Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Программная инженерия»

Кафедра инженерной психологии и эргономики

Учебная дисциплина «Основы алгоритмизации и программирования »

Отчет

по лабораторной работе No7

«Структуры и файлы»

Вариант 17

Подготовил: Ковальчук А.А.

Проверил: Усенко Ф.В.

Минск 2024

**Цель работы:** сформировать навыки и умения обработки структурированных типов данных, организованных в виде структур и файлов.

Задание: Описать структуру с именем STUDENT, содержащую следующие поля: - фамилия и инициалы; - номер группы; - успеваемость (массив из пяти элементов). Написать программу, выполняющую следующие действия: - ввод с клавиатуры данных; записи должны быть упорядочены по алфавиту. Предусмотреть запись в файл.

#include <iostream>

#include <fstream>

#include <cstring>

#include <windows.h>

#include <algorithm>

using namespace std;

struct STUDENT {

char lastName[50];

int groupNumber;

int grades[5];

STUDENT() {

lastName[0] = '\0'; // Инициализация первого символа массива нулевым символом, чтобы указать, что строка пустая.

groupNumber = 0; // начальное значение этого поля при создании объекта структуры STUDENT

for (int i = 0; i < 5; ++i) { //Цикл для инициализации всех элементов массива нулем

grades[i] = 0;

}

}

}

// Функция для сравнения 2-х студентов

bool compareStudents(const STUDENT& a, const STUDENT& b) {

return strcmp(a.lastName, b.lastName) < 0;

}

int main() {

setlocale(LC\_ALL, "RU");

SetConsoleCP(1251);

// Массив для хранения студентов

STUDENT students[200];

int count = 0;

// Считывание данных из файла

ifstream file("student.txt");

if (file.is\_open()) {

while (file) { //Начало цикла для чтения данных из файла

char temporary[50]; //Объявление временного массива для чтения строк

// Считываем фамилию и инициалы

file.getline(temporary, 50, ':'); // Чтение строки до :

file.ignore(1); // Пропускаем пробел после двоеточия

if (!file.getline(students[count].lastName, 50)) break; // Чтение номера группы и проверка на успешное чтение.

// Считываем номер группы

file.getline(temporary, 50, ':');

file.ignore(1); // Пропускаем пробел после двоеточия

if (!(file >> students[count].groupNumber)) break;

file.ignore(256, '\n'); // Пропускаем оставшуюся часть строки

// Считываем успеваемость

file.getline(temporary, 50, ':');

file.ignore(1); // Пропускаем пробел после двоеточия

for (int j = 0; j < 5; ++j) {

if (!(file >> students[count].grades[j])) break;

}

file.ignore(256, '\n'); // Игнорируем оставшуюся часть строки

count++;

// Проверка на переполнение массива

if (count >= 200) {

cout << "Превышено максимальное количество студентов." << endl;

break;

}

}

file.close();

}

else {

cout << "Не удалось открыть файл student.txt для чтения." << endl;

}

char choice[10]; // Объявление массива для хранения ответа пользователя на вопрос о добавлении нового студента.

do {

if (count >= 200) {

cout << "Превышено максимальное количество студентов." << endl;

break;

}

cout << endl;

cout << "Введите данные студента.\n";

cout << "Фамилия и инициалы: ";

cin.getline(students[count].lastName, 50);

cout << "Номер группы: ";

cin >> students[count].groupNumber;

cout << "Успеваемость (5 оценок): ";

for (int j = 0; j < 5; ++j) {

cin >> students[count].grades[j];

}

cin.ignore(); // Игнорировать оставшийся символ новой строки

count++; //Увеличение счетчика студентов после успешного ввода данных.

cout << endl;

cout << "Хотите ввести данные еще одного студента? (да/нет): ";

cin.getline(choice, 10);

} while (strcmp(choice, "да") == 0 || strcmp(choice, "Да") == 0); //Проверка, если ответ пользователя "да", повторный ввод данных для следующего студента.

