

АННОТАЦИЯ

Пояснительная записка: с 10 рис.,2 библиограф. источника, 3 приложения

Ключевые слова: работа с документами, обработка текста программы, С++.

В курсовой работе разработан алгоритм и соответствующая ему программа, которая осуществляет обработку текста файла в формате txt, в котором описана программа на языке С++. Создается двумерный массив символьного типа, в который копируется текст открытого документа. В телах трех функций выполняются основные задачи: подсчет количества строк и символов, подсчет максимальной вложенности циклов, удаление комментариев программиста.

Программа написана на языке Visual C++ и функционирует в операционной системе Windows’08.

Содержание

[Постановка задачи 5](#_Toc43647437)

[Разработка контрольного примера и построение словесного алгоритма 5](#_Toc43647438)

[Назначение и область применения проекта 8](#_Toc43647439)

[Словесный алгоритм решения задачи 8](#_Toc43647440)

[Выбор и обоснование метода решения 9](#_Toc43647441)

[Алгоритмизация 9](#_Toc43647442)

[Блок-схема главной программы: 11](#_Toc43647443)

[Блок-схема функции подсчета строк и символов: 12](#_Toc43647444)

[Блок-схема функции подсчета максимальной вложенности циклов: 14](#_Toc43647445)

[Блок-схема функции удаления комментариев из текста: 15](#_Toc43647446)

[Программная реализация разработанных алгоритмов 17](#_Toc43647447)

[Тестирование 17](#_Toc43647448)

[Заключение 17](#_Toc43647449)

[Библиографический список 18](#_Toc43647450)

# Постановка задачи

В рамках данной курсовой работы требуется разработать проект, который позволяет открыть в программе документ с текстом программы на С++, обработать текст данного документа, а именно:

1) найти количество символов, отличных от пробела

2) найти общее количество строк,

3) удалить все комментарии из текста

4) разработать функцию подсчета максимальной вложенности циклов

Текст программы, который является объектом обработки разрабатываемого приложения, должен удовлетворять следующим условиям, определяемым ЕСПД.

1. Каждый оператор начинается с новой строки.
2. В тексте допустимы только однострочные комментарии.

# Разработка контрольного примера и построение словесного алгоритма

Рассмотрим алгоритм работы программы на примере фрагмента текста исходного документа (рис. 1). Основной задачей является обработка строк и символов документа. Алгоритму нужно найти строки, начинающиеся со служебных слов for, while, do, для этого будет реализован поиск совпадений символьных последовательностей. Чтобы реализовать подсчет вложенных циклов, необходимо, чтобы текст пользователя соответствовал требованиям:

1. Циклы записаны без табуляций и пробелов, с самого начала строки
2. Каждое тело цикла заключено в фигурные скобки

Таким образом, по фигурной скобке “}” можно определить конец цикла. Пока программой не будет найдена данная скобка, будет осуществляться поиск служебных последовательностей for, while, do, по которым можно обнаружить наличие вложенного цикла. В примере исходного текста документа обозначен один строчный комментарий, один вложенный цикл. Текст документа, представляющий собой объект обработки, приведён на рис. 1), он соответствует ранее заявленным требованиям.

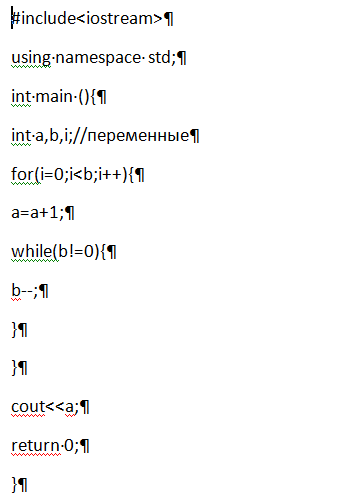


Рис.1 Пример текста программы, который поступает на обработку: один вложенный цикл, один строчный комментарий

Анализ обрабатываемого текста показывает, что после запуска программы должны быть выведены на консоль значения, которые подсчитаны вручную:

Количество символов: 135

Количество строк: 13

Максимальное количество вложенных циклов:1

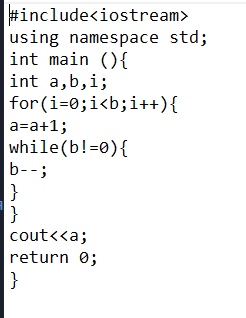


Рис. 2 Пример выходного файла без строчного комментария

На рис. 2 представлен пример измененного исходного файла, который перезаписан в выходной файл без строчных комментариев программиста.

