

Министерство образования и науки РФ
Государственное образовательное учреждение
высшего профессионального образования

Тульский государственный университет

КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

ФАЙЛЫ. РАБОТА С ТЕКСТОМ.

Лабораторная работа № 5
по курсу «Программирование на ЯВУ»

Вариант № 4

Выполнил: студент группы 220601

(подпись) Белым А.А.

Проверил:

(подпись) Сулимова В.В.

Тула 2011

Цель работы

Цель работы заключается в том, что необходимо научиться работать с файлами и обрабатывать данные, находящиеся в них. Требуется написать программу, считывающую данные из текстового файла; затем данные записываются в файл прямого доступа (компоненты файла являются целого или вещественного типа (если входные данные символы, то сначала нужно перевести их в коды, а потом записать в файл)); обработать данные (использовать прямой доступ к компонентам файла) и записать результат в текстовый файл.

Задание

Определить, какая буква чаще всего встречается в заданном тексте.

Схема алгоритма

На рисунке 1 представлена схема алгоритма ввода данных, поиска самых частых букв и их вывода.

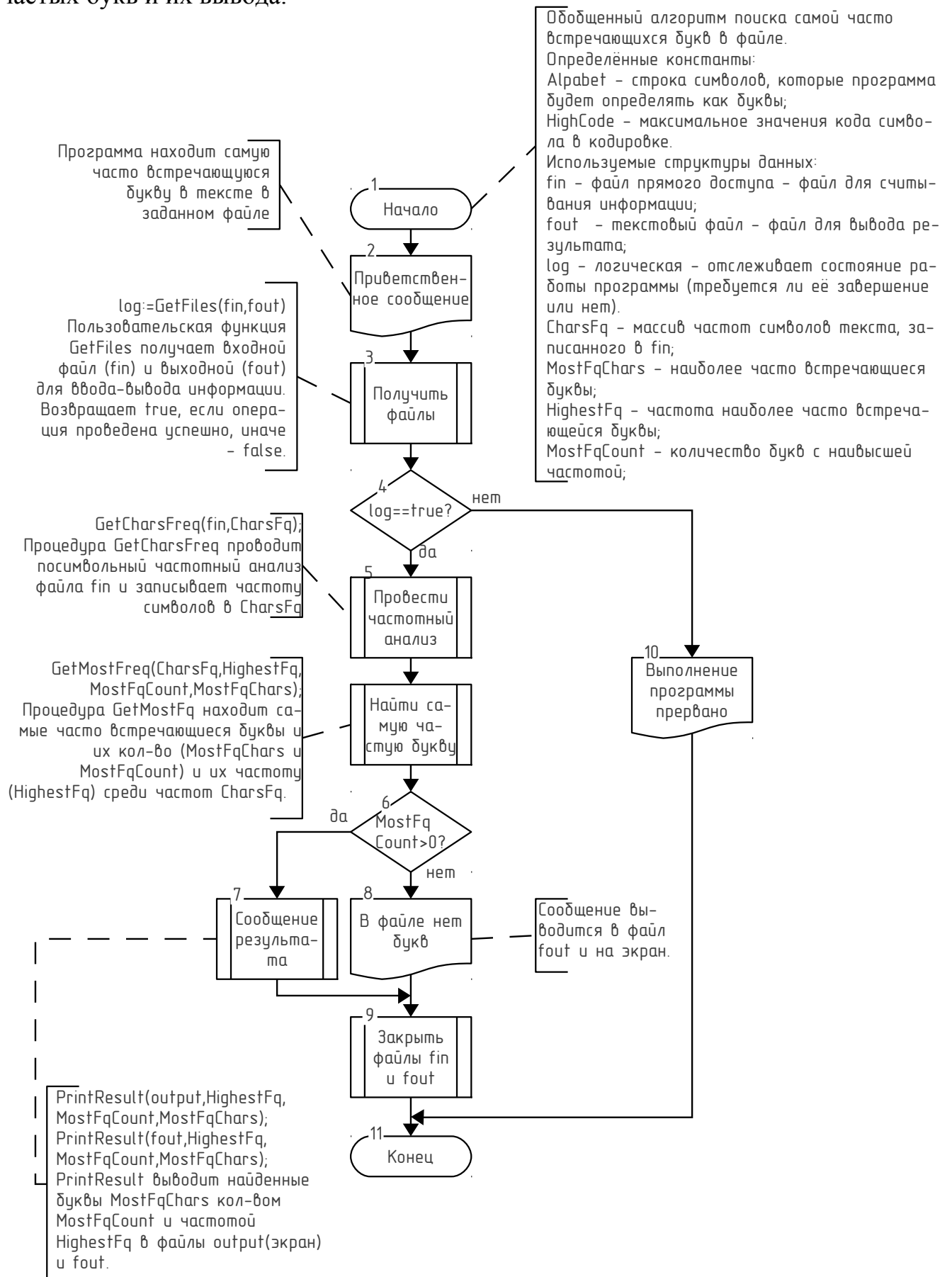


Рисунок 1 — Блок-схема обобщенного алгоритма поиска самой часто встречающейся буквы

На рисунке 2 представлена схема алгоритма получения необходимых для работы файлов.

Пользовательская функция GetFiles получает входной файл и выходной для ввода-вывода информации. Возвращает true, если операция проведена успешно, иначе - false.

function GetFiles(var fin:bytefile;var fout:text):boolean;
Используемые параметры:
fin - мультированный файл, в который записываются коды символов исходного текста
fout - текстовый файл для вывода результата
Используемые локальные переменные:
error - код ошибки ввода-вывода(0 - нет ошибки).
fsrc - текстовый исходный файл
ch - текущий символ из исходного файла
i - код текущего символа

error:=GetInputFile(fin);
Функция GetInputFile получает файл последовательного доступа для записи и в результате считывания кодов символов. Возвращает код возникшей ошибки ввода-вывода.

error:=GetOutputFile(fout)
Функция GetOutputFile получает текстовый файл для вывода в него самой часто встречающейся буквы.. Возвращает код возникшей ошибки ввода-вывода;

error:=GetSourceFile(fsrc)
Функция GetSourceFile получает текстовый файл для считывания из него символов.. Возвращает код возникшей ошибки ввода-вывода.

