СОДЕРЖАНИЕ

1. ЗАДАНИЕ 3

2. ОПИСАНИЕ ПРИМЕНЕНИЯ 3

2.1. Запуск программы 3

2.2. Входные данные 3

2.3. Выходные данные 3

2.4. Сообщения программы 4

3. ОПИСАНИЕ ПРОГРАММЫ 4

3.1. Метод решения задачи 4

3.2. Структура программы 5

3.3. Описание функций 5

3.3.1. main – главная функция 5

3.3.2. spisok\_uchenikov - вывод фамилии и имени

учеников введенного класса. 7

3.3.3. kolvo\_marki - подсчет и вывод количества учеников в введенном классе и параллельных классах. 9

ЛИТЕРАТУРА 13

Приложение 1. Текст программы 14

Приложение 2. Пример входного файла shkola.txt 18

Приложение 3. Результаты тестирования программы 19

**1. ЗАДАНИЕ**

Дан файл, содержащий сведения об учениках начальной школы. Структура записи файла: фамилия и имя ученика, название класса. Записи в файле не упорядочены.

### Написать программу, которая

* печатает список учеников заданного класса;
* выдает информацию о количестве учеников в данном классе и в каждом из параллельных классов.

**2. ОПИСАНИЕ ПРИМЕНЕНИЯ**

**2.1 Запуск программы**

Запуск программы производится в среде программирования CodeBlocks

**2.2 Входные данные**

Входные данные находятся в файле “shkola.txt”. Число строк в файле - произвольное. Каждая строка содержит фамилию с инициалами (20 символов), № класса (2 символа), букву класса (2 символа). Пример строки файла:

Штольц Никита, 3 Б.

| 20 |2 |2|

Пример входного файла приведен в приложении 2.

По запросу программы с клавиатуры необходимо ввести номер пункта меню (см. раздел 2.3).

**2.3. Выходные данные**

Программа выводит на экран меню:

===========================================================

**Выберите номер пункта меню:**

**1 - список учеников заданного класса**

**2 - кол-во учеников в данном классе и параллельных классах**

**3 - выход**

При выборе пункта 1 на экран выводится запрос в виде:

**Введите № класса:**

Введем, к примеру, “3”. После ввода на экран выводится запрос в виде:

**Введите букву класса:**

Введем, кпримеру, “А”. После ввода на экран выводится результат в виде:

**№ Фамилия Имя**

**---------------------------**

**1 Мартыненко Сергей**

При отсутствии файла выводится сообщение:

**Файл не может быть открыт!**

Примеры выходных данных для остальных пунктов меню см. в приложении 3.

**2.4. Сообщения программы**

Ниже приводится перечень возможных сообщений программы:

**1.Файл не может быть открыт!**

**2.Выберите номер пункта меню:**

**3.Нужно вводить номер пункта от 1 до 4**

**4.Для продолжения нажмите любую клавишу**

**5.Файл пуст!**

**6.№ Фамилия Имя**

**7.Введите № класса:**

**8.Введите букву класса:**

**9.В классе А учатся учеников.**

**10.В классе Б учатся учеников.**

**11.В классе В учатся учеников.**

**3. ОПИСАНИЕ ПРОГРАММЫ**

**3.1. Метод решения задачи**

Задачу можно разбить на две отдельные подзадачи:

1. Вывод фамилии и имени учеников введенного класса.

2. Подсчет и вывод количества учеников в введенном классе и параллельных классах.

Каждая подзадача решается методом линейного поиска (последовательного просмотра элементов таблицы).

Чтобы пользователь мог выбирать, какие подзадачи решать и в каком порядке, программа выводит на экран меню (см. раздел 2.3).

**3.2. Структура программы**

spisok\_uchenikov

main

kolvo\_uchenikov

Рис.1. Функциональная структура программы

Программа состоит из трех функций: главной функции main и двухподпрограмм.

spisok\_uchenikov - выводит фамилию и имя учеников введенного класса.

kolvo\_uchenikov - подсчитывает и выводит количество учеников в введенном классе и параллельных классах.

