

### 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

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.

```
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

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

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

#### 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.
```

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
```

#### 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.
```

### 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.
```

#### Задача 4.

Вывести треугольник из символов:

Аяяяяяяяя...я Бюююю...ю Вээээээ...э

Эввв Юбб

Яа

```
Program Z4;
Var
i,j : char;
n,k: integer;
Begin
j:='Я';
n:=32;
for i:='A' to 'A' 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	A	160	а	192	L	224	p
129	Б	161	б	193	1	225	c
130	В	162	В	194	Т	226	T
131	Г	163	r	195	1	227	У
132	Д	164	д	196	÷	228	Φ
133	E	165	e	197	+	229	×
134	ж	166	280	198	F	230	ц
135	3	167	3	199	ŀ	231	ч
136	И	168	и	200	L	232	ш
137	Й	169	й	201	F	233	щ
138	K	170	ĸ	202	1	234	ъ
139	Л	171	л	203	ī	235	ы
140	M	172	M	204	Ī	236	ь
141	H	173	н	205		237	э
142	0	174	0	206	<u> </u>	238	ю
143	П	175	п	207		239	я
144	P	176	\$ <b>\$</b>	208	1	240	Ë
145	С	177	~	209	₹	241	ë
146	T	178		210		242	ε
147	У	179	T	211	$\mathbb{I}$	243	ε
148	Φ	180	6-E	212	L	244	Ϊ
149	X	181	8 <b>4</b> 8	213	F	245	ï
150	ц	182		214	Г	246	ў
151	ч	183	1	215	#	247	ў
152	Ш	184	7	216	¥	248	۰
153	Щ	185	4	217	: -	249	
154	ъ	186		218	ſ	250	12
155	ы	187		219		251	٧
156	ь	188	]	220	_	252	N:
157	Э	189	Ш	221		253	×
158	Ю	190		222	1	254	
159	я	191	1	223		255	

#### 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:
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.'

<sup>'284'</sup>

" – 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:

# 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:

# 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	M	a	M	a		M	Ы	Л	a		p	a	M	y			

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

```
Progrem 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 языка 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 ( =, <, >, <=, >=, <>) 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' <> 'Maмa' TRUE

'nana' = ' nana' FALSE

'Cat' = 'Cat' TRUE

'Cat and Dog' <= 'Cat' FALSE

'Rita' >= 'RITA' TRUE

Задача 7.

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

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

Задача 8.

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

```
Program Z8;
Var
     s: string;
     i: integer;
Begin
ReadIn(s);
for i:=1 to length(s)-3 do
     if (s[i]='m') and (s[i+1]='y') and (s[i+2]='x') and (s[i+3]='a') then
           begin
           s[i]:='c'; s[i+1]:='\pi'; s[i+2]:='o'; s[i+3]:='H';
           end:
Writeln(s);
End.
```

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.
```

#### 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!".

```
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 Upcase

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:=copy(s,5,3);
...
```

# Функция Сору

```
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 О.

```
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');
 readIn (q);
 p:= Pos(q,s);
 if p <> 0
  then
   begin
    write ('Фрагмент < +q+'> содержится в строке < +s); writeIn ('>, 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.

# 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.
```