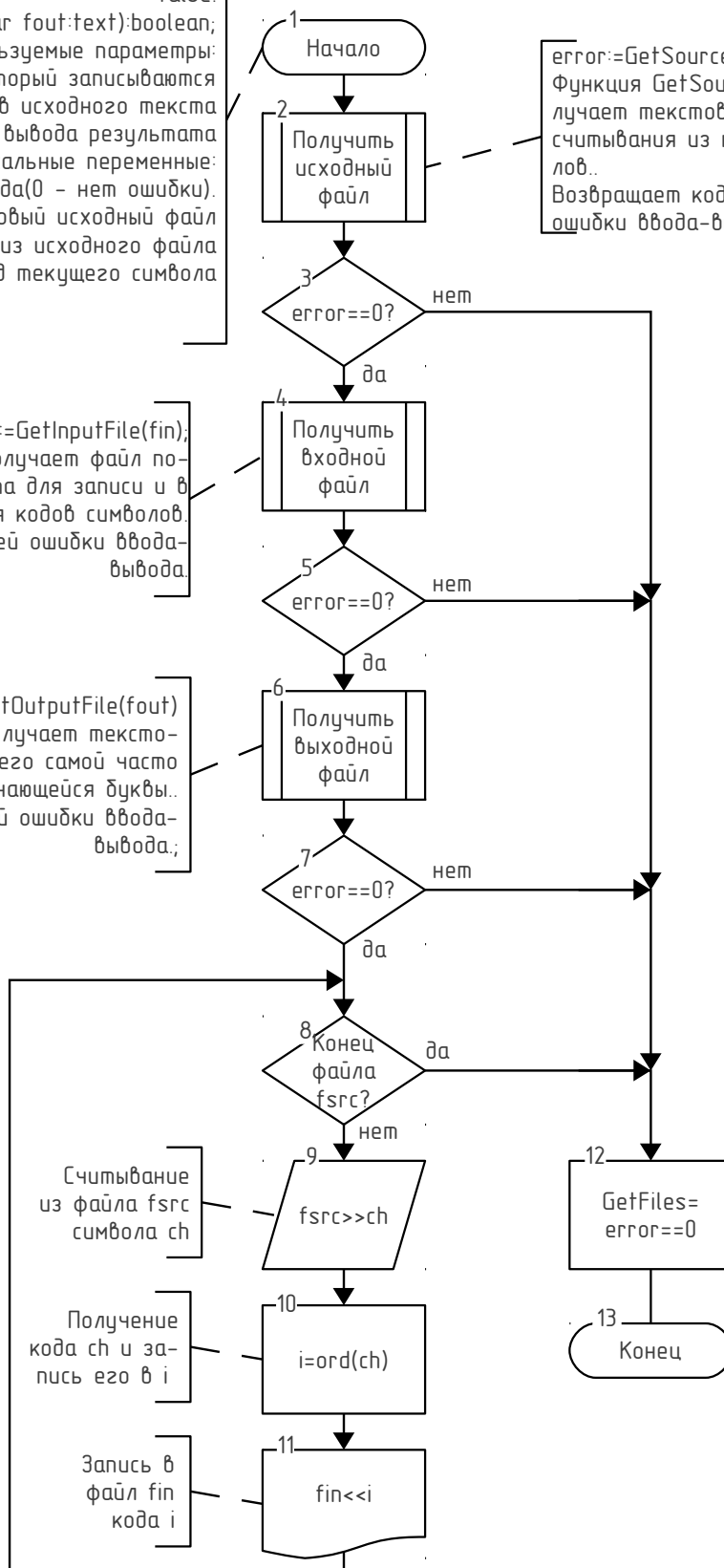


Рисунок 2 — Блок-схема алгоритма получения необходимых файлов

На рисунке 3 представлена схема алгоритма получения файла-источника.

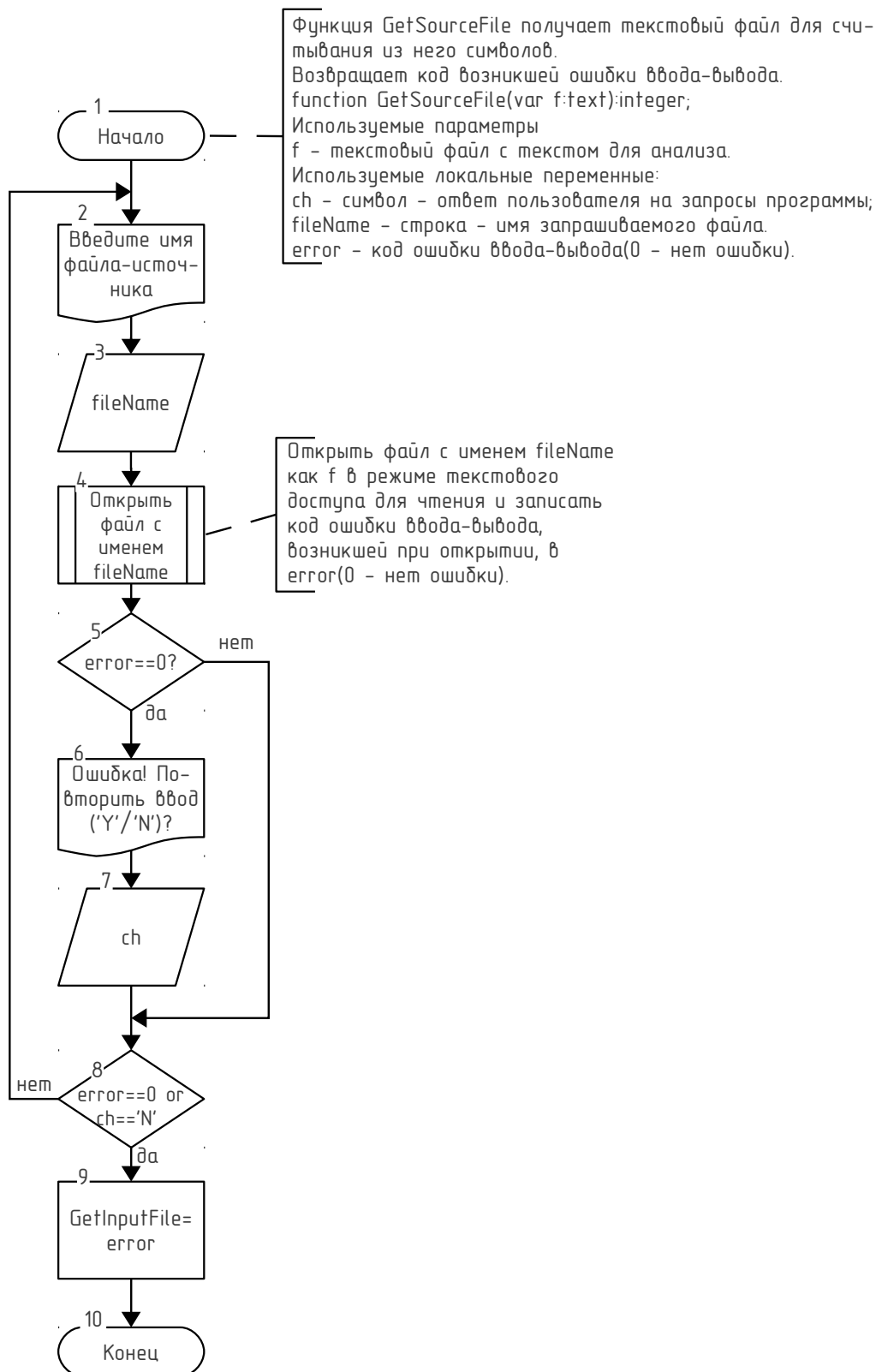


Рисунок 3 — Блок-схема алгоритма получения файла-источника

На рисунке 4 представлена схема алгоритма получения входного файла.

