Министерство образования и науки Российской Федерации

Федеральное агентство по образованию

Марийский государственный технический университет

Кафедра информатики и

системного программирования

Курсовая работа по дисциплине "Информатика"

на тему:

Моделирование иных процессов

вариант №11

Выполнил: студент ИВТ-11

Андреев Маским Анатольевич

Проверила: старший преподаватель

Кафедры ИиСП Иванова И.Н.

Йошкар – Ола

2008

**Постановка задачи**

Вариант №11

Построить модель решения задачи автоматического переноса слов по правилам русской орфографии.

**Содержание**

Введение

1. Теоретическая часть

1.1 Теория

1.2 Описание метода решения

2. Практическая часть

2.1 Алгоритм решения задачи

2.2 Листинг программы

2.3 Тестирование программы

3. Вывод

Список литературы

**Введение**

В данной курсовой работе показан один из способов использования математических методов для решения задачи по переносу слов по правилам русской орфографии. Для реализации поставленной задачи я использовал теоретические сведения из курса информатики, а также полученные ранее знания в области программирования.

**Теоретическая часть**

* 1. **Теория**

Для начала определим, какие существуют правила переносов в русской орфографии.

Основными правилами являются:

* При переносе слов нельзя ни оставлять в конце строки, ни переносить на другую сторону часть слова, не составляющую слога; например, нельзя переносить: просмо-тр, ст-рах
* Нельзя отделять согласную от следующей за ней гласной.
* Нельзя отрывать буквы ъ и ь от предшествующей согласной.
* Нельзя отрывать букву й от предшествующей гласной
* Нельзя оставлять в конце строки или переносить на другую строку одну букву
* При переносе слов с приставками нельзя разбивать односложную приставку, если за приставкой идёт согласный.
* При переносе слов с приставками нельзя оставлять в конце строки при приставке начальную часть корня, не составляющую слога.
* При переносе сложных слов нельзя оставлять в конце строки начальную часть второй основы, если эта часть не составляет слога
* Нельзя оставлять в конце строки или переносить в начало следующей две одинаковые согласные, стоящие между гласными
* Нельзя разбивать переносом односложную часть сложносокращённого слова

**1.2 Описание метода решения**

Методом решения данной задачи является алгоритм, в основе которого лежит «каретка», длиной в 4 символа. Принцип работы «каретки» следующий : в слове берутся первые 4 символа, подсчитываются гласные и согласные буквы, и, если в данном месте в слове можно сделать перенос по правилам, то перенос делается, если же ни одно правило не подходит, то «каретка» сдвигается на один символ и происходит тот же алгоритм проверки, и так до конца слова.

**2. Практическая часть**

**2.1 Алгоритм решения задачи**

Первоначально, для удобства пользователя, предоставим возможность прописать путь к файлу собственноручно. Когда путь к файлу введен пользователь жмет «выполнить».

После этого производятся следующие расчеты:

а) для начала создадим алгоритм, который будет считать количество гласных и согласных букв в слове. Для этого создаем двумерный массив, в котором содержатся все гласные и согласные буквы в первой ячейке, во второй же содержатся цифры «1» и «2», где гласные приравниваем «1», а согласные «2». Далее создаем цикл, который считает символы от начала строки до первого пробела, затем от пробела до пробела, и так до конца строки.

б) создаем цикл, который в каждом слове части по четыре символа проверяет на возможность переноса, если между первыми четырьмя символами перенос сделать нельзя, то «каретка» сдвигается на один символ вправо и так далее до конца слова.

в) заключительный этап программы - это вывод текста, в котором сделаны все возможные переносы в отдельный файл. Файл будет иметь прежнее название + ”2.txt”

**2.2 Листинг программы**

**Программа переноса слов по правилам русской орфографии.**

Для оптимизации кода, создаем массив “al” .