Для хранения текста документа создается глобальная переменная – двумерный символьный массив 100\*80, а для хранения числа строк в документе создается глобальная числовая переменная.

На основе анализа контрольного примера становится очевидным, что для выполнения задачи потребуется реализовать следующие функции:

**Функция №1**- подсчет общего количества символов, все символы текста без пробелов и выводится информации об находящихся в глобальной переменной – двумерном символьном массиве, возврат значения кол-ва символов, подсчет количества строк и возврат значения их количества. Алгоритм работы этой функции заключается в том, что складываются их количестве. Складываются все строки файла и выводится их количество.

**Функция №2 -**подсчет максимальной вложенности циклов, путем обработки глобальной переменной – двумерного символьного массива построчно, возврат количества максимальной вложенности циклов. Алгоритм работы этой функции заключается в поиске циклов и дальнейшем их анализе на вложенность, после чего это значение выводится на экран.

**Функция №3-** удаление из исходного документа комментариев программиста. Алгоритм работы этой функции заключается в обработке символов двумерного символьного массива и перезаписью текста файла без комментариев программиста в новый файл.

# Назначение и область применения проекта

Данный проект может быть применен для редактирования кодов программиста, если ему необходимо представить код без комментариев, или если необходимо рассчитать примерное время обработки программы компилятором, зная количество строк и количество символов, а также максимальную вложенность циклов.

Данный проект может также помочь соблюсти заданные программисту ограничения по количеству строк или символов.

# Словесный алгоритм решения задачи

На основе анализа контрольного примера можно сделать вывод, что для реализации задач потребуются следующие алгоритмы:

• работа с двумерным массивом (в данном массиве копия текста документа);

• подсчет количества символов (алгоритм введения счетчика);

• поиск максимальной вложенности циклов (поиск по совпадению последовательностей символов);

• удаление символов (все символы, идущие после ‘\\’);

• в кодах разрабатываемых функций предусматривается введение временных переменных.

Математическое решение данных задач выглядит следующим образом:

*1) Подсчет строк в документе*

Открываем файл и пока не достигли конца файла

* считываем строку в глобальный двумерный символьный массив (для копирования символов документа)
* увеличиваем накопительную переменную к – количество строк

*2) Подсчет символов отличных от нуля*

* в цикле for идем построчно по двумерному символьному массиву
* начинаем с нулевого символа строки
* пока не конец строки
* если символ не равен пробелу, увеличиваем накопительную переменную s.

*3) Поиск максимальной вложенности циклов*

* создаем три константных символьных массива, содержащих служебные слова начала цикла, для работы со встроенной функцией по поиску последовательностей символов
* в цикле for идем с нулевой строки двумерного массива с текстом документа
* если в строке встречается последовательность символов for/ do/ while то
* пока не встретится строка с символом закрытия цикла «}»
* снова ищем последовательность символов for/ do/ while
* если находится такая последовательность увеличиваем накопительную переменную
* если нет, переходим к следующей строке

*4) Удаление комментариев программиста*

* в цикле for начинаем проход построчно по символьному массиву с текстом документа с нулевой строки
* осуществляем проход по символам строки
* если встречается символ « /» начала комментария
* и следующий после него символ так же равен « /»
* присваиваем данному символу символ конца строки
* проходим весь массив
* создаем и открываем новый файл
* заполняем его отредактированным массивом

# Выбор и обоснование метода решения

Для реализации данного алгоритма выбран метод накопления суммы, который является наиболее удобным для подсчета количества символов, отличных от пробела, а также количества строк в документе. Для реализации поиска вложенности циклов, используется метод применения строковых функций для поиска символьных последовательностей служебных слов «for, while, do». Для обработки двумерного символьного массива используется метод построчной обработки и посимвольной его обработки, путем циклического прохода с нулевой строки по последнюю, и с нулевого символа по последний.

