ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Обработка данных в файлах |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4135К |  |  |  | Сенин А.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**1.Цель работы:** Целью работы является изучение файловых операций ввода-вывода

**2.Задачи работы:**

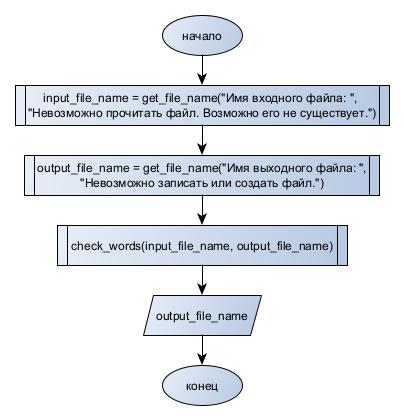
Написать программу, которая считывает текст из файла и записывает в другой файл строки, не содержащие двузначные числа. **3.Описание функций:**  
**1)Имя: main для fstream и cstdio**  
**Назначение:** считывание текста из файла и запись в файл строки, которая не содержит двузначные числа

**Входные данные:** нет  
**Выходные данные:** нет  
**Побочный эффект:** отсутствует.

**Тестовые данные:**   
**Ввод: 24 343**

**Вывод:** **343**  
**Прототип:** int main()  
**Псевдокод:**вводим имена файлов (с проверкой)

подсчитываем найденные слова

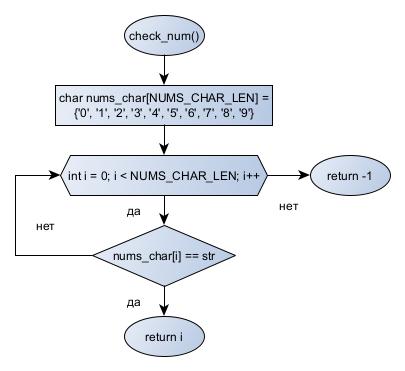
запись результата в файл  
**Блок-схема:  
**

**2)Имя: check\_num()**  
**Назначение:** функция для определения числа

**Входные данные:** str  
**Выходные данные:** i  
**Побочный эффект:** отсутствует.   
**Прототип:** int check\_num()  
**Псевдокод:**перебор символа и сравнения его с массивом чисел

Если сравнение верное, то возврат в main числа

Если нет, то возврат в main (-1)  
**Блок-схема:**

****

**3)Имя: check\_words() для fstream**  
**Назначение:** функция для подсчета слов в предложениях

**Входные данные:** \*input\_file\_name, \*output\_file\_name   
**Выходные данные:** нет  
**Побочный эффект:** отсутствует.   
**Прототип:** void check\_words()  
**Псевдокод:**открываем файл

Цикл по всему файлу

Берем из файла первый символ

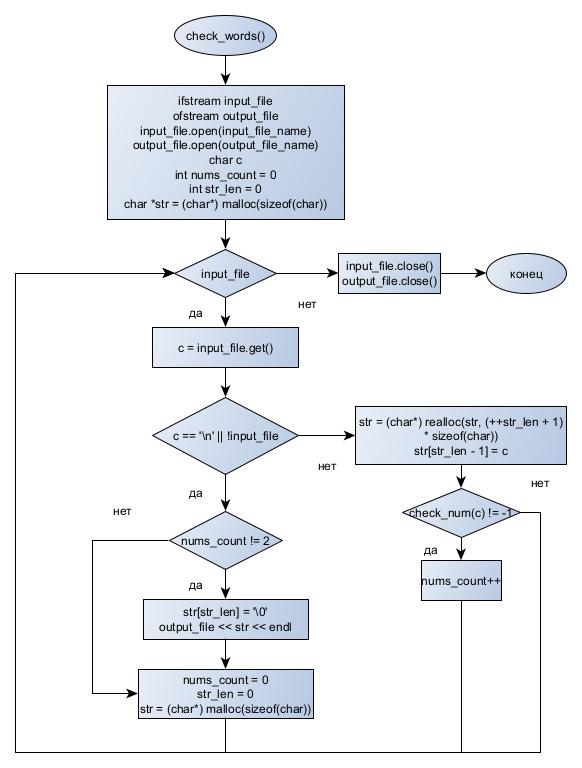
записываем строку, если это конец строки или файла

После чего обнуляем все переменные

Иначе записываем строку в массив

если в строке есть цифра, то прибавляем к счётчику 1

закрываем файл  
**Блок-схема:**



**4)Имя: check\_words() для cstdio**  
**Назначение:** функция для подсчета слов в предложениях

**Входные данные:** \*input\_file\_name, \*output\_file\_name   
**Выходные данные:** нет  
**Побочный эффект:** отсутствует.   
**Прототип:** void check\_words()  
**Псевдокод:**открываем файл