**3.3. Описание функций**

**3.3.1. main – главная функция**

Заголовок функции:

int main ()

Значение функции:

0 – в случае успешного завершения;

1 – если входной файл не найден.

Рабочие данные:

f – указатель на структуру с информацией о входном файле;

n – номер пункта меню.

n

чтение номера пункта меню n

вывод меню

f=NULL

ifstream f(”shkola.txt”)

да нет

вывод “Файл не может быть открыт!”

spisok\_uchenikov

'1'

kolvo\_uchenikov

'2'

n!='4'

да

нет

закрытие файла

Рис.1 Блок-схема функции main()

**3.3.2. spisok\_uchenikov - вывод фамилии и имени учеников введенного класса.**

Заголовок функции:

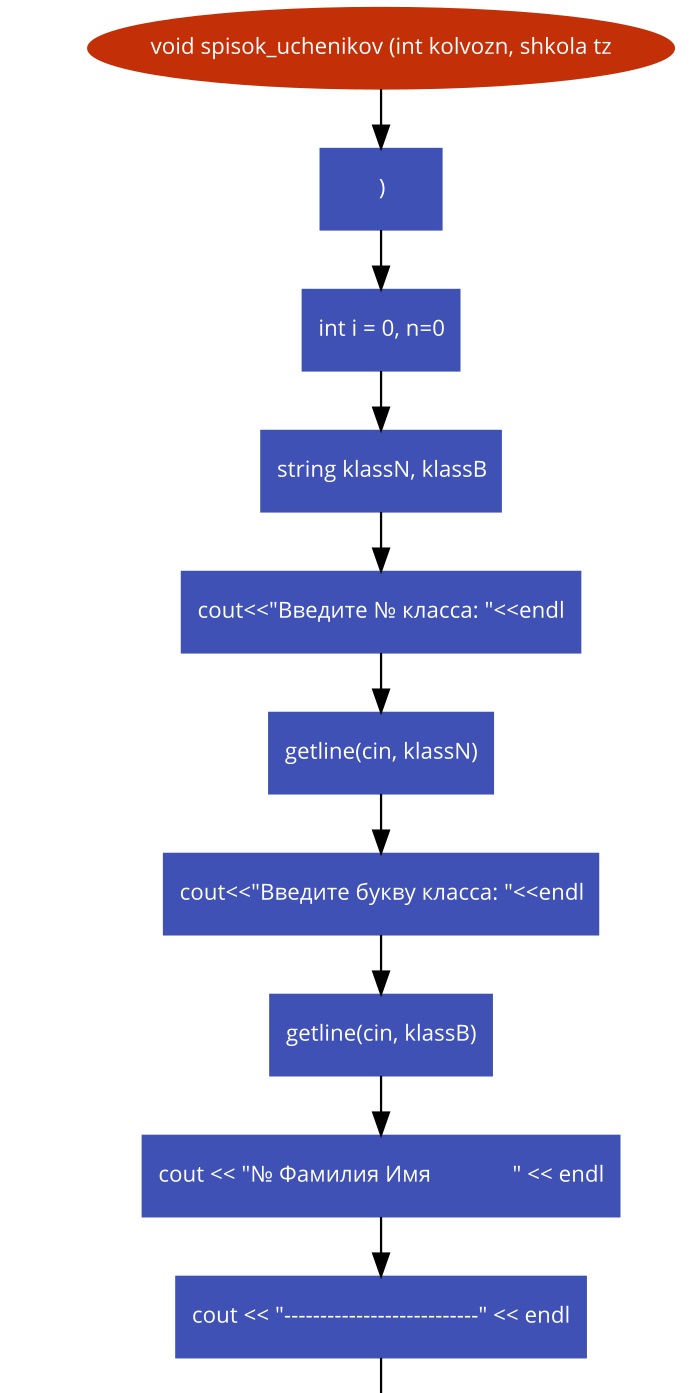
void spisok\_uchenikov (int kolvozn, shkola tz[])