# Алгоритмизация

На основе анализа контрольных примеров алгоритм задачи будет состоять из следующих шагов:

1.Открываем файл и считываем текст;

2. Выполняем подсчёт символов и строк;

3. Выводим на экран их количество;

4. Выполняем подсчет максимального количества вложенных циклов;

5. Выводим на экран их количество;

6. Удаляем комментарии;

7. Записываем отредактированный текст в новый файл;

Из анализа контрольного примера и словесного алгоритма можно выделить повторяющиеся фрагменты программы и логически целостные вычислительные блоки. Эти блоки реализуются в виде функций, предварительные прототипы которых были представлены в разделе Разработка контрольного примера.

Функция **poisk** осуществляет считывание с файла текста программы, копирование этого текста в символьный двумерный массив, подсчет количества символов и количества строк, вывод их на экран. Функция осуществляется без параметров, т.к двумерный массив и количество строк – глобальные переменные.

Функция **max\_vlogh** осуществляет поиск максимальной вложенности циклов, выводит их количество на экран. Функция осуществляется без параметров, т.к двумерный массив и количество строк – глобальные переменные.

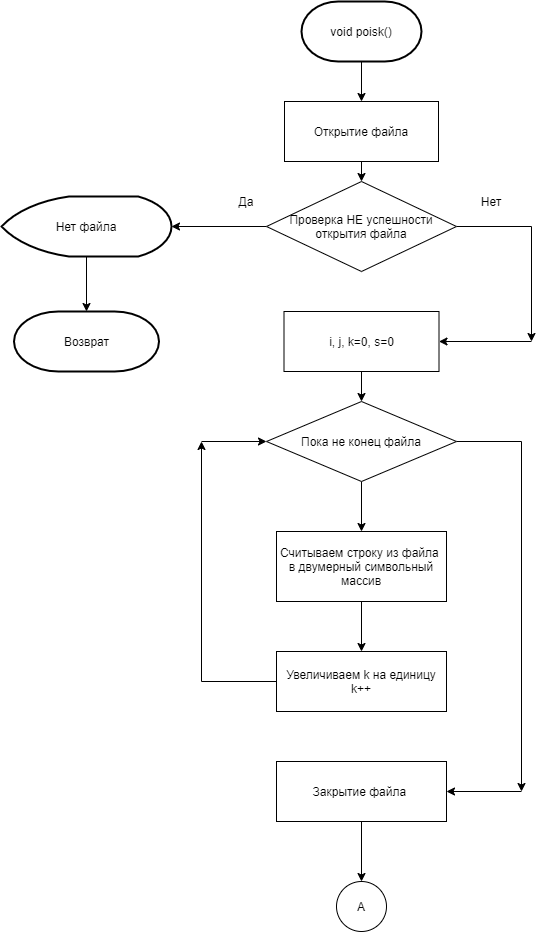
Функция **del\_kom** позволяет удалить комментарии из копии текста ( двумерного символьного массива) и записать отредактированный текст в новый документ. Функция осуществляется без параметров, т.к двумерный массив и количество строк – глобальные переменные.

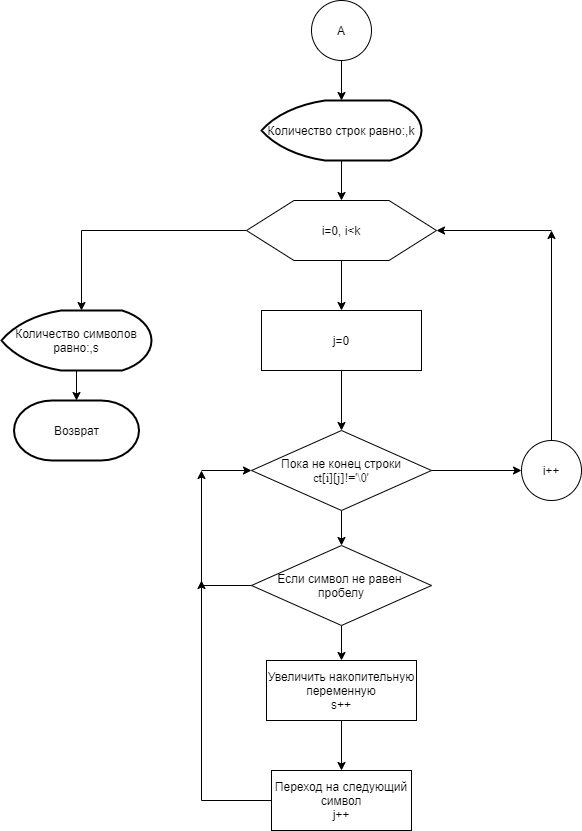
Стратегия решения данного алгоритма представлены на следующих блок-схемах:

