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. ne\_doljen – вывода Фамилии И. О. студентов, не имеющих задолжностей. 7

3.3.3. procent\_ne\_uspevaushih - вывод процента неуспевающих студентов.  9

ЛИТЕРАТУРА 11

Приложение 1. Текст программы 12

Приложение 2. Пример входного файла sessia.txt 16

Приложение 3. Результаты тестирования программы 17

**1. ЗАДАНИЕ**

Дан файл, содержащий сведения о сдаче студентами сессии. Структура записи файла: фамилия и инициалы студента, оценки по четырем экзаменам и пяти зачетам («з» - зачет, «н» - незачет).

Написать программу, выдающую следующую информацию:

-фамилии студентов, не имеющих задолженностей;

-процент неуспевающих студентов;

**2. ОПИСАНИЕ ПРИМЕНЕНИЯ**

**2.1 Запуск программы**

Запуск программы производится в среде программирования CodeBlocks

**2.2 Входные данные**

Входные данные находятся в файле “sessia.txt”. Число строк в файле - произвольное. Каждая строка содержит фамилию с инициалами (20 символов), оценки по 4 экзаменам (8 символов), оценки по 5 зачетам (6 символов). Пример строки файла:

Прахов Н. В., 4 3 2 4 ззззз.

| 20 | 8 | 6 |

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 студентов без задолжностей**

**2 - Процент неуспевающих студентов**

**3 - Выход**

При выборе пункта 1 на экран выводится результат в виде:

**№ Фамилия И.О. зачет экзамен**

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

**1 Суслов Г. К.**

**2 Маулина А. Д.**

**3 Капустин В. Н.**

**Вывод завершен**

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 И.О. зачет экзамен**

**7.Вывод завершен**

**8.процент неуспевающих =**

**3. ОПИСАНИЕ ПРОГРАММЫ**

**3.1. Метод решения задачи**

Задачу можно разбить на две отдельные подзадачи:

1. Вывод Фамилии И. О. студентов, не имеющих задолжностей.

2. Подсчет и вывод количества процента неуспевающих.

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**

ne\_doljen

main

procent\_ne\_uspevaushih

Рис.1. Функциональная структура программы

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

ne\_doljen - выводит Фамилию И. О. студентов, не имеющих задолжностей.

procent\_ne\_uspevaushih - выводит процент неуспевающих студентов.

**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 – объект класса ifstream, с помощью которого получают доступ к указанному файлу;

n – номер пункта меню.

n

чтение номера пункта меню n

вывод меню

вывод “Файл не найден!”

n

чтение номера пункта меню n

вывод меню

вывод “Файл не может быть открыт!”

f=NULL

ifstream f(”sessia.txt”)

да нет

ne\_doljen

'1'

procent\_ne\_uspevaushih

'2'

n!='4'

да

нет

закрытие файла

Рис.1 Блок-схема функции main()

**3.3.2. ne\_doljen – вывода Фамилии И. О. студентов, не имеющих задолжностей.**

Заголовок функции: Структура записи файла:

void ne\_doljen (int kolvozn, sessia tz[])

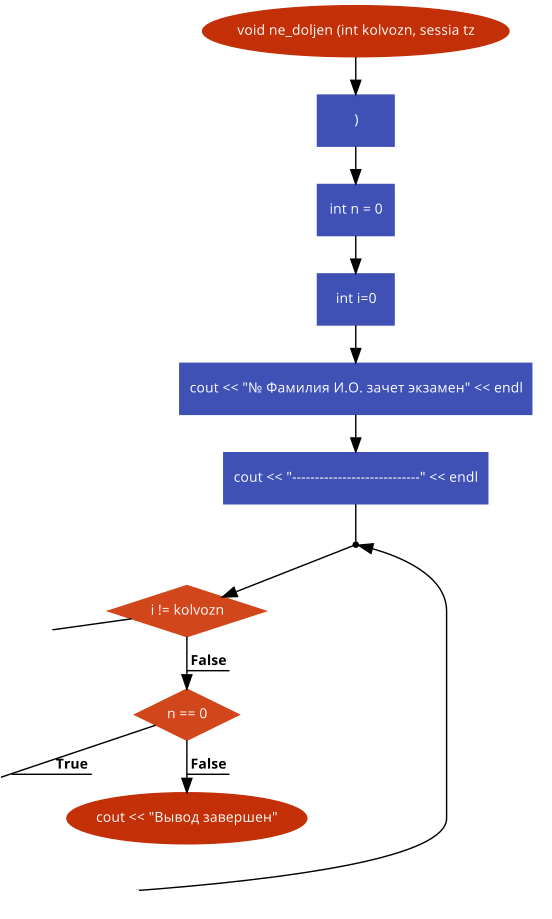
fio - фамилия и инициалы студента zachet - оценки по зачетам

z1, z2, z3, z4, z5 - оценки по

экзаменам

Входные данные:

