Содержание

[1 Постановка задачи 3](#_Toc133920638)

[2 Методика решения задачи 4](#_Toc133920639)

[2.1 Подпрограммы 4](#_Toc133920640)

[2.2 Сохранение и создание файлов 4](#_Toc133920641)

[2.3 Сохранение пар слов 4](#_Toc133920642)

[2.4 Замена слов 5](#_Toc133920643)

[3 Описание алгоритмов решения 6](#_Toc133920644)

[4 Результаты расчетов 8](#_Toc133920645)

[Приложение А 10](#_Toc133920646)

[Приложение B 14](#_Toc133920647)

# Постановка задачи

Файл F1 содержит произвольный текст. Слова в тексте разделены пробелами и знаками препинания. Файл F2 содержит не более 40 слов, которые разделены запятыми. Эти слова образуют пары: каждое первое слово считается заменяемым, каждое второе слово – заменяющим. Найти в файле F1 все заменяемые слова и заменить их на соответствующие заменяющие. Результат поместить в файле G.

# Методика решения задачи

## Подпрограммы

Для решения поставленной задачи необходимо реализовать подпрограммы для открытия и создания файлов, сохранения пар слов (заменяемых и заменяющих) и замены слов заключается в прохождении N div 2 раз (N – количество слов в файле F2) по тексту файла F1, чтобы проверить все пары слов. Чтобы заменить все слова в файле F1 можно использовать следующие подпрограммы:

Рассмотрим реализацию каждой из этих подпрограмм.

## Сохранение и создание файлов

Для корректной работы с русскими символами необходимо выбрать одинаковые кодировки в программе и текстовом документе (в нашем случае ANSI). Для работы с файлами необходимо связать путь, указанный пользователем, с файловой переменной. Для проверки указанного пользователем пути используется функция FileExists. Для создания нового файла G необходимо связать директорий, указанный пользователем, с файловой переменной.

## Сохранение пар слов

Так как количество слов в файле F2 неизвестно, то для работы с парами слов (заменяемых и заменяющих) можно использовать динамическую структуру однонаправленный список и создать подпрограммы для создания главного звена, добавления нового звена и удаления списка.

Для поиска слов из файла F2 можно использовать функцию Pos (SubStr, Str), где SubStr содержит искомую часть (в нашем случае “,”) строки Str. После нахождения позиции запятой можно сохранить слово в однонаправленный список и удалить его из строки Str. Данный алгоритм можно использовать в цикле, пока в строке будет запятая.

Так как слов может быть нечётное количество, необходимо сохранить только чётное количество слов. Для этого необходимо подсчитать количество запятых и выполнить следующую проверку:

if I mod 2 = 1 then

NewZveno(Adres,Temp);

## Замена слов

Для замены слов в файле F1 необходимо пройти N-1 div 2 раз (N – количество звеньев в списке) по тексту файла для поиска каждого заменяющего слова. Для замены слова можно использовать 2 подпрограммы:

* функция StringReplace (Str, S1, S2, [Flags]), где Str – исходная строка, S1 – заменяемое слово, S2 – заменяющее слово, Flags используются для замены всех слов в строке и без учёта регистра. Данная подпрограмма доступна с подключением модуля StrUtils.
* подпрограммы PosAfter (SubStr, Str, Index), Delete (Str, Index, Amount), Insert (SubStr, Str, Index). Подпрограмма PosAfter позволяет вывести индекс входа подстроки, начиная с символа Index. Процедура Delete удаляет из строки количество символов Amount с символа Index. Процедура Insert вставляет в строку Str строку SubStr в позицию Index.

