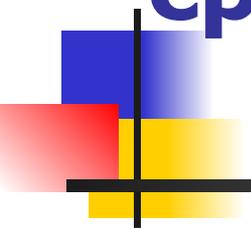
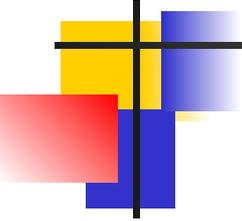


Обработка символов и строк средствами языка Турбо-Паскаль

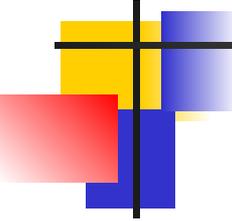


План лекции



- 1. Символьные и строковые константы и переменные.**
- 2. Операции над строками.**
- 3. Встроенные функции и процедуры обработки строк.**
- 4. Примеры программ обработки строк**

Символьные и строковые константы и переменные

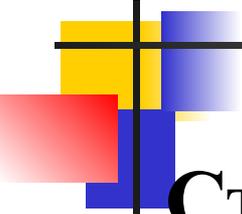


Символьные константы – это любой символ персонального компьютера, заключенный в апострофы.

Строковые константы – определяются последовательностью произвольных символов, заключенных в апострофы.

Символьный тип (char). Значениями символьного типа является множество всех символов ПК. Для кодировки используется код ASCII (American Standart Code for Information Interchange – американский стандартный код для обмена информацией).

Строковые переменные



Строка — это последовательность символов.

При использовании в выражениях строка обязательно заключается в апострофы.

Количество символов в строке (длина строки) может динамически изменяться от 0 до 255. Определение строкового типа устанавливает максимальное количество символов, которое может содержать строка.

Формат:

1-ый способ:

Type

<имя типа> = string [максимальная длина строки];

Var

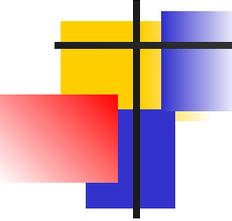
<идентификатор> : <имя типа>;

2-ой способ:

Var

<идентификатор> : string [максимальная длина строки];

Пример:



Type

```
Stroka = string [50];
```

Var

```
St : Stroka;
```

```
St1: string;
```

```
St2 : string [25];
```

Const

```
St : string = 'Сегодня хорошая погода !';
```

Array [0..n] of char

Например:

Var

st : string;

Begin

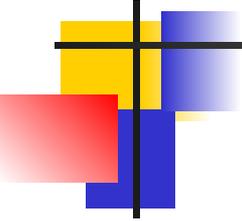
.....

if st [5] = 'A' then

.....

End.

Операции над строками



Операция сцепления (+) применяется для сцепления нескольких строк в одну результирующую строку.

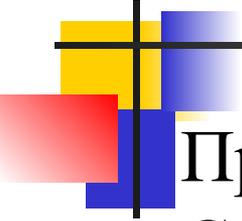
Выражение

'E'+ 'C'+ ' 18' + '40'

Результат

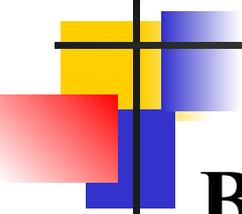
'ЕС 1840'

Операции отношения (=, <>, <, >, >=, <=)



Проводят сравнение двух строковых операндов. Сравнение строк производится слева направо до первого несовпадающего символа, и та строка считается больше, в которой первый несовпадающий символ имеет больший номер в стандартной таблице обмена информацией. Результат выполнения операций отношения над строковыми операндами всегда имеет булевский тип и принимает значение True, если выражение истинно, и False, если выражение ложно.