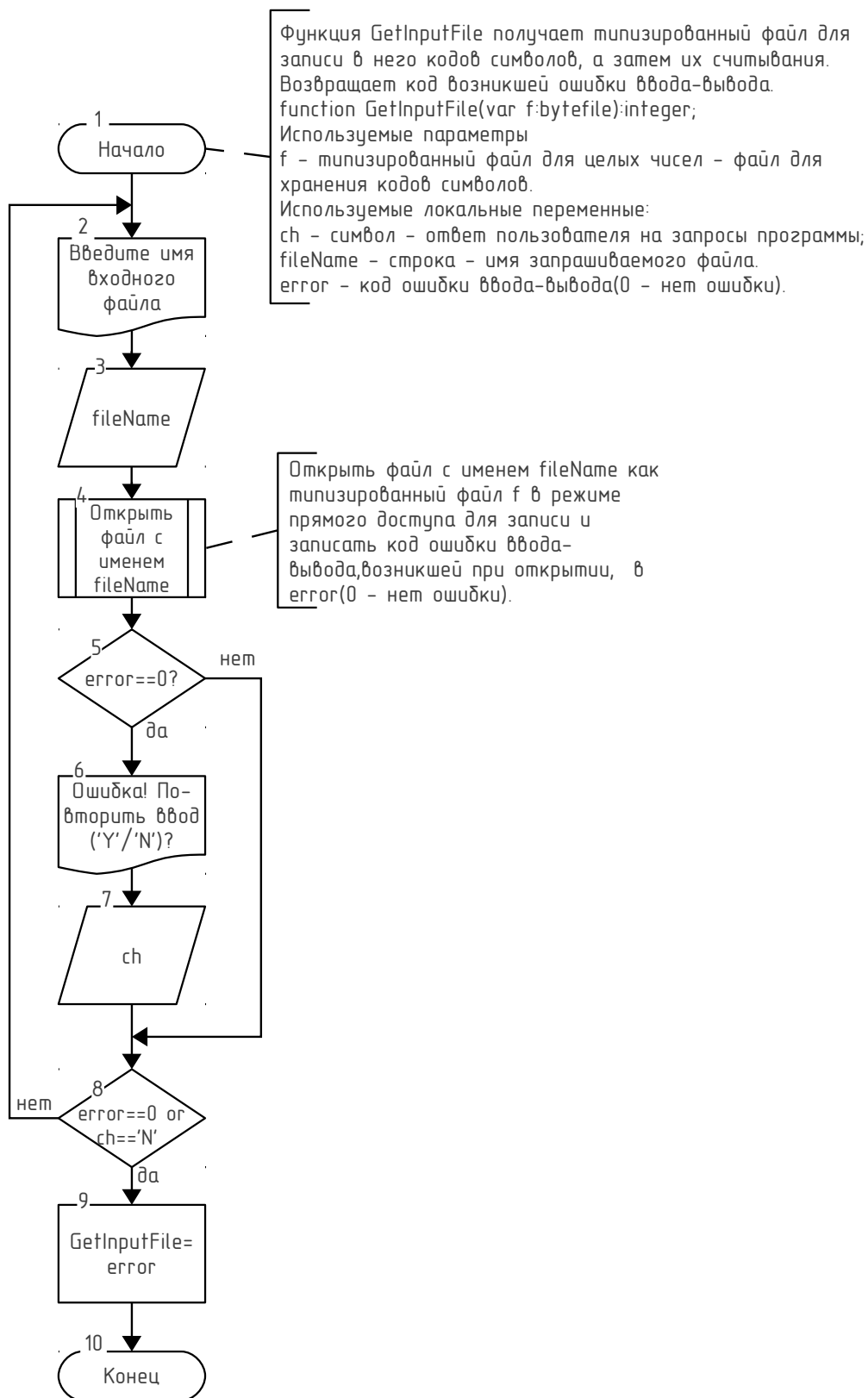


Рисунок 4 — Блок-схема алгоритма получения входного файла

На рисунке 5 представлена схема алгоритма получения выходного файла.