Сама программа имеет следующий вид:

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ComCtrls, ShellCtrls;

type

TForm1 = class(TForm)

Button1: TButton;

Edit1: TEdit;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

t,tt,t2:string;

f:Tstrings;

r,i,rs,j,q,rc,m,n,l,qw:integer;

al:array[1..66,1..2] of string;

kl:array[1..1000] of integer;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

f:=Tstringlist.create();

f.Loadfromfile(form1.Edit1.Text);

for qw:=0 to f.count-1 do begin

tt:='';

t2:='';

t:=f.strings[qw];

i:=1;

while t[i]<>'' do

i:=i+1;

r:=i-1;

for i:=1 to r do begin

q:=1;

for j:=1 to 66 do begin

if t[i]=al[j,1] then begin

tt:=tt+al[j,2];

q:=2;

end;

end;

if q=1 then begin

if t[i]=' ' then begin tt:=tt+'4'; q:=2; end;

if t[i]='-' then begin tt:=tt+'5'; q:=2; end;

if q=1 then tt:=tt+'6';

end;

end;

//---------------------------

i:=1;

while i<=r do begin

q:=0;

m:=0;

n:=0;

for j:=0 to 4 do begin

if tt[i+j]='1' then q:=q+1;

if (tt[i+j]='2') and (j<>4) then m:=m+1;

if ((tt[i+j]='4')or(tt[i+j]='5')or(tt[i+j]='6')) and (j<>4) then n:=n+1;

end;

if ((q+m>=4)and(q>=2)) and

(tt[i+2]<>tt[i+3])and(n=0)and(tt[i+1]+tt[i+2]<>'21')and((t[i+3]<>'ь') and(t[i+3]<>'ъ')) then begin

t2:=t2+t[i]+t[i+1]+'-';

i:=i+1;

end else begin

t2:=t2+t[i];

end;

i:=i+1;

end;

//---------------------------

f.strings[qw]:=t2;

end;

f.SaveToFile(form1.Edit1.Text+'2.txt');

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

al[1 , 1 ]:='а' ;

al[2 , 1 ]:='б' ;

al[3 , 1 ]:='в' ;

al[4 , 1 ]:='г' ;

al[5 , 1 ]:='д' ;

al[6 , 1 ]:='е' ;

al[7 , 1 ]:='ё' ;

al[8 , 1 ]:='ж' ;

al[9 , 1 ]:='з' ;

al[10 , 1 ]:='и' ;

al[11 , 1 ]:='й' ;

al[12 , 1 ]:='к' ;

al[13 , 1 ]:='л' ;

al[14 , 1 ]:='м' ;

al[15 , 1 ]:='н' ;

al[16 , 1 ]:='о' ;

al[17 , 1 ]:='п' ;

al[18 , 1 ]:='р' ;

al[19 , 1 ]:='с' ;

al[20 , 1 ]:='т' ;

al[21 , 1 ]:='у' ;

al[22 , 1 ]:='ф' ;

al[23 , 1 ]:='х' ;

al[24 , 1 ]:='ц' ;

al[25 , 1 ]:='ч' ;

al[26 , 1 ]:='ш' ;

al[27 , 1 ]:='щ' ;

al[28 , 1 ]:='ъ' ;

al[29 , 1 ]:='ы' ;

al[30 , 1 ]:='ь' ;

al[31 , 1 ]:='э' ;

al[32 , 1 ]:='ю' ;

al[33 , 1 ]:='я' ;

al[34 , 1 ]:='А' ;

al[35 , 1 ]:='Б' ;

al[36 , 1 ]:='В' ;

al[37 , 1 ]:='Г' ;

al[38 , 1 ]:='Д' ;

al[39 , 1 ]:='Е' ;

al[40 , 1 ]:='Ё' ;

al[41 , 1 ]:='Ж' ;

al[42 , 1 ]:='З' ;

al[43 , 1 ]:='И' ;

al[44 , 1 ]:='Й' ;

al[45 , 1 ]:='К' ;

al[46 , 1 ]:='Л' ;

al[47 , 1 ]:='М' ;

al[48 , 1 ]:='Н' ;

al[49 , 1 ]:='О' ;

