

Практическая работа № 6

Работа со строками на Паскале

Цель работы: Изучить строковый тип `string`. Изучить стандартные функции работы со строками, научиться выполнять стандартные операции со строками.

Содержание работы.

Основные понятия.

1 Строка представляет собой особую форму одномерного массива символов, которая имеет существенное отличие. Массив символов имеет фиксированную длину (количество элементов), которая определяется при описании. Строка имеет две разновидности длины:

- Общая длина строки, которая характеризует размер памяти, выделяемый строке при описании;
- Текущая длина строки (всегда меньше или равна общей длине), которая показывает количество смысловых символов строки в каждый конкретный момент времени.

2 Строка в Паскале – упорядоченная последовательность символов. Количество символов в строке называется ее длиной. Длина строки в Паскале может лежать в диапазоне от 0 до 255. Каждый символ строковой величины занимает 1 байт памяти и имеет числовой код в соответствии с таблицей кодов ASCII.

3 Для описания строковых переменных в Паскале существует предопределенный тип `string`. В общем виде описание строковой переменной будет выглядеть следующим образом:

```
Var <имя_переменной>: string[<максимальная длина строки>]
```

4 Операция слияния (сцепления, конкатенации) применяется для соединения нескольких строк в одну, обозначается знаком «+». Операция слияния применима для любых строковых выражений, как констант, так и переменных.

5 Операции отношения позволяют сравнивать строки на отношение равенства (=), неравенства (<>), больше (>), меньше (<), больше или равно (>=), меньше или равно (<=). В результате сравнения двух строк получается логическое значение (`true` или `false`). Сравнение строк производится слева направо посимвольно до первого несовпадающего символа, большей считается та строка, в которой первый несовпадающий символ имеет больший код в таблице кодировки. Если строки имеют различную длину, но в общей части символы совпадают, считается, что короткая строка меньше. Строки равны, если они имеют равную длину и соответствующие символы совпадают.

6 Стандартные функции для работы со строками в Паскале

– `Сору (S, poz, n)` выделяет из строки `S`, начиная с позиции `poz`, подстроку из `n` символов. Здесь `S` – любое строковое выражение, `poz`, `n` – целочисленные выражения.

- Concat (s_1, s_2, \dots, s_n) выполняет слияние строк s_1, s_2, \dots, s_n в одну строку.
- Length(S) определяет текущую длину строкового выражения S . Результат – значение целого типа.
- Pos(sub S, S) определяет позицию первого вхождения подстроки sub S в строку S . Результат – целое число, равное номеру позиции, где находится первый символ искомой подстроки. Если вхождение подстроки не обнаружено, то результат функции будет равен 0.

7 Стандартные процедуры для работы со строками в Паскале:

- Delete (S, roz, n) удаляет из строки S , начиная с позиции roz, подстроку из n символов. Здесь S – строковая переменная (в данном случае нельзя записать никакое другое строковое выражение, кроме имени строковой переменной, т.к. только с именем переменной связана область памяти, куда будет помещен результат выполнения процедуры); roz, n – любые целочисленные выражения.
- Insert(sub S, S, roz) вставляет в строку S , начиная с позиции roz, подстроку sub S . Здесь sub S – любое строковое выражение, S – строковая переменная (именно ей будет присвоен результат выполнения процедуры), roz – целочисленное выражение.

8 Процедуры преобразования типов в Паскале:

- Str(x, S) преобразует число x в строковый формат. Здесь x – любое числовое выражение, S – строковая переменная. В процедуре есть возможность задавать формат числа x . Например, str($x: 8: 3, S$), где 8 – общее число знаков в числе x , а 3 – число знаков после запятой.
- Val(S, x, kod) преобразует строку символов S в число x . Здесь S – строковое выражение, x – числовая переменная (*именно туда будет помещен результат*), kod – целочисленная переменная (*мина integer*), которая равна номеру позиции в строке S , начиная с которой произошла ошибка преобразования, если преобразование прошло без ошибок, то переменная kod равна 0.

Задание

- 1 Составить и записать алгоритм решения задачи в графическом и словесно-формульном виде
- 2 Написать программу на языке Паскаль
- 3 Ввести программу и запустить её на исполнение.
- 4 Протестировать алгоритм с различными данными.
- 5 Отладить программу (найти и исправить возможные ошибки).
- 6 Проанализировать программу и сделать вывод.

Исходные данные:

Дана строка. Определить, сколько раз в нее входит группа букв ешь.