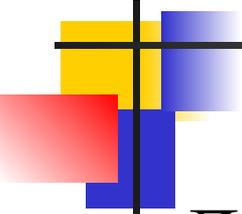
Операции отношения (=, <>, <, >, >=, <=)



Выражение	Результат
'COSM1' < 'COSM2'	True

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

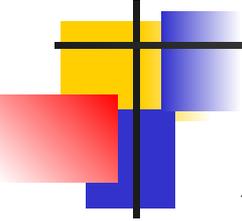
Строки считаются равными, если они полностью совпадают по длине и содержат одни и те же символы.



Допускается смешение в одном выражении операндов строкового и символьного типа.

Если при этом символьной переменной присваивается значение строкового типа, длина строки должна быть равна единице, иначе возникает ошибка выполнения.

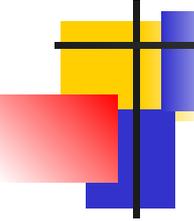
Процедуры обработки строк



Delete (St, Poz, N) — удаление N символов строки St, начиная с позиции Poz.

Значение St	Выражение	Результат
'абвгде'	Delete (St, 4, 2)	'абве'

Процедуры обработки строк



Insert (Str1, Str2, Poz) — вставка строки Str1 в строку Str2, начиная с позиции Poz.

Пример:

```
Var S1, S2, S3 : string[11];
```

...

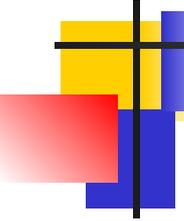
```
S1 := ' EC ';
```

```
S2 := 'ЭВМ1841';
```

```
S3 := Insert (S1, S2, 4);
```

В результате выполнения последнего выражения значение строки S3 станет равным 'ЭВМ EC 1841'.

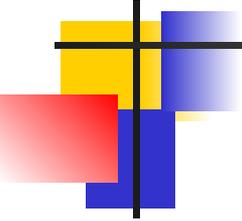
Процедуры обработки строк



Str (I, St) — преобразование числового значения величины **I** (целого или вещественного типа) и помещение результата в строку **St**.

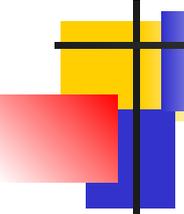
После **I** может записываться формат, аналогичный формату вывода. Если в формате указано недостаточное для вывода количество разрядов, поле вывода расширяется автоматически до нужной длины.

Str (I, St)



Значение I	Выражение	Результат
1500	Str (I:6, St)	' 1500'

Процедуры обработки строк

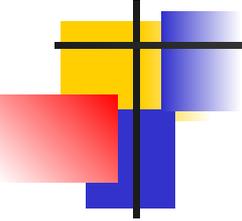


Val (St, I, Cod) — преобразует значение St в величину целочисленного или вещественного типа и помещает результат в I.

Значение St не должно содержать незначащих пробелов в начале и в конце.

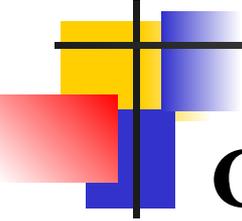
Cod — целочисленная переменная. Если во время операции преобразования ошибки не обнаружено, значение Cod равно нулю, если ошибка обнаружена (например, литерное значение переводится в цифровое), Cod будет содержать номер позиции первого ошибочного символа, а значение I не определено.

Val (St, I, Cod)



Значение St	Выражение	Результат
'1450'	Val (St, I, Cod)	1450 Cod=0

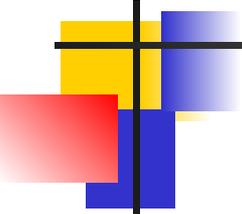
Функции для обработки строк



Copy (St, Poz, N) — выделяет из строки St подстроку длиной N символов, начиная с позиции Poz.

Значение St	Выражение	Результат
'ABCDEFGFG'	Copy (St, 2, 3)	'BCD'

Функции для обработки строк



Concat (Str1, Str2, ..., StrN) — выполняет сцепление строк Str1, Str2, ..., StrN в том порядке, в каком они указаны в списке параметров.