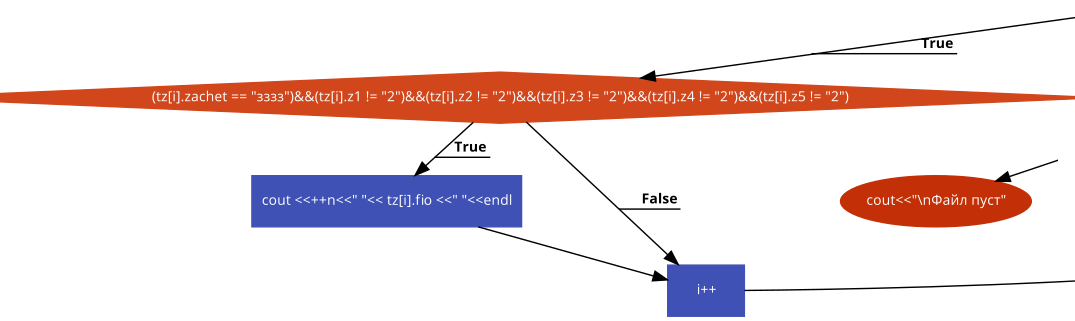
f - объект класса ifstream, с помощью которого получают доступ к указанному файлу;



3

2

1



2

1

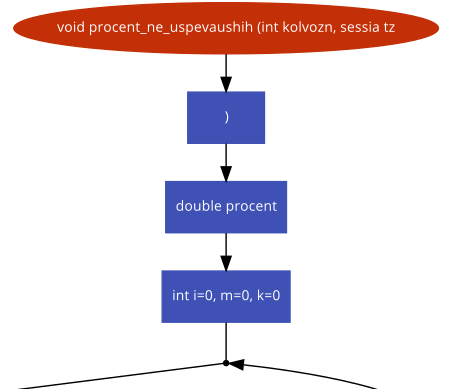
Рис.2. Схема функции вывода Фамилии И. О. студентов, не имеющих задолжностей.

i++

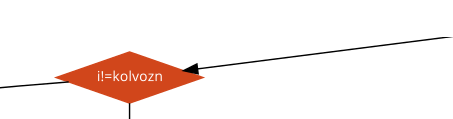
**3.3.3. procent\_ne\_uspevaushih - вывод процента неуспевающих студентов.**

Заголовок функции:

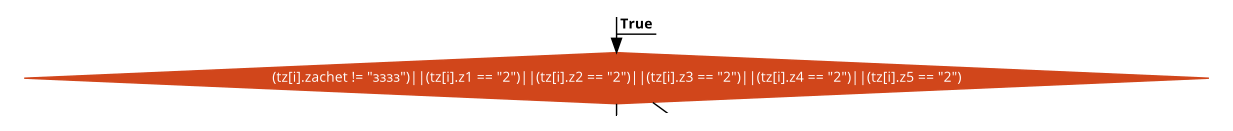
void procent\_ne\_uspevaushih (int kolvozn, sessia tz[])



1

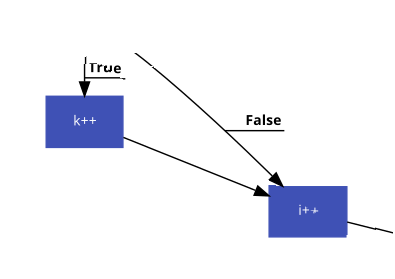
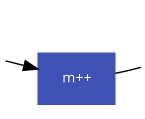


2



4

3

1

3

4

2

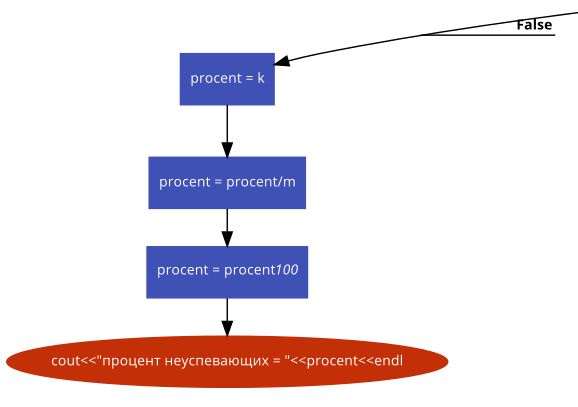


Рис.3. Блок-схема функции вывода процента неуспевающих студентов.

**Литература.**

1. Задания к курс. работе.

2. Пример кода программы.

3. Пример титульного листа.