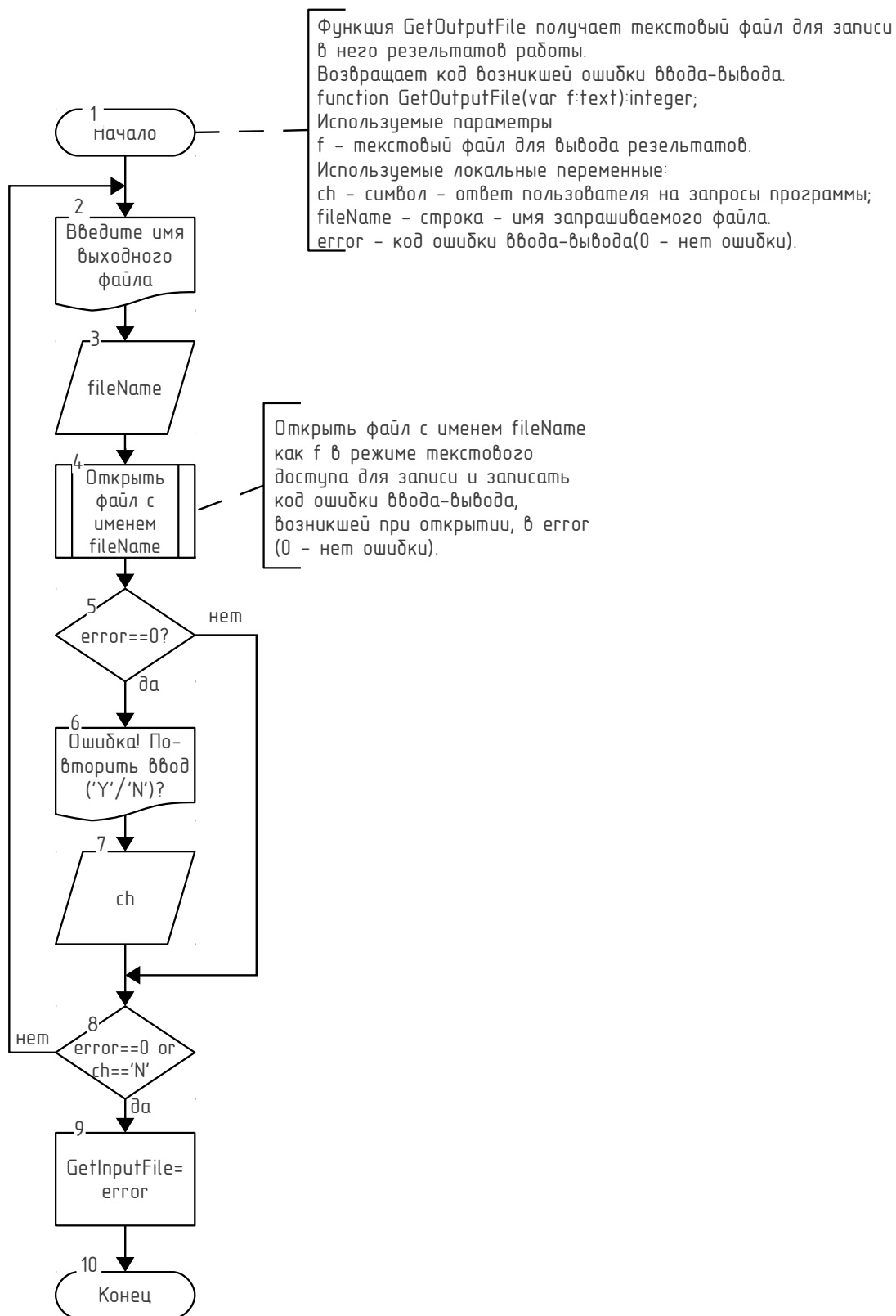


Рисунок 5 — Блок-схема алгоритма получения выходного файла

На рисунке 6 представлена схема посимвольного частотного анализа исходного текста с использованием кодов символов из типизированного файла.

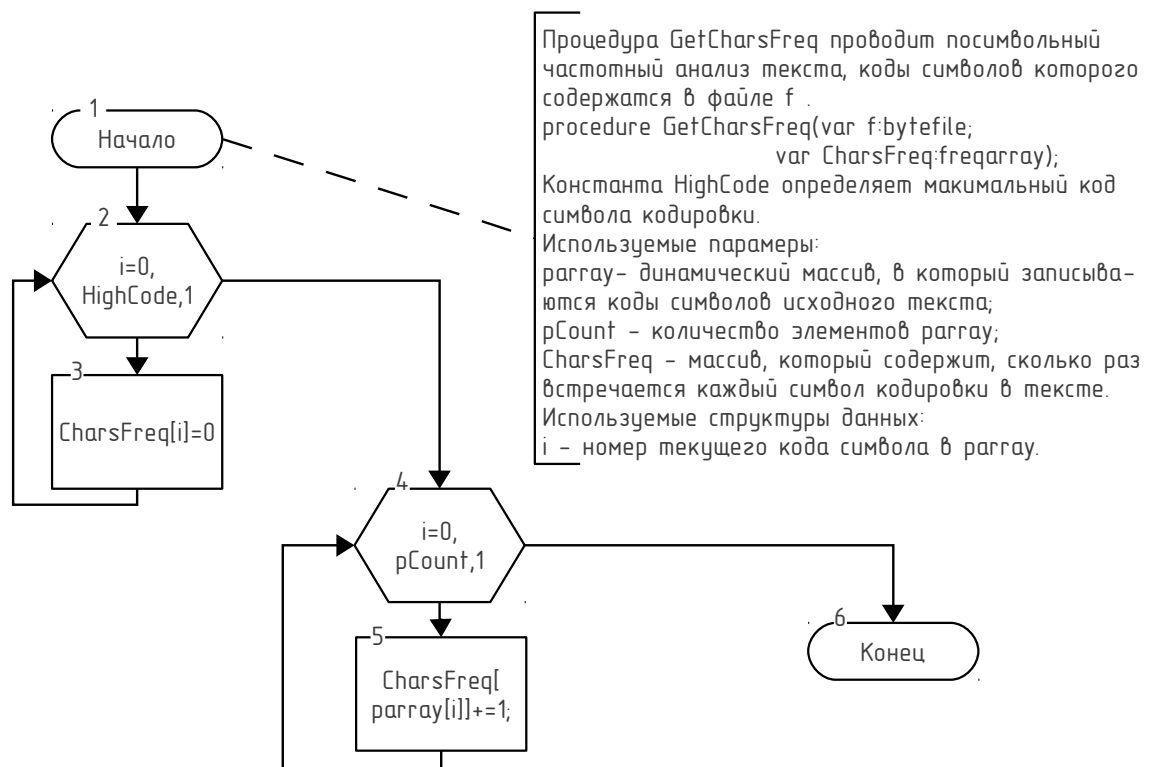


Рисунок 6 — Блок-схема алгоритма посимвольного частотного анализа

На рисунке 7 представлена схема алгоритма поиска самых частых букв при помощи данных частотного анализа.