Структура записи файла:

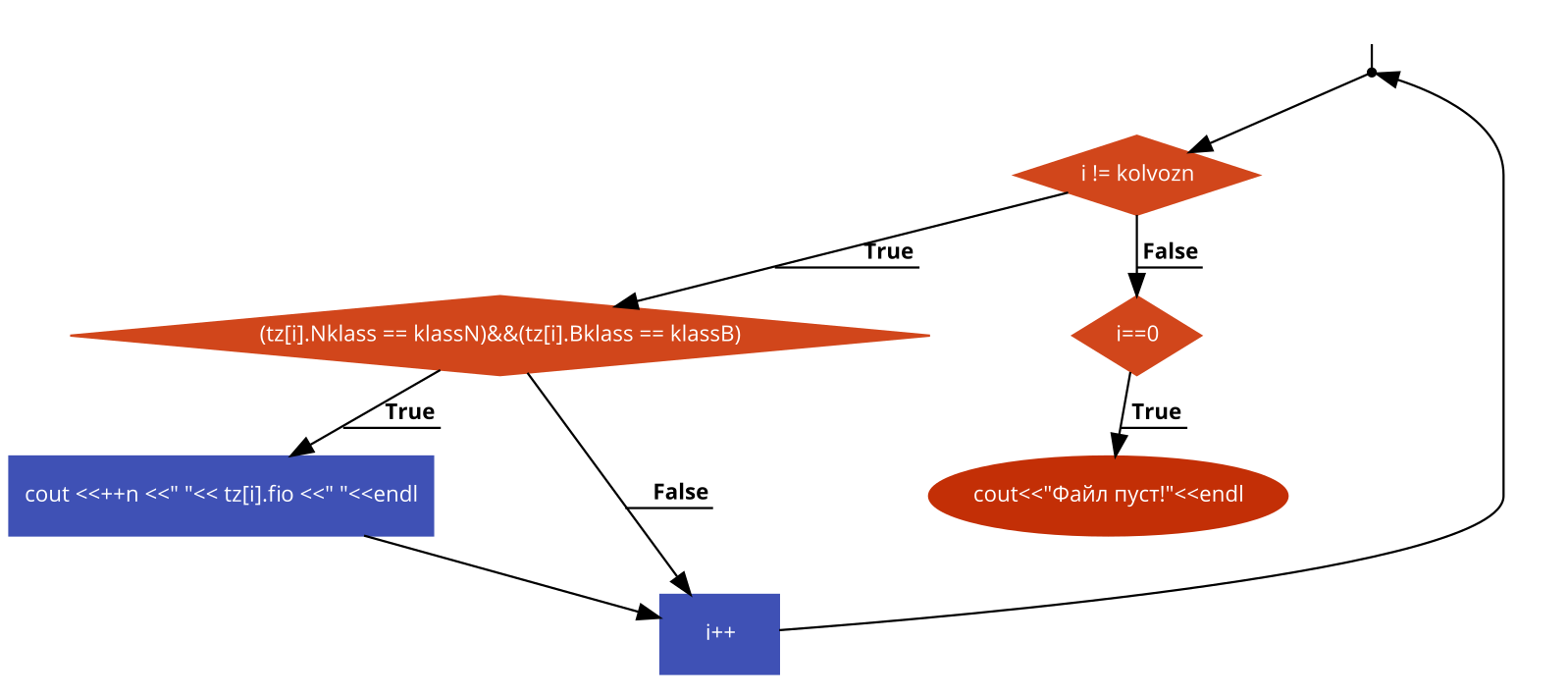
fio - фамилия и инициалы ученика Nklass - номер класса

