1. Лабораторная работа № 7

«Программирование операций над записями и типизированными файлами»

* 1. Цель работы

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

## Постановка задачи

Изучить особенности работы записей и типизированных файлов.

* 1. Ход выполнения работы

Вариант выполнения задания – 19.

* + 1. Описание алгоритма решения задачи
       1. Определение входных и выходных данных

Первоначально программа принимает путь к файлу. Затем, если путь верен, программа ждёт, пока пользователь выберет действие, введя соответствующее действие.

Если пользователь выбирает ввод данных в файл, то программа принимает две строки (Название товара и Название магазина) и целое число (Стоимость товара). Если требуется завершить ввод, пользователь должен ввести «\*». Кроме подсказок, программа ничего не выводит на экран.

Если пользователь выбирает чтение данных из файла, то программа выводит на экран сохранённые в файле данные в виде таблицы. Ввод от пользователя не требуется.

Если пользователь выбирает вывод информации о товарах, продающихся в данном магазине, то программа ожидает ввода строки (Название магазина), а затем выводит требуемую информацию в виде таблицы.

* + - 1. Структурная схема алгоритма

Структурная схема программы представлена Рисунках 7.1, 7.2, 7.3.

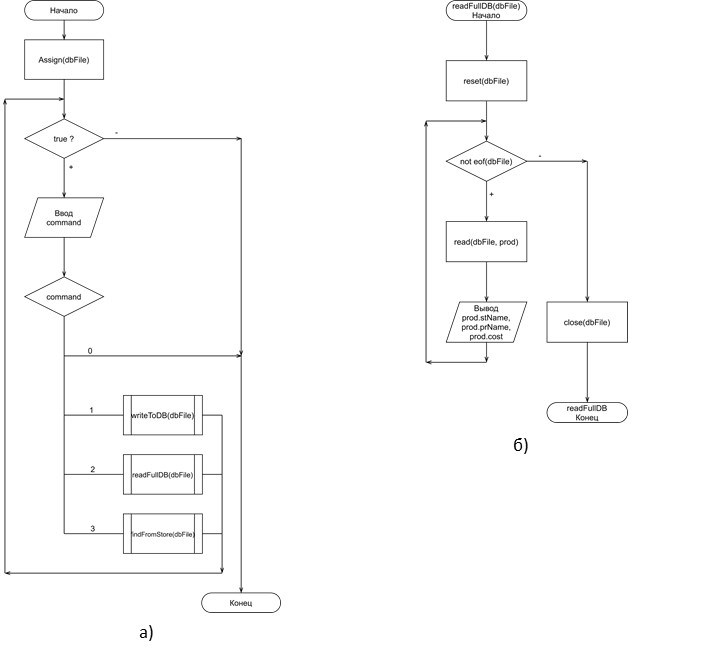


Рисунок 7.1 – Структурные схемы: а) основная программа;

б) процедуры readFullDB

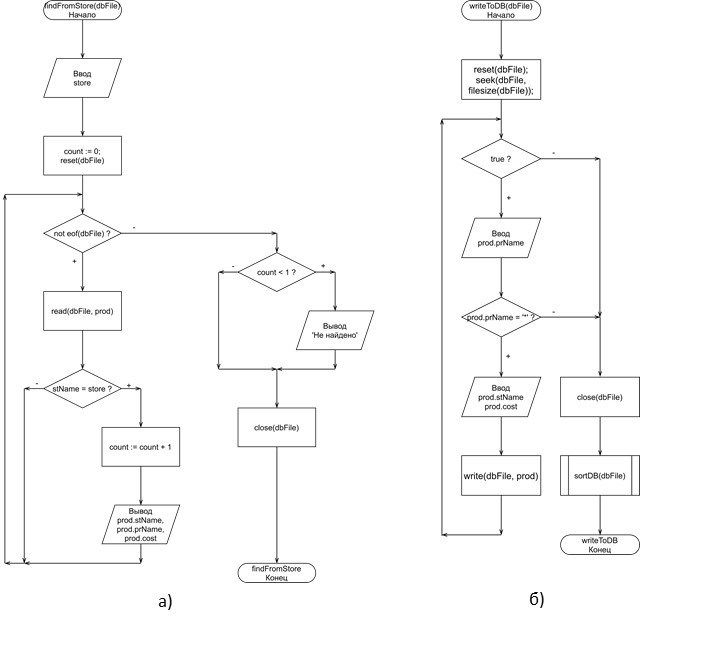


Рисунок 7.2 – Структурные схемы: а) процедуры findFromStore;

