Министерство образования и науки Российской Федерации

Севастопольский Государственный Университет

Кафедра ИС

ОТЧЕТ

по лабораторной работе №6

ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ

НАД СТРУКТУРАМИ И БИНАРНЫМИ ФАЙЛАМИ

Выполнил: ст. гр. ИТб-11

Куркчи А.Э.

Проверил:

Забаштанский А.К.

Севастополь

2015

# 1. Цель

Изучение способов описания структур данных на языке C. Исследование особенностей обработки бинарных файлов, хранящих структурные типы данных.

2. Постановка задач

Вариант №2

Описать структуру с именем STUDENT, содержащую следующие поля:

* Фамилия и Инициалы
* Номер группы
* Успеваемость (массив из пяти элементов)

Написать программу, выполняющую следующие действия с помощью функций:

* Ввод с клавиатуры данных в файл, состоящий из структур типа STUDENT; записи должны быть упорядочены по возрастанию среднего балла;
* Чтение данных из этого файла;
* Вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5;
* Если таких студентов нет, вывести соответствующее сообщение;

3. Структурная схема

На рисунке 1 изображена схема алгоритма работы программы.

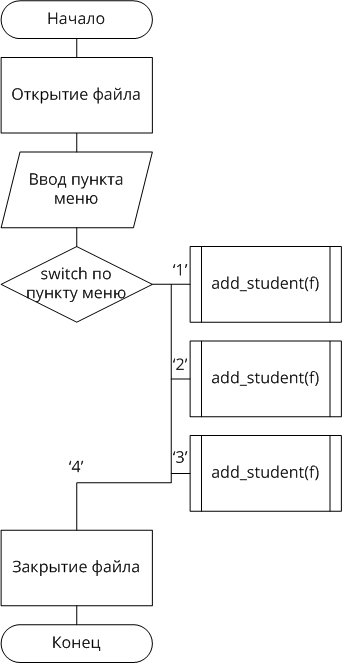
****

Рисунок 1 – Структурная схема

На рисунке 2 изображена схема алгоритма работы подпрограммы add\_student.

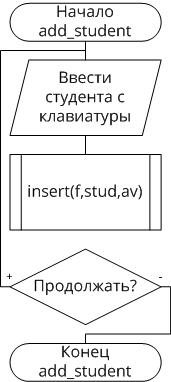
****

Рисунок 2 – Структурная схема add\_student

На рисунке 3 изображена схема алгоритма работы подпрограммы show\_all.

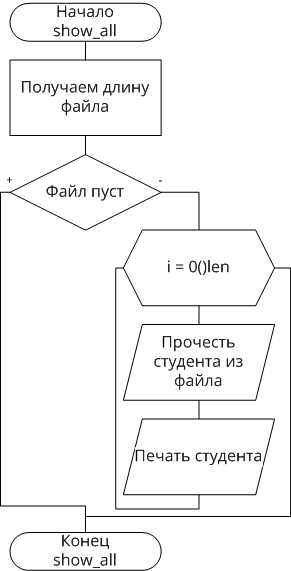
****

Рисунок 3 – Структурная схема show\_all

На рисунке 4 изображена схема алгоритма работы подпрограммы insert.

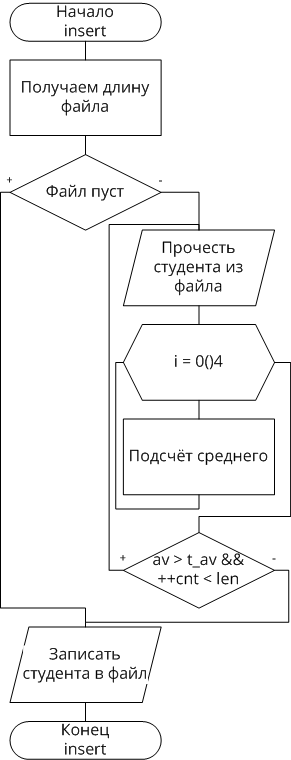
****

Рисунок 4 – Структурная схема insert

На рисунке 5 изображена схема алгоритма работы подпрограммы show\_best.

