

Государственное бюджетное образовательное учреждение
дополнительного профессионального образования
«НИЖЕГОРОДСКИЙ ИНСТИТУТ РАЗВИТИЯ ОБРАЗОВАНИЯ»

М. Ю. ВТЮРИН, М. И. ГОЛУНОВА, Э. В. ЕФИМОВА

ПРАКТИКУМ
по ПРОГРАММИРОВАНИЮ
на ЯЗЫКЕ ПАСКАЛЬ



Часть III

Нижний Новгород
Нижегородский институт развития образования
2015

УДК 372.016:004

ББК 74.263.2

В87

Авторы

М. Ю. Втюрин, канд. физ.-мат. наук, доцент кафедры теории и методики обучения информатике ГБОУ ДПО НИРО;

М. И. Голунова, канд. пед. наук, зав. кафедрой теории и методики обучения информатике ГБОУ ДПО НИРО;

Э. В. Ефимова, канд. пед. наук, доцент кафедры теории и методики обучения информатике ГБОУ ДПО НИРО

*Рекомендовано к изданию
научно-методическим экспертным советом
ГБОУ ДПО НИРО*

Втюрин, М. Ю.

В87 Практикум по программированию на языке Паскаль / М. Ю. Втюрин, М. И. Голунова, Э. В. Ефимова. — Н. Новгород : Нижегородский институт развития образования, 2015. — 21 с.

ISBN 978-5-7565-0671-6

Предлагаемое пособие предназначено для освоения работы со строковым типом данных и содержит примеры использования основных процедур и функций для работы со строками, а также упражнения для самостоятельной работы. Алгоритмы представлены в виде фрагментов программ на языке программирования Паскаль.

Издание адресовано слушателям курсов повышения квалификации в Нижегородском институте развития образования и призвано помочь учителям информатики освоить основы алгоритмизации и программирования с помощью языка Паскаль.

УДК 372.016:004

ББК 74.263.2

ISBN 978-5-7565-0671-6

© ГБОУ ДПО «Нижегородский институт развития образования», 2015

Предлагаемое пособие предназначено для освоения основных процедур и функций обработки строковых данных. Изложение материала основано на использовании языка программирования Паскаль. В первой главе пособия приведено описание данных строкового типа, основные процедуры и функции обработки строковых переменных, даны примеры программ, реализующих их. Во второй главе практикума рассматриваются два алгоритма выделения слов заданной строки. Первый алгоритм приведен для случая, когда слова строки разделены одним или несколькими пробелами. Второй алгоритм выделения слов строки приведен для случая, когда слова разделены наперед заданным фиксированным набором символов. Изложение материала ведется по принципу — сначала описываются исходные данные, указывается требуемый результат выполнения алгоритма, перечисляются (при необходимости) вспомогательные переменные, а затем приводится алгоритм в виде фрагмента программы на языке программирования Паскаль, после чего рассматривается пример решения конкретной задачи с применением соответствующего алгоритма. В конце каждой главы приведены упражнения для самостоятельной работы.

Приступая к работе, сначала изучите приведенные в соответствующем разделе

программы, попытайтесь реализовать их самостоятельно на компьютере, отладить и протестировать с различными исходными данными. После этого рекомендуется самостоятельно приступать к решению задач из упражнений по изучаемой теме.

Пособие предназначено, в первую очередь, для работы на практических занятиях, а также для самостоятельной практической работы. Более полную информацию о языке программирования Паскаль, среде программирования и принципах структурного программирования читатель может получить из источников, представленных в разделе «Литература» данного издания.

Строки в языке Паскаль

Строка — последовательность символов. При использовании в выражениях строка заключается в апострофы. Длина строки до 255 символов.

var <имя_переменной>: **string**[n];

n — длина строковой переменной; если n не указано, то длина строки 255 символов.

Пример:

```
var
  x: string[7];
  st: string;
```

В языке Паскаль строку можно рассматривать как массив символов и получить доступ к отдельному символу по его номеру в строке. Например, для строки st запись st[i] означает символ этой строки, расположенный на i-й позиции.

Строки, не содержащие ни одного символа, в Паскале обозначаются двумя подряд идущими апострофами.

