

**ТЕМА УРОКА:**  
**СТРОКОВЫЙ И**  
**СИМВОЛЬНЫЙ ТИП**  
**ДАнных**



# ПЛАН УРОКА

1. Основные сведения о символьных величинах.
2. Стандартные функции, процедуры для работы с символьными величинами.
3. Решение задач.
4. Домашнее задание.



# ОСНОВНЫЕ СВЕДЕНИЯ О СИМВОЛЬНЫХ ВЕЛИЧИНАХ

- С помощью ЭВМ можно решать весьма разнообразные задачи обработки текстов: от составления платежных ведомостей до автоматической верстки газет. Для того, чтобы ЭВМ могла обрабатывать тексты, она должна уметь оперировать не только с числами, но и со словами.
- **Текст** - это произвольная последовательность символов некоторого алфавита.
- **Алфавит** - любое множество символов, например  $(0, 1, 2, \dots)$ ,  $(A, B, V, \dots)$ ,  $(A, B, C, \dots)$ .



# СТРОКА СИМВОЛОВ

- Символьная (строковая, текстовая) константа - последовательность символов заключенных в кавычки (апострофы).
- Строка символов может состоять из одного или нескольких символов, а также не содержать ни одного символа (пустая строка, или строка нулевой длины).
- Максимальная длина текстовой строки 255 символов.
- Для обработки на компьютере данных того или иного типа используются переменные.
- Тип переменной определяется типом данных, которые она представляет.



# КОДОВАЯ ТАБЛИЦА ASCII

Наиболее распространенная международная согласованная система кодирования всех символов - система ASCII.

- Символы с кодами от 0 до 127 представляют так называемую основную таблицу кодов ASCII. Эта часть идентична на всех IBM-совместимых компьютерах. Коды с символами от 128 до 255 представляют национальную часть.
- Так как символы языка упорядочены, то к символьным данным применимы операции сравнения. Операция сравнения осуществляется следующим образом: из двух символов меньше тот, который встречается в таблице ASCII раньше.





# ТИП ПЕРЕМЕННЫХ CHAR

- Символьный тип данных (Char)
- Диапазон значений: любой символ — это буквы, цифры, знаки препинания и специальные символы.
- Каждому символу алфавита соответствует индивидуальный числовой код от 0 до 255.
- Ord(x) - возвращает порядковый номер символа x
- Chr(i) – определяет символ с порядковым номером i
- В Turbo Pascal 7.0 значения для переменных типа char задаются в апострофах: `ch := '*'; a := '3'; letter := 'G'`
- Кроме того, имеется возможность задавать значения указанием непосредственного числового значения ASCII-кода: `kd:=#65 {символ 'A'}; s:=#10 {клавиша <Enter>}`.

[программа](#)

[исполнение](#)



# СТРОКОВЫЙ ТИП ДАННЫХ (STRING)

- Как правило, одно целое число или один символ занимают в памяти ЭВМ два байта. В то же время для изображения символа достаточно одного байта. С целью экономии памяти машины при использовании символьных данных в языке Паскаль введено понятие строки.
- **Строка** - последовательность символов определенной длины. Элементы строки хранятся по два в двух байтах памяти ЭВМ.
- Переменные типа **string** могут быть объявлены следующим образом:

```
Var Str1: String[30];  
    Str2: String.
```



# ОПЕРАЦИЯ СЛОЖЕНИЯ

**1. Операция сложения (конкатенация)** позволяет строить из двух символьных строк третью, состоящих из символов первой, и второй строки. Обозначается эта операция знаком "+".

Пример Описываем строковые переменные

```
Var Str1, Str2, Str3: String[20].
```

```
Str1:= 'У Егорки';
```

```
Str2:='всегда отговорки';
```

```
Str3:=Str1+' '+Str2.
```

Строка Str3 имеет значение 'У Егорки всегда отговорки'.

**Примечание.** В данном примере итоговая строка может состоять максимум из 20 символов, если она будет состоять из большего числа, то будут взяты в качестве значения только первые 20, а остальные рассматриваться не будут.





# СРАВНЕНИЕ

Сравнение происходит посимвольно слева направо: сравниваются коды соответствующих символов до тех пор, пока не нарушится равенство, при этом сразу делается вывод о знаке неравенства. Две строки называются равными, если они равны по длине и совпадают посимвольно.

## Пример

'Balkon' < 'balkon' (Ord('B')<Ord('b'));

'balkon' > 'balken' (Ord('o')>Ord('e'));

“balkon” > “balk” длина первой строки больше 'balkon' > 'balk'

Можно использовать любые сравнения (>, <, =, <>, >=, <=) и их комбинации в условных операторах. Их результат — это одно из двух значений: True или False.

Для доступа к отдельному символу в строке необходимо указать имя строки и в квадратных скобках номер позиции элемента (символа) в строке.

При этом по отношению к отдельному символу строки возможны все те же операции, что и к переменной типа Char.



# ДЛИНА СТРОКИ - LENGTH(STR)

**Это количество введенных символов**, но она не может превышать максимально возможной длины (в описательной части). Это значение можно определить при помощи функции, результат которой целое число, равное количеству символов.

