Министерство образования Российской Федерации

Севастопольский государственный университет

Кафедра ИС

ОТЧЕТ

По лабораторной работе №10

ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ НАД

ЗАПИСЯМИ И ТИПИЗИРОВАННЫМИ ФАЙЛАМИ

Выполнил: ст. гр. ИТб-11

Куркчи А. Э.

Проверил:

Забаштанский А.К.

Севастополь

2014

1. Цель работы

Изучение способов обработки логически связанных данных различных типов. Создание программ, использующих записи и типизированные файлы.

2. Постановка задач

Вариант №2

Описать структуру с именем student, содержащую следующие поля:

* Фамилия и инициалы
* Номер группы
* Успеваемость

Написать программу, выполняющую следующие действия с помощью процедур или функций:

* Ввод с клавиатуры данных в файл, состоящий из структур типа student; записи должны быть упорядочены по возрастанию среднего бала;
* 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.

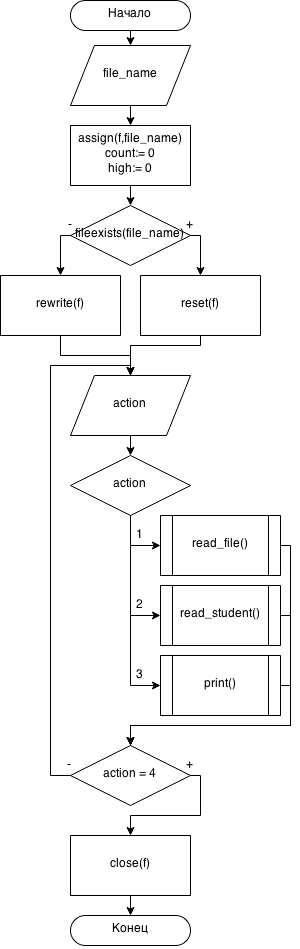


Рисунок 1 – структурная схема

На рисунке 2 изображена схема алгоритма работы процедуры print.

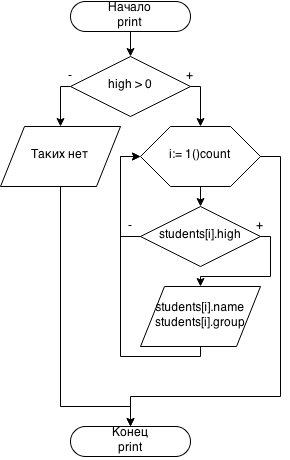


Рисунок 2 – структурная схема процедуры print

На рисунке 3 изображена схема алгоритма работы процедуры read\_file.

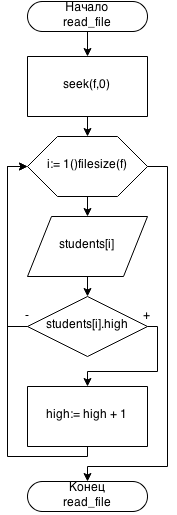


Рисунок 3 – структурная схема процедуры read\_file

На рисунке 4 изображена схема алгоритма работы процедуры read\_student.

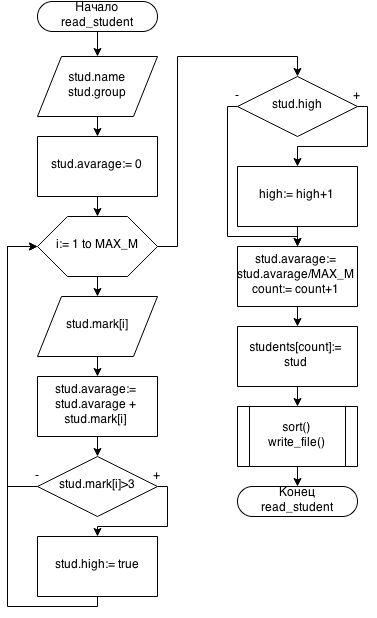


Рисунок 4 – структурная схема процедуры read\_student

На рисунке 5 изображена схема алгоритма работы процедуры sort.

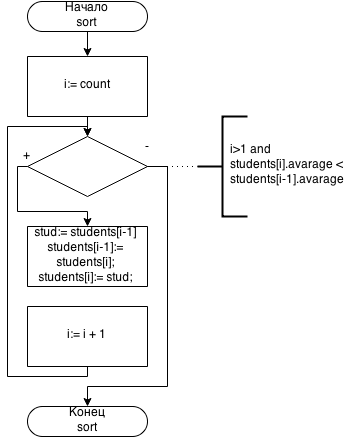


Рисунок 5 – структурная схема процедуры sort

На рисунке 6 изображена схема алгоритма работы процедуры sort.

