Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное учреждение высшего образования «Омский государственный технический университет»

Кафедра «Автоматизированные системы обработки информации и управления»

**ОТЧЁТ**

**ПО РАСЧЁТНО-ГРАФИЧЕСКОЙ РАБОТЕ**

по дисциплине «Проектная деятельность»

студента Федорова Вячеслава Николаевича, группа ПИН-221

**Пояснительная записка**

Шифр работы От-2068998-43-ПИН-221-45-22 ПЗ

Направление 09.03.04

Старший преподаватель Р.Н. Богатов

Студент В.Н. Федоров

Омск 2023

РЕФЕРАТ

Отчет 13 с.,1ч., 5 рис., 1 табл., 3 источ., 1 прил.

С++20, VISUAL STUDIO 2022, БИБЛИОТЕКИ IO.H; IOSTREAM.H; FCNTL.H; WINDOWS.H; SOURCE1.TXT; SOURCE2.TXT

Объектом разработки является алгоритм записи слов из двух файлов в один поочередно.

Цель работы - разработка программы для записи слов из двух файлов в один поочередно.

В процессе работа проводилась разработка записи слов из двух файлов в один поочередно и тестирование данного алгоритма.

В результате работы создана программа, выполняющая условия задачи.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc125043719)

[1 Постановка задачи 5](#_Toc125043720)

[2 Схема алгоритма 6](#_Toc125043721)

[3 Особенности реализации программы 7](#_Toc125043722)

[4 Методика тестирования: 8](#_Toc125043723)

[ЗАКЛЮЧЕНИЕ 9](#_Toc125043724)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10](#_Toc125043725)

ПРИЛОЖЕНИЕ А Исходный код …………………….………..…………………11

# ВВЕДЕНИЕ

Расчетно-графическая работа включает в себя решение задачи: даны текстовые файлы F и G. Переписать в файл Q слова из файлов F и G поочередно, одно слово из F, другое из G. После окончания слов в одном из входных файлов, запись прекращается.

Выполнение работы осуществляется на языке С++ в среде VisualStudio 2022, которая позволяет разрабатывать консольные приложения.

Исходный код решения в приложении А.

## 1 Постановка задачи

1.1 Задание

Даны текстовые файлы F и G. Переписать в файл Q слова из файлов F и G поочередно, одно слово из F, другое из G. После окончания слов в одном из входных файлов, запись прекращается.

1.2 Условия задачи

Написать программу, выполняющую следующие действия:

- открыть заданные файлы проверить файлы на возможность их открытияс использованием методов языка C++

- инициализировать символьные массивы

- записать слова в общий файл в соответствии с заданием

- вывести готовый текст в консоль VS.

1.3 Конечные цели

- разработка алгоритма для поочередной записи слов из двух файлов в один в соответствии с заданием;

- реализовать вывод готового текста в консоль VS.

1.4 Формы выдачи результатов

- Результаты работы программы записывают в общий заранее заданный файл.

- Результаты работы программы выводятся в консоль VS.

1.5 Описание используемых данных

- Исходные данные находятся в заранее заданных файлах F и G и с помощью программы поочередно записываются в файл Q.

## 2 Схема алгоритма

На рисунке 1 представлена схема алгоритма.

