Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение

высшего образования

«**Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные

системы»

**ОТЧЁТ**

**по лабораторной работе №9**

Дисциплина: «Информатика»

Тема: Структурное программирование. Файловые потоки

Вариант 17

Выполнил:

Студент группы ИВТ-20-2б

Сафронов Владислав Владиславович

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

**Пермь, 2021**

**Цель задачи**

Работа с текстовыми файлами, ввод-вывод текстовой информации и ее хранение на внешних носителях.

**Постановка задачи**

1. Создать текстовый файл F1 не менее, чем из 10 строк и записать в него информацию.
2. Скопировать из файла F1 в файл F2 все строки, начинающиеся на букву «А» расположенные между строками с номерами N1 и N2, а затем все строки от N2+3 и до последней.
3. Определить количество слов в последней строке файла F2.

**Анализ задачи**

1. Для решения задачи необходимо:
   1. Проверить, существует ли файл ввода и в случае его отсутствия сообщить об ошибке.

if (input.is\_open())

{

…

}

else

cout << “Не удалось открыть файл F1.txt\n”

* 1. Разработать функцию rowCounter для подсчета количества строк в файле. Через функцию getline() считывается каждая строка файла, к переменной count каждый раз пребалсяется единица. Затем функция переводит курсор обратно в начало файла и возвращается переменная count, хранящая количество строк в файле.

int rowCounter(ifstream& file) {

string line;

int count = 0;

file.clear();

file.seekg(0);

while (getline(file, line))

{

count++;

}

file.clear();

file.seekg(0);

return count;

}

* 1. С помощью функции rowCounter проверить, что в файле F1.txt содержится более 10 строк. В противном случае сообщить, что файл не соответствует условию задачи.

if (rowCount < 10)

{

cout << "Количество строк в исходном файле должно быть не менее 5!\n";

cout << "Измените файл и запустите программу снова.";

output.close();

}

* 1. Разработать функцию wordsCounter, чтобы посчитать количество слов в последней строке файла.

int wordsCounter(string row)

{

int count = 0;

int i = 0;

while (i < row.length())

{

while (row[i] == ' ') i++;

while (row[i] != ' ' && row[i] != '\n' && row[i] != '\0') i++;

count++;

}

return count;

}

* 1. Реализовать применение этих функций для каждой строки файла F1.txt.

cout << "Количество строк в исходном файле: " << rowCount - 4 << endl;

cout << "Введите число N1 (1-" << rowCount - 4 << "): ";

cin >> n1;

cout << "\nВведите число N2 (" << n1 + 1 << "-" << rowCount - 3 << "): ";

cin >> n2;

int rowNum = 1;

while (rowNum <= rowCount)

{

string row;

string\* words;

int wordsInRow;

if (rowNum >= n1 && rowNum <= n2)

{

getline(input, row);

if (row[0] == 'А' || row[0] == 'а')

{

output << row << endl;

}

}

else if (rowNum >= (n2 + 3) && rowNum <= rowCount)

{

getline(input, row);

output << row << endl;

if (rowNum == rowCount)

{

lastRowWordsCount = wordsCounter(row);

}

}

else getline(input, row);

rowNum++;

}

cout << "\nФайл успешно построен.\n";

cout << "Количество слов в последней строке: " << lastRowWordsCount;

output.close();

1. В ходе работы были использованы следующие типы данных:
   1. Для получения данных из файла испкользуется класс ifstream:

ifstream input(“F1.txt”);

* 1. Для вывода данных в файл используется класс ofstream:

ofstream output(“F2.txt”);

* 1. Для функции wordsCounter используются следующие аргументы:

1. Тип string: входная строка, в которой будет считаться количество слов.

Сама функция имеет тип int для возвращения количества слов в строке:

int wordsCounter(string row)

* 1. Для функции rowCounter были использованы следующие типы данных:

1. Ссылка на ifstream: файл, количество строк которого необходимо посчитать.

Сама функция имеет тип int для возвращения количества строк в файле:

int rowCounter(ifstream& file)

3. Для операций ввода и вывода использовались следующие операторы и функции:

3.1. Ввод данных из файла F1.txt реализован с помощью функции getline.

3.2. Вывод данных на консоль реализован с помощью оператора cout.