# Блок-схема главной программы:



# Блок-схема функции подсчета строк и символов:

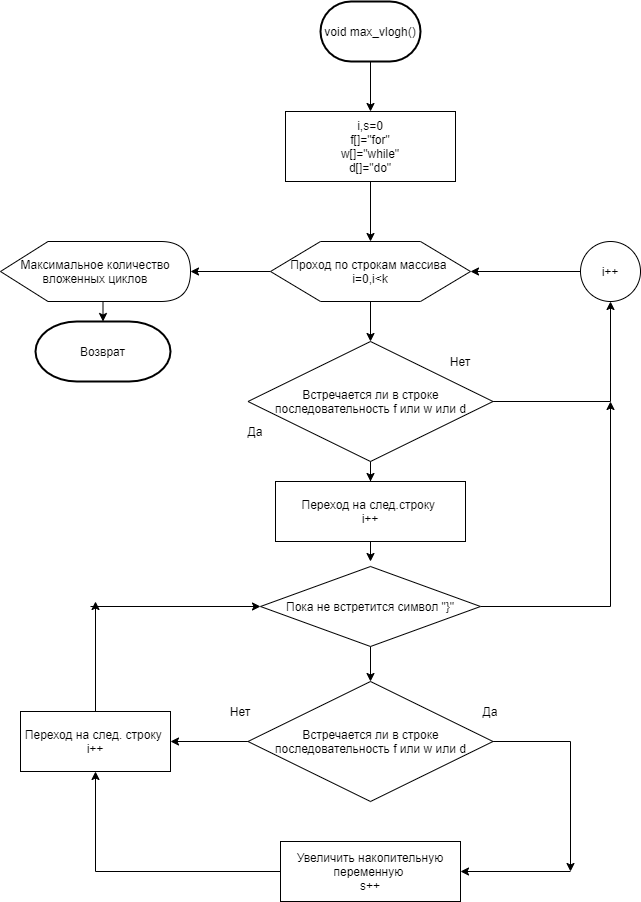




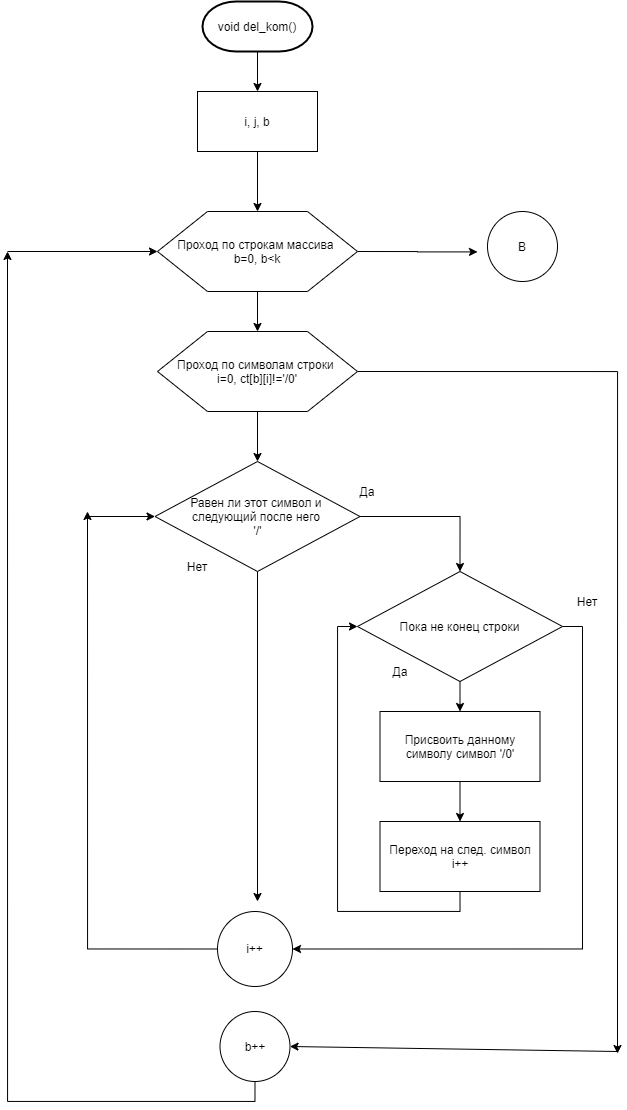
Нет

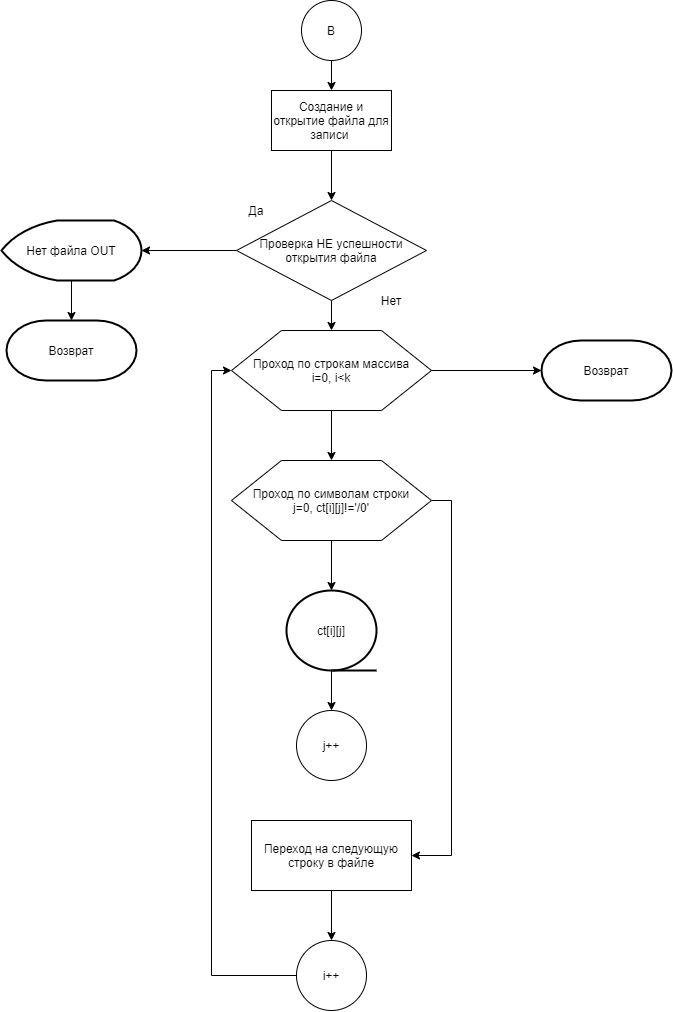
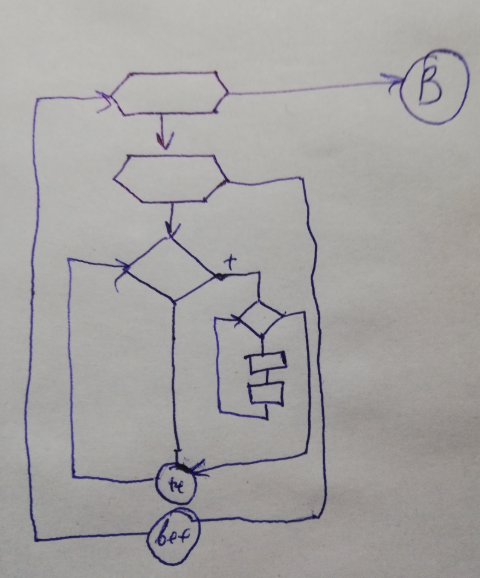
Да

# Блок-схема функции подсчета максимальной вложенности циклов:



# Блок-схема функции удаления комментариев из текста:





# Программная реализация разработанных алгоритмов

Реализация данного алгоритма выполнена в виде программ, написанных на языке C++ среди Visual Studio 2010. Программы написаны хорошим стилем программирования, читабельны: блоки программы оформлены отступами, операторы снабжаются комментариями. Листинг программы приведен в приложении 1.