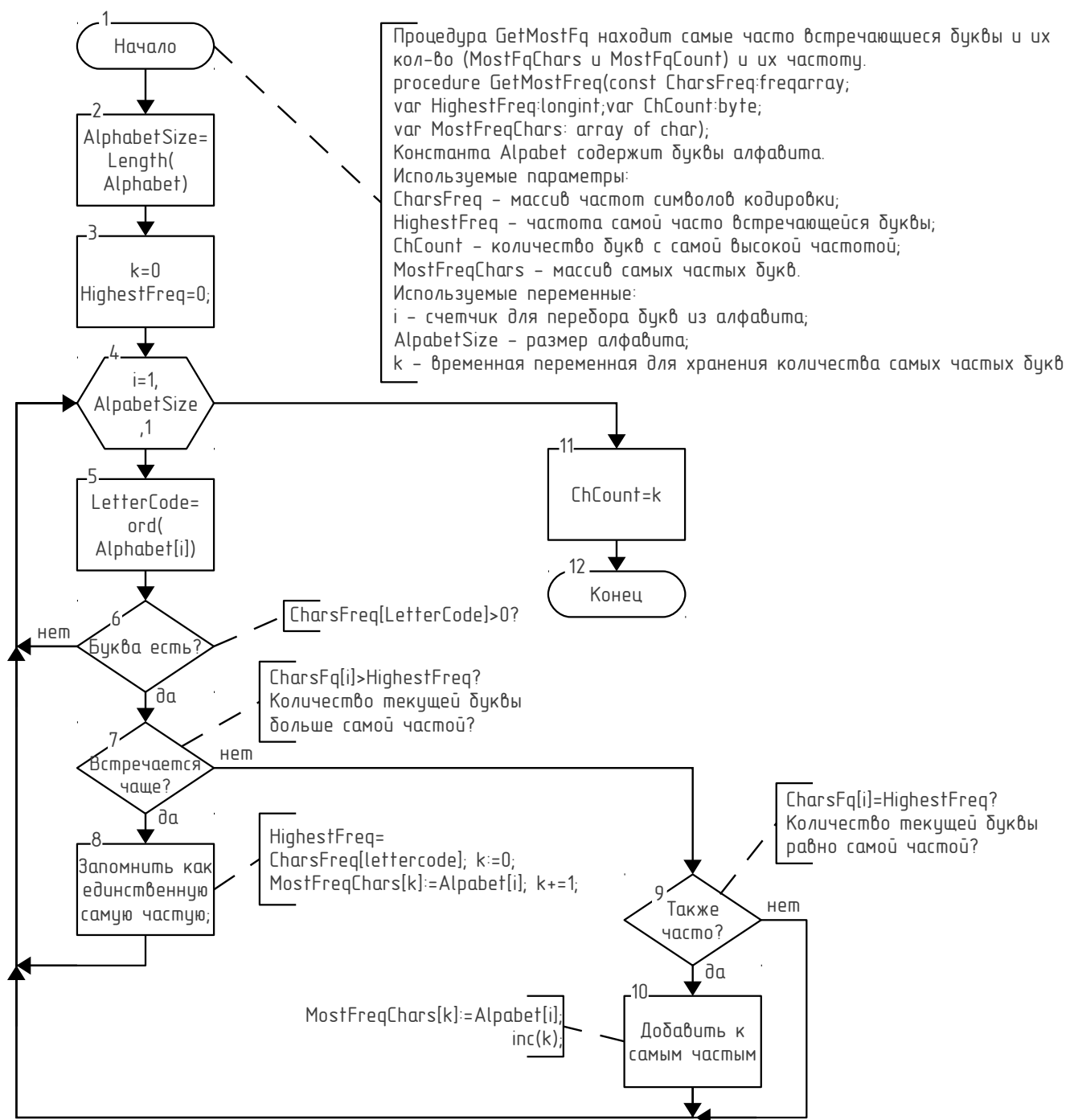


Рисунок 7 — Блок-схема алгоритма поиска самых часто встречающихся букв

На рисунке 8 представлена схема алгоритма вывода в файл самых часто встречающихся букв исходного текста.

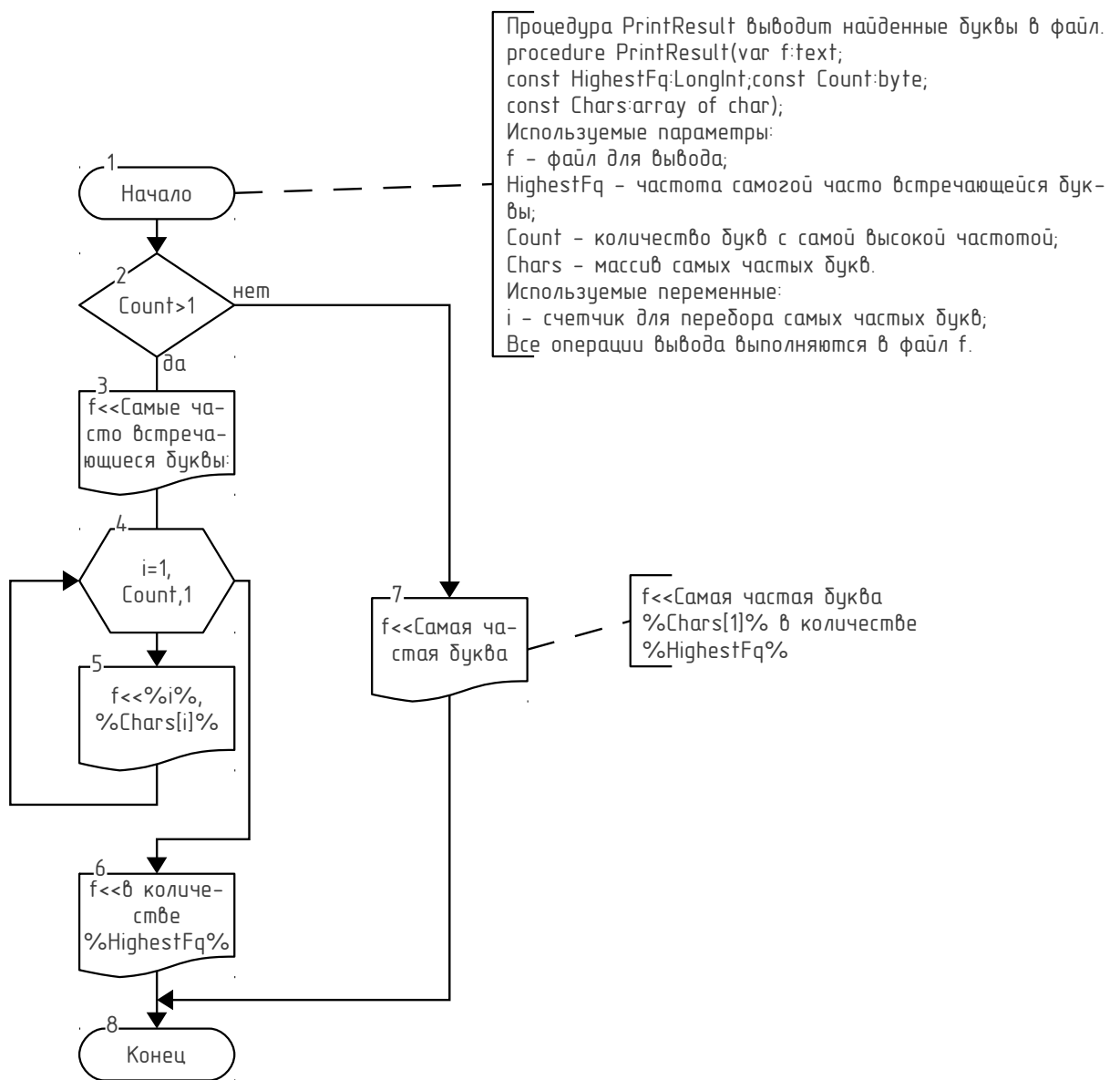


Рисунок 8 — Схема алгоритма вывода в файл самых частых букв

Инструкция пользователю

Данная программа позволяет найти самую часто используемую букву в текстовом файле.

Для работы программы необходимо ввести имена некоторых файлов. Первый файл - это файл, в котором находится текст для поиска. Его имя длиной не более 255 символов нужно передать программе. Затем передайте программе имя временного файла для кодов символов исходного текста. Внимание! Если файл с указанным именем существует, то вся информация в нем будет стерта! Пожалуйста, проверьте имя файла, так как восстановить потерянные данные будет невозможно. Далее передайте программе имя файла для вывода результатов поиска. Внимание! Если файл с указанным именем существует, то вся информация в нем будет стерта! Пожалуйста, проверьте имя файла, так как восстановить потерянные данные будет невозможно. В случае неправильного ввода имени файла-источника (если такой не существует), или при невозможности получить доступ к какому-нибудь файлу, имеется возможность ввести заново имя этого файла, ответив на запрос программы о продолжении ввода 'Y', или отказаться от повторного ввода и завершить программу, ответив 'N'.

