139	Ë	171	ë	203	ℓ	235	ы
140	Ě	172	ě	204	ℓ	236	ь
141	Ď	173	ď	205	=	237	э
142	Ě	174	ě	206	ℓ	238	ю
143	Ħ	175	ħ	207	ℓ	239	я
144	Ï	176	ï	208	ℓ	240	Ё
145	Ĵ	177	ĵ	209	ℓ	241	ё
146	Ŧ	178	ŧ	210	ℓ	242	Є
147	Ũ	179	ũ	211	ℓ	243	е
148	Ų	180	ų	212	ℓ	244	ї
149	Ẁ	181	ẁ	213	ℓ	245	і
150	Ẃ	182	ẃ	214	ℓ	246	ÿ
151	Ẅ	183	ẅ	215	ℓ	247	ÿ
152	Ẇ	184	ẇ	216	ℓ	248	°
153	Ẹ	185	ẹ	217	ℓ	249	·
154	ƒ	186	ƒ	218	ℓ	250	·
155	Ƀ	187	Ƀ	219	█	251	√
156	Ʉ	188	Ʉ	220	█	252	№
157	Ʌ	189	Ʌ	221	█	253	х
158	Ɇ	190	Ɇ	222	█	254	■
159	ɇ	191	ɇ	223	█	255	

СИМВОЛЬНЫЙ ТИП ДАННЫХ

Задача 5.

Получить массив символов размером 20 из случайных символов, заглавных букв латинского алфавита. Подсчитать количество символов 'S' в массиве.

```
Program Z5;
Var
  a : array [1..20] of char;
  i,k : integer;
Begin
  k:=0;
  Randomize;
  For i:=1 to 20 do
    begin
      a[i]:=chr(random(26)+65);
      write(a[i]);
      if a[i]='S' then
        inc(k);
      end;
    end;
  writeln;
  writeln(k);
End.
```

Строковый тип данных

Строка (string) – это последовательность символов.

Тип данных (string) определяет строки с максимальной длиной 255 символов.

Переменная этого типа может принимать значения переменной длины.

Пример :

```
Var  
    s : string;  
Begin  
    Read(s);  
    Writeln(s);  
End.
```

Строковый тип данных

Строковая переменная может иметь **атрибут длины**, определяющий ее максимальную длину.

Пример :

Var

```
s : string[50];
```

Begin

```
s:='От топота копыт пыль по полю летит.';
```

```
write(s);
```

End.

Строковые константы

Строковая константа – последовательность символов, заключенная в апострофы.

Пример :

'От топота копыт пыль по полю летит.'

'284'

'' – пустая строковая константа

Длина строки

Текущая длина строковой переменной может быть определена с помощью встроенной функции `length`.

Для заданного значения типа `string` эта функция возвращает целое значение, показывающее количество символов в строке.

Пример:

Var

s : string;

n : integer;

Begin

s:='расписание занятий';

n:=length(s);

Writeln (n);

End.

Результат исполнения программы:

18

Обращение к символам строки

К отдельным символам строки можно обратиться по номеру (индексу) данного символа в строке.

Пример:

```
Var  
    s : string;  
Begin  
s:='паровоз';  
Writeln (s[3]);  
End.
```

Результат работы программы:

р

Внутренняя структура представления строк в памяти

Для хранения строковых переменных выделяется память, на единицу большая максимальной длины строки. Начальный байт этой памяти отводится для хранения текущей длины строки, следующие байты – для символов самой строки. Так как элементы строк стандартно нумеруются целыми числами, начиная с единицы, байт с длиной строки можно считать нулевым ее элементом.

Такая структура памяти допускает прямой доступ к ее элементам.

Индексы	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		255
stroka	14	М	а	м	а		м	ы	л	а		р	а	м	у				

Длина строки

stroka[7]

пустые символы

Длина строки

Задача 6-1.

Напишите программу, которая после ввода строки, выводит ее символы в обратном порядке.

Пример :

```
А РОЗА УПАЛА НА ЛАПУ АЗОРА  
АРОЗА УПАЛ АН АЛАПУ АЗОР А
```

Program Z6;

Var

s: string;

i,n: integer;

Begin

Readln(s);

n:= length(s);

for i:=n downto 1 do

write(s[i]);

End.

Операции над строковыми выражениями

Над строками можно выполнять операции :

1. Конкатенацию (сцепление строк).
2. Операции отношения ($>$, $<$, $>=$, $<=$, $=$, $<>$)

Конкатенация

Конкатенация – сцепление нескольких строк в одну результирующую строку.

На языке Паскаль записывается как +

Пример :

Var

 a : string;

Begin

a:= 'Мама' + ' мыла' + ' раму.';

Writeln(a);

End.

Длина результирующей строки не должна превышать 255 символов.

Конкатенация

Задача 6-2.

Напишите программу, которая после ввода строки, получает новую строку из исходной, записью ее символов в обратном порядке.