# Тестирование

Тестирование программы произведено путем сравнения результатов её работы с контрольным примером на рис.1. По скриншоту консоли, представленном в приложении 3, можно удостовериться в правильности подсчета программой строк, символов и вложенных циклов, а скриншот файла Output показывает правильность работы функции удаления комментариев программиста.

В Тестировании №2 в исходном файле (рис. 5) нет комментариев программиста и нет вложенных циклов. Скриншот консоли (рис. 6) и выходного файла (рис. 7) также показывает верные результаты выполнения программы.

В Тестировании №3 (рис 8.) проверяется действенность программы с двумя комментариями программиста и двумя вложенными циклами. Сравнивая данные, выведенные на консоль (рис. 9), и подсчитанные вручную данные, можно сделать вывод о правильности работы программы.

Все тестирования прошли проверку и могут считаться достоверными, программа работает верно.

# Заключение

В данной работе была создана программа, позволяющая подсчитать количество символов и строк в заданном документе, максимальное количество вложенных циклов и осуществить удаление комментариев программиста.

Перед началом процесса программирования, были обозначены такие важные моменты, как назначение и область применения алгоритма, были осуществлены такие этапы как постановка самой задачи, математическая формулировка её условия, выбор и обоснование метода решения.

На этапе алгоритмизации были словесно описаны шаги алгоритма, которые в дальнейшем мы представили в виде блок-схем.

Далее были описаны все функции по таким критериям, как формальные параметры, возвращаемое значение и логика работы.

После этого мы осуществили процесс программирования, т.е. запись алгоритма решаемой задачи на конкретном языке программирования. В конечном итоге, была произведена отладка написанной программы на основе контрольных примеров.

# Библиографический список

Павловская Т. А. C/C++. Программирование на языке высокого уровня: для магистров и бакалавров: учебник для вузов / Т. А. Павловская. – Санкт Петербург [и др.]: Питер, 2013 - 460 с.

Петрова Н.К. Конспект лекций «Программирование на С++». Электронный документ <https://lms.kgeu.ru/course/view.php?id=2594>.

**ПРИЛОЖЕНИЕ 1. Листинг программы**

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

//объявление глобальных переменных

int k; //хранение кол-во строк

char ct[50][80]; //хранение копии текста документа

//прототипы функций

void poisk(); //функция подсчета символов и строк

void max\_vlogh (); //функция подсчета вложенности циклов

void del\_kom(); //функция удаления комментариев программиста

//главная программа

int main( )

{

setlocale(LC\_ALL, "Russian");

poisk();

max\_vlogh ();

del\_kom();

return 0;

}

void poisk(){ //функция подсчета строк и символов

ifstream inp( "Input.txt " ); // открытие файла для чтения

if(!inp){ //проверка НЕ успешности открытия файла

std:: cout << "Нет файла" ;return;

}

int i,j;

k=0;

long int s=0; //хранение кол-во символов

while( !inp.eof( ) ) // цикл до окончания файла

{ inp.getline(ct[k], 79); //получение строки из файла

k++; } // увеличение номера строки на единицу

inp.close( ); // закрытие файла

std::cout<< "Количество строк равно: "<<k-1<<endl;

for(int i = 0; i < k; i++) {

j=0;

while(ct[i][j]!='\0'){ //пока не обнаружится символ строки

if (ct[i][j]!=' '){ //если символ не равен пробелу

s++; //увеличение накопительной переменной

}

j++; //переход на следующий символ

}

}

std::cout<< "Количество символов равно: "<<s<<endl;

}

void max\_vlogh (){ //функция подсчета вложенности циклов

int i,s=0;

const char f[] = "for";  
const char w[] = "while";  
const char d[] = "do";

for(i=0;i<k;i++){ // с первой по последнюю строку

if( (strstr(ct[i],f))|| (strstr(ct [i],w)) ||(strstr(ct [i],d)) ){ //если встречается

i++; //переходим на следующую строку

while (ct [i][0]!='}') { //пока не найден конец цикла

if( (strstr(ct [i],f))|| (strstr(ct [i],w)) ||(strstr(ct [i],d)) ){

s++; //увеличиваем кол-во вложенных циклов

}

i++; //переход на следующую строку

}

}

}

std::cout<< "Максимальное количество вложенных циклов: "<<s<<endl;

