ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

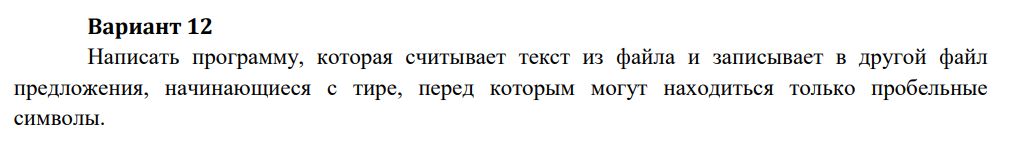
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

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

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4135К |  |  |  | Столяров Н.С. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**1.Цель работы:** Целью работы является изучение файловых операций ввода-вывода.  
**2.Задачи работы:**  
**3.Описание функций:  
Имя: main**   
**Назначение:** апись в файл предложений начинающихся с тире.  
**Входные данные:** Название входного файла и выходного.   
**Выходные данные:** файл.  
**Побочный эффект:** отсутствует.   
**Тестовые данные:**

|  |  |
| --- | --- |
| Входная файл | Выходной файл |
| -sadfsf. -1asdfdsf.  -4sdaf. b-hgkuhgusdfhgu.  sadfsadfasdfas.  - sadfsadf. | -sadfsf.  -1asdfdsf.  -4sdaf.  - sadfsadf. |

**Прототип:** int main()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| Ввод входного файла Ввод выходного файла  Формирование выходного файла. |  |

**Имя: get\_file\_name**   
**Назначение:** Проверка файла.  
**Входные данные:** Название файла.   
**Выходные данные:** Название файла.  
**Побочный эффект:** отсутствует.   
**Прототип:** char \*get\_file\_name()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| * Ввод названия файла * Открываем файл * Если файла не существует, либо его нельзя записать, то повторяем всё заново * Закрываем файл Возвращаем название файла | **fstream** |
| **Cstdlib** |

**Имя: check\_words**   
**Назначение:** Запись в файл предложений начинающихся с тире.  
**Входные данные:** Название входного фала, название выходного файла.   
**Выходные данные:** нет.  
**Побочный эффект:** Запись в файл.   
**Прототип:** void check\_words()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| * Открываем файлы * цикл по всему файлу * если в предложение начинается с тире и перед ним ничего нет, то сохраняем его в выходной файл * после цикла закрываем файлы­ | **fstream** |
| **Cstdlib** |

**Имя: read\_value**  
**Назначение:** проверка переменной a на корректность ввода  
**Входные данные:** Отсутствуют.   
**Выходные данные:** x(double).  
**Побочный эффект:** отсутствует.   
**Прототип:** double read\_value()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| Бесконечный цикл  ввод строки  проверка на пробелы и удаление их  проверка на запятые и замена их на точки  проверка на минус  проверка на лишние символы  если не было проблем, то останавливаем бесконечный цикл­  конец бесконечного цикла  перевод строки в число  вывод числа |  |

**ЛИСТИНГ КОДА:**

**fstream.cpp  
/\*   
   Написать программу, которая считывает текст из файла и записывает в другой файл   
 предложения, начинающиеся с тире, перед которым могут находиться только пробельные   
 символы.   
\*/   
#include <iostream>   
using namespace std;   
  
#include "libs/lib.h"   
#include <cmath>   
#include <time.h>   
#include <fstream>   
#include <cctype>   
  
// проверка ввода   
#include "libs/simple\_char.h"   
#include "libs/input\_validation.h"   
  
// набор функций для этой лабы   
#include "fstream\_files.h"   
  
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 существу  
ет.");   
 output\_file\_name = get\_file\_name("Имя выходного файла: ", "Невозможно записать или создать файл.");   
  
 check\_words(input\_file\_name, output\_file\_name, true);   
  
 draw\_line(20);   
  
 cout << "В файл " << output\_file\_name << " был успешно записан результат." << endl;   
  
 return 0;   
}  
fstream\_files.h**

#include <iostream>

using namespace std;

#include <cmath>

#include <time.h>

#include <fstream>

#include <cstdio>

#include <cstdlib>

#include <cctype>

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

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()) {

cout << error\_promt << endl;

ok = false;