Входные данные: Bklass - буква класса

f - ссылка на входной файл



1



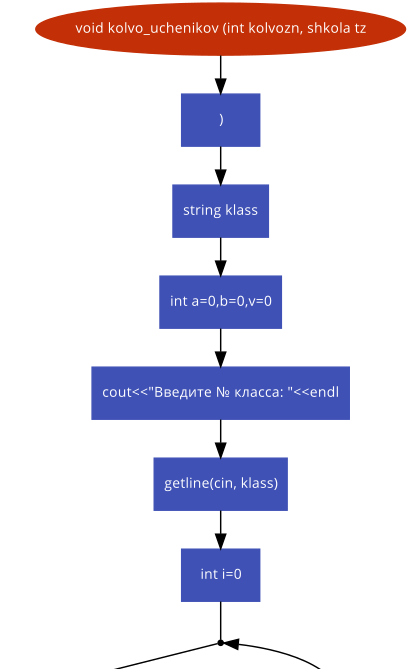
1

Рис.2. Схема функции вывода фамилии и имени учеников введенного класса.

**3.3.3. kolvo\_uchenikov - подсчет и вывод количества учеников в введенном классе и параллельных классах.**

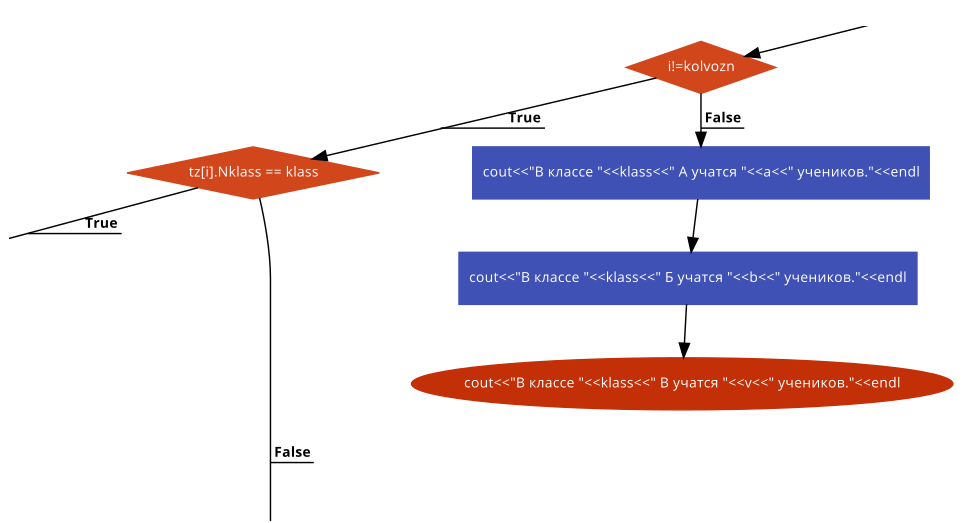
Заголовок функции:

void kolvo\_uchenikov (int kolvozn, shkola tz[])



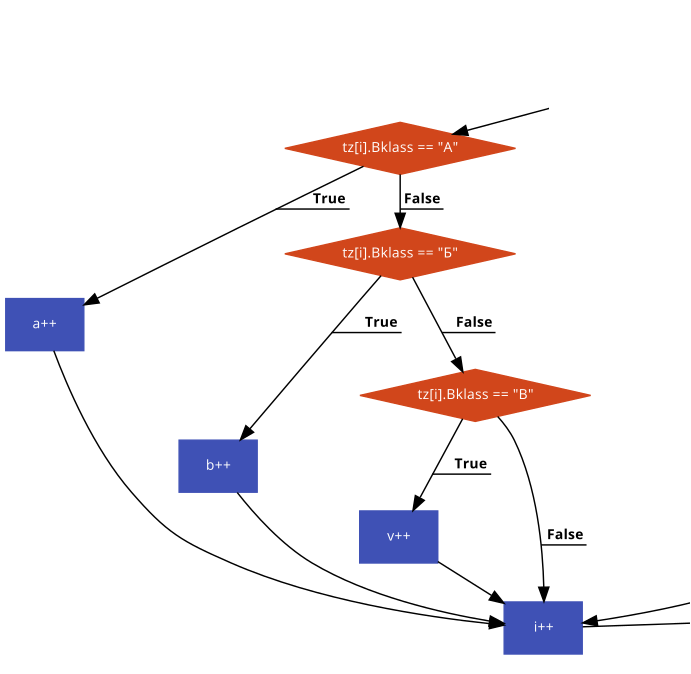
1

1



4

3



3

4

2

Рис.3. Блок-схема функции подсчета и вывода количества учеников в введенном классе и параллельных классах.

