

ГУАП
КАФЕДРА №43

ОТЧЁТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

К.Т.Н., доц. _____
должность, уч. степень, звание

подпись, дата

А.В. Туманова _____
инициалы, фамилия

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №8

ОБРАБОТКА ДАННЫХ В ФАЙЛАХ

по дисциплине: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. Z7431

 31.12.2018
подпись, дата

 М.Д.Семочкин
инициалы, фамилия

Санкт-Петербург
2018

1. Цель работы

Целью работы является изучение файловых операций ввода-вывода.

2. Задание

Согласно варианту №14,

Написать программу, которая считывает текст из файла и записывает его в другой файл, заменив цифры от 0 до 9 на слова «ноль», «один», ..., «девять», начиная каждое предложение с новой строки.

3. Описание созданных функций

Для реализации задания нам потребуются следующие функции:

Имя: replaceDigitsWithWordsAndAddNewlines

Назначение: считать из первого файла текст, заменить цифры на слова, добавить переносы строк после каждого предложения, записать результат во второй файл

Входные данные: нет

Выходные данные: нет

Побочный эффект: записывает текст в файл file2.txt

Тестовые данные:

file1.txt	file2.txt
Прошел 1 день, затем прошло 2 дня. Потом прошло 3 дня! Когда пройдет 4 дня? 1, 2, 3, 4, 5 вышел зайчик погулять. 6 7 8? 9 0!	Прошел один день, затем прошло два дня. Потом прошло три дня! Когда пройдет четыре дня? один, два, три, четыре, пять вышел зайчик погулять. шесть семь восемь? девять ноль!

Прототип: void replaceDigitsWithWordsAndAddNewlines()

Алгоритм:

- псевдокод

открыть первый файл

определить длину файла, создать буфер нужного размера

записать файл в буфер

создать второй буфер

для каждого элемента первого буфера

 если он содержит символ, подлежащий замене

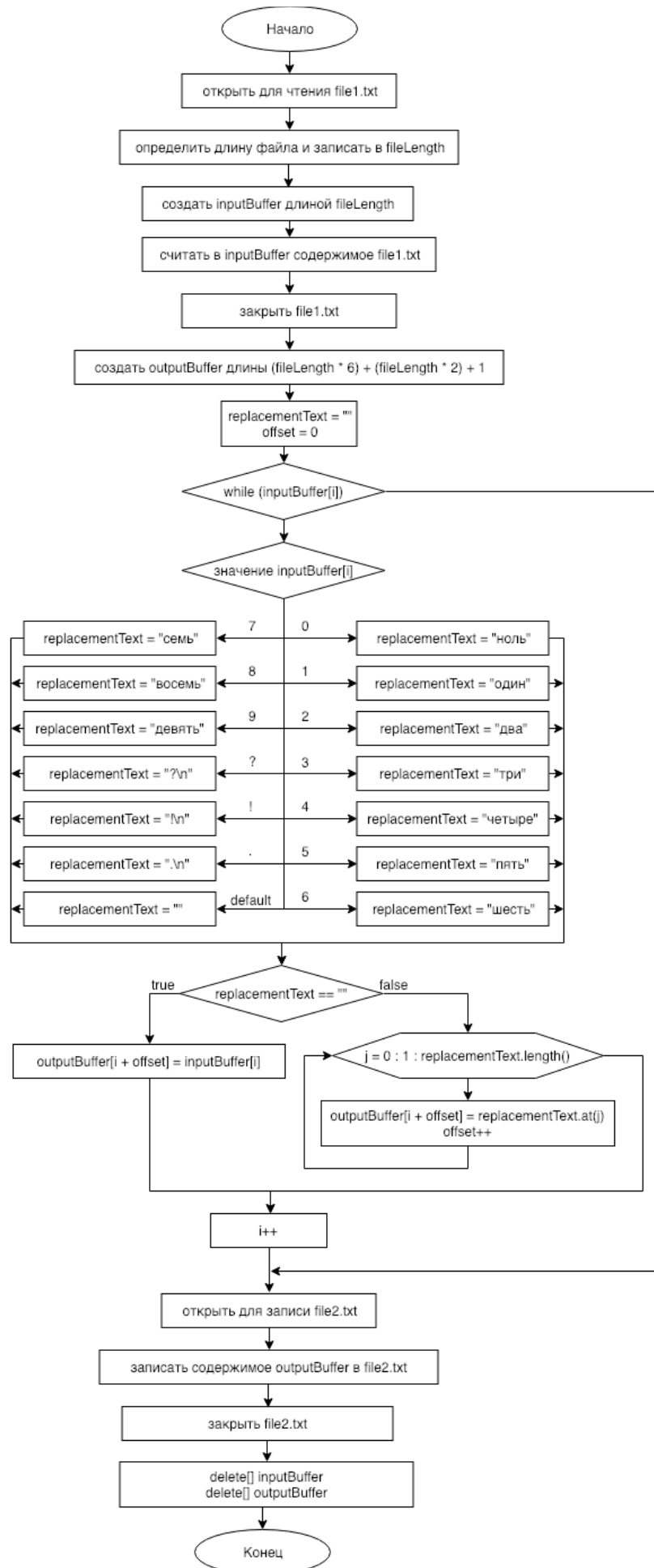
 записать во второй буфер замененное значение