Пример :

Шалаш Казак Телефон
нофелеТ казаК шалаШ

```
Program Z6_2;  
Var  
    s, q : string;  
    i,n: integer;  
Begin  
    Readln(s);  
    n:= length(s);  
    q:= "";  
    for i:=n downto 1 do  
        q:=q+s[i];  
    Writeln(q);  
End.
```

Операции отношения

Операции отношения ($=$, $<$, $>$, \leq , \geq , \leftrightarrow) позволяют производить сравнение двух строк, в результате чего получается логическое значение.

Сравнение строк производится слева направо **до первого несовпадающего символа**, и та строка считается больше, в которой первый несовпадающий символ имеет больший номер в таблице ASCII-кодов.

Если строки имеют различную длину, но в общей части символы совпадают, считается, что более короткая строка меньше, чем более длинная.

Строки равны если они полностью совпадают по длине и содержат одинаковые символы.

Операции отношения

Определите значение логических выражений :

'True1' < 'True2'

'Month' > 'MONTH'

'мама' <> 'Мама'

'папа' = ' папа'

'Cat' = 'Cat'

'Cat and Dog' <= 'Cat'

'Rita' >= 'RITA'

Операции отношения

Определите значение логических выражений :

'True1' < 'True2'	TRUE
'Month' > 'MONTH'	TRUE
'мама' <> 'Мама'	TRUE
'папа' = ' папа'	FALSE
'Cat' = 'Cat'	TRUE
'Cat and Dog' <= 'Cat'	FALSE
'Rita' >= 'RITA'	TRUE

Решение задач

Задача 7.

Дана строка. Подсчитать в ней количество вхождений буквы 'w'.

```
Program Z7;  
Var  
  s : string;  
  i,n: integer;  
Begin  
  Readln(s);  
  n:=0;  
  for i:=1 to length(s) do  
    if s[i]='w' then  
      inc(n);  
  Writeln(n);  
End.
```

Решение задач

Задача 8.

Дана строка. Заменить в ней все вхождения последовательностей символов 'муха' на 'слон'.

```
Program Z8;  
Var  
    s : string;  
    i: integer;  
Begin  
    Readln(s);  
    for i:=1 to length(s)-3 do  
        if (s[i]='м') and (s[i+1]='у') and (s[i+2]='х') and (s[i+3]='а') then  
            begin  
                s[i]:='с'; s[i+1]:='л'; s[i+2]:='о'; s[i+3]:='н';  
            end;  
    Writeln(s);  
End.
```

Решение задач

Задача 9.

Дана строка. Получите новую строку из символов исходной строки, стоящих в четных позициях исходной строки.

```
Program Z8;  
Var  
    s,q : string;  
    i: integer;  
Begin  
    Readln(s);  
    q:='';  
    for i:=1 to length(s) div 2 do  
        q:=q+s[2*i];  
    Writeln(q);  
End.
```

Решение задач

Задача 10.

Заставьте компьютер случайным образом загадать заглавную букву русского алфавита. Предложите пользователю отгадать загаданную букву, помогая ему следующим образом. Если в очередной попытке пользователем введена буква, стоящая ближе к загаданной, чем предыдущая, то выводите пользователю сообщение "Горячее!", а если дальше - "Холоднее!".

Решение задач

```
Program Z10;
```

```
Var
```

```
  z,x : char;
```

```
  n,m : integer;
```

```
Begin
```

```
  Randomize;
```

```
  z:=chr(random(33)+128);
```

```
  writeln(z);
```

```
  n:=32;
```

```
  repeat
```

```
    readln(x);
```

```
    if x<z then
```

```
      m:=ord(z)-ord(x)
```

```
    else
```

```
      m:=ord(x)-ord(z);
```

```
  if m<n then
```

```
    writeln ('Горячее !')
```

```
  else
```

```
    writeln ('Холоднее !');
```

```
  n:=m;
```

```
  until x=z;
```

```
  writeln ('Вы отгадали. Это буква  
    '+z);
```

```
End.
```

Функция Upcase

Функция Upcase позволяет преобразовывать символ из строчного в прописной.

Эта функция рассчитана на обработку отдельного символа. Поэтому для обработки строки символов с помощью этой функции приходится организовывать цикл.

Пример :

```
Program Z11;  
  Var  
    s: string;  
    i: Byte;  
  Begin  
    s := 'фирма Microsoft';  
    for i := 1 to length (s) do  
      s[i] := UpCase (s[i]);  
    writeln(s); {выводится текст 'фирма MICROSOFT'}  
  End.
```

Функция Uppercase

Русские символы не могут обрабатываться этой функцией.

Для того, чтобы преобразовать в заглавные строчные буквы русского алфавита, применяют оператор выбора:

```
...  
  case s[i] of  
    'a' : s[i] := 'A';  
    'б' : s[i] := 'Б';  
    'в' : s[i] := 'В';  
    ...  
  end;
```

```
...
```

Функция Сору

Функция `Сору` позволяет копировать фрагмент некоторой строки из одной переменной в другую.