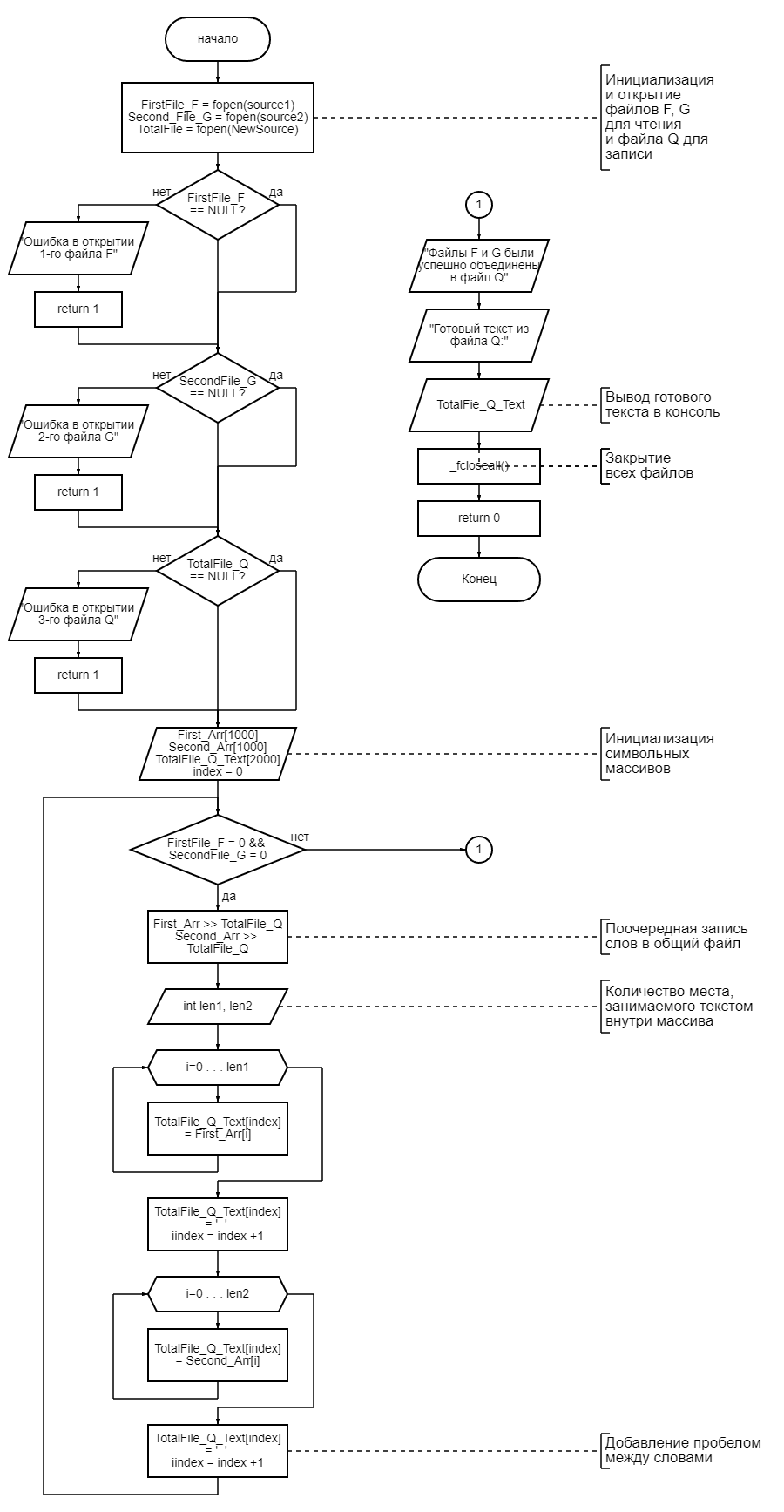


Рисунок 1 – Схема алгоритма

## 3 Особенности реализации программы

Этап реализации всегда включает в себя процессы проектирования и программирования. Первой особенностью разработки проекта является депрекация и борьба с ней. Депрекация – это неодобрение использования стандартных методов ввода/вывода и работы с файлами.

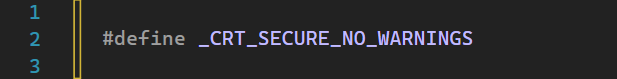


Рисунок 2 – Метод избавления от депрекации

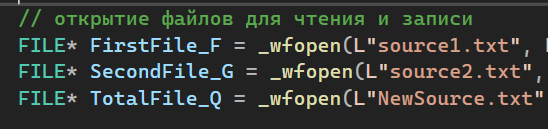


Рисунок 3 – Метод открытия и создания файла

Второй особенностью является использование «широких» строк для правильного вывода текста в консоль VS. Использование широких строк необходимо

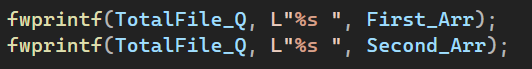


Рисунок 4 – Использование методов записи в файл с «широкими» строками

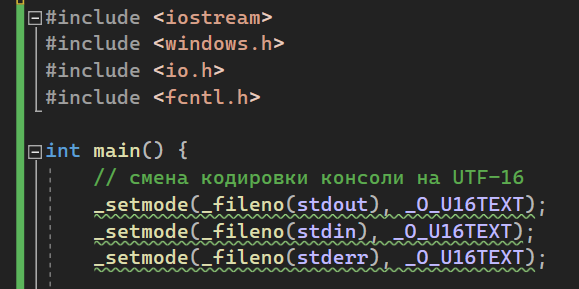


Рисунок 5 – Библиотеки и методы для смены кодировки консоли на UTF-16

А так же в ходе тестирования была обнаружена необходимость использования методов, совместимых с «широкими» строками.

