

УДК 378.016:004.42

С. В. Журавлёв, М. С. Можаров

S. V. Zhuravlev, M. S. Mozharov

Журавлёв Сергей Владимирович, соискатель кафедры ТиМПИ НФИ КемГУ, г. Новокузнецк.

Можаров Максим Сергеевич, кандидат педагогических наук, профессор, НФИ КемГУ, г. Новокузнецк.

Zhuravlev Sergei Vladimirovich, competitor of the Department of TiMPI at the NFI KemGU, Novokuznetsk.

Mozharov Maksim Sergeevich, candidate of pedagogical Sciences, Professor, NFI KemGU, Novokuznetsk.

РАБОТА СО СТРОКАМИ В КУРСЕ ДИСЦИПЛИНЫ «ПРОГРАММИРОВАНИЕ» В СРЕДЕ LAZARUS

WORKING WITH STRINGS IN THE COURSE OF DISCIPLINE «PROGRAMMING» IN THE ENVIRONMENT LAZARUS

Аннотация. В статье рассматриваются особенности обучения студентов педагогических вузов дисциплине «Программирование» в среде Lazarus. Приводится пример изучения темы «Строки».

Annotation. The article describes the features of teaching of students of pedagogical high schools to subject «Programming» in the environment «Lazarus». Authors give the example of training on theme «Strings».

Ключевые слова: обучение программированию, среда разработки Lazarus, методы решения задач, строка.

Keywords: teaching to programming, development environment «Lazarus», methods of solving problems, string.

Вариативная часть ФГОС ВО направления подготовки 44.03.01 «Педагогическое образование» включает в себя обязательную дисциплину «Программирование». Мы будем использовать в процессе обучения среду Lazarus. Целесообразность работы в данной среде заключается в том, что Lazarus является средой визуального программирования, освоив которую, студенты, которые решат в будущем заниматься программированием на более высоком уровне, смогут лучше усвоить сложные разделы, такие как, например, объектно-ориентированное программирование [2, 5]. Кроме того эта среда свободно распространяемая, т. е. каждый может установить её бесплатно.

Рассмотрим методику изучения дисциплины «Программирование» на примере темы «Строки».

М. С. Можаров, С. В. Журавлев 2018-04-24

При решении задач по данной теме студенты используют специальные функции и процедуры:

1) function length(s: string): integer – определяет длину строки s.

Пример.

```
s:='мама мыла раму';
```

```
k:=length(s);
```

В результате переменная k будет иметь значение 14;

2) function pos(s1, s2: string): integer – определяет первую позицию вхождения строки s1 в строку s2.

Пример.

```
s:='мама мыла раму';
```

```
i:=pos('ма',s);
```

В результате переменная i будет иметь значение 1;

3) function copy(s: string; i, k: integer): string – выводит часть строки s, начиная с позиции i, длиной k символов.

Пример.

```
s:='мама мыла раму';
```

```
s1:=copy(s, 6, 4);
```

В результате переменная s1 будет иметь значение 'мыла';

4) procedure delete(var s: string; i, k: integer) – удаляет часть строки s, начиная с позиции i, длиной k символов.

Пример.

```
s:='мама мыла раму';
```

```
delete(s, 11, 4);
```

В результате переменная s будет иметь значение 'мама мыла';

5) procedure insert(s1: string; var s2: string; i: integer) – вставляет строку s1 в строку s2, начиная с позиции i.

Пример.

```
s:='мама раму';
```

```
s1:=' мыла';
```

```
insert(s1, s, 5);
```

В результате переменная *s* будет иметь значение 'мама мыла раму'.

Также студенты используют правило сложения строк.

Пример.

```
s1:='abc';
```

```
s2:='def';
```

```
s:=s1+s2;
```

В результате переменная *s* будет иметь значение 'abcdef'.

При объяснении правила сложения строк необходимо обратить внимание студентов на то, что операция сложения строк не коммутативна, т. е. $s1+s2 \neq s2+s1$ [1].

Рассмотрим пример решения задачи.

Даны две строки *s1* и *s2*, имеющие одинаковую длину. Образовать строку *s*, в которой должны чередоваться символы строк *s1* и *s2*.

Разместим на форме компоненты Edit1 (текстовое окно) для ввода строки *s1*, Edit2 для ввода строки *s2*, Label1 (метка) для вывода строки *s*, Button1 (кнопка) для инициализации решения задачи. Вид формы показан на рисунке 1.