 иначе

 записать во второй буфер значение из первого буфера

записать второй буфер во второй файл

- блок-схема



4. Листинг программы

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

void replaceDigitsWithWordsAndAddNewlines() {

    // Открыть первый файл
    ifstream file1("file1.txt");
    if (!file1) {
        cout << "Не удалось открыть файл file1.txt" << endl;
        return;
    }

    // Определить длину файла, создать буфер нужного размера
    file1.seekg(0, ios::end);
    const long fileLength = file1.tellg();
    char *inputBuffer = new char [fileLength + 1];

    // Записать файл в буфер
    file1.seekg(0, ios::beg);
    file1.read(inputBuffer, fileLength);
    inputBuffer[fileLength] = '\0';

    file1.close();

    // Создать второй буфер
    // В худшем случае длина текста после обработки будет
    const long outputBufferLength = (fileLength * 6) + (fileLength * 2) + 1;
    // Если все цифры будут заменены на слово из 6 букв ("девять")
    // и каждый символ будет отдельным предложением
    char *outputBuffer = new char [outputBufferLength];
    outputBuffer[outputBufferLength] = '\0';

    long i = 0;
    string replacementText;
    // Поскольку один символ заменяется несколькими, позиция чтения
    // из первого буфера и позиция записи во второй буфер различаются
    long offset; // Смещение позиции записи во второй буфер

    while (inputBuffer[i]) {

        switch(inputBuffer[i]) {
            case '0': replacementText = "ноль"; break;
            case '1': replacementText = "один"; break;
            case '2': replacementText = "два"; break;
            case '3': replacementText = "три"; break;
            case '4': replacementText = "четыре"; break;
            case '5': replacementText = "пять"; break;
            case '6': replacementText = "шесть"; break;
            case '7': replacementText = "семь"; break;
            case '8': replacementText = "восемь"; break;
            case '9': replacementText = "девять"; break;
            case '?': replacementText = "?\n"; break;
            case '!': replacementText = "!\n"; break;
            case '.': replacementText = ".\n"; break;
            default: replacementText = "";
        }
    }
```

```

        if (replacementText == "") {
            outputBuffer[i + offset] = inputBuffer[i];
        } else {
            for (int j = 0; j < replacementText.length(); j++) {
                outputBuffer[i + offset] = replacementText.at(j);
                offset++;
            }
        }

        i++;
    }

    // Открыть второй файл
    ofstream file2("file2.txt");
    if (!file2) {
        cout << "Не удалось открыть файл file2.txt" << endl;
        return;
    }

    // Записать второй буфер в файл
    file2.write(outputBuffer, outputBufferLength);
    file2.close();

    // Удалить буферы
    delete[] inputBuffer;
    delete[] outputBuffer;
}

int main() {
    setlocale(LC_ALL, "russian");
    replaceDigitsWithWordsAndAddNewlines();
    return 0;
}

```

5. Пример выполнения программы

Ниже показан пример выполнения программы.

```

MacBook-Pro-Mikhail:lab-8 ms$ make lab-8
c++ lab-8.cpp -o lab-8
MacBook-Pro-Mikhail:lab-8 ms$ cat file1.txt
Прошел 1 день, затем прошло 2 дня. Потом прошло 3 дня! Когда пройдет 4 дня? 1, 2, 3, 4, 5 вышел зайчик погулять. 6 7 8? 9 0!
MacBook-Pro-Mikhail:lab-8 ms$ ./lab-8
MacBook-Pro-Mikhail:lab-8 ms$ cat file2.txt
Прошел один день, затем прошло два дня.
Потом прошло три дня!
Когда пройдет четыре дня?
один, два, три, четыре, пять вышел зайчик погулять.
шесть семь восемь?
девять ноль!
MacBook-Pro-Mikhail:lab-8 ms$

```

Видно, что результаты расчётов совпадают с тестовыми данными.

6. Используемая литература

- Татьяна Александровна Павловская - C/C++. Структурное и объектно ориентированное программирование: практикум (Питер, 2011)

7. Анализ результатов и выводы

К достоинствам программы можно отнести:

- Программа выполняет поставленную задачу и работает без ошибок (для корректных тестовых данных).
- Задание реализовано в виде отдельной функции.