cout << endl;

sort(students, students + count, compareStudents);// Сортировка массива студентов по фамилии

ofstream outFile("student.txt");// Запись отсортированных данных в файл

if (outFile.is\_open()) {

for (int i = 0; i < count; ++i) { //Начало цикла для записи данных в файл.

if (students[i].groupNumber != 0) { // Проверяем, чтобы не записывать пустые записи

outFile << "Фамилия и инициалы: " << students[i].lastName << endl;

outFile << "Номер группы: " << students[i].groupNumber << endl;

outFile << "Успеваемость: ";

for (int j = 0; j < 5; ++j) {

outFile << students[i].grades[j] << " ";

}

outFile << endl;

}

}

outFile.close();

cout << "Данные успешно записаны в файл student.txt." << endl;

}

else {

cout << "Не удалось открыть файл student.txt для записи." << endl;

}

return 0;

}

Результат работы программы представлен на рисунке 1.

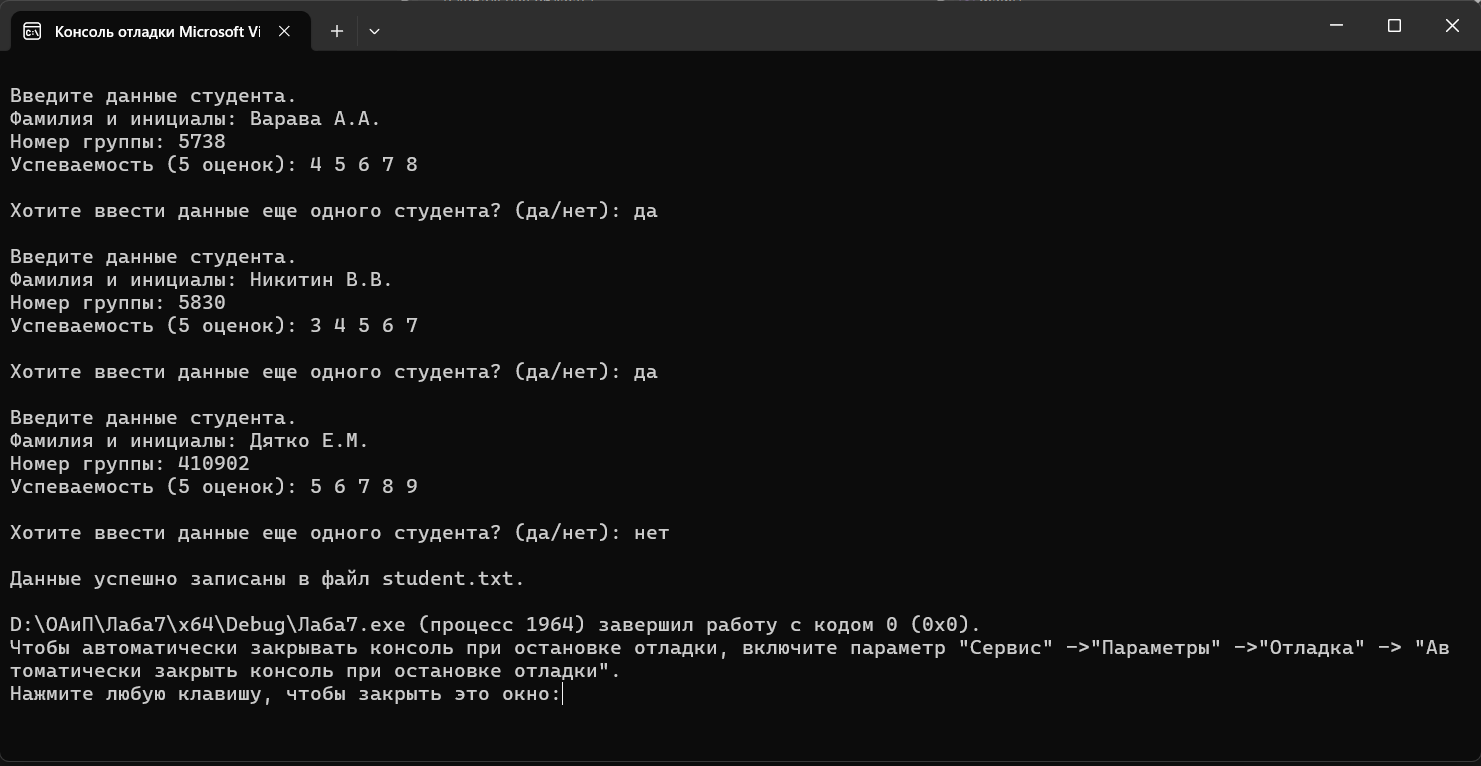


Рисунок 1

Результат записи в файл представлен на рисунке 2.

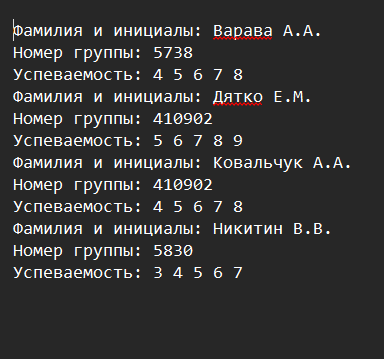


Рисунок 2

Блок-схема работы программы представлена на рисунке 3.

STUDENT

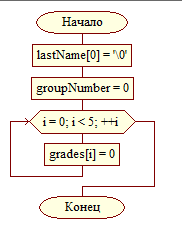


Рисунок 3

compareStudents

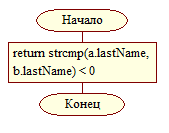


Рисунок 3

Main

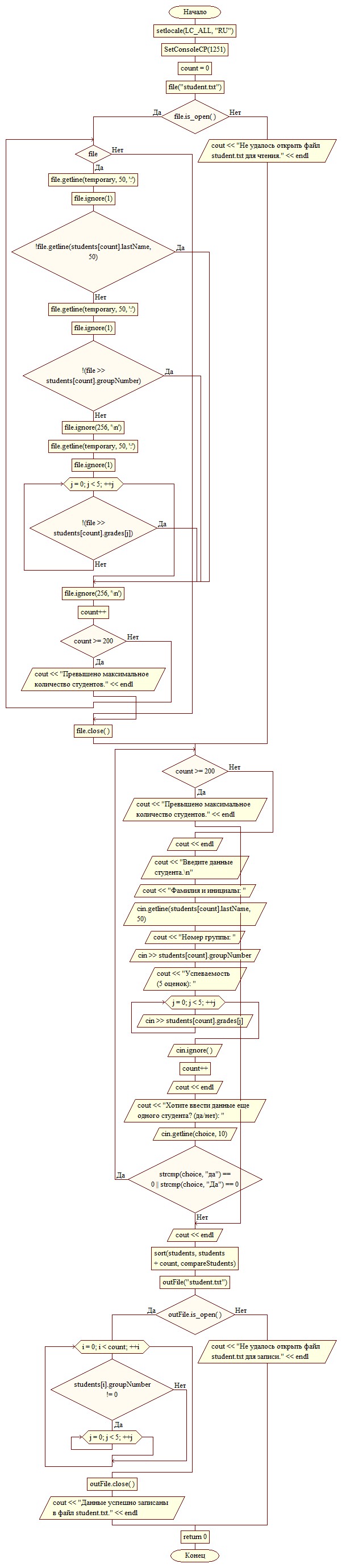


Рисунок 3

**Вывод:** в ходе выполнения работы была достигнута цель данной лабораторной работы: сформировать навыки и умения обработки структурированных типов данных, организованных в виде структур и файлов.