4. Пример отчета по курсовой работе.

**Приложение 1. Текст программы**

#include <fstream>

#include <string>

#include <iostream>

#include <sstream>

#include <windows.h>

using namespace std;

struct sessia

{

string fio;

string zachet;

string z1, z2, z3, z4;

};

void ne\_doljen (int kolvozn, sessia tz[]);

void procent\_ne\_uspevaushih (int kolvozn, sessia 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("sessia.txt");

if (!f.is\_open())

{

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

cin.get();

return 1;

}

else if (!f.eof())

{

sessia 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].z1;

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

{

f>>tz[i].z2;

}

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

{

f>>tz[i].z3;

}

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

{

f>>tz[i].z4;

}

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

{

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

}

cout << tz[i].fio<<" "<<" " << tz[i].z1<<" "<<" " << tz[i].z2<<" " << tz[i].z3<<" " << tz[i].z4<<" " << tz[i].zachet <<" "<<endl;

}

int kolvozn = i - 1;

do

{

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

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

cout << "1 - Список студентов без задолжностей " << endl;

cout << "2 - Процент неуспевающих студентов "<<endl;

cout << "3 - Выход"<<endl;

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

n=cin.get();

cin.sync();

switch (n)

{

case '1': ne\_doljen(kolvozn,tz); break;

case '2': procent\_ne\_uspevaushih(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 ne\_doljen (int kolvozn, sessia tz[])

{

int n = 0;

int i=0;

cout << "№ Фамилия И.О. зачет экзамен" << endl;

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

while (i != kolvozn)

{

if ((tz[i].zachet == "ззззз")&&(tz[i].z1 != "2")&&(tz[i].z2 != "2")&&(tz[i].z3 != "2")&&(tz[i].z4 != "2"))

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

i++;

}

if (n == 0) cout<<"\nФайл пуст";

else cout << "Вывод завершен";

}

void procent\_ne\_uspevaushih (int kolvozn, sessia tz[])

{

double procent;

int i=0, m=0, k=0;

while(i!=kolvozn)

{

if ((tz[i].zachet != "ззззз")||(tz[i].z1 == "2")||(tz[i].z2 == "2")||(tz[i].z3 == "2")||(tz[i].z4 == "2"))

k++;

i++;

m++;

}

procent = k;

procent = procent/m;

procent = procent\*100;

if (k!=0)

cout<<"процент неуспевающих = "<<procent<<endl;

else

cout<<"Файл пуст"<<endl;

}

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

Прахов Н. В., 4 3 2 4 ззззз.

Суслов Г. К., 3 3 4 3 ззззз.

Гайнутдинова А. Р., 4 4 3 5 зззнз.

Маулина А. Д., 5 4 5 5 ззззз.

Крикунов Д. В., 2 4 3 2 ннззз.

Капустин В. Н., 5 5 5 5 ззззз.

**Приложение 3. Результаты тестирования программы**

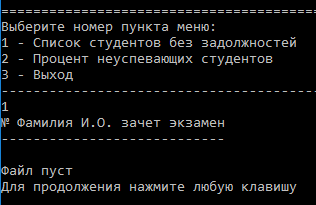
**Тест 1.** Входного файла нет в текущем каталоге.

Результат:

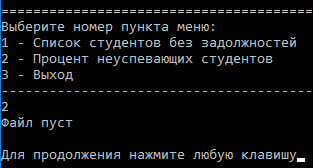


В следующих тестах используется файл из приложения 2.

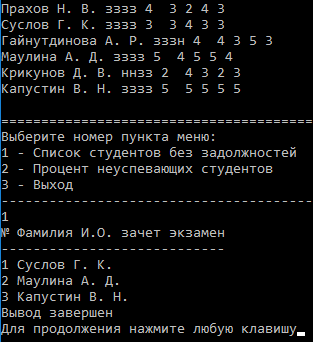
**Тест 2.**



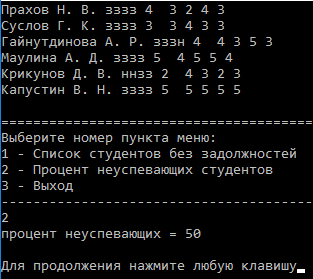
**Тест 3.**

****

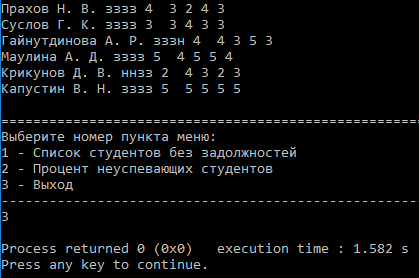
**Тест 4.**

****

**Текст 5.**

****

**Тест 6.**

****