# Описание алгоритмов решения

Таблица 1 – Описание алгоритмов решения задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п. | Наименование алгоритма | Название алгоритма | Формальные  параметры | Предпола-гаемый тип реализации |
| 1 | Основной алгоритм | Вызов следующих подпрограмм:  ListCreate, GetReplace, Replaces1, Replaces2, ListClear |  |  |
| 2 | ListCreate  (  Adres  ) | Создаёт главное звено списка и выводит его адрес | Adres – получают от фактического параметра адрес, возвращаемый параметр | Функция  (Res – возвращаемый параметр) |
| 3 | NewZveno  (  Addr,  S  ) | Добавляет звено со строкой S и адресом nil в список, в Addr^.AdrCled записывает адрес этого звена | S – получает от фактического параметра адрес c защитой;  Addr – получает от фактического параметра адрес, возвращаемый параметр | Процедура |
| 4 | ListClear  (  AdrStart  ) | Удаляет все звенья списка | AdrStart – получает от фактического параметра адрес, возвращаемый параметр | Процедура |
| 5 | GetFile  (  F  ) | Связывает файловую переменную F c путём, который введёт пользователь | F – получает от фактического параметра адрес, возвращаемый параметр | Процедура |
| 6 | GetReplace  (  Adres,  File2  ) | Записывает слова из файла File2 в список звено после Adres, вызывая процедуру NewZveno | Adres, File2 – получают от фактического параметра адрес, возвращаемые параметры | Процедура |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п. | Наименование алгоритма | Название алгоритма | Формальные  параметры | Предпола-гаемый тип реализации |
| 7 | Replaces1  (  File2,  File3,  Adres  ) | Заменяет слова из файла File2 словами из списка, записывая результат в файл File3, используя подпрограмму StringReplace | File2, File3, Adres – получают от фактического параметра адрес, возвращаемые параметры | Процедура |
| 8 | PosAfter  (  S,  Per,  After  ) | Находит вхождение подстроки S в строку Per с символа After и выводит индекс | S, Per – получают от фактического параметра значение;  After – получает от фактического параметра адрес, возвращаемый параметр | Функция  (Res – возвращаемый параметр) |
| 9 | Replaces2  (  File2,  File3,  Adres  ) | Заменяет слова из файла File2 словами из списка, записывая результат в файл File3. Вызывает следующие подпрограммы: PosAfter, Delete, Insert | File2, File3, Adres – получают от фактического параметра адрес, возвращаемые параметры | Процедура |