Решение:

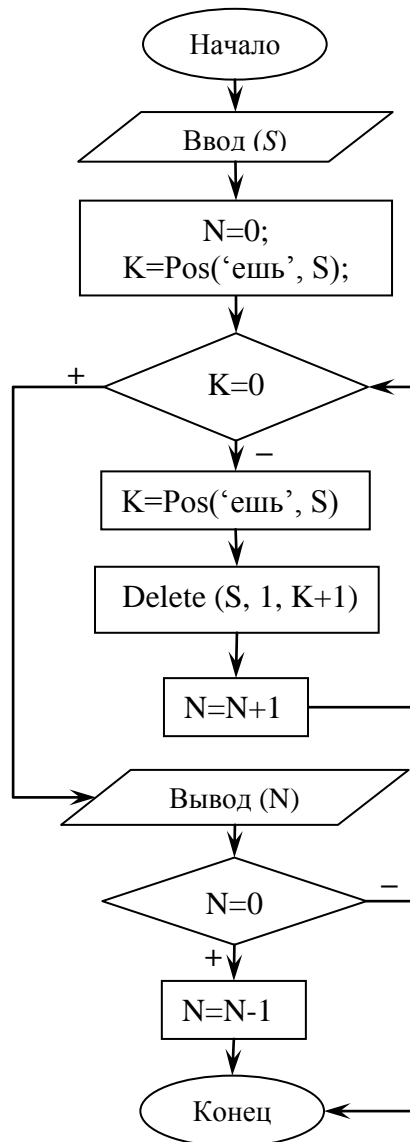
- 1 Алгоритм задачи:

а) для поиска первого вхождения сочетания «ешь» используем функцию Pos, она покажет первый символ к;

б) дальше надо отрезать от строки символы с первого по к+1 и снова воспользоваться указанной функцией;

в) выполнять эти действия, пока значение Pos не станет нулевым, но когда это произошло, N все равно увеличивается на 1, поэтому печатаем N – 1.

Получим алгоритм:



1. Начало;
2. Ввод (S);
3. N=0;
4. K=Pos('ешь', S);
5. Пока K <> 0 выполнять
нц
K=Pos('ешь', S);
Delete (S, 1, K+1);
N=N+1;
кц
6. Если N>0 то N=N+1;
7. Вывод (N-1);
8. Конец.

2 Программа на языке Паскаль:

```
program stroka1;  
var K, N:integer;  
    S:string;  
begin  
    writeln('Введите текст');  
    readln(S);  
    N:=0;
```

```

K:=Pos('ешь', S);
while K<>0 do
begin
  K:=Pos('ешь', S);
  N:=N+1;
  Delete (S, 1, K+1);
end;
if N>0 then N:=N-1;
writeln('В тексте ешь встречается ', N, ' раз');
end.

```

3 Программный код для запуска 4 Тестирование программы с различными данными:

```

*Program1.pas
program stroka1;
var K, N:integer;
    S:string;
begin
  writeln('Введите текст');
  readln(S);
  N:=0;
  K:=Pos('ешь', S);
  while K<>0 do
  begin
    K:=Pos('ешь', S);
    N:=N+1;
    Delete (S, 1, K+1);
  end;
  if N>0 then N:=N-1;
  writeln('В тексте ешь встречается ', N, ' раз');
end.

```

```

| <
Введите текст
переживаешь? передаешь? садись на диету!
В тексте ешь встречается 2 раз

| \
Введите текст
Раз, два, три, четыре, пять! Вышел зайчик погулять
В тексте ешь встречается 0 раз

```

6 Программа выдает верное значение, если заданная комбинация букв присутствует в тексте и если не присутствует.

- 19 Дана строка. Определить, сколько в ней «,»
- 20 Дана строка. Определить, сколько в ней «;»
- 21 Удалить часть символьной строки в скобках (вместе со скобками)
- 22 Удалить часть символьной строки в кавычках (вместе с кавычками)
- 23 Дана строка, содержащая кавычки. Выдать сообщение, правильно ли расставлены кавычки (количество открывающихся соответствует количеству закрывающихся)
- 24 Дана строка. Найти количество слов, начинающихся с буквы А
- 25 Дана строка. Найти количество слов, начинающихся с буквы А
- 26 Дана строка. Заменить первую букву слов, начинающихся с буквы А, на малую букву
- 27 Дана строка. Заменить последнюю букву слов, заканчивающихся на е, на большую букву
- 28 Дана строка. Найти количество букв к.
- 29 Дана строка. Определить, сколько раз в нее входит группа букв ель
- 30 Дана строка. Определить, сколько раз в нее входит группа букв ого