Пример:

```
st:=""; {Пустая строка (аналог «числового нуля») }
```

Строковые величины можно сравнивать между собой с помощью операций отношения <, >, <=, >=, =, <>.

Процедуры и функции для обработки строк

1. Определение длины L строки S:

```
L:=Length(S);
```

Пример:

```
S:='Мама мыла раму';  
L:=Length(S);  
Результат: переменной L присвоится значение 14.
```

2. Выделение подстроки S1 строки S длиной L (или до конца) начиная с K-того символа

```
S1:=Copy(S, K, L);
```

Пример:

```
S:='Мама мыла раму';  
S1:=Copy(S, 5, 6);  
Результат: переменной S1 присвоится значение ' мыла '.
```

3. Поиск подстроки S1 в строке S

```
P:=Pos(S1, S);
```

Пример:

```
S:='Мама мыла раму';  
S1:='мыла';  
P:=Pos(S1, S);  
Результат: переменной P присвоится значение 6.
```

4. Удаление фрагмента строки S длиной L начиная с позиции K

```
Delete(S, K, L);
```

Пример:

```
S:='Мама мыла раму';  
Delete(S, 6, 7);  
Результат: переменной S присвоится значение 'Мама му'.
```

Примеры программ

Пример: подсчитать количество цифровых символов в заданной строке текста.

```
program n1;
var
  S: string; {исходная строка}
  i:integer; {параметр}
  k:integer; {количество цифровых символов}
begin
  {Ввод исходных данных}
  writeln ('Исходная строка: ');
  readln(S);
  {Решение}
  k:=0;
  for i:=1 to length(S) do
    if (S[i]='0') or (S[i]='1') or (S[i]='2') or
      (S[i]='3') or (S[i]='4') or (S[i]='5') or
      (S[i]='6') or (S[i]='7') or (S[i]='8') or
      (S[i]='9') then
      k:=k+1;
  {Вывод результатов}
  writeln ('Результат: ');
  if k>0 then
    writeln ('Количество цифр ',k)
  else
    writeln ('Цифр в строке нет');
end.
```

Пример: удалить из заданной строки все запятые.

```
program n2;
var
  s1:string; {исходная строка}
  i:integer; {параметр}
  s2:string; {строка — решение}
```

```

begin
  {Ввод исходных данных}
  writeln ('Исходная строка: ');
  readln (s1);
  {Решение задачи}
  {формирование новой строки без запятых}
  s2:=""; {пустая строка}
  for i:=1 to length(s1) do
    if s1[i]<>',' then
      s2:=s2+s1[i];
  {Вывод результатов}
  writeln ('Строка — решение: ',s2);
end.

```

Пример: задан массив слов. Объединить все слова в единую строку, разделяя соседние слова пробелом.

```

program n3;
const
  nmax=50; {максимальный размер массива}
var
  n:integer; {количество слов в массиве}
  i:integer; {параметр}
  w:array [1..nmax] of string; {массив слов}
  s: string; {строка слов}
begin
  {Ввод исходных данных}
  writeln ('Количество слов: ');
  readln (n);
  for i:=1 to n do
    begin
      writeln ('Слово: ');
      readln (w[i]);
    end;
  {Решение задачи}
  s:=""; {пустая строка}

```



```
for i:=1 to n do
  s:=s+w[i]+' ';
  {Вывод результатов}
  writeln ('Строка слов: ',s);
end.
```

Упражнения

1. Найти позицию первого появления заданного символа в заданной строке.
2. Найти позицию третьего появления заданного символа в заданной строке.
3. Удалить все символы пробелов в строке.
4. Подсчитать, сколько раз среди символов строки встречается символ x .
5. Подсчитать наибольшее число подряд идущих пробелов в строке.
6. Заменить все восклицательные знаки в строке точками.
7. Удалить символы строки, стоящие перед символом $*$.
8. Распечатать все символы строки, расположенные между двумя запятыми.
9. Удалить все символы строки, не являющиеся буквами.
10. Исключить все символы строки между круглыми скобками. Рассмотреть случай вложенных скобок.
11. Распечатать все символы строки, расположенные между двумя запятыми.
12. Подсчитать число вхождений в строку группы символов abc .
13. Удалить из каждой группы идущих подряд цифр строки нули.
14. Удалить из строки все запятые, предшествующие первой точке.