# Результаты расчетов

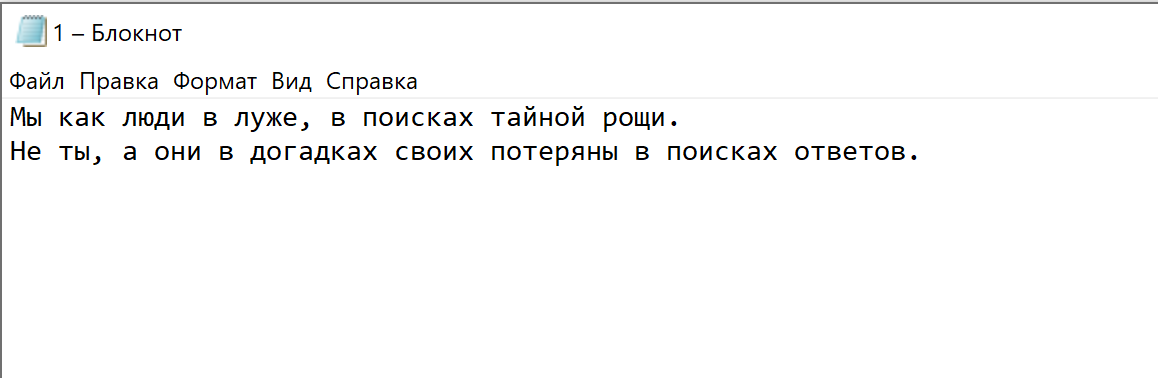


Рисунок 1 – Исходный файл

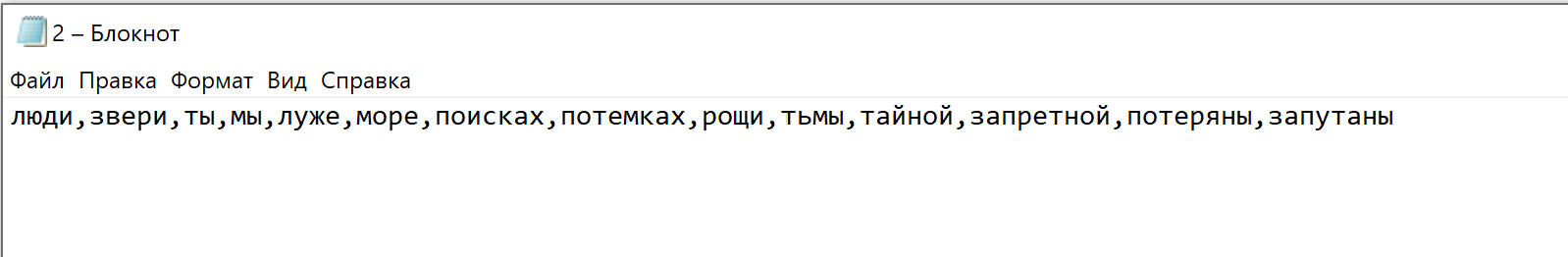


Рисунок 2 – Файл с парами слов

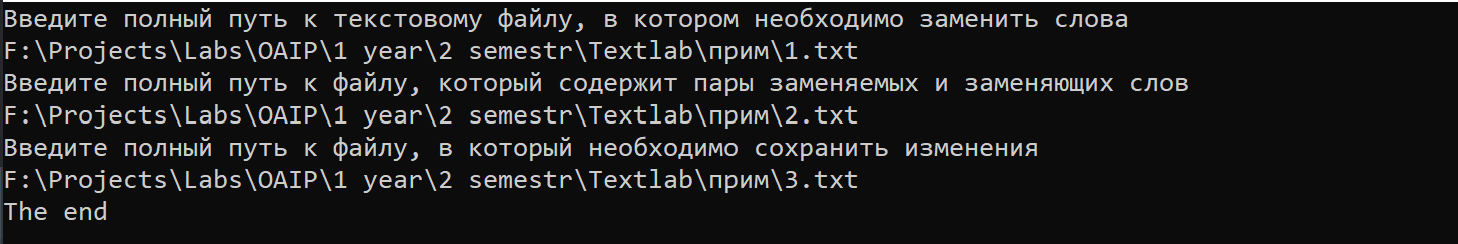


Рисунок 3 – Ввод данных в программу

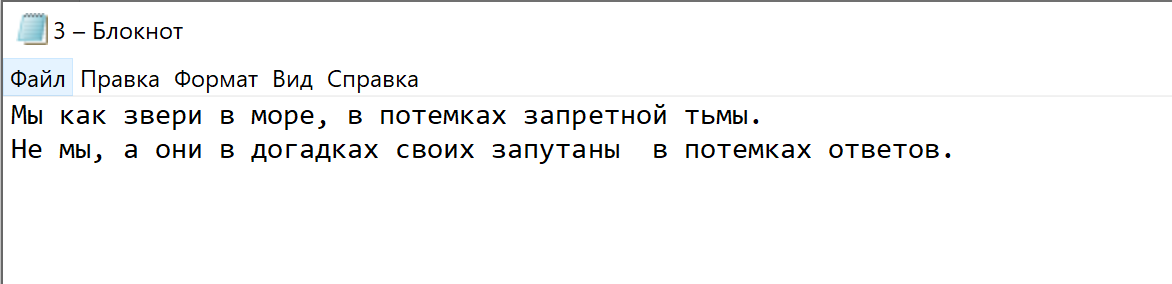


Рисунок 4 – Выходной файл

Приложение А

(обязательное)

Код программы с использованием процедуры StringReplace

program train\_text;

{

Создать программу для изменения текстового файла F1,

изменяя слова из файла F2, сохранив изменения в файл G

}

uses

System.SysUtils,

StrUtils;

// Declaring singly-directed list

Type

Adr = ^Rep;

Rep = record

Words: String;

AdrCled: Adr;

end;

var

F1, F2, G: TextFile;

Adr1, AdrRep: Adr;

// F1, F2 – beginning text files

// G – ending text file

// Adr1, AdrRep – pointers to the first and current link

// Subprogram for creating first link

// of singly-directed list

function ListCreate (var Adres: Adr) :Adr;