**Литература.**

1. Задания к курс. работе.

2. Пример титульного листа.

3. Пример отчета по курсовой работе.

4. Пример кода программы.

**Приложение 1. Текст программы**

#include <fstream>

#include <string>

#include <iostream>

#include <sstream>

#include <windows.h>

using namespace std;

struct shkola

{

string fio;

string Nklass;

string Bklass;

};

void spisok\_uchenikov (int kolvozn, shkola tz[]);

void kolvo\_uchenikov (int kolvozn, shkola tz[]);

int main()

{

ifstream f; /\* ссылка на входной файл \*/

char n; /\* номер пункта меню \*/

int i;

char sl;

string sl1;

SetConsoleCP(1251);// установка кодовой страницы win-cp 1251 в поток ввода

SetConsoleOutputCP(1251); // установка кодовой страницы win-cp 1251 в поток вывода

f.open("shkola.txt");

if (!f.is\_open())

{

cout << "Файл не может быть открыт!\n";

cin.get();

return 1;

}

else if (!f.eof())

{

shkola tz[88];

for (i = 0; f.good();i++)

{

if ((sl=f.get())=='\n'&&i!=0)

getline(f, tz[i].fio,',');

else

{

tz[i].fio += sl;

getline(f, sl1, ',');

tz[i].fio += sl1;

}

f >> tz[i].Nklass;

if ((sl = f.get()) == ' ')

getline(f, tz[i].Bklass,'.');

cout << tz[i].fio<<" "<< tz[i].Nklass <<" "<< tz[i].Bklass <<" "<<endl;

}

int kolvozn = i - 1;

do

{

cout << "========================================================"<<endl;

cout << "Выберите номер пункта меню:"<<endl;

cout << "1 - список учеников заданного класса" << endl;

cout << "2 - кол-во учеников в данном классе и параллельных классах"<<endl;

cout << "3 - выход"<<endl;

cout << "------------------------------------------------------"<<endl;

n=cin.get();

cin.sync();

switch (n)

{

case '1': spisok\_uchenikov(kolvozn,tz); break;

case '2': kolvo\_uchenikov(kolvozn,tz); break;

case '3': break;

default: cout << "\nНужно вводить номер пункта от 1 до 4";

}

if (n != '3')

{

cout << "\nДля продолжения нажмите любую клавишу";

cin.get();

}

} while (n != '3');

f.close();

}

else

cout << "Файл пуст!\n";

//cin.get();

return 0;

}

void spisok\_uchenikov (int kolvozn, shkola tz[])

{

int i = 0, n=0;

string klassN, klassB;

cout<<"Введите № класса: "<<endl;

getline(cin, klassN);

cout<<"Введите букву класса: "<<endl;

getline(cin, klassB);

cout << "№ Фамилия Имя " << endl;

cout << "---------------------------" << endl;

while (i != kolvozn)

{

if ((tz[i].Nklass == klassN)&&(tz[i].Bklass == klassB))

cout <<++n <<" "<< tz[i].fio <<" "<<endl;

i++;

}

if(i==0)

cout<<"Файл пуст!"<<endl;

}

void kolvo\_uchenikov (int kolvozn, shkola tz[])

{

string klass;

int a=0,b=0,v=0;

cout<<"Введите № класса: "<<endl;

getline(cin, klass);

int i=0;

while (i!=kolvozn)

{

if(tz[i].Nklass == klass)

{

if (tz[i].Bklass == "А")

a++;

else

if (tz[i].Bklass == "Б")

b++;

else

if (tz[i].Bklass == "В")

v++;

}

i++;

}

cout<<"В классе "<<klass<<" А учатся "<<a<<" учеников."<<endl;

cout<<"В классе "<<klass<<" Б учатся "<<b<<" учеников."<<endl;

cout<<"В классе "<<klass<<" В учатся "<<v<<" учеников."<<endl;

}

**Приложение 2. Пример входного файла shkola.txt**

Штольц Никита, 3 Б.