3.3. Вывод даннах в файл реализован с помощью оператора вывода в файл

4. Поставленные задачи будут решены следующими действиями:

4.1. Для работы с информацией из файла построчно, все операции со строкой проводятся внутри цикла while, после каждого полного прохождения тела цикла производится считывание следующей строки. Повтроение цикла продолжается, пока переменная-счетчик rowNum не станет равна количеству строк в файле.

while rowNum <= rowCount)

{

…

rowNum++;

}

4.2. Поиск и копирование строк из файла реализован с помощью проверки с использованием условного оператора if:

if (rowNum >= n1 && rowNum <= n2)

{

getline(input, row);

if (row[0] == 'А' || row[0] == 'а')

{

output << row << endl;

}

}

else if (rowNum >= (n2 + 3) && rowNum <= rowCount)

{

getline(input, row);

output << row << endl;

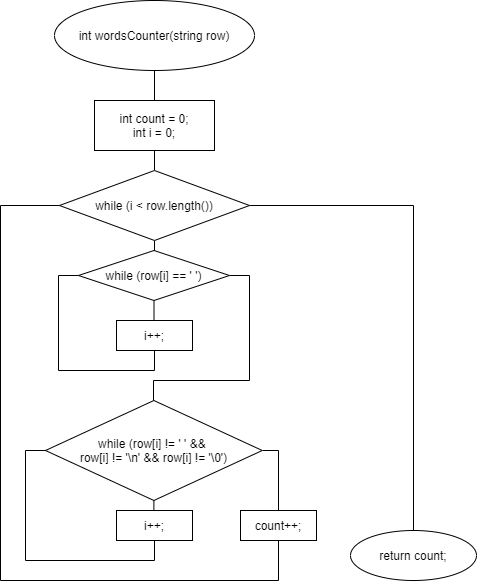
if (rowNum == rowCount)

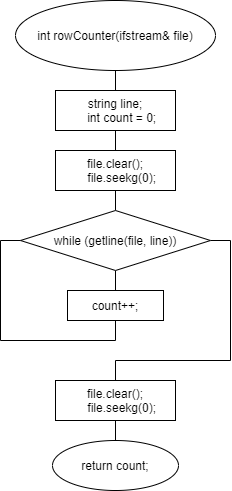
{

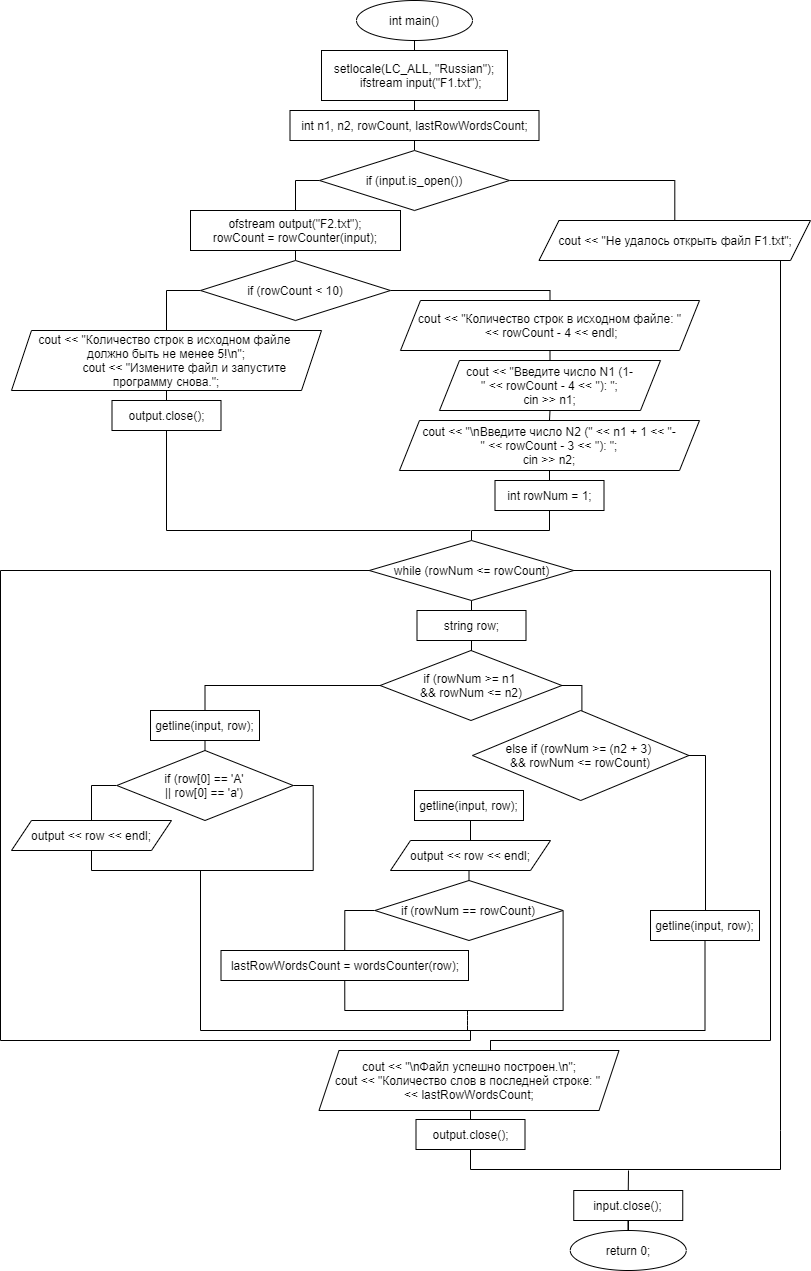
lastRowWordsCount = wordsCounter(row);