}

void del\_kom() { //функция удаления комментариев программиста

int i, j,b;

for(b=0;b<k;b++){ //с первой по последнюю строку

for (i = 0; ct[b][i] !='\0'; i++){ //с первого по последний символ строки

if ((ct[b][i] == '/' )&& (ct[b][i+1] == '/')){ //если найден символ начала комментария и следующий за ним символ

while (ct[b][i]!='\0 '){

ct[b][i] = '\0'; //присваиваем символ конца строки

i++;

}

}

}

ofstream out("output.txt"); // создание и открытие файла для записи

if(!out){ // проверка успешности открытия файла

std:: cout << "Нет файла OUT" ; return;

}

for ( i = 0; i < k; i++ ) { //с первой по последнюю строку

for (j=0;ct[i][j]!='\0';j++){ //с первого по последний символ

out << ct[i][j]; //выводим символ в новый текстовый документ

}

out<<'\n' ; //переход на след. строку

}

out.close( ); // закрытие файла

}

}

**ПРИЛОЖЕНИЕ 2. Руководство пользователю**

Пользователь создает проект в Visual Studio, в папку с этим проектом загружает файл с текстом программы Input.txt . Если файл не найден, программа выдает сообщение на консоль. Программа анализирует текст файла, на консоль выводит количество строк, символов в документе, а также количество вложенных циклов. В папке с проектом создается файл с именем Output.txt, куда записывается текст файла без комментариев программиста.

**ПРИЛОЖЕНИЕ 3. Тестирование**

**Тестирование 1.**

Консоль:

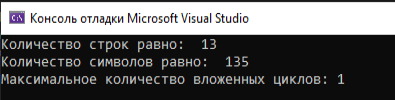


Рис. 3

Файл Output:

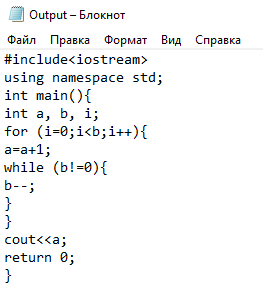
****

Рис. 4

**Тестирование 2.**

Входной файл (без комментария программиста, без вложенного цикла):

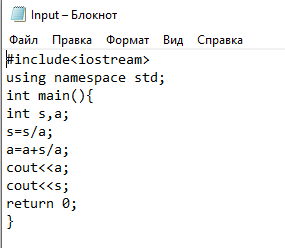
****

Рис. 5

Что должно быть выведено на консоль:

Количество строк равно: 10

Количество символов равно: 92

Максимальное количество вложенных циклов: 0

Консоль:

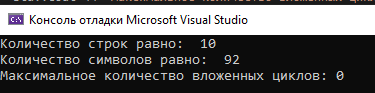


Рис. 6

Выходной файл Output:

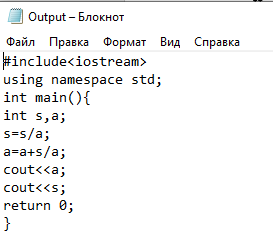


Рис. 7

**Тестирование 3.**

Входной файл (два строчных комментария, два вложенных цикла):

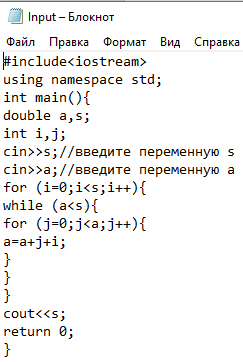


Рис. 8

Что должно быть выведено на консоль:

Количество строк равно: 18

Количество символов равно: 224

Максимальное количество вложенных циклов равно: 2

Консоль:

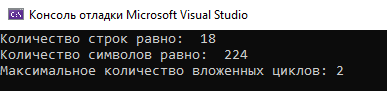


Рис. 9

Выходной файл Output:

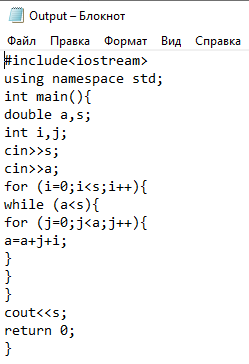


Рис. 10