Begin

New(Adres);

Result:=Adres;

Adres^.AdrCled:=Nil;

End;

// Subprogram for deleating all links

// in singly-directed list

procedure ListClear (var AdrStart: Adr);

var

Adres: Adr;

Begin

while AdrStart^.AdrCled<>Nil do

Begin

Adres:=AdrStart^.AdrCled;

AdrStart^.AdrCled:=Adres^.AdrCled;

Dispose(Adres);

End;

End;

// Subprogram for adding links to singly-directed

// list

Procedure NewZveno (var Addr: Adr;

S: String);

Begin

New(Addr^.AdrCled);

Addr:=Addr^.AdrCled;

Addr^.Words:=S;

Addr^.AdrCled:=Nil;

End;

// Subprogram for connecting file variable and

// full path, given by user

procedure GetFile (var F: TextFile);

var

Path: String;

Begin

// Checking is the path exist

Repeat

Readln(Path);

If FileExists(Path) = False then

Write('Некорректный путь, введите заново: ');

Until FileExists(Path);

AssignFile(F, Path);

End;

// Subprogram for recieving couple of changing and

// сhanged words

procedure GetReplace (var Adres: Adr;

var File2: TextFile);

var

I, J: Integer;

Temp: String;

Begin

Reset(File2);

Readln(File2, Temp);

I:=0;

// Index of a commar in string

J:=Pos(',',Temp);

// Cycle for adding a word, commar in the string

while J <> 0 do

Begin

NewZveno(Adres, Copy(Temp,1,J-1));

Delete(Temp,1,J);

Inc(I);

J:=Pos(',',Temp);

End;

// Checking number of words on even

if I mod 2 = 1 then

NewZveno(Adres,Temp);

// Subprogram for changing words

// StringReplace

procedure Replaces1 (var File1, File3: TextFile;

Adres: Adr);

var

AdrStart: Adr;

Temp: String;

Begin

// Saving an addres of the first link

AdrStart:=Adres;

Reset(File1);

ReWrite(File3);

// Cycle for going throw hall text file

while not Eoln (File1) do

Begin

Readln(File1, Temp);

// Cycle for going throw all changing words

While Adres<>nil do

Begin

Temp:=StringReplace(Temp,Adres^.Word,Adres^.AdrCled^.

Words, [rfReplaceAll, rfIgnoreCase]);

Adres:=Adres^.AdrCled^.AdrCled;

End;

Writeln(File3, Temp);

// Brings the pointer in the begining of the link

Adres:=AdrStart;

End;

end;

// Main program

begin

Writeln('Введите полный путь к текстовому файлу, в

котором необходимо заменить слова');

GetFile(F1);

Writeln('Введите полный путь к файлу, который содержит

пары заменяемых и заменяющих слов');

GetFile(F2);

Writeln('Введите файл, в который необходимо сохранить

изменения');

GetFile(G);

AdrRep:=ListCreate(Adr1);

GetReplace(AdrRep, F2);

AdrRep:=Adr1^.AdrCled;

Replaces1(F1, G, AdrRep);

ListClear(Adr1);

Write('The end');

Close(F1);

Close(F2);

Close(G);

Readln;

end.

Приложение B

(обязательное)

Код программы с использованием подпрограммы PosAfter

program train\_text;

{

Создать программу для изменения текстового файла F1,

изменяя слова из файла F2, сохранив изменения в файл G

}

uses

System.SysUtils;

// Объявление типа однонаправленного списка

Type

Adr = ^Rep;

Rep = record

Words: String;

AdrCled: Adr;

end;

var

F1, F2, G: TextFile;

Adr1, AdrRep: Adr;

// F1, F2 – исходные текстовые файлы

// G – выходной текстовый файл

// Adr1, AdrRep – укаатели на первое и текущее звено

// списка

// Подпрограмма для создания первого звена

// однонаправленного списка

function ListCreate (var Adres: Adr) :Adr;

Begin

New(Adres);

Result:=Adres;

Adres^.AdrCled:=Nil;

End;