Если во время записи кодов символов программа не сможет записать очередной код, то она завершит свою работу. Следует проверить доступность файла и свободное место, а затем перезапустить программу и ввести заново данные.

Если найдена всего одна самая часто встречающаяся буква, то будет выведена она и её частота в указанный файл и на экран; если существует несколько букв с самой высокой частотой появления, то они будут выведены нумерованным списком, и в конце будет выведена их частота, в указанный файл и на экран.

Инструкция программисту

При создании программы поиска однокоренных слов были предприняты следующие действия.

В основной части программы определены константы:

1. Alphabet - строка, символы которой должны определяться как буквы;
2. HighCode - максимальное значения кода символа в кодировке. Программа работает корректно с русскими символами только в 8-битных кодировках, поэтому значение HighCode, скорее всего должно быть нестрого меньше 255.

Определены типы:

1. bytearray как file of byte - типизированный файл, куда записываются и впоследствии считываются коды символов исходного текста.
2. freqarray как array [0..HighCode] of longint - тип массива частот символов кодировки в тексте.

Были введены структуры данных, описание которых представлено в таблице 1.

Таблица 1 - Структуры данных, используемые в в основной части программы поиска самой частой буквы в тексте

имя	тип	предназначение
fin	bytefile	Входной файл с кодами символов.
fout	text	Выходной файл для записи результатов работы
HighestFq	longint	Частота наиболее часто встречающейся буквы.
MostFqCount	byte	Количество наиболее часто встречающихся букв.
CharsFq	freqarray	Массив частот символов кодировки в тексте.
MostFqChars	array [byte] of char	Наиболее часто встречающиеся буквы текста.
log	boolean	Указывает на отсутствие ошибок (значение true) ввода данных или желания пользователя прервать программу.

Кроме того, в процессе создания вышеуказанной программы были определены следующие подпрограммы:

1. Функция GetFiles запрашивает у пользователя все необходимые для работы файлы, и записывает во входной файл коды символов из файла-источника. Возвращает true, если все необходимые файлы получены, иначе возвращается false.

```
function GetFiles(var fin:bytefile;var fout:text):boolean;
```

В теле функции через локальные функции GetSourceFile, GetInputFile, GetOutputFile запрашиваются файл-источник с текстом, входной типизированный файл для кодов символов, текстовый выходной файл соответственно. Если на каком-то этапе возникла ошибка, то последующие этапы не выполняются и возвращается false. Далее из файла-источника считывается каждый символ, и коды этих символов записываются во входной типизированный файл. После завершения этой операции файл-источник закрывается, а типизированный файл переоткрывается для чтения. Далее функция завершает работу и возвращает true.

Используемые функцией параметры-переменные приведены в таблице 2, локальные переменные - в таблице 3.

Таблица 2 - Параметры-переменные функции получения файлов

имя	тип	предназначение
fin	bytefile	Входной файл с кодами символов .
fout	text	Выходной файл для записи результатов.

Таблица 3 - Локальные переменные функции получения файлов

имя	тип	предназначение
error	integer	Код ошибки ввода-вывода, возникшей при открытии файла(код 0 - нет ошибки) .
fsrc	text	Файл-источник с исходным текстом.
i	byte	Код текущего символа, считанного из текста.
ch	char	Текущий символ, считанный из текста.

Для получения файла каждого типа были введены следующие функции:

1.1 Функция GetSourceFile запрашивает у пользователя имя файла-источника с анализируемым текстом, и, если возможно, открывает его для чтения. Возвращает код возникшей при открытии ошибки ввода-вывода (код 0 - нет ошибки).


```
function GetSourceFile(var f:text):integer;
```

В теле функции в цикле с постусловием происходит запрос у пользователя имени файла-источника, и проводится попытка его открытия в режиме для чтения. Если файл открылся успешно, функция завершает работу и передает открытый файл , и значение функции, равное 0. Если нет, то производится запрос пользователя о продолжении работы. Если пользователь ответил 'N', функция завершает работу и возвращает значение ошибки ввода-вывода, Иначе операция повторяется до достижения двух вышеуказанных условий.

Используемые функцией параметры-переменные приведены в таблице 4, локальные переменные - в таблице 5.

Таблица 4 - Параметры-переменные функции получения файла-источника

имя	тип	предназначение
f	text	Файл-источник с исходным текстом.

Таблица 5 - Локальные переменные функции получения файла-источника

имя	тип	предназначение
ch	char	Содержит ответ пользователя на запросы программы о повторении ввода данных.
fileName	string	Имя файла-источника.
error	integer	Код ошибки ввода-вывода, возникшей при открытии файла(код 0 - нет ошибки) .

1.2 Функция GetInputFile запрашивает у пользователя имя входного файла, и пытается открыть его для записи. Возвращает код возникшей при открытии ошибки ввода-вывода (код 0 - нет ошибки).

```
function GetInputFile(var f:bytefile):integer;
```

В теле функции в цикле с постусловием происходит запрос у пользователя имени входного файла, и проводится попытка его открытия в режиме для записи. Если файл открылся успешно, функция завершает работу и передает открытый файл , и значение функции, равное 0. Если нет, то производится запрос пользователя о продолжении работы. Если пользователь ответил 'N', функция завершает работу и возвращает значение ошибки ввода-вывода, Иначе операция повторяется до достижения двух вышеуказанных условий.

Используемые функцией параметры-переменные приведены в таблице 6, локальные переменные - в таблице 7.

Таблица 6 - Параметры-переменные функции получения типизированного входного файла

имя	тип	предназначение
f	text	Входной типизированный файл для кодов символов файла источника.

Таблица 7 - Локальные переменные функции получения типизированного входного файла

имя	тип	предназначение
ch	char	Содержит ответ пользователя на запросы программы о повторении ввода данных.
fileName	string	Имя входного файла.
error	integer	Код ошибки ввода-вывода, возникшей при открытии файла(код 0 - нет ошибки) .

1.3 Функция GetOutputFile запрашивает у пользователя имя выходного файла, и пытается открыть его для записи. Возвращает код возникшей при открытии ошибки ввода-вывода (код 0 - нет ошибки).

```
function GetOutputFile(var f:text):integer;
```

В теле функции в цикле с постусловием происходит запрос у пользователя имени выходного файла, и проводится попытка его открытия в режиме для записи. Если файл открылся успешно, функция завершает работу и передает открытый файл , и значение функции, равное 0. Если нет, то производится запрос пользователя о продолжении работы. Если пользователь ответил 'N', функция завершает работу и возвращает значение ошибки ввода-вывода, Иначе операция повторяется до достижения двух вышеуказанных условий.

Используемые функцией параметры-переменные приведены в таблице 8, локальные переменные - в таблице 9.