15. Проверить, является ли строка палиндромом.
16. Преобразовать строку так, чтобы в ней сначала шли все цифры, а затем все буквы исходной строки.
17. Подсчитать частоту встречаемости каждой цифры в строке.
18. Подсчитать наибольшее число букв А, идущих подряд в строке.
19. Все запятые заданной строки окаймить пробелами.
20. Преобразовать строку, удалив из нее каждый символ * и повторив каждый символ, отличный от *.
21. Заменить все разделители («. , - ! ? ; :») пробелами.
22. Найти номер позиции первого появления частицы *не* в строке. Слева и справа от частицы должны быть пробелы.
23. Инvertировать строку (расположить символы строки в обратном порядке).
24. Разделить исходную строку на несколько строк длины K.
25. Заменить все русские буквы в строке их порядковыми номерами в русском алфавите.
26. Определить, содержит ли строка дату в виде xx.xx.xx

Алгоритмы выделения слов строки

1. Разделителем слов является один или несколько пробелов, любое количество пробелов может стоять в начале или в конце строки

⇒ **Исходные данные:**

Str — строка слов, разделенных пробелами.

⇒ **Требуемый результат:**

kw — количество слов в строке Str;

w(kw) — массив слов строки Str.

⇒ **Рабочая переменная:**

p — позиция очередного пробела в строке.

⇒ **Фрагмент программы:**

```
Str:=Str+' '; {добавляем завершающий пробел}
```

```
kw:=0;
```

```
while Str<>" do {пока строка не пустая}
```

```
begin
```

```
  p:=Pos(' ',Str); {определяем позицию пробела}
```

```
  if p>1 then {если пробел не первый}
```

```
  begin
```

```
    kw:=kw+1;
```

```
    w[kw]:=Copy(Str, 1, p-1); {копируем все до пробела (слово)}
```

```
  end;
```

```
  Delete(Str, 1, p); {удаляем все до пробела и пробел}
```

```
end;
```

Пример: задана строка, состоящая из слов, между которыми, а также в начале и конце строки, может быть любое количество пробелов. Выделить все слова, начинающиеся на гласную латинскую букву, и поместить их в массив.

```

program n4;
const
  nmax=30; {максимальный размер массива}
var
  s:string;           {исходная строка}
  kw:integer;        {количество слов в строке}
  p:integer;         {позиция пробела}
  w:array[1..nmax] of string; {массив слов строки}
  i:integer;         {параметр}
  k:integer;         {количество искомым слов}
  wa:array[1..nmax] of string; {слова на гласную букву}
begin
  {Ввод исходных данных}
  writeln ('Исходная строка: ');
  readln (s);
  {Решение задачи}
  s:=s+' ';
  kw:=0;              {количество слов в строке}
  while s<>" do
    begin
      p:=pos(' ',s); {номер позиции, на которой первый
                     раз встречается пробел}
      if p>1 then
        begin
          kw:=kw+1;
          w[kw]:=copy(s,1,p-1);
          end;
        delete(s, 1, p);
      end;
    {Выделяем слова, начинающиеся на гласную латинскую букву}
  
```

```

k:=0;
for i:=1 to kw do
  if (copy(w[i],1,1)='a') or (copy(w[i],1,1)='e') or (copy
(w[i], 1,1)='y') or (copy(w[i],1,1)='u') or (copy
(w[i],1,1)='o') or (copy(w[i],1,1)='i') then
  begin
    k:=k+1;
    wa[k]:=w[i];
  end;
  {Вывод результатов}
writeln ('Результат: ');
if k=0 then
  writeln ('Слов на гласную букву в строке нет')
else
  for i:=1 to k do
    writeln (wa[i]);
end.

```

Пример: в словах заданной строки, которые содержат заданное ключевое слово, найти самое длинное слово. Между словами, а также в начале и в конце строки может быть любое количество пробелов.

Контрольный пример:

Стр о к а: мирный род помирить родина народ миры.

К л ю ч: мир.

