Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)

Факультет «Робототехника и комплексная автоматизация»

Кафедра «Системы автоматизированного проектирования»

**Отчет по лабораторной работе**

По курсу «Объектно-ориентированное программирование»

Выполнил: Студент Сергеева Д.К.

Группа РК6-26Б

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2020 г.

**ПЯВУ. Вариант F20**

**Задание:**

Разработать объектно-ориентированную программу для исключения из любого текстового файла повторений пустых строк, чтобы из каждой группы пустых соседних строк оставалась только одна. При этом пустыми считаются строки, где отсутствуют любые другие символы, кроме необязательных пробелов и табуляций. Имя исходного текстового файла должно передаваться программе через аргумент командной строки ее вызова. Результирующий текст после исключений повторов пустых строк должен отображаться через поток стандартного вывода (сout). При разработке программы необходимо использовать методы бесформатного ввода файловых потоков и исключить ее зависимость от длины строк заданного файла.

**Алгоритм:**

Читаем наш исходный файл, считываем строки из этого файла и записываем во временно созданный файл эти строки, пропуская дублированные пустые строки. Далее удаляем всю информацию из исходного файла и записываем туда наши считаные строки, которые временно хранятся во 2 созданном нами файле. Временный файл удаляем.

**Входные данные:**

Имя файл, в котором надо удалить дублированные пустые строки.

**Выходные данные:**

Строки в нашем файле, где мы удалили дублированные пустые строки.

**Текст программы:**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main(int argc, char\* argv[])

{

if (argc != 2)

{

cout << "Wrong number of arguments." << endl;

return (-1);

}

ifstream f\_in;

f\_in.open(argv[1]);

if (!f\_in.is\_open())

{

cout << "File could not open." << endl;

return (-1);

}

ofstream f\_out;

f\_out.open("new.txt");

bool prevLine = false;

while (!f\_in.eof())

{

string buffer;

getline(f\_in, buffer);

int i = 0;

bool lineBlank = true;

while ((lineBlank) && (buffer[i] != '\0'))

{

if ((buffer[i] != ' ') && (buffer[i] != ' '))

{

lineBlank = false;

}

++i;

}

if (!lineBlank)

{

if (f\_in.eof())

f\_out << buffer;

else

f\_out << buffer << endl;

prevLine = false;

}

else if (!prevLine)

{

if (f\_in.eof())

f\_out << buffer;

else

f\_out << buffer << endl;

prevLine = true;

}

}

f\_in.close();

f\_out.close();

f\_out.open(argv[1]);

f\_in.open("new.txt");

while (!f\_in.eof())

{

string buffer;

getline(f\_in, buffer);

if (f\_in.eof())

{

f\_out << buffer;

cout << buffer;

}

else

{

f\_out << buffer << endl;

cout << buffer << endl;

}

}

f\_in.close();

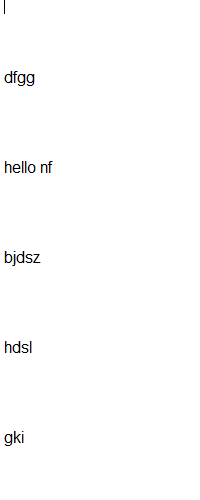
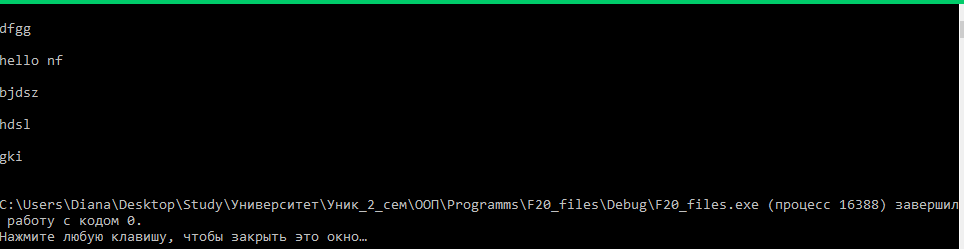
f\_out.close();

remove("new.txt");

return 0;

}

**Тесты:**

Файл:**** ****

**Список использованной литературы:**

* Волосатова Т.М., Родионов С.В. Лекции по курсу «Объектно-ориентированное программирование»
* bigor.bmstu.ru