Таблица 8 - Параметры-переменные функции получения текстового выходного файла

имя	тип	предназначение
f	text	Выходной текстовый файл.

Таблица 9 - Локальные переменные функции получения текстового выходного файла

имя	тип	предназначение
ch	char	Содержит ответ пользователя на запросы программы о повторении ввода данных.
fileName	string	Имя выходного файла.
error	integer	Код ошибки ввода-вывода, возникшей при открытии файла(код 0 - нет ошибки) .

2. Функция GetCharsFreq проводит частотный анализ текста, коды которого записаны в типизированном файле. Использует константу HighCode.

```
procedure GetCharsFreq(var f:bytefile;var CharsFreq:freqarray);
```

В начале массив частот символов кодировки (от 0 до HighCode) обнуляется. Затем из типизированного файла считываются коды символов, и элемент массива частот, имеющий индекс, равный коду символа, увеличивается на единицу.

Используемые функцией параметры-переменные приведены в таблице 10; локальные переменные - в таблице 11.

Таблица 10 - Параметры-переменные процедуры посимвольного частотного анализа

имя	тип	предназначение
f	bytefile	Входной файл с кодами символов .
CharsFreq	freqarray	Массив частот символов кодировки в тексте.

Таблица 11 - Локальные переменные процедуры посимвольного частотного анализа

имя	тип	предназначение
i	byte	Код текущего символа, считанный из типизированного файла.

3. Функция GetMostFreq с помощью данных частотного анализа находит самые частые буквы в тексте. Использует константу Alpabet.

```
procedure GetMostFreq(const CharsFreq:freqarray; var HighestFreq:longint;
var ChCount:byte; var MostFreqChars:array of char);
```

При инициализации количество самых частых букв и наивысшая частота обнуляются.

В цикле с параметром перебираются буквы из строки Alphabet, и если частота её появления не равна 0 (элемент массива частот имеющий индекс, равный коду символа, не равен 0), то тогда если её частота больше максимальной, то эта частота запоминается как максимальная, количество букв устанавливается в 1, и буква добавляется в массив; если равна, то количество букв увеличивается на 1, и буква добавляется в массив.

Используемые функцией параметры-константы приведены в таблице 12; параметры-переменные - в таблице 13; локальные переменные - в таблице 14.

Таблица 12 - Параметры-константы процедуры поиска самых часто встречающихся букв

имя	тип	предназначение
CharsFreq	freqarray	Массив частот символов кодировки в тексте.

Таблица 13 - Параметры-переменные процедуры поиска самых часто встречающихся букв

имя	тип	предназначение
HighestFreq	longint	Частота наиболее часто встречающейся буквы.
ChCount	byte	Количество наиболее часто встречающихся букв.
MostFreqChars	array of char	Наиболее часто встречающиеся буквы текста.

Таблица 14 - Локальные переменные процедуры поиска самых часто встречающихся букв

имя	тип	предназначение
i	byte	Переменная-счетчик для перебора ,букв - элементов строки Alphabet.
k	byte	Временная переменная для хранения количества наиболее часто встречающихся букв.
AlphabetSize	byte	Количество букв в Alphabet.
lettercode	byte	Код текущей буквы из Alphabet.

4. Процедура PrintResult выводит найденные самые частые буквы в тексте.

```
procedure PrintResult(var f:text;const HighestFq:LongInt;const Count:byte;
                     const Chars:array of char);
```

Если в переданном массиве всего одна буква, то она и её частота просто выводятся в соответствующем сообщении; иначе выводятся нумерованный список букв, а в конце - частота их появлению.

Используемые процедурой параметры-константы приведены в таблице 15, параметры-константы - в таблице 16, локальные переменные - в таблице 17 .

Таблица 15 - Параметры-константы процедуры вывода самых часто встречающихся букв

имя	тип	предназначение
HighestFq	LongInt	Частота наиболее часто встречающейся буквы.
Count	byte	Количество наиболее часто встречающихся букв.
Chars	array of char	Наиболее часто встречающиеся буквы текста.

Таблица 16 - Параметры-переменные процедуры вывода самых часто встречающихся букв

имя	тип	предназначение
f	text	Выходной файл для записи результатов.

Таблица 17 - Локальные переменные процедуры вывода самых часто встречающихся букв

имя	тип	предназначение
i	byte	Переменная-счетчик для обработки массива.

Текст программы

Ниже представлен текст программы, написанной на языке Turbo Pascal 7, которая находит самую часто встречающуюся букву в тексте.

```
const
    alphet='qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM'+
        'ёйцукенгшщзхъфывапролджэячсмитьбюЁЙЦУКЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБЮ';
const highcode=255;
type bytefile=file of byte;
type freqarray=array [0..highcode] of longint;
{-----ФУНКЦИЯ ПОЛУЧЕНИЯ ФАЙЛОВ-----}
function GetFiles(var fin:bytefile;var fout:text):boolean;

{-----ФУНКЦИЯ ПОЛУЧЕНИЯ ФАЙЛА ИСТОЧНИКА-----}
function GetSourceFile(var f:text):integer;
var ch:char;error:integer;
    fileName:string;
begin
    repeat
        WriteLn('Введите имя файла для анализа');
        Write('Файл:');ReadLn(fileName);
        Assign(f,fileName);
        {$I-}
        Reset(f);
        {$I+}
        error:=ioresult;
        if error<>0 then begin
            WriteLn('Неправильное имя файла! Повторить ввод? <Y>/<N>');
            ReadLn(ch);
        end;
    until (UpCase(ch)='N') or (error=0);
    GetSourceFile:=error;
end;

{-----ФУНКЦИЯ ПОЛУЧЕНИЯ ВХОДНОГО ФАЙЛА-----}
function GetInputFile(var f:bytefile):integer;
var ch:char; error:integer;
    fileName:string;
begin
    repeat
        WriteLn('Введите имя временного файла');
        Write('Файл:');ReadLn(fileName);
        Assign(f,fileName);
        {$I-}
```

```

        Rewrite(f);
    {$I+}
    error:=ioresult;
    if error<>0 then begin
        WriteLn('Ошибка при создании файла! Повторить ввод? <Y>/<N>');
        ReadLn(ch);
    end;
until (UpCase(ch)='N') or (error=0);
    GetInputFile:=error;
end;

{-----ФУНКЦИЯ ПОЛУЧЕНИЯ ВЫХОДНОГО ФАЙЛА-----}
function GetOutputFile(var f:text):integer;
var ch:char; error:integer;
    fileName:string;
begin
    repeat
        WriteLn('Введите имя файла-результата. ');
        Write('Файл: '); ReadLn(fileName);
        Assign(f, fileName);
    {$I-}
        Rewrite(f);
    {$I+}
    error:=ioresult;
    if error<>0 then begin
        WriteLn('Ошибка при создании файла! Повторить ввод? <Y>/<N>');
        ReadLn(ch);
    end;
until (UpCase(ch)='N') or (error=0);
    GetOutputFile:=error;
end;

var error:integer; fsrc:text; i:byte; ch:char;
begin
    error:=GetSourceFile(fsrc);
    if error=0 then
        error:=GetInputFile(fin);
    if error=0 then
        error:=GetOutputFile(fout);
    GetFiles:=error=0;
    if error=0 then begin
        Writeln('Считываю данные во временный файл...');
    {$I-}
        while not EOF(fsrc) and (ioresult=0) do begin