Цикл по всему файлу

Берем из файла первый символ

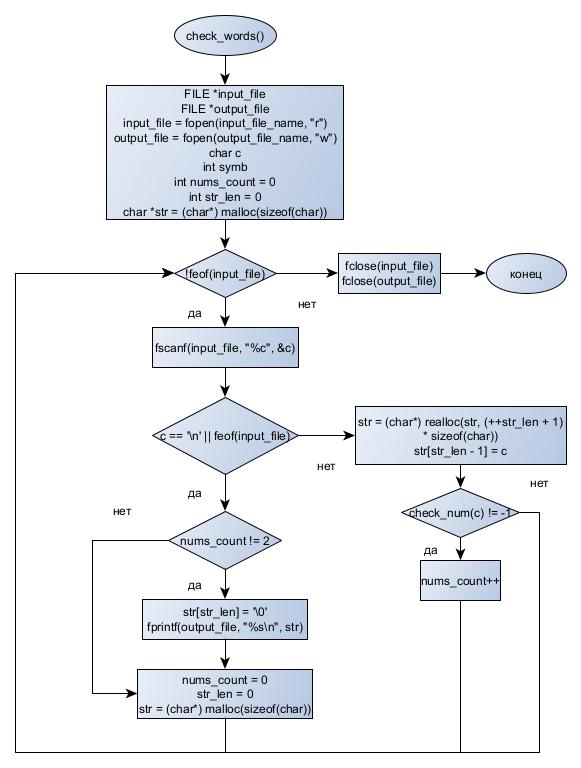
записываем строку, если это конец строки или файла

После чего обнуляем все переменные

Иначе записываем строку в массив

если в строке есть цифра, то прибавляем к счётчику 1

закрываем файл  
**Блок-схема:**

****

**5.ЛИСТИНГ КОДА:  
1)Для fstream**

#include <iostream>

using namespace std;

#include "libs/lib.h"

#include <cmath>

#include <time.h>

#include <fstream>

#include <cctype>

v

#include "libs/simple\_char.h"

#include "fstream\_files.h"

char \*get\_file\_name(const char \*promt = "", const char \*error\_promt = "", bool new\_file = false) {

ifstream input\_file;

int len;

char \*input\_file\_name;

bool ok = false;

while (!ok) {

cout << promt;

input\_file\_name = get\_string(&len);

if (len > 0) {

ok = true;

input\_file.open(input\_file\_name);

if (!input\_file.good()) {

if (new\_file) {

ofstream ost(input\_file\_name);

} else {

cout << error\_promt << endl;

ok = false;

}

}

input\_file.close();

}

}

return input\_file\_name;

}

void check\_words(char \*input\_file\_name, char \*output\_file\_name) {

ifstream input\_file;

ofstream output\_file;

// открываем файлы

input\_file.open(input\_file\_name);

output\_file.open(output\_file\_name);

// буферная переменная для хранения считанного символа

char c;

int nums\_count = 0;

int str\_len = 0;

// массив для сохранения считанной строки

char \*str = (char\*) malloc(sizeof(char));

// цикл по всему файлу

while (input\_file) {

// берём из файла 1 символ

c = input\_file.get();

// записываем строку, если это конец строки или файла

if (c == '\n' || !input\_file) {

if (nums\_count != 2) {

str[str\_len] = '\0';

output\_file << str << endl;

}

// обнуляем все переменные

nums\_count = 0;

str\_len = 0;

str = (char\*) malloc(sizeof(char));

} else {

// зписываем строку в массив

str = (char\*) realloc(str, (++str\_len + 1) \* sizeof(char));

str[str\_len - 1] = c;

// если в строке есть цифра, то прибавляем к счётсчику 1

if (check\_num(c) != -1) nums\_count++;

}

}

// закрываем файлы

input\_file.close();

output\_file.close();

}

int check\_num(char str) {

char nums\_char[NUMS\_CHAR\_LEN] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};

for (int i = 0; i < NUMS\_CHAR\_LEN; i++) {

if (nums\_char[i] == str) {

return i;

}

}

return -1;

}

int main() {

system("chcp 65001");

//clear\_scr();

char \*input\_file\_name;

char \*output\_file\_name;

// вводим имена файлов (с проверкой)

input\_file\_name = get\_file\_name("Имя входного файла: ", "Невозможно прочитать файл. Возможно его не существует.");

output\_file\_name = get\_file\_name("Имя выходного файла: ", "Невозможно записать или создать файл.", true);

// подсчитываем найденые слова

check\_words(input\_file\_name, output\_file\_name);

draw\_line(20);

cout << "В файл " << output\_file\_name << " был успешно записан результат." << endl;

return 0;

}

**2)Для cstdio**

#include <iostream>

using namespace std;

#include "libs/lib.h"

#include <cmath>

#include <time.h>

#include <cstdlib>

#include <cctype>

// проверка ввода

#include "libs/simple\_char.h"

// набор функций для этой лабы

#include "cstdio\_files.h"

char \*get\_file\_name(const char \*promt = "", const char \*error\_promt = "") {

FILE \*input\_file;

int len;

char \*input\_file\_name;

bool ok = false;

while (!ok) {

cout << promt;

input\_file\_name = get\_string(&len);

if (len > 0) {

ok = true;

input\_file = fopen(input\_file\_name, "rw"); // если выдаёт ошибку поставь "r" вместо "rw"

if (input\_file == NULL) {

cout << error\_promt << endl;

ok = false;

}

fclose(input\_file);