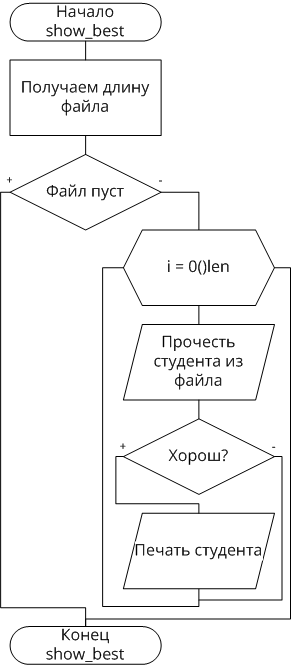
****

Рисунок 5 – Структурная схема show\_best

4. Текст программы

#include <stdio.h>

#include <iostream>

#include <iomanip>

#define NAME\_SIZE 32

#define GROUP\_SIZE 16

#define MARKS\_SIZE 5

**using** **namespace** std**;**

struct student **{**

char name**[**NAME\_SIZE**];**

char group**[**GROUP\_SIZE**];**

int marks**[**MARKS\_SIZE**];**

**};**

const int size\_s **=** **sizeof(**student**);**

// Вставка студента в файл, сохраняя упорядоченность

void insert**(**FILE **\***f**,**student stud**,**double avarage**)** **{**

fseek**(**f**,**0**,**SEEK\_END**);**

long len **=** ftell**(**f**)/**size\_s**;**

rewind**(**f**);**

**if(**len **==** 0**)** **{**

fwrite**(&**stud**,**size\_s**,**1**,**f**);**

**}** **else** **{**

student tmp**;**

double tmp\_avarage**;**

long cnt **=** 0**;**

**do** **{**

fread**(&**tmp**,**size\_s**,**1**,**f**);**

tmp\_avarage **=** 0.0**;**

**for(**int i**=**0**;**i**<**MARKS\_SIZE**;**i**++)** **{**

tmp\_avarage **+=** tmp**.**marks**[**i**];**

**}**

tmp\_avarage **/=** MARKS\_SIZE**;**

**}** **while((**avarage **>** tmp\_avarage**)** **&&** **(++**cnt **<** len**));**

**if(**tmp\_avarage **>** avarage**)** **{**

fseek**(**f**,-**size\_s**,**SEEK\_CUR**);**

**}**

fwrite**(&**stud**,**size\_s**,**1**,**f**);**

**}**

**}**

// Добавить студента

void add\_student**(**FILE **\***f**)** **{**

char c**;**

**do** **{**

double avarage **=** 0.0**;**

student stud**;**

cout **<<** "Student name:" **<<** endl**;**

cin**.**getline**(**stud**.**name**,**NAME\_SIZE**);**

cout **<<** "Student group:" **<<** endl**;**

cin**.**getline**(**stud**.**group**,**GROUP\_SIZE**);**

cout **<<** "Student marks:" **<<** endl**;**

**for(**int i**=**0**;**i**<**MARKS\_SIZE**;**i**++)** **{**

cin **>>** stud**.**marks**[**i**];**

avarage **+=** stud**.**marks**[**i**];**

**}**

avarage **/=** MARKS\_SIZE**;**

insert**(**f**,**stud**,**avarage**);**

cout **<<** "Continue addition? (y/n): "**;**

cin **>>** c**;**

cin**.**get**();**

**}** **while(**c **==** 'y'**);**

**}**

// Показать всех студентов

void show\_all**(**FILE **\***f**)** **{**

fseek**(**f**,**0**,**SEEK\_END**);**

long len **=** ftell**(**f**)/**size\_s**;**

rewind**(**f**);**

**if(**len **==** 0**)** **{**

cout **<<** "No students" **<<** endl**;**

cout **<<** "Press enter to continue..." **<<** endl**;**

cin**.**get**();**

**return;**

**}** **else** **{**

cout **<<** "╔════════════════════════════════╦════════════════╦═════════╗" **<<** endl**;**

cout **<<** "║ Name ║ Group ║ Avarage ║" **<<** endl**;**

student stud**;**

double avarage**;**

**for(**int i**=**0**;**i**<**len**;**i**++)** **{**

fread**(&**stud**,**size\_s**,**1**,**f**);**

avarage **=** 0.0**;**

**for(**int j**=**0**;**j**<**MARKS\_SIZE**;**j**++)** **{**

avarage **+=** stud**.**marks**[**j**];**

**}**

avarage **/=** MARKS\_SIZE**;**

cout **<<** "╠════════════════════════════════╬════════════════╬═════════╣" **<<** endl**;**

cout**.**precision**(**2**);**

cout **<<** "║" **<<** setw**(**32**)** **<<** stud**.**name **<<** "║" **<<** setw**(**16**)** **<<** stud**.**group **<<** "║" **<<** setw**(**9**)** **<<** avarage **<<** "║" **<<** endl**;**