```

```

        Read(fsrc,ch);
        i:=ord(ch);
        Write(fin,i);

    end;
    {$I+}
    Close(fin);
    if EOF(fsrc) then begin
        WriteLn('Выполнено!');
        Reset(fin);
    end else begin
        GetFiles:=false;
    end;
    Close(fsrc);
end;
end;

{-----ФУНКЦИЯ ПОСИМВОЛЬНОГО ЧАСТОТНОГО АНАЛИЗА-----}
procedure GetCharsFreq(var f:bytefile;var CharsFreq:freqarray);
var i:byte;
begin
    for i:=0 to highcode do
        CharsFreq[i]:=0;

    WriteLn('Выполнено:');
    Write(0:3,'%');
    while not EOF(f) do begin
        Read(f,i);Write(#13,FilePos(f)/FileSize(f)*100:3:0,'%');
        inc(CharsFreq[i]);

    end;

    WriteLn;
end;

{-----ФУНКЦИЯ ПОЛУЧЕНИЯ САМЫХ ЧАСТЫХ БУКВ-----}
procedure GetMostFreq(const CharsFreq:freqarray;
                        var HighestFreq:longint;var ChCount:byte;
                        var MostFreqChars:array of char);
var k,i:byte; AlpabetSize,lettercode:byte;
begin
    AlpabetSize:=length(alpabet);
    k:=0;
    HighestFreq:=0;

```



```

for i:=1 to AlpabetSize do begin
    lettercode:=ord(Alpabet[i]);
    if (CharsFreq[lettercode]>0) then begin

        if (CharsFreq[lettercode]>HighestFreq) then begin
            HighestFreq:=CharsFreq[lettercode];
            k:=0;
            MostFreqChars[k]:=Alpabet[i]; inc(k);
        end else
            if CharsFreq[lettercode]=HighestFreq then begin

                MostFreqChars[k]:=Alpabet[i]; inc(k);
            end;
        end;
    end;
    ChCount:=k;
end;

{-----ФУНКЦИЯ ВЫВОДА САМЫХ ЧАСТЫХ БУКВ НА ЭКРАН-----}
procedure PrintResult(var f:text;const HighestFq:LongInt;const Count:byte;
    const Chars:array of char);
var i:byte;
begin
    if Count>1 then begin
        WriteLn(f,'Наиболее часто встречаются буквы');
        for i:=0 to Count-1 do begin
            Write(f,i+1,'. ',Chars[i],' ');
        end;
        WriteLn(f,#13#10,'в количестве ',HighestFq,' шт.');
```

```

    end else begin
        WriteLn(f,'Наиболее часто встречается буква',#10#13,
            ' ',Chars[0],' в количестве ',HighestFq,' шт.');
```

```

    end;
end;

{-----ОСНОВНАЯ ПРОГРАММА-----}
var log:boolean;MostFqCount:byte; HighestFq:LongInt; fin:bytefile; fout:text;
    CharsFq:freqarray;MostFqChars:array[byte] of char;
BEGIN
    WriteLn('Программа находит самую часто встречающуюся букву в тексте
    следующим образом:');
    WriteLn('Программа считывает символы из текстового файла-источника;');
    WriteLn('затем записывает их коды во входной типизированный файл;');
```

```

        WriteLn('далее открывает этот файл, считывает данные, и находит самую частую
букву;');

        WriteLn('выводит результат на экран и в текстовый файл.');
```

log:=GetFiles(fin,fout);

if log then begin

 WriteLn('Выполняется поиск самой часто встречающейся буквы.');

 GetCharsFreq(fin,CharsFq);

 Close(fin);

 GetMostFreq(CharFq,HighestFq,MostFqCount,MostFqChars);

 if MostFqCount<>0 then begin

 {\$I-}

 PrintResult(output,HighestFq,MostFqCount,MostFqChars);

 PrintResult(fout,HighestFq,MostFqCount,MostFqChars);

 {\$I+}

 end else begin

 {\$I-}

 WriteLn('В тексте нет букв.');

 WriteLn(fout,'В тексте нет букв.');

 {\$I+}

 end;

 Close(fout);

end else

 WriteLn('Программа завершена пользователем.');

 WriteLn('Нажмите <Enter>...');

 ReadLn;

END.

Тестовые примеры

На рисунке 8 представлен пример работы программы для текста на русском языке размером примерно 90 Кбайт.

```
Программа находит самую часто встречающуюся букву в тексте следующим образом:  
Программа считывает символы из текстового файла-источника;  
затем записывает их коды во входной типизированный файл;  
далее открывает этот файл, считывает данные, и находит самую частую букву;  
выводит результат на экран и в текстовый файл.  
Введите имя файла для анализа  
файл:lab5\labirint.txt  
Введите имя временного файла  
файл:in  
Введите имя файла-результата.  
файл:out  
Считываю данные во временный файл...  
Выполнено!  
Выполняется поиск самой часто встречающейся буквы.  
Выполнено:  
100%  
Наиболее часто встречается буква  
'o' в количестве 6707 шт.  
Нажмите <Enter>...
```

Рисунок 8 - Пример работы программы для произвольного файла

На рисунке 9 представлен пример работы программы для файла, содержащего строку “sssdddfffggghj”.

```
Программа находит самую часто встречающуюся букву в тексте следующим образом:  
Программа считывает символы из текстового файла-источника;  
затем записывает их коды во входной типизированный файл;  
далее открывает этот файл, считывает данные, и находит самую частую букву;  
выводит результат на экран и в текстовый файл.  
Введите имя файла для анализа  
файл:lab5\test.txt  
Введите имя временного файла  
файл:in  
Введите имя файла-результата.  
файл:out  
Считываю данные во временный файл...  
Выполнено!  
Выполняется поиск самой часто встречающейся буквы.  
Выполнено:  
100%  
Наиболее часто встречаются буквы  
1.'s' 2.'d' 3.'f' 4.'g'  
в количестве 3 шт.  
Нажмите <Enter>...
```

Рисунок 9 - Пример работы программы для файла с несколькими самыми частыми буквами

Вывод

В ходе выполнения данной лабораторной работы я научился использовать файлы различных типов при разработке программ. Файлы позволяют долговременно хранить большие объёмы данных и передавать их между совершенно различными программами. К сожалению, работа с файлами ведется медленнее, чем с оперативной памятью.