al[50 , 1 ]:='П' ;

al[51 , 1 ]:='Р' ;

al[52 , 1 ]:='С' ;

al[53 , 1 ]:='Т' ;

al[54 , 1 ]:='У' ;

al[55 , 1 ]:='Ф' ;

al[56 , 1 ]:='Х' ;

al[57 , 1 ]:='Ц' ;

al[58 , 1 ]:='Ч' ;

al[59 , 1 ]:='Ш' ;

al[60 , 1 ]:='Щ' ;

al[61 , 1 ]:='Ъ' ;

al[62 , 1 ]:='Ы' ;

al[63 , 1 ]:='Ь' ;

al[64 , 1 ]:='Э' ;

al[65 , 1 ]:='Ю' ;

al[66 , 1 ]:='Я' ;

al[1 , 2 ]:='1' ;

al[2 , 2 ]:='2' ;

al[3 , 2 ]:='2' ;

al[4 , 2 ]:='2' ;

al[5 , 2 ]:='2' ;

al[6 , 2 ]:='1' ;

al[7 , 2 ]:='1' ;

al[8 , 2 ]:='2' ;

al[9 , 2 ]:='2' ;

al[10 , 2 ]:='1' ;

al[11 , 2 ]:='1' ;

al[12 , 2 ]:='2' ;

al[13 , 2 ]:='2' ;

al[14 , 2 ]:='2' ;

al[15 , 2 ]:='2' ;

al[16 , 2 ]:='1' ;

al[17 , 2 ]:='2' ;

al[18 , 2 ]:='2' ;

al[19 , 2 ]:='2' ;

al[20 , 2 ]:='2' ;

al[21 , 2 ]:='1' ;

al[22 , 2 ]:='2' ;

al[23 , 2 ]:='2' ;

al[24 , 2 ]:='2' ;

al[25 , 2 ]:='2' ;

al[26 , 2 ]:='2' ;

al[27 , 2 ]:='2' ;

al[28 , 2 ]:='1' ;

al[29 , 2 ]:='1' ;

al[30 , 2 ]:='1' ;

al[31 , 2 ]:='1' ;

al[32 , 2 ]:='1' ;

al[33 , 2 ]:='1' ;

al[34 , 2 ]:='1' ;

al[35 , 2 ]:='2' ;

al[36 , 2 ]:='2' ;

al[37, 2 ]:='2' ;

al[38 , 2 ]:='2' ;

al[39 , 2 ]:='1' ;

al[40 , 2 ]:='1' ;

al[41 , 2 ]:='2' ;

al[42 , 2 ]:='2' ;

al[43 , 2 ]:='1' ;

al[44 , 2 ]:='1' ;

al[45 , 2 ]:='2' ;

al[46 , 2 ]:='2' ;

al[47 , 2 ]:='2' ;

al[48 , 2 ]:='2' ;

al[49 , 2 ]:='1' ;

al[50 , 2 ]:='2' ;

al[51 , 2 ]:='2' ;

al[52 , 2 ]:='2' ;

al[53 , 2 ]:='2' ;

al[54 , 2 ]:='1' ;

al[55 , 2 ]:='2' ;

al[56 , 2 ]:='2' ;

al[57 , 2 ]:='2' ;

al[58 , 2 ]:='2' ;

al[59 , 2 ]:='2' ;

al[60 , 2 ]:='2' ;

al[61 , 2 ]:='1' ;

al[62 , 2 ]:='1' ;

al[63 , 2 ]:='1' ;

al[64 , 2 ]:='1' ;

al[65 , 2 ]:='1' ;

al[66 , 2 ]:='1' ;

end;

end.

**2.3 Тестирование программы**

Задав слова русского языка получаем на выходе слова с расставленными переносами.

**Вывод**

# При выполнении этой курсовой работы, был составлен наиболее оптимальный алгоритм для расстановки переносов в словах по правилам русской орфографии. Эта программа необходима всем людям, которые сомневаются, как переносить то или иное слово по слогам.