**}**

cout **<<** "╚════════════════════════════════╩════════════════╩═════════╝" **<<** endl**;**

**}**

cout **<<** "Press enter to continue..." **<<** endl**;**

cin**.**get**();**

**}**

// Показать лучших студентов

void show\_best**(**FILE **\***f**)** **{**

fseek**(**f**,**0**,**SEEK\_END**);**

long len **=** ftell**(**f**)/**size\_s**;**

rewind**(**f**);**

**if(**len **==** 0**)** **{**

cout **<<** "No students" **<<** endl**;**

cout **<<** "Press enter to continue..." **<<** endl**;**

cin**.**get**();**

**return;**

**}** **else** **{**

bool found **=** **false;**

student stud**;**

double avarage**;**

**for(**int i**=**0**;**i**<**len**;**i**++)** **{**

fread**(&**stud**,**size\_s**,**1**,**f**);**

avarage **=** 0.0**;**

bool best **=** **false;**

**for(**int j**=**0**;**j**<**MARKS\_SIZE**;**j**++)** **{**

**if(**stud**.**marks**[**i**]** **>=** 4**)** **{**

best **=** **true;**

**}**

avarage **+=** stud**.**marks**[**j**];**

**}**

avarage **/=** MARKS\_SIZE**;**

**if(**best**)** **{**

**if(!**found**)** **{**

cout **<<** "╔════════════════════════════════╦════════════════╦═════════╗" **<<** endl**;**

cout **<<** "║ Name ║ Group ║ Avarage ║" **<<** endl**;**

**}**

found **=** **true;**

cout **<<** "╠════════════════════════════════╬════════════════╬═════════╣" **<<** endl**;**

cout**.**precision**(**2**);**

cout **<<** "║" **<<** setw**(**32**)** **<<** stud**.**name **<<** "║" **<<** setw**(**16**)** **<<** stud**.**group **<<** "║" **<<** setw**(**9**)** **<<** avarage **<<** "║" **<<** endl**;**

**}**

**}**

**if(**found**)** **{**

cout **<<** "╚════════════════════════════════╩════════════════╩═════════╝" **<<** endl**;**

**}** **else** **{**

cout **<<** "No such students" **<<** endl**;**

**}**

**}**

cout **<<** "Press enter to continue..." **<<** endl**;**

cin**.**get**();**

**}**

int main**(**int argc**,** char**\*\*** argv**)** **{**

FILE **\***f**;**

f **=** fopen**(**"data.bin"**,**"r+b"**);**

**if(!**f**)** **{**

f **=** fopen**(**"data.bin"**,**"w+b"**);**

**if(!**f**)** **{**

cout **<<** "Can`t open (create) file" **<<** endl**;**

**return** 1**;**

**}**

**}**

char c**;**

**while(**1**)** **{**

cout **<<** "1 - Add student" **<<** endl**;**

cout **<<** "2 - Show all students" **<<** endl**;**

cout **<<** "3 - Show best students" **<<** endl**;**

cout **<<** "4 - Exit" **<<** endl**;**

cout **<<** "Choose your destiny: "**;**

cin **>>** c**;**

cin**.**get**();**

**switch(**c**)** **{**

**case** '1'**:**

add\_student**(**f**);**

**break;**

**case** '2'**:**

show\_all**(**f**);**

**break;**

**case** '3'**:**

show\_best**(**f**);**

**break;**

**case** '4'**:**

fclose**(**f**);**

**return** 0**;**

**}**

**}**

fclose**(**f**);**

**return** 0**;**

**}**

5. Тестовые примеры

На рисунке 6 представлено меню программы

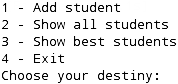


Рисунок 6 – Меню программы

На рисунке 7 представлен вывод студентов в таблице



Рисунок 7 – Вывод студентов

Вывод

В ходе лабораторной работы были изучены способы описания структур данных на языке C. Исследованы особенности обработки бинарных файлов, хранящих структурные типы данных.