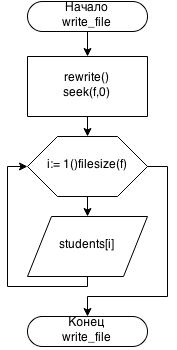


Рисунок 6 – структурная схема процедуры write\_file

4. Текст программы

**program** lab10;

**uses** crt;

**const** MAX\_N=1000;

MAX\_M=5;

**type** student = **record**

name,group:string[20];

mark:**array** [1..MAX\_M] **of** real;

avarage:real;

high:boolean;

**end**;

**var** file\_name:string;

students:**array** [1..MAX\_N] **of** student;

f:**file of** student;

count,high:integer;

**procedure** read\_file();

**var** i:integer;

**begin**

seek(f,0);

**for** i:=1 **to** filesize(f) **do begin**

read(f,students[i]);

**if**(students[i].high) **then**

high:= high+1;

**end**;

count:= filesize(f);

**end**;

**procedure** write\_file();

**var** i:integer;

**begin**

rewrite(f);

seek(f,0);

**for** i:=1 **to** count **do**

write(f,students[i]);

**end**;

**procedure** sort();

**var** i:integer;

stud:student;

**begin**

i:= count;

**while**((i>1) **and** (students[i].avarage < students[i-1].avarage)) **do begin**

stud:= students[i-1];

students[i-1]:= students[i];

students[i]:= stud;

i:= i-1;

**end**;

**end**;

**procedure** read\_student();

**var** stud:student;

i:integer;

**begin**

writeln('Введите Ф.И.О.:');

readln(stud.name);

writeln('Введите группу:');

readln(stud.group);

stud.avarage:= 0;

**for** i:=1 **to** MAX\_M **do begin**

writeln('Введите оценку #',i);

readln(stud.mark[i]);

stud.avarage:= stud.avarage + stud.mark[i];

**if**(stud.mark[i]>3) **then**

stud.high:= true;

**end**;

stud.avarage:= stud.avarage/MAX\_M;

**if**(stud.high) **then**

high:= high+1;

count:= count+1;

students[count]:= stud;

sort();

write\_file();

**end**;

**procedure** print();

**var** i:integer;

**begin**

**if**(high > 0) **then begin**

writeln('+--------------------+--------------------+');

writeln('|','Ф.И.О.':13,'|':8,'Группа':13,'|':8);

**for** i:= 1 **to** count **do**

**if**(students[i].high) **then begin**

writeln('+--------------------+--------------------+');

writeln('|',students[i].name:20,'|',students[i].group:20,'|');

**end**;

writeln('+--------------------+--------------------+');

**end else begin**

writeln('Таких студентов нет.');

**end**;

writeln('Для продолжения нажмите Enter');

readln;

**end**;

**var** action:integer;

**begin**

writeln('Добро пожаловать! Введите имя файла с данными:');

readln(file\_name);

assign(f,file\_name);

**if**(**not**(fileexists(file\_name))) **then**

rewrite(f)

**else**

reset(f);

count:= 0;

high:= 0;

**repeat**

clrscr();

writeln('Выберите одно из доступных действий:');

writeln('1 - Чтение файла');

writeln('2 - Запись элемента в файл');

writeln('3 - Вывод всех студентов с оценками 4 и 5');

writeln('4 - Выход');

readln(action);

**case** action **of**

1:read\_file();

2:read\_student();

3:print();

**end**;

**until** action = 4;

writeln('Выключение...');

close(f);

**end**.

5. Тестовые примеры

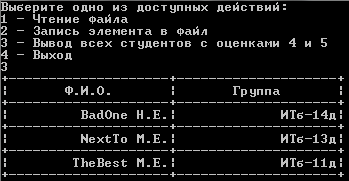


Рисунок 7 – пример

На рисунке 7 изображена работа программы с файлом из 4 записей:

1. TheBest M.E. ИТб-11д с оценками: 5,5,5,5,5
2. TheWorst H.E. ИТб-14д с оценками: 1,1,1,1,1
3. NextTo M.E. ИТб-13д с оценками: 5,5,5,4,4
4. BadOne H.E. ИТб-14д с оценками: 1,2,3,4,5

Под условия вывода подходят все студенты кроме второго, так как у него нет оценок 4 или 5. Отсортированный порядок вывода: 4,3,1;

Вывод

В ходе лабораторной работы были исследованы способы обработки логически связанных данных различных типов. Создана программа, использующая записи и типизированные файлы. Также были получены навыки по использованию записей для формирования логических структур.