## Пример **Length(Str)**

```
Str1:='ABCDEFGH';
```

```
Str2:= 'Мама мыла раму';
```

```
k1:=Length(Str1);
```

```
k2:=Length(Str2).
```

В результате значения целых переменных будут равны: k1=8, k2=14.



# КОПИРОВАНИЕ - COPY(STR,N,M)

Функция **Copy(Str, n, m)**—начиная с позиции *n*, копируют *m* символов из строки *str*, при этом исходная строка не меняется. Результат этой функции можно присваивать другой строке или сразу выводить на экран.

## **Пример**      **Copy(Str,n,m)**

```
Srt1:='ABCDEFGH';
```

```
Str2:='abcdefgh';
```

```
Str3:=Copy(Str1, 4, 3);
```

**Результат: DEF**

```
Writeln(Str3);
```

```
Writeln(Copy(Str2, 4, 3));
```

**Результат: DEF**



# УДАЛЕНИЕ - DELETE (STR, N,M)

Для этого используется процедура **Delete (Str, n,m)**, которая вырезает из строки Str m символов, начиная с n-го, таким образом сама строка изменяется.

Пример Дан фрагмент программы:

```
Str1:='ABCDEFGH';
```

```
Delete(Str1, 3, 4);
```

```
Writeln(Str1);
```

После выполнения этих операторов из строки будут удалены четыре символа, начиная с третьего, то есть строка будет такой: **Str1='ABGH'**.



# ВСТАВКА - INSERT(STR1,STR2,N)

Это можно сделать, применяя процедуру **Insert(Str1,Str2,n)** — вставка строки Str1 в строку Str2, начиная с n-го символа, при этом первая строка остается такой же, как и была, а вторая получает новое значение.

## Пример

**Str1:='ABCDEFGH':**

**Str2:='abcdefgh';**

**Insert(Str1, Str2, 3);**

Результат: **Str2='abABCDEFGHcdefgh'.**





# ПОДСТРОКА

- Результатом определения позиции подстроки в строке функций — **целое число**, оно определяет номер первого элемента вхождения подстроки в строку. Если такой подстроки нет, то значение функции равно 0.
- **Примеры**

<b>Pos(str1, str2)</b>	<b>Ответы</b>
Str1:= 'CDE'; Str2:= 'ABCDEFGH';	
k1:=Pos(Str1, Str2);	K1=3
k2:=Pos(Str2, Str1);	K2=0

# ПРЕОБРАЗОВАНИЕ ЧИСЕЛ В СТРОКИ

- ✓ Надо заметить, что число 13 и строка 13 - это не одно и то же. Для работы с числами и строками применяются две процедуры.
- ✓  $\text{Str}(N,S)$  — переводит числовое значение  $N$  в строковое и присваивает результат строке  $S$ , причем, можно переводить *как целые числа, так и вещественные*.
- ✓  **$\text{Str}(N,S)$**  – число  **$N$**  преобразуется в строку  **$S$**
- ✓ Пример  
 $\text{Str}(1234, S)$   
 $S='1234';$



**Распечатайте символы соответствующие различным кодам и одновременно определите, какой максимальный код можно использовать**

**Program** prim1;

Uses Crt;

Var i,n: byte; {описание данных}

**Begin**

clrscr; n:=255;

for i:= 0 to n do

writeln(i, ' ', chr(i));

**end.**

- ✓ Возьмем  $N = 300$  заметим, что после 255 символы уже не появляются. Таким образом, мы выяснили, что максимальный код может быть 255.

Вывод  
СИМВОЛОВ  
рядами

программа1

Вывод  
СИМВОЛОВ  
пирамидой

программа2

программа

исполнение



**Ввести слово и:** а) напечатать его по вертикали,  
б) переставить буквы слова в обратном порядке.

- а) Будем рассматривать слово слева направо и делать посимвольную вырезку (*копию*), а затем печатать вырезанный символ.
- б) Двигаясь справа налево и делая посимвольную вырезку, мы получим нужное слово.

### Программа 1

### Исполнение 1

```
Program p1;  
uses crt;  
var i: byte; a: string;  
begin  
  clrscr;  
  writeln('a='); readln(a) ;  
  for i:=1 to length(a) do  
    writeln(copy(a,i,1));  
  readln;  
end.
```

### Программа 2

### Исполнение 2

```
Program p2;  
uses crt;  
var i: byte; a: string;  
begin  
  clrscr;  
  writeln('a='); readln(a) ;  
  for i:=length(a) downto 1 do  
    writeln(copy(a,i,1));  
  readln;  
end.
```



# ДОМАШНЕЕ ЗАДАНИЕ

Ответить на вопросы.

1. Что является результатом каждой функции, предназначенной для выполнения операций над строками?
2. Что является результатом действия функции `LEN(a)`, `(length(a))`?
3. Какие функции позволяют определить код по символу и символ по коду?
4. Для каждого ли символа имеется код?
5. Может ли значение функции `LEN(a)`, `(length(a))` быть равным 300?
6. Верно ли что символы "А" и "а" имеют одинаковые коды?