б) процедуры writeToDB

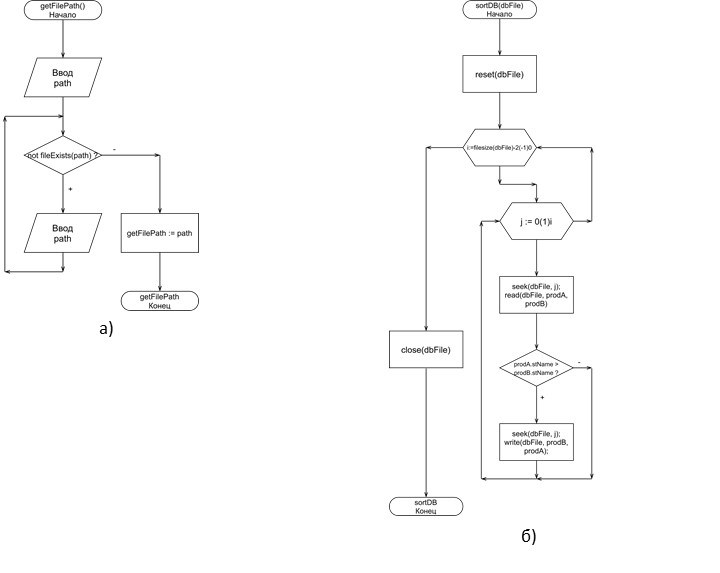


Рисунок 7.3 – Структурные схемы: а) функции getFilePath;

б) процедуры sortDB

* + 1. Текст программы

Пример реализации программы на языке программирования Pascal.

Program LW\_7;

Uses Crt;

Type tPrice = record

prName, stName: string[20]; cost: 1..maxint;

end;

tFile = file of tPrice;

tPath = string[255];

Var dbFile: tFile;

command: 0..3;

{Сортирует-базу-данных----------------------------------------------------}

procedure sordDB(var dbFile: tFile);

Var prodA, prodB: tPrice;

i, j: integer;

begin

reset(dbFile);

for i := filesize(dbFile) - 2 downto 0 do

begin

for j := 0 to i do

begin

seek(dbFile, j); read(dbFile, prodA, prodB);

if prodA.stName > prodB.stName then

begin

seek(dbFile, j);

write(dbFile, prodB, prodA);

end;

end;

end;

close(dbFile)

end;

{Запись-данных-в-файл------------------------------------------------------}

procedure writeToDB(var dbFile: tFile);

Var prod: tPrice;

begin

reset(dbFile);

seek(dbFile, filesize(dbFile));

with prod do

while true do

begin

clrscr();

writeln('Введите Наименование товара (\* для выхода):');

write(': '); readln(prName);

if prName = '\*' then break;

writeln('Введите Название магазина:');

write(': '); readln(stName);

writeln('Введите Цену товара:');

write(': '); readln(cost);

write(dbFile, prod);

end;

close(dbFile);

sordDB(dbFIle);

end;

{Чтение-всех-данных-из-файла--------------------------------------------------------}

procedure readFullDB(var dbFile: tFile);

Var prod: tPrice;

begin

clrscr;

reset(dbFile);

writeln('|', ' Название магазина ', '|', 'Наименование товара ', '|', 'Стоимость товара', '|');

while not eof(dbFile) do

begin

read(dbFile, prod);

with prod do

begin

writeln('|', stName:20, '|', prName:20, '|', cost:16, '|');

end;

end;

readkey;

close(dbFile);

end;

{Получает-от-пользователя-путь-к-файлу-----------------------------------------------}

function getFilePath(): tPath;

Var path: tPath;

begin

writeln(' Введите путь к файлу: ');

write(': '); readln(path);

while not fileExists(path) do

begin

writeln(' (\*) Ошибка: файл не найден.');

writeln(' Введите путь к файлу ещё раз:');

write(': '); readln(path)

end;

getFilePath := path

end;

{Вывод-товаров-данного-магазина-------------------------------------------------------}

procedure findFromStore(var dbFile: tFile);