## 4 Методика тестирования

В таблице 1 приведены результаты тестирования:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тест | Результат | Итог |
| 1 | Два текста на русском языке длиной не более 100 слов | Общий текст на русском языке с поочередно записанными словами длиной не более 200 слов | + |
| 2 | Текст на русском языке и текст на английском языке. Оба длиной не более 100 слов. | Общий текст с поочередно записанными русскими и английскими словами длиной не более 200 слов | + |
| 3 | Два русских текста со знаками препинания и числами длиной не более 100 слов и чисел | Общий текст на русском языке с поочередно записанными словами(или числами) длиной не более 200 слов и чисел. Знаки препинания являются частью слова или числа, с которым они не разделены пробелами | + |
| 4 | Русский и английский тексты со знаками препинания и числами длиной не более 100 слов и чисел | Общий текст на русском и английском языках с поочередно записанными словами(или числами) длиной не более 200 слов и чисел. Знаки препинания являются частью слова или числа, с которым они не разделены пробелами | + |
| 5 | Два русских текста с переносами строк длиной не более 100 слов или чисел | Общий русский текст. Переносы строк не сказываются на работе программы. | + |
| 6 | Русский и английский тексты переносами строк длиной не более 100 слов или чисел | Общий текст на русском и английском языках . Переносы строк не сказываются на работе программы. | + |
| 7 | Два русских текста без пробелов | Общий текст на русском языке, обьединенный в один поочередно | + |
| 8 | Английский и русский тексты без пробелов | Общий текст на русском и английском языках, обьединенный в один поочередно | + |
| 9 | Два пустых текста | Программа ничего не вывела в консоль | + |
| 10 | Один пустой текст | Программа ничего не вывела в консоль | + |

Таблица 1 – методика тестирования

# ЗАКЛЮЧЕНИЕ

В результате выполнения данной расчетно-графической работы был разработан алгоритм, удовлетворяющий условиям задачи. Работа по поочередной записи слов из двух файлов F и П в один файл Q и выводу результата в консоль VS. В ходе тестирования программа прошла все необходимые тесты без ошибок.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Литвиненко Н. А. Технология программирования на С++. Win32 API-приложения. — СПб.: БХВ-Петербург, 2010. — 288 с.: ил. — (Учебное пособие) ISBN 978-5-9775-0600-7
2. Занятия по С++.- URL: https://vk.com/asoiu\_cpp
3. Справочный материал о библиотеке CTIME.

–URL: http://cppstudio.com/cat/309/326/

ПРИЛОЖЕНИЕ А

(обязательное)

Исходный код

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include <io.h>

#include <fcntl.h>

int main() {

// смена кодировки консоли на UTF-16

\_setmode(\_fileno(stdout), \_O\_U16TEXT);

\_setmode(\_fileno(stdin), \_O\_U16TEXT);

\_setmode(\_fileno(stderr), \_O\_U16TEXT);

// открытие файлов для чтения и записи

FILE\* FirstFile\_F = \_wfopen(L"source1.txt", L"rt, ccs=UTF-8"); //открытие 1-го файла F с 1-м исходным текстом

FILE\* SecondFile\_G = \_wfopen(L"source2.txt", L"rt, ccs=UTF-8"); //открытие 2-го файла G со 2-м исходным текстом

FILE\* TotalFile\_Q = \_wfopen(L"NewSource.txt", L"wt, ccs=UTF-8"); //открытие файла Q для записи нового текста

// проверка файлов на открытие

if (FirstFile\_F == NULL)

{

std::wcout << L"Ошибка в открытии 1-го файла F" << std::endl;

return 1;

}

if (SecondFile\_G == NULL)

{

std::wcout << L"Ошибка в открытии 2-го файла G" << std::endl;

return 1;

}

if (TotalFile\_Q == NULL)

{

std::wcout << L"Ошибка в открытии общего файла" << std::endl;

return 1;

}

// инициализация символьных массивов для записи слов

wchar\_t First\_Arr[1000], Second\_Arr[1000], TotalFile\_Q\_Text[2000];

int index = 0;

// очистка памяти, занимаемой массивом TotalFileQ\_Text

ZeroMemory(TotalFile\_Q\_Text, sizeof(TotalFile\_Q\_Text));

// цикл, внутри которого слова из двух файлов записываются в общий файл, пока в одном из исходных файлов не закончатся слова

while (fwscanf(FirstFile\_F, L"%s", First\_Arr) != EOF && fwscanf(SecondFile\_G, L"%s", Second\_Arr) != EOF)

{

fwprintf(TotalFile\_Q, L"%s ", First\_Arr);

fwprintf(TotalFile\_Q, L"%s ", Second\_Arr);

// запись слов из двух массивов в TotalFile\_Q\_Text поочередно

int len1 = lstrlenW(First\_Arr);

int len2 = lstrlenW(Second\_Arr);

// циклы посимвольной записи слов в общий массив

for (int i = 0; i < len1; i++)

{

TotalFile\_Q\_Text[index] = First\_Arr[i];

index++;

}

TotalFile\_Q\_Text[index] = ' ';

index++;

for (int i = 0; i < len2; i++)

{

TotalFile\_Q\_Text[index] = Second\_Arr[i];

index++;

}

TotalFile\_Q\_Text[index] = ' '; // добавляем пробел между словами

index++;

}

std::wcout << L"\nФайлы F и G были успешно объединены в файл Q!\n" << std::endl;

std::wcout << L"Готовый текст из файла Q:" << std::endl;

std::wcout << L"\n" << TotalFile\_Q\_Text << std::endl;

//закрытие всех файлов

\_fcloseall();

return 0;

}