}

}

return input\_file\_name;

}

// функция для определения цифры

int check\_num(char str) {

char nums\_char[NUMS\_CHAR\_LEN] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};

for (int i = 0; i < NUMS\_CHAR\_LEN; i++) {

if (nums\_char[i] == str) {

return i;

}

}

return -1;

}

void check\_words(char \*input\_file\_name, char \*output\_file\_name) {

FILE \*input\_file;

FILE \*output\_file;

input\_file = fopen(input\_file\_name, "r");

output\_file = fopen(output\_file\_name, "w");

char c;

int symb;

int nums\_count = 0;

int str\_len = 0;

// массив для сохранения считанной строки

char \*str = (char\*) malloc(sizeof(char));

while (!feof(input\_file)) {

fscanf(input\_file, "%c", &c);

if (c == '\n' || feof(input\_file)) {

if (nums\_count != 2) {

str[str\_len] = '\0';

fprintf(output\_file, "%s\n", str);

}

// обнуляем все переменные

nums\_count = 0;

str\_len = 0;

str = (char\*) malloc(sizeof(char));

} else {

// зписываем строку в массив

str = (char\*) realloc(str, (++str\_len + 1) \* sizeof(char));

str[str\_len - 1] = c;

// если в строке есть цифра, то прибавляем к счётсчику 1

if (check\_num(c) != -1) nums\_count++;

}

}

fclose(input\_file);

fclose(output\_file);

}

int main() {

// смена кодировки

system("chcp 65001");

// очистка терминала

//clear\_scr();

char \*input\_file\_name;

char \*output\_file\_name;

input\_file\_name = get\_file\_name("Имя входного файла: ", "Невозможно прочитать файл. Возможно его не существует.");

output\_file\_name = get\_file\_name("Имя выходного файла: ", "Невозможно записать или создать файл.");

check\_words(input\_file\_name, output\_file\_name);

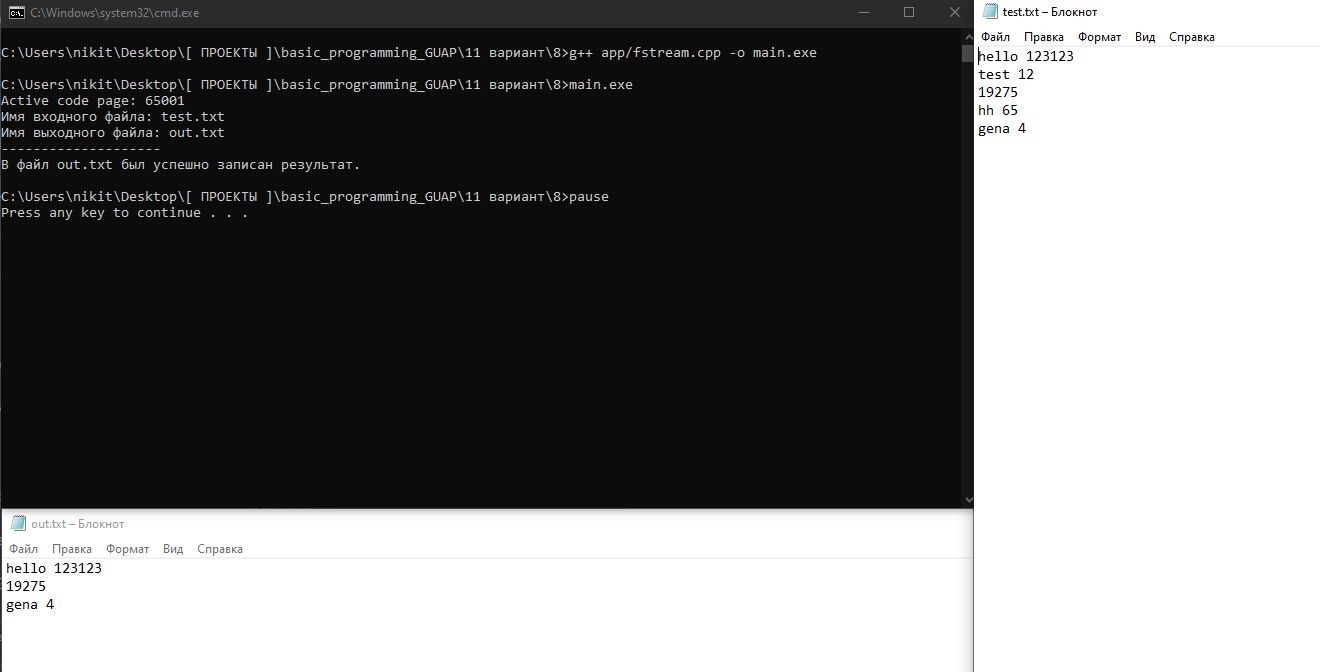
draw\_line(20);

cout << "В файл " << output\_file\_name << " был успешно записан результат." << endl;

return 0;

}

**6.Пример выполнения программы:**



**7.Анализ результатов и выводы:**

В ходе этой лабораторной работы мы научились работать с файлами.

Из плюсов можно отметить, что программа работает корректно.

Существенных минусов не обнаружено.