// Подпрограмма для удаления всех звеньев

// однонаправленного списка

procedure ListClear (var AdrStart: Adr);

var

Adres: Adr;

Begin

while AdrStart^.AdrCled<>Nil do

Begin

Adres:=AdrStart^.AdrCled;

AdrStart^.AdrCled:=Adres^.AdrCled;

Dispose(Adres);

End;

End;

// Подпрограмма для добавления звена в однонаправленный

// список

Procedure NewZveno (var Addr: Adr;

S: String);

Begin

New(Addr^.AdrCled);

Addr:=Addr^.AdrCled;

Addr^.Words:=S;

Addr^.AdrCled:=Nil;

End;

// Подпрограмма для связывания файловой переменной и

// полного пути, введённого пользователем

procedure GetFile (var F: TextFile);

var

Path: String;

Begin

//Writeln('Укажите полный путь к файлу c расширением

''txt'': ');

// Проверка пути файла

Repeat

Readln(Path);

If FileExists(Path) = False then

Write('Некорректный путь, введите заново: ');

Until FileExists(Path);

AssignFile(F, Path);

End;

// Подпрограмма для получения пары заменяемых и заменяющих

// слов

procedure GetReplace (var Adres: Adr;

var File2: TextFile);

var

I, J: Integer;

Temp: String;

Begin

Reset(File2);

Readln(File2, Temp);

I:=0;

// Индекс запятой в строке

J:=Pos(',',Temp);

// Цикл для добавления слова, пока в строке есть запятая

while J <> 0 do

Begin

NewZveno(Adres, Copy(Temp,1,J-1));

Delete(Temp,1,J);

Inc(I);

J:=Pos(',',Temp);

End;

// Проверка количества слов на чётность

if I mod 2 = 1 then

NewZveno(Adres,Temp);

End;

// Подпрограмма для вывода индекса входа подстроки

Function PosAfter (S: string;

Per: String;

var After: Integer): Integer;

var

I1, I2: Integer;

T1: String;

Begin

// Копируем строку, начиная с индекса After

T1:=Copy(Per,After,Length(Per)-After+1);

// Переводим всю строку в нижний регистр

T1:=AnsiLowerCase(T1);

Result:=Pos(S,T1);

End;

// Подпрограмма для замены слов с помощью подпрограммы

// PosAfter, Insert, Copy

procedure Replaces2 (var File1, File3: TextFile;

Adres: Adr);

var

J: Integer;

Temp: String;

K, L: Integer;

AdrStart, Adr2: Adr;

Begin

Reset(File1);

ReWrite(File3);

K:=1;

// Сохранение адреса первого звена

AdrStart:=Adres;

// Цикл для прохода по всем строкам файла

while not Eoln (File1) do

Begin

Readln(File1, Temp);

// Цикл для прохода по всем заменяемым словам

While Adres<>nil do

Begin

J:=PosAfter(Adres^.Words,Temp,K);

// Цикл для прохода по строке, пока есть

// заменяемые слова

while J <> 0 do

Begin

Adr2:=Adres^.AdrCled;

L:=Length(Adres^.Words);

Delete(Temp,J,L);

Insert(Adres^.AdrCled^.Words,Temp,J);

J:=PosAfter(Adres^.Words,Temp,K);

End;

Adres:=Adres^.AdrCled^.AdrCled;

End;

// Изменение текущего звена на первое

Adres:=AdrStart;

Writeln(File3, Temp);

End;

End;

// Основная программа

begin

Writeln('Введите полный путь к текстовому файлу, в

котором необходимо заменить слова');

GetFile(F1);

Writeln('Введите полный путь к файлу, который содержит

пары заменяемых и заменяющих слов');

GetFile(F2);

Writeln('Введите папку, в которую необходимо сохранить

изменения');

GetFile(G);

AdrRep:=ListCreate(Adr1);

GetReplace(AdrRep, F2);

AdrRep:=Adr1^.AdrCled;

Replaces2(F1, G, AdrRep);

ListClear(Adr1);

Write('The end');

Close(F1);

Close(F2);

Close(G);

Readln;

end.