Var count: integer; prod: tPrice; store: string[20];

begin

clrscr();

count := 0;

writeln(' Введите название магазина:');

write(': '); readln(store);

writeln('|', ' № ', '|', ' Название магазина ', '|', 'Наименование товара ', '|', 'Стоимость товара', '|');

reset(dbFile);

while not eof(dbFile) do

begin

read(dbFile, prod);

with prod do

if stName = store then

begin

count := count + 1;

writeln('|', count:4, '|', stName:20, '|', prName:20, '|', cost:16, '|');

end;

end;

if count < 1 then writeln('Магазин "', store, '" не найден.');

close(dbFile);

readkey;

end;

{Основная-программа-----------------------------------------------------------------------}

BEGIN

// Получение-пути-к-файлу

assign(dbFile, getFilePath() );

while true do

begin

clrscr();

writeln(' Меню ');

writeln(' 1 - ввод данных. ');

writeln(' 2 - вывод содержимого базы данных.');

writeln(' 3 - поиск по магазину. ');

writeln(' 0 - выход из программы. ');

write(': '); readln(command);

case (command) of

0: break;

1: writeToDB(dbFile);

2: readFullDB(dbFile);

3: findFromStore(dbFile);

end;

end;

// Конец-работы-программы

writeln();

writeln(' Выход из программы...');

readkey;

halt(0)

END.

* + 1. Описание тестовых примеров

Тестовые примеры представлены на Рисунках 7.4 – 7.10.

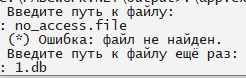


Рисунок 7.4 – Ввод неверного пути к файлу

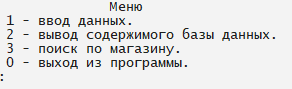


Рисунок 7.5 – Реализация меню

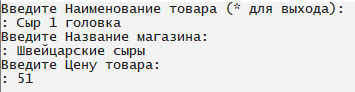


Рисунок 7.6 – Ввод товара



Рисунок 7.7 – Отмена ввода товара через «\*»

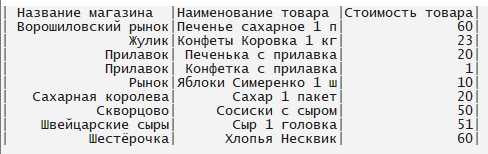


Рисунок 7.8 – Вывод содержимого файла

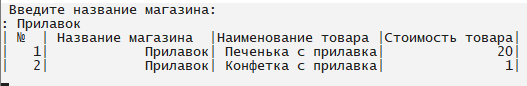


Рисунок 7.9 – Вывод товаров в данном магазине (магазин существует)

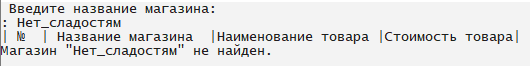


Рисунок 7.10 – Вывод товаров в данном магазине (магазин отсутствует)

Рисунок 7.4 показывает, что программа определяет, существует ли файл по введённому пути или нет. Это исключает ошибку открытия несуществующего файла.

Рисунок 7.5 показывает, что реализовано удобное меню.

Рисунки 7.6 и 7.7 показывают процесс ввода информации о товарах. Сначала был введён товар, он сохранён в файл. Затем введено «\*», тем самым ввод завершён.

На Рисунке 7.8 продемонстрировано содержимое файла в табличном виде. Видно, что данные отсортированы в алфавитном порядке по названиям магазинов. Также видно, что введённый в прошлом пункте товар тоже отсортирован.

На Рисунках 7.9 и 7.10 показана информация о товарах, продающихся в данном магазине. На Рисунке 7.9 показано, что введённый магазин существует в файле, а значит выведена информация о товарах. На Рисунке 7.10 показано, что введённый магазин в файле не существует, а значит выводится соответствующее сообщение.

## Вывод

В ходе данной лабораторной работы были получены навыки разработки приложений, умеющих работать с бинарными типизированными файлами: чтение данных из файла, запись в файл и сортировка этих данным по определённому критерию, поиск в бинарном файле с помощью процедуры Seek, получение размера файла с помощью FileSize. Также получены навыки работы с записями, использованию модуля Crt и встроенных функций, например, ClrScr, readkey.