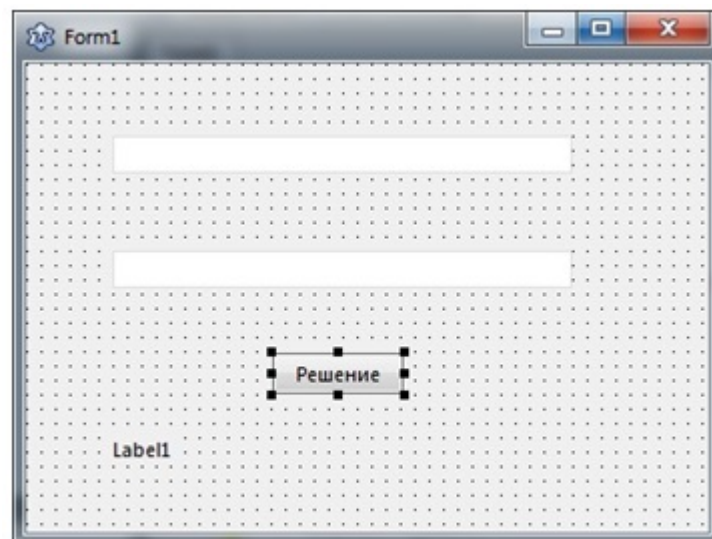


Рис. 1. Вид формы для задачи 1

Ниже приведён алгоритм решения задачи.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var s, s1, s2: string;
```

```
i: integer;
```

```
begin
```

```
s1:=Edit1.Text;  
s2:=Edit2.Text;  
s:='';  
for i:=1 to length(s1) do  
s:=s+s1[i]+s2[i];  
Label1.Caption:=s;  
end; [4]
```

Большинство задач на строки связаны с выделением слов из строки. Рассмотрим пример решения задачи, в котором реализован алгоритм разбиения строки на слова.

Дана строка *s*, состоящая из слов (последовательностей символов, не содержащих пробелов внутри себя), разделённых между собой одним или несколькими пробелами. Вывести на экран все слова этой строки и посчитать их количество.

Разместим на форме компоненты Edit1 для ввода строки *s*, Label1 для вывода количества, ListBox1 (многострочное текстовое окно) для вывода слов, Button1 для инициализации решения задачи. Вид формы показан на рисунке 2.

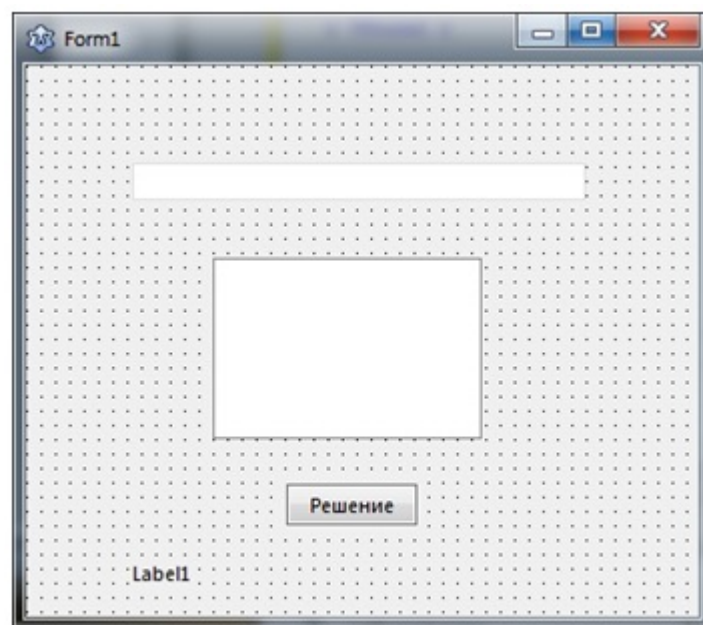


Рис. 2. Вид формы для задачи 2

Ниже приведён алгоритм решения задачи.

```
procedure TForm1.Button1Click(Sender: TObject);  
  
var s, s1: string;
```

```
i, k: integer;
```

begin

s:=Edit1.Text+' '; // пробел добавлен, чтобы последнее слово, если после него нет пробела, обрабатывалось как предыдущее

s1:='';

k:=0;

ListBox1.Items.Clear;

for i:=1 **to** length(s) **do**

begin

if s[i]<>' ' **then**

s1:=s1+s[i];

if (s[i]=' ') **and** (s[i-1]<>' ') **then**

begin

ListBox1.Items.Add(s1);

k:=k+1;

s1:='';

end;

end;

Label1.Caption:=IntToStr(k);

end; [4]

Теперь рассмотрим более сложную задачу.

Дана строка s, состоящая из слов (последовательностей символов, не содержащих пробелов внутри себя), разделённых между собой одним или несколькими пробелами. Определить число слов в строке, состоящих из нечётного числа символов.

Разместим на форме компоненты Edit1 для ввода строки s, Label1 для вывода количества, Button1 для инициализации решения задачи. Вид формы показан на рисунке 3.

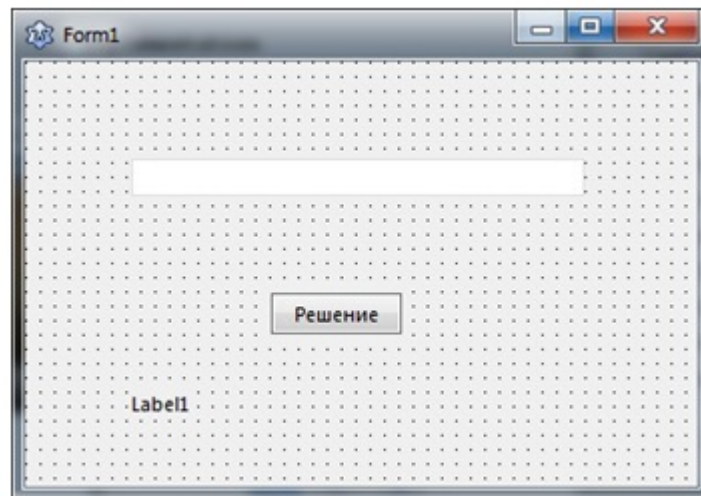


Рис. 3. Вид формы для задачи 3

Ниже приведён алгоритм решения задачи.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var s: string;
```

```
i, k, k1: integer;
```

```
begin
```

```
s:=Edit1.Text+' ';
```

```
k:=0;
```

```
k1:=0;
```

```
for i:=1 to length(s) do
```

```
begin
```

```
if s[i]<>' ' then
```

```
k:=k+1;
```

```
if (s[i]=' ') and (s[i-1]<>' ') then
```

```
begin
```

```
if k mod 2<>0 then
```

```
k1:=k1+1;
```

```
k:=0;
```

```
end;
```

```
end;
```