}}

**Блок-схема программы**







**Код программы**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int wordsCounter(string row)

{

int count = 0;

int i = 0;

while (i < row.length())

{

while (row[i] == ' ') i++;

while (row[i] != ' ' && row[i] != '\n' && row[i] != '\0') i++;

count++;

}

return count;

}

int rowCounter(ifstream& file)

{

string line;

int count = 0;

file.clear();

file.seekg(0);

while (getline(file, line))

{

count++;

}

file.clear();

file.seekg(0);

return count;

}

int main()

{

setlocale(LC\_ALL, "Russian");

ifstream input("F1.txt");

int n1, n2, rowCount, lastRowWordsCount;

if (input.is\_open())

{

ofstream output("F2.txt");

rowCount = rowCounter(input);

if (rowCount < 10)

{

cout << "Количество строк в исходном файле должно быть не менее 5!\n";

cout << "Измените файл и запустите программу снова.";

output.close();

}

else

{

cout << "Количество строк в исходном файле: " << rowCount - 4 << endl;

cout << "Введите число N1 (1-" << rowCount - 4 << "): ";

cin >> n1;

while (n1 <= 0 || n1 > rowCount - 4)

{

cout << "N1 введено неверно.\n";

cout << "Введите число N1 (1-" << rowCount << "): ";

cin >> n1;

}

cout << "\nВведите число N2 (" << n1 + 1 << "-" << rowCount - 3 << "): ";

cin >> n2;

while (n2 <= 0 || n2 <= n1)

{

cout << "N2 введено неверно.\n";

cout << "\nВведите число N2 (" << n1 + 1 << "-" << rowCount - 3 << "): ";

cin >> n2;

}

int rowNum = 1;

while (rowNum <= rowCount)

{

string row;

if (rowNum >= n1 && rowNum <= n2)

{

getline(input, row);

if (row[0] == 'А' || row[0] == 'а')

{

output << row << endl;

}

}

else if (rowNum >= (n2 + 3) && rowNum <= rowCount)

{

getline(input, row);

output << row << endl;

if (rowNum == rowCount)

{

lastRowWordsCount = wordsCounter(row);

}

}

else getline(input, row);

rowNum++;

}

cout << "\nФайл успешно построен.\n";

cout << "Количество слов в последней строке: " << lastRowWordsCount;

output.close();

}

}

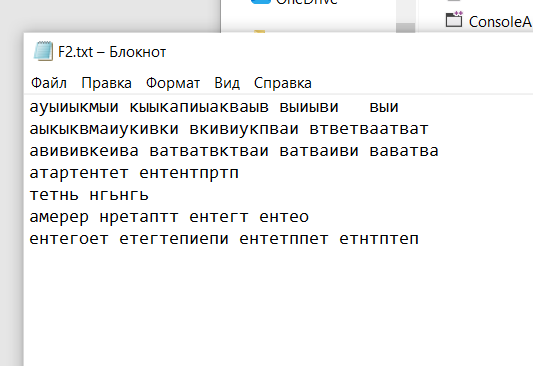
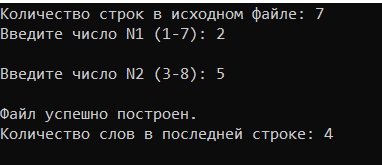
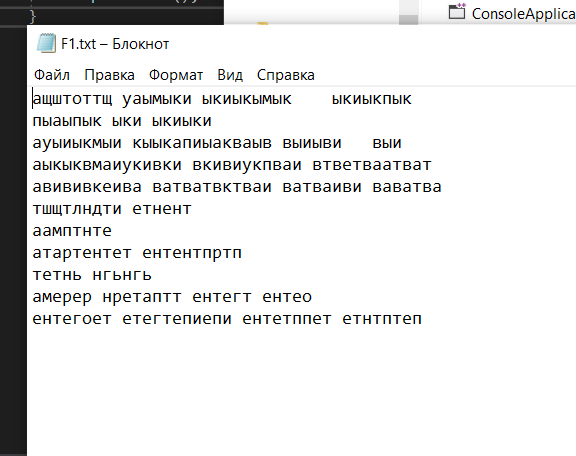
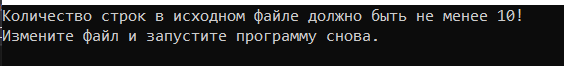
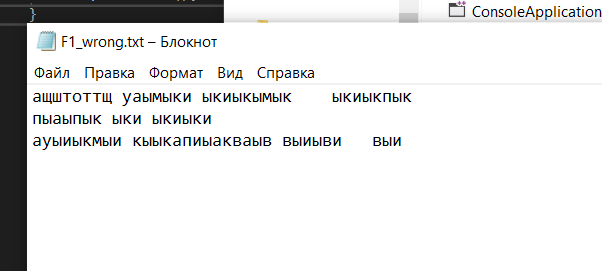
else cout << "Не удалось открыть файл F1.txt";

input.close();

return 0;

}

**Скриншоты результатов работы программы**

****