}

input\_file.close();

}

}

return input\_file\_name;

}

void check\_words(char \*input\_file\_name, char \*output\_file\_name, bool data = false) {

ifstream input\_file;

ofstream output\_file;

// открываем файлы

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

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

char c;

bool space = true;

bool state = false;

if (data) {

draw\_line(20);

cout << "Отладочная информация." << endl;

draw\_line(20);

}

while (input\_file) {

c = input\_file.get();

if (!space) { // проверка на пробел перед тире

if (isspace(c))

space = true;

else

space = false;

} else if (!state) { // проверка на тире после пробела

if (c == '-') {

state = true;

output\_file << c;

} else {

space = false;

state = false;

}

} else if (state && space) { // записываем предложение в файл

if (data) cout << c;

output\_file << c;

if (c == '.' || c == '?' || c == '!') { // проверка на конец предложения

state = false;

space = false;

if (data) cout << endl;

output\_file << endl;

}

}

}

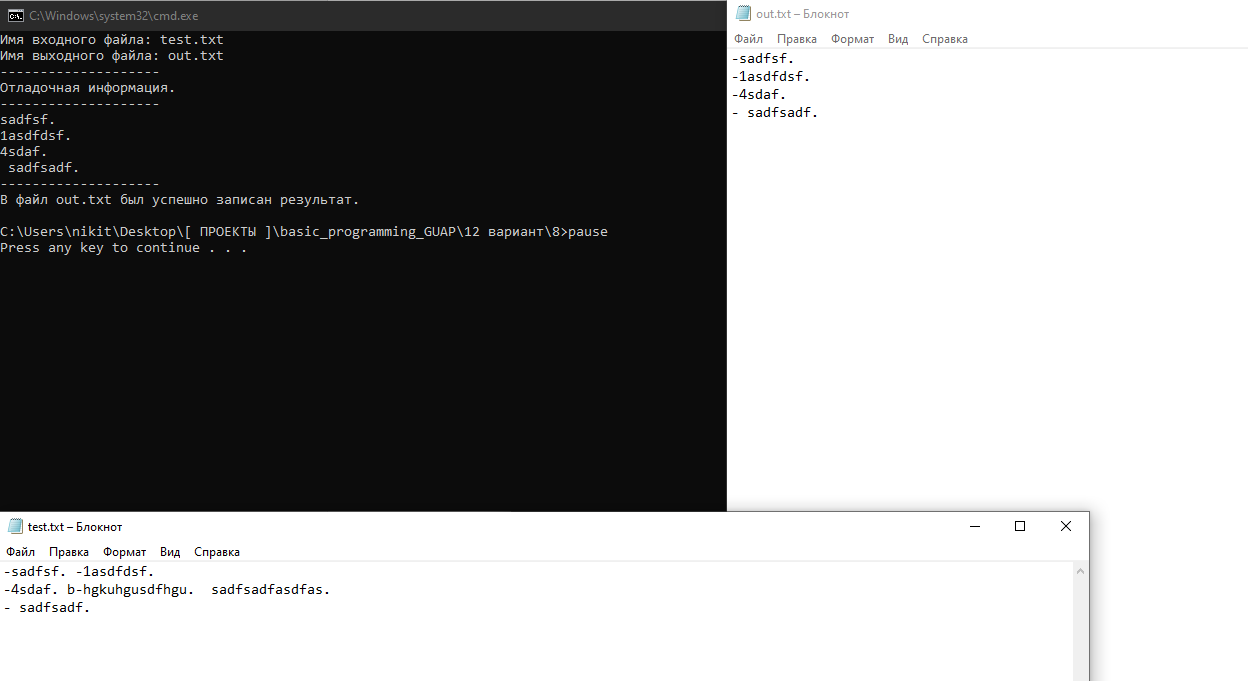
// закрываем файлы

input\_file.close();

output\_file.close();

}

**Пример выполнения программы:**



Видно, что результаты расчётов совпадают с тестовыми данными.

**Анализ результатов и выводы:**

В ходе этой лабораторной работы мы научились работать с файлами (читать и записывать)  
Из плюсов можно отметить её идеальную(может и не идеальную) работу.

Из минусов можно отметить то, что программа написана в двух экземплярах.