Label1.Caption:=IntToStr(k1);

end; [3]

Задачи для самостоятельного решения.

1. Даны две строки s1 и s2. Посчитать, сколько раз в строке s2 встречается первый символ строки s1.
2. Дана строка s, состоящая из слов (последовательностей символов, не содержащих пробелов внутри себя), разделённых между собой одним или несколькими пробелами. Определить количество слов в этой строке, содержащих хотя бы одну букву «е».
3. Дана строка s, состоящая из слов (последовательностей символов, не содержащих пробелов внутри себя), разделённых между собой одним или несколькими пробелами. Определить, содержит ли данная строка слова, заканчивающиеся символом *.
- 4.* Дана строка s, состоящая из слов (последовательностей символов, не содержащих пробелов внутри себя), разделённых между собой одним или несколькими пробелами. Если количество слов в строке s чётно, то поменять местами её средние слова. В противном случае на место среднего слова строки s поместить пробелы.

Предложенный фрагмент изучения раздела «Строки» является базой для дальнейшего изучения структурного программирования, от полученных студентами компетенций зависит не только возможность дальнейшего изучения программирования под Windows, требующего освоить работу с литерными переменными, но и изучение курса «Методика обучения (информатика)», а также овладение студентами методикой подготовки школьников к единому государственному экзамену, в котором работе со строками уделяется большое внимание.

Также следует отметить, что использование среды Lazarus в обучении студентов должно повысить их интерес к изучению программирования, способствовать стремлению научиться создавать сложные проекты, имеющие практическое применение, что, несомненно, будет способствовать повышению уровня их конкурентоспособности на рынке труда по окончании ВУЗа.

В настоящее время представленные материалы внедряются в образовательный процесс на физико-математическом и технологическом факультете в рамках изучения курсов: «Программирование», «Решение задач по программированию повышенной сложности», «Методика обучения (информатика)». В следующей статье будет дана оценка их использования на занятиях.

Список литературы

1. Журавлёв, С. В. Методика проведения занятий по дисциплине «Практикум по решению задач на ЭВМ» на тему «Строки» на основе компетентностного, информационного и деятельностного подходов (этап ориентирования) [Электронный ресурс] / С. В. Журавлёв //

Информационно-коммуникационные технологии в педагогическом образовании: электронный научный журнал. – 2012. – № 02 (17) март-апрель. – Режим доступа: journal.kuzspa.ru/articles/74/ (дата обращения 18.01.2018).

2. Можаров, М. С. Дидактические принципы формирования готовности к профессиональной деятельности IT-специалиста в образовании [Текст] / М. С. Можаров, Ю. В. Коровина // Вестник Московского университета. Серия 20: Педагогическое образование. – 2011. – № 4. – С. 8387.
3. Можаров, М. С. Введение в структурное программирование: учебное пособие [Текст] / М. С. Можаров, Г. Н. Бойченко. – 2-е изд., стереот. – Новокузнецк: Изд-во КузГПА, 2014. – 203 с.
4. Можаров, М. С. Практикум по решению задач в среде Lazarus с использованием модульно-рейтинговой системы оценивания: Учебно-методическое пособие для студентов [Текст] / М. С. Можаров, Ю. И. Валеева, Л. В. Попова. – Новокузнецк: Изд-во КузГПА, 2013. – 150 с.
5. Можаров, М. С. Профессиональная мобильность как квалификационная характеристика будущего учителя информатики [Текст] / М. С. Можаров // Педагогическое образование и наука. – 2007. – №4. – С. 44-49.