Прахов Илья, 1 А.

Габидуллин Ильнар, 4 Б.

Морозов Юрий, 1 Б.

Сафина Алина, 2 Б.

Хабибуллин Адель, 2 А.

Мартыненко Сергей, 3 А.

Кондрашова Евгения, 4 А.

Хакимуллина Ирина, 1 В.

Баранов Евгений, 1 А.

**Приложение 3. Результаты тестирования программы**

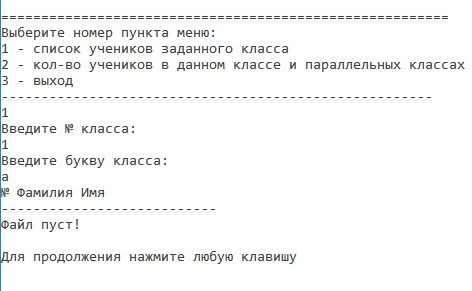
**Тест 1.** Входного файла нет в текущем каталоге.

Результат:

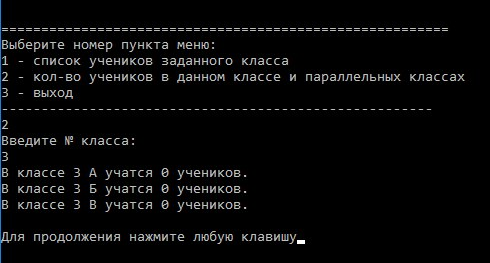


В следующих тестах используется файл из приложения 2.

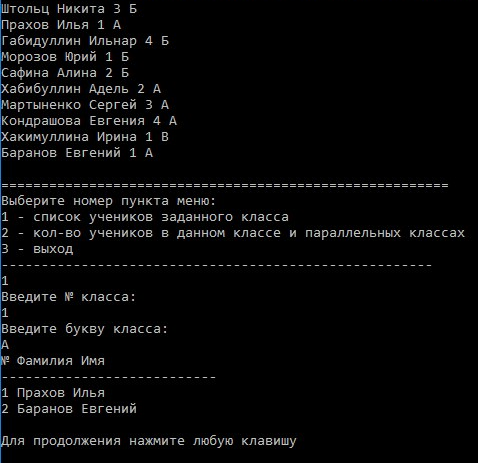
**Тест 2.**



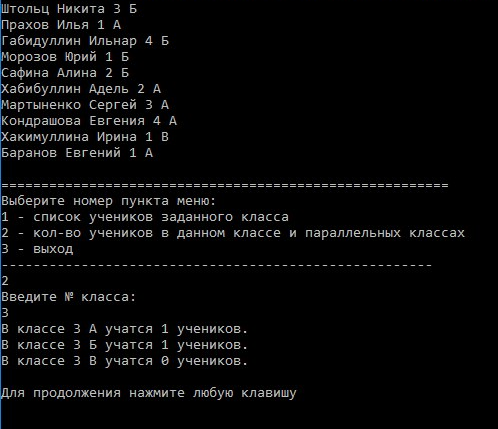
**Тест 3.**

****

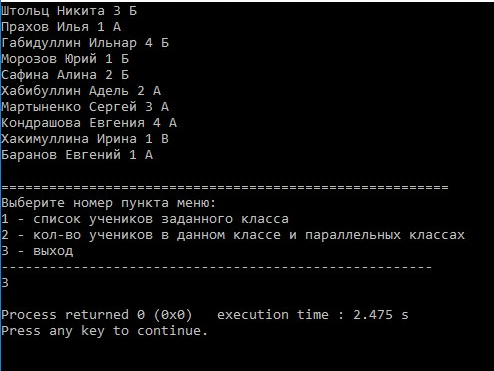
**Тест 4.**

****

**Тест 5.**

****

**Тест 6.**

****