

Строки

1. Строковые типы;
2. ASCII и UNICODE;
3. Работа со строками.

1. Строковые типы

Символьный тип данных **Char** — тип данных, значениями которого являются одиночные символы.

В зависимости от языка программирования символ в кодировки ASCII (1 байт) или UNICODE (2 байт)

Строки имеют тип **string**, состоят из набора последовательно расположенных символов **char** и используются для представления текста.

Для задания значения строковым переменным используются апострофы “

пример

```
st := 'строка'
```

```
st := "
```

2. ASCII и UNICODE

ASCII (англ. American standard code for information interchange) — название таблицы (кодировки, набора), в которой некоторым распространённым печатным и непечатным символам сопоставлены числовые коды. Таблица была разработана и стандартизована в США в 1963 году.

Таблица ASCII определяет коды для символов:

- десятичных цифр;
- латинского алфавита;
- национального алфавита;
- знаков препинания;
- управляющих символов.

Строки

1. Строковые типы;
2. **ASCII и UNICODE;**
3. Работа со строками.

2. ASCII и UNICODE

Первые 128 символов (коды 0-127):

- управляющие символы
- знаки препинания
- цифры
- латинские символы

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

2. ASCII и UNICODE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ъ	Ґ	,	Ґ	„	…	†	‡	■	‰	Љ	<	Њ	Ќ	ћ	џ
9	ђ	`	´	˘	•	–	—	■	™	љ	>	њ	ќ	ћ	џ	
A		Ў	ў	Ј	ѡ	Г	;	§	Ё	©	Є	«	¬	—	®	İ
B	°	±	І	і	Г	μ	¶	·	ё	№	є	»	ј	ѕ	ѕ	ї
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

2. ASCII и UNICODE

Юникод — стандарт кодирования символов, позволяющий представить знаки почти всех письменных языков.

Коды в стандарте Юникод разделены на несколько областей. Область с кодами от U+0000 до U+007F содержит символы набора ASCII с соответствующими кодами. Далее расположены области знаков различных письменностей, знаки пунктуации и технические символы. Часть кодов зарезервирована для использования в будущем. Под символы кириллицы выделены области знаков с кодами от U+0400 до U+052F, от U+2DE0 до U+2DFF, от U+A640 до U+A69F

2. ASCII и UNICODE

Нулевая плоскость - отведена для символов практически всех современных письменностей и большого числа специальных

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

- Латинская письменность
- Нелатинские европейские письменности
- Письменности Африки
- Письменности Среднего Востока и Юго-Западной Азии
- Письменности Южной и Центральной Азии
- Письменности Восточной Азии
- Письменности Юго-Восточной Азии
- Идеограммы ККЯ
- Письменности Индонезии и Океании
- Письменности Америки
- Системы нотописи
- Знаки
- Область для частного использования
- Суррогатные пары UTF-16
- Свободные кодовые позиции

По состоянию на версию Юникода 9.0

Строки

1. Строковые типы;
2. **ASCII и UNICODE;**
3. Работа со строками.

3. Работа со строками

Тип String похож на одномерный массив символов ARRAY [0..N] OF CHAR

var

st : string;

st1 : string[7];

begin

st := 'Hello world!!!';

writeln(st);

st[2] := 'E';

writeln(st);

st1:=st;

writeln(st1);

end.

Hello world!!!

HEllo world!!!

HEllo w

Можно применять операцию «+» - сцепление строк

st := 'a'+ 'b';

st := st+'c';

3. Работа со строками

1) **CONCAT(S1, [, S2, ... SN])** – функция типа **STRING**; возвращает строку, представляющую собой сцепление из строк-параметров **S1, S2, ..., SN**

```
st1 := 'Hello';  
st2 := ' world!!!';  
st3 := CONCAT(st1,st2);  
writeln(st3);
```

Hello world!!!

2) **COPY(ST, INDEX, COUNT)** – функция типа **STRING**; копирует из строки **ST** **COUNT** символов, начиная с символа с номером **INDEX**

```
st1 := 'Hello world!!!';  
st2 := copy(st1,3,4);  
writeln(st2);
```

llo

3) **DELETE(ST, INDEX, COUNT)** – процедура; удаляет **COUNT** символов из строки **ST**, начиная с символа с номером **INDEX**.

```
st1 := 'Hello world!!!';  
delete(st1,3,4);  
writeln(st1);
```

Heworld!!!

3. Работа со строками

4) **POS(SUBST, ST)** – функция типа INTEGER; отыскивает в строке ST первое вхождение подстроки SUBST и возвращает номер позиции с которой она начинается; если подстрока не найдена, возвращается ноль

```
st1 := 'Hello world!!!';  
writeln(pos('ata',st1));      0  
writeln(pos('ll',st1));      3
```

5) **INSERT(SUBST, ST, INDEX)** – процедура; вставляет подстроку SUBST в строку ST, начиная с символа с номером INDEX.

6) **LENGTH(ST):integer** – функция типа INTEGER; возвращает длину строки ST.

3. Работа со строками

7) **STR(X [:WIDTH [:DECIMALS]],ST)** – процедура; преобразует число X любого вещественного или целого типов в строку символов ST, так, как это делает процедура WRITELN перед выводом.

var

st:string; x : real;

begin

x := 10102.02341;

str(x:12:2,st);

write(st);

str(x:11:3,st);

write(st);

end.

10102.02 10102.023

3. Работа со строками

8) `chr(X:byte):char` – функция, преобразует код в символ в кодировке Windows .

```
var  
  i : integer;  
begin  
  for i := 32 to 255 do writeln(i, ' - ', Chr(i));  
end.
```

```
..  
189 - S  
190 - s  
191 - ï  
192 - A  
193 - Б  
194 - В  
195 - Г  
196 - Д  
197 - Е  
198 - Ж  
199 - З  
200 - И  
..
```

9) `ChrUnicode(X:word) : char` – функция, преобразует код в символ в кодировке Unicode.

3. Работа со строками

10) Ord(a: char): byte - функция преобразует символ в код в кодировке Windows .

11) OrdUnicode(a: char): word - функция преобразует символ в код в кодировке Unicode.

3. Работа со строками

Пример. Сравнение строк

```
var st1,st2 : string;  
begin  
  write('Строка 1:'); readln(st1);  
  write('Строка 2:'); readln(st2);  
  if st1=st2 then writeln(st1,'=',st2)  
    else if st1>st2 then writeln(st1,'>',st2) else writeln(st1,'<',st2);  
end.
```

```
Строка 1:123  
Строка 2:123  
123=123
```

```
Строка 1:123  
Строка 2:12  
123>12
```

```
Строка 1:123  
Строка 2:1234  
123<1234
```


3. Работа со строками

Пример. Удаление повторяющихся пробелов из строки символов.

```
var
  st: string;
begin
...
while Pos(' ', st)>0 do
  Delete(st, Pos(' ', st), 1 );
...
end.
```

3. Работа со строками

Пример. Записать слова в обратном порядке.

Словом считается последовательность символов между пробелами.

```
var
  Source, Temp, Dest, OneWord: string;
  i: integer;
begin
  writeln('Введи строку из нескольких слов:');
  readln(Source);
  Temp := Source;
  Dest := '';
  while pos(' ',Temp)>0 do
    begin
      i:= pos(' ',Temp);
      OneWord := copy(Temp, 1, i);
      Dest := OneWord + Dest;
      Delete( Temp, 1, i );
    end;
  Dest := Temp + ' ' + Dest;
  writeln(Dest);
end.
```

Введи строку из нескольких слов:
раз два три четыре
четыре три два раз