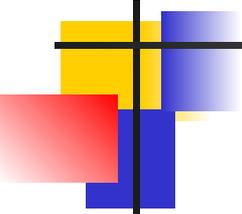
Выражение

Concat ('AA', 'XX', 'Y')

Результат

'AAXXY'

Функции для обработки строк



Length (St) — вычисляет длину в символах строки St.

Значение St

'123456789'

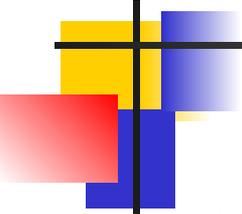
Выражение

Length (St)

Результат

9

Функции для обработки строк

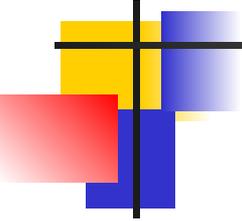


Pos (Str1, Str2) — обнаруживает первое появление в строке Str2 подстроки Str1.

Результат имеет целочисленный тип и равен номеру той позиции, где находится первый символ подстроки Str1. Если в Str2 подстроки Str1 не найдено, результат равен 0.

Значение Str	Выражение	Результат
'abcdef'	Pos('de', Str)	4
'abcdef'	Pos('r', Str)	0

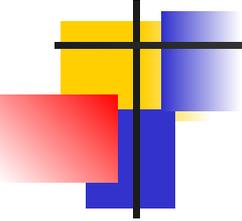
Функции для обработки строк



UpCase(Ch) - преобразует строчную букву в прописную.

Параметр и результат имеют литерный тип.
Обрабатывает буквы только латинского алфавита.

Примеры программ обработки строк



Дана строка. Слова в строке разделены одним пробелом, в конце строки точка.
Распечатать слово максимальной длины.

Program maxdl;

var

st, st1 : string; n, i, ns, ds, d, d_max : integer;

Begin

Writeln ('Введите строку');

Readln (St);

d := length (st); ns := 1; d_max := 0;

for i := 1 to d do

if ((St [i] = ' ') or (St [i] = '.')) Then

begin

ds := i - ns;

If ds > d_max then

begin

d_max := ds;

st1 := copy (st, ns, ds);

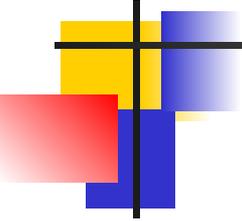
end;

ns := i + 1;

end;

Writeln ('Слово максимальной длины: ', st1);

End.



Дан текст Txt.

Заменить любое вхождение строки St1 на строку St2.

program zam; (*замена *)

var

d2,k:integer; s,str1, str2,txt:string;

begin

writeln('Введите исходный текст:'); readln (txt);

writeln ('Введите замещающий текст:'); readln (str1);

**writeln ('Введите текст, вместо которого нужна замена:');
readln (str2);**

d2 := length(str2);

repeat

k := pos(str2, txt);

if k <> 0 then

begin

delete (txt, k, d2);

insert(str1, txt, k);

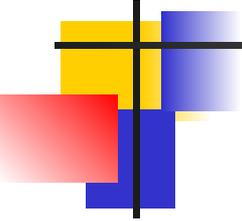
end;

until k = 0;

writeln('Полученный текст: ');

writeln (txt);

end.



Дан текст Txt.

Удалить строку St из текста

Program udal; (* удаление *)

var

str, txt : string; n, k : integer;

begin

 Writeln ('Введите исходный текст: '); readln (txt);

 writeln ('Введите удаляемый текст: '); readln (str);

 n := length(str);

 repeat

 k := pos(str, txt);

 if k <> 0 then

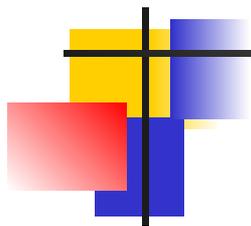
 delete (txt, k, n);

 until k = 0;

 writeln('Полученный текст: ');

 writeln (txt);

end.



Спасибо за внимание!