Вызывая эту функцию нужно указать следующие параметры:

- имя строки, из которой должен извлекаться копируемый фрагмент;
- позицию в строке, начиная с которой будет копироваться фрагмент;
- число копируемых символов.

Пример :

```
...  
s:='паровоз';  
q:=сору(s,5,3);  
...
```

Функция Copy

```
Program Z12;  
Var  
  s : string;  
  q : string[20];  
Begin  
  s := 'фирма Microsoft';  
  writeln(s); {выводится текст 'фирма MICROSOFT'}  
  q := Copy (s,1,5);  
  writeln(q); {выводится текст 'фирма'}  
End.
```

Если начальная или конечная позиции копируемого текста находятся вне пределов исходной строки символов, то сообщение об ошибке не выдается. Результатом выполнения операции в первом случае будет строка нулевой длины, во втором – фрагмент от начальной позиции копирования до конца исходной строки.

Функция Pos

С помощью функции Pos Вы можете осуществить поиск некоторого фрагмента в строке. Если заданный фрагмент в строке присутствует, то функция возвращает номер позиции, с которой он начинается. Если фрагмент не найден, то функция возвращает 0.

Пример:

```
...  
s:='информатика'  
x:=pos('форма',s);  
...
```

Результат :

x=3

Функция Pos требует полного совпадения искомого фрагмента и фрагмента строки, в которой производится поиск. Причем большие и маленькие буквы считаются различными символами.

Функция Pos

```
Program Z13;
Var
  s : string;
  q : string[20];
  p : Byte;
Begin
  s := 'фирма Microsoft';
  writeln(s); {выводится текст 'фирма MICROSOFT'}
  writeln ('Введите искомый текст ');
  readln (q);
  p:= Pos(q,s);
  if p <> 0
  then
    begin
      write ('Фрагмент <'+q+'> содержится в строке <'+s);
      writeln ('>, начиная с позиции ',p);
    end
  else
    writeln('Фрагмент <'+q+'> не содержится в строке <'+s);
End.
```

Функция Concat

Функция `Concat (Str1,Str2,...,StrN)` выполняет конкатенацию (или сцепление) строк `Str1,Str2,...,StrN` в том порядке, в каком они указаны в списке параметров.

Сумма символов всех сцепленных строк не должна превышать 255.

Пример:

```
Program Z14;
```

```
Var
```

```
  s : string;
```

```
  q1, q2 : string[20];
```

```
Begin
```

```
  q1 := 'фирмы ';
```

```
  q2 := 'Microsoft';
```

```
  s := Concat('Программы ',q1,q2);
```

```
  writeln(s); {выводится текст 'Программы фирмы Microsoft'}
```

```
End.
```

Процедура Insert

Процедура Insert вставляет в исходную строку, начиная с указанной позиции, какую-либо другую строку.

Оператор Insert (str1,str2,5) указывает, строку str1 необходимо вставить в строку str2, начиная с 5-ой позиции.

Пример:

```
Program Z15;
```

```
Var
```

```
    d,m : string;
```

```
Begin
```

```
m:= 'дарованный нам свыше';
```

```
d:= 'От ветра пламя пуще полыхает, так и талант не чахнет в  
испытаниях, а крепчает';
```

```
insert (m,d,42);
```

```
Write(d);
```

```
End.
```

Процедура Delete

Процедура Delete удаляет в исходной строке фрагмент определенной длины, начиная с указанной позиции.

Так, оператор Delete(str1,2,3) удаляет из указанной строки фрагмент, длиной в три символа, начиная со второго.

Пример:

```
Program Z16;
```

```
Var
```

```
    m : string;
```

```
Begin
```

```
m:='мухомор';
```

```
delete (m,4,3);
```

```
Write(m);
```

```
End.
```

Процедура Str

Общий вид Str(Chislo,Stroka)

Процедура Str преобразовывает числовое значение переменной Chislo в строковую переменную Stroka. После первого параметра может указываться формат, аналогичный формату вывода.

```
Program Z17;  
Var  
  x : string;  
  y : integer;  
Begin  
  y := 1560;  
  Str(y,x);  
  writeln(x); {выводится строка ' 1560'}  
End.
```

Процедура Val

Общий вид Val(Stroka,Chislo,Code)

Процедура Val преобразует значение строки Stroka в величину целочисленного или вещественного типа и помещает результат в Chislo. Значение строковой переменной Stroka не должно содержать пробелов в начале и в конце.

Code целочисленная переменная. Если во время операции преобразования ошибки не обнаружено, значение Code равно нулю, если же ошибка обнаружена, Code будет содержать номер позиции первого ошибочного символа, а значение Chislo будет не определено.

```
Program Z18;  
Var  
  s: string;  
  x, Code : integer;  
Begin  
  writeln('Введите строку цифр ');  
  readln(s);  
  Val(s, x, Code); {преобразование строки в число}  
  if Code <> 0  
  then  
    writeln('Ошибка! В позиции ',Code,' не цифра!')  
  else  
    write(x);  
End.
```