Р е з у л ь т а т: помирить.

```

program n5;
const
  nmax=40;           {максимальный размер
                    массива}

var
  s:string;         {заданная строка}
  sk:string;       {ключ}
  kw:integer;      {количество слов в строке}
  p:integer;       {позиция пробела}
  w:array[1..nmax] of string; {массив слов строки}

```

i:integer;	{параметр}
lmax:integer;	{длина искомого слова}
nom:integer;	{номер искомого слова в массиве слов}

```

begin
  {Ввод исходных данных}
  writeln ('Исходная строка: ');
  readln (s);
  writeln ('Ключ: ');
  readln (sk);
  {Алгоритм выделения слов строки}
  s:=s+' ';
  kw:=0;
  while s<>" do
    begin
      p:=pos(' ',s);
      if p>1 then
        begin
          kw:=kw+1;
          w[kw]:=copy(s, 1, p-1);
        end;
      delete(s, 1, p);
    end;
  {Алгоритм поиска максимального элемента (условный)}
  lmax:=0;
  nom:=0;
  for i:=1 to kw do
    if pos(sk,w[i])>0 then {если ключ в слове}
      if length(w[i])>lmax then {слово максимальной длины}
        begin
          lmax:=length(w[i]);
          nom:=i;
        end;
  end;

```

```

{Вывод результатов}
writeln ('Результат: ');
if nom=0 then
  writeln ('В строке нет слова с заданным ключом')
else
  writeln ('Искомое слово: ',w[nom]);
end.

```

2. Разделителем слов являются знаки препинания, заданные в виде набора символов

▣ Исходные данные:

Str — строка, состоящая из слов, разделенных заданным набором символов-разделителей;

Dlm — строка символов-разделителей («. , ! ? - : ;»).

▣ Требуемый результат:

kw — количество слов в строке Str;

w(kw) — массив слов строки Str;

d(kw) — массив разделителей слов строки Str.

▣ Фрагмент программы:

```

kw:=0;
while (Str<>") do
begin
  kw:=kw+1;
  w[kw]:= ""; {сумматор символов очередного
                слова}
  d[kw]:= ""; {сумматор разделителей после
                очередного слова}
  while (pos(Str[1],Dlm)=0) and (Str<>") do
  {Пока первый символ строки Str — не
  разделитель}
  begin
    w[kw]:=w[kw]+Str[1];
    delete(Str, 1, 1);

```

```

end;
while (pos(Str[1],Dlm)>0) and (Str<>") do
begin
d[kw]:=d[kw]+Str[1];
delete(Str, 1, 1);
end;
end;

```

Пример: задана строка, состоящая из слов. Разделителями слов являются знаки препинания, заданные в виде набора символов («. , ! ? - : ;») или набор знаков препинания (например несколько пробелов или несколько точек). Выделить все слова и поместить их в массив слов. Выделить разделители слов и поместить их в массив разделителей.

```

program n6;
const
  nmax=30; {максимальный размер массива}
  Dlm=' .,!?-:;'; {строка символов-разделителей}
var
  Str:string;           {исходная строка}
  kw:integer;          {количество слов в строке}
  w:array[1..nmax] of string; {массив слов строки}
  d:array[1..nmax] of string; {массив разделителей слов}
  i:integer;
begin
  {Ввод исходных данных}
  writeln ('Исходная строка: ');
  readln (Str);
  {Решение задачи}
  kw:=0;
  while (Str<>") do
  begin
    kw:=kw+1;
    w[kw]:="; {сумматор символов очередного слова}

```



```

d[kw]:= ""; {сумматор разделителей после очеред-
              ного слова}
while (pos(Str[1],Dlm)=0) and (Str<>") do
{Пока первый символ строки Str — не разделитель}
begin
  w[kw]:=w[kw]+Str[1];
  delete (Str,1,1);
end;
while (pos(Str[1],Dlm)>0) and (Str<>") do
begin
  d[kw]:=d[kw]+Str[1];
  delete(Str, 1, 1);
end;
end;
{Вывод результата}
{Вывод массива слов}
for i:=1 to kw do
  writeln(w[i]);
{Вывод массива разделителей}
for i:=1 to kw do
  writeln(d[i]);
end.

```

Упражнения

1. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Сформировать массив слов строки.

2. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Найти самое длинное слово строки.

3. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Найти самое короткое слово строки.

4. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Вывести на экран слова строки, упорядочив их по алфавиту.

5. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Вывести на экран слова строки, упорядочив их по возрастанию количества символов.

6. Дан словарь в виде списка (массива) слов и строка, слова в которой разделены пробелами. Определить, какие слова из словаря встречаются в данной строке.

7. Дан словарь в виде списка (массива) слов и строка, слова в которой разделены пробелами. Вывести те слова строки, которых нет в словаре.

8. Слова в строке разделены символами «пробел, . - ! ? : ;». Сформировать массив слов строки.

9. Слова в строке разделены разделителями «пробел, . - ! ? : ;». Сформировать массив слов текста.

10. Задан текст в виде массива строк. Предложения в тексте завершаются одним из символов «. ! ?». Сформировать массив предложений.

11. Задан текст в виде массива строк. Слова в строке разделены символами «пробел, . - ! ? : ;». Сформировать частотный словарь текста — список слов текста в алфавитном порядке с указанием частоты встречаемости каждого слова в тексте.

12. Дан текст в виде массива строк. Отформатиро-

вать текст на заданную длину строки с выравниванием по левому краю.

13. Дан текст в виде массива строк. Отформатировать текст на заданную длину строки с выравниванием по правому краю.

14. Дан текст в виде массива строк. Абзацы в тексте начинаются со строки, в начале которой не менее пяти пробелов. Распечатать абзацы в порядке убывания количества строк в абзацах.

15. Дан текст на некотором языке и алфавит этого языка. Распечатать слова текста в алфавитном порядке (по алфавиту заданного языка).

1. *Алексеев, Е. Р.* Турбо Паскаль 7. 0 / Е. Р. Алексеев, О. В. Чеснокова, В. Н. Павлыш, Л. В. Славинская. — М. : АСТ : НТ Пресс, 2004.
2. *Громницкий, В. С.* Сборник задач по программированию / В. С. Громницкий, В. А. Гусев, Н. А. Паршин. — Н. Новгород : Издательство ННГУ, 2005.
3. *Дагене, В. А.* 100 задач по программированию : книга для учащихся / В. А. Дагене, Г. К. Григас, К. Ф. Аугутис. — М. : Просвещение, 1993.
4. *Кетков, Ю. Л.* Свободное программное обеспечение: FREE PASCAL для студентов и школьников / Ю. Л. Кетков, А. Ю. Кетков. — СПб. : БХВ-Петербург, 2011.
5. *Культин, Н. Б.* Turbo Pascal в задачах и примерах / Н. Б. Культин. — СПб. : БХВ-Петербург, 2007.
6. *Павловская, Т. А.* Паскаль. Программирование на языке высокого уровня : практикум / Т. А. Павловская. — СПб. : Питер, 2006.
7. *Пильщиков, В. Н.* Сборник упражнений по языку Паскаль : учебное пособие для вузов / В. Н. Пильщиков — М. : Наука, 1989.
8. Информационный портал для разработчиков Free Pascal [Электронный ресурс]. — Режим доступа URL: www.freepascal.ru.

Содержание _____

Введение

⇒ 3

Строки в языке Паскаль

⇒ 5

Алгоритмы выделения слов строки

⇒ 11

Литература

⇒ 20

Учебное издание

*ВТЮРИН Максим Юрьевич, ГОЛУНОВА Марина Ивановна,
ЕФИМОВА Элеонора Владимировна*

**ПРАКТИКУМ
по программированию на языке Паскаль**



Часть III

Редактор *И. М. Морева*
Компьютерная верстка *Л. И. Половинкиной*

Оригинал-макет подписан в печать 15.12.2015 г.
Формат $60 \times 84 \frac{1}{16}$. Бумага офсетная. Гарнитура «Times».
Печать офсетная. Усл.-печ. л. 1,4. Тираж 100 экз. Заказ 2299.
ГБОУ ДПО «Нижегородский институт развития образования»
603122, Н. Новгород, ул. Ванеева, 203.
www.niro.nnov.ru

Отпечатано в издательском центре учебной
и учебно-методической литературы ГБОУ ДПО НИРО

