**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

дистанционная форма обучения

Кафедра автоматизированных систем управления (АСУ)

Программирование

Курсовой проект

«телефонный справочник»

Дата выполнения работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### Дата проверки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

И.О.Фамилия преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Содержание:**

Содержание

Введение

Постановка задачи

Используемые методы

Описание алгоритма программы

Заключение

Список литературы

Приложение 1. Листинг программы

**Введение:**

Целью курсовой работы является углубление знаний и расширение навыков по разработке алгоритмов и их реализации на персональном компьютере.

**Постановка задачи:**

Необходимо написать программу, которая будет работать с базой телефонных номеров. При этом программа должна уметь добавлять, изменять, удалять, искать записи. Также, для упрощения работы с программой, необходимо создать интуитивно понятный интерфейс.

1. **Используемые методы:**

В программе используется следующий алгоритм работы с данными: все данные хранятся в файле phone.dat и загружаются в программу при вызове той или иной процедуры. При удалении или изменении записи создается временный файл с именем temp.dat, в него записываются все данные из файла phone.dat кроме тех которые были удалены или записываются уже измененные данные, затем файл phone.dat удаляется, а файл temp.dat переименовывается в phone.dat.

1. **Описание алгоритма программы.**

Создадим запись Phone, в этой записи создадим четыре поля с именами:

* SurnameS – строкового типа длинной 15 символов, данное поле будет содержать фамилию;
* NameS – строкового типа длинной 15 символов, данное поле будет содержать имя;
* PhoneS – строкового типа длинной 15 символов, данное поле будет содержать номер телефон.
* AddressS – строкового типа длинной 50 символов, данное поле будет содержать адрес.

Создадим переменные:

* PhoneF – файлового типа, для работы с файлами данных;
* PhoneR – произвольного типа (Phone), для работы с данными.
* а – массив от 1 до 600 типа Phone, данный массив будет необходим для организации сортировки записей.

Процедура WritePhone – данная процедура выводит все записи из файла phone.dat на экран. При помощи оператора Assign устанавливается связь между файлом и переменной файлового типа PhoneF. С помощью команды Reset открываем файл для чтения и переводим указатель в первую позицию. Далее создаем цикл размером от первой позиции в файле до последней, в этом цикле выводим на экран значение полей: фамилия, имя, номер телефона при помощи оператора Writeln. Закрываем файл при помощи оператора Close, это необходимо для того, чтобы другие процедуры могли с ним работать.

procedure WritePhone;

begin

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Все записи:');

writeln;

while not Eof(PhoneF) do

begin

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

read(PhoneF,PhoneR);

writeln;

write(PhoneR.SurnameS);

write(' ',PhoneR.NameS);

write('',PhoneR.PhoneS);

write('',PhoneR.AddressS);

end;

Close(PhoneF);

writeln;

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

Процедура AddPhone – данная процедура добавляет запись. С помощью операторов Assign и Reset открываем файл phone.dat и устанавливаем указатель в первую позицию. С помощью операторов Writeln и Readln создаем интерфейс для ввода фамилии, имени, номера телефона с клавиатуры и записываем введенные значения в файл. Закрываем файл.

procedure AddPhone;

begin

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

while not Eof(PhoneF) do Read(PhoneF,PhoneR);

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

ClrScr;

writeln('Добавление записи:');

writeln;

Write('Введите Фамилию:');

readln(PhoneR.SurnameS);

write('Введите Имя:');

readln(PhoneR.NameS);

write('Введите номер телефона:');

readln(PhoneR.PhoneS);

write('Введите адрес:');

readln(PhoneR.AddressS);

write(PhoneF,PhoneR);

Close(PhoneF);

writeln;

writeln('Запись успешно добавлена в базу!!!');

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

Процедура DeletePhone – удаляет запись. Для работы данной процедуры нам потребуются дополнительные переменные:

* temp – файлового типа, в данной переменной будут хранится измененные данные;
* Nazv – строкового типа длинной 15 символов, в этой переменной будет храниться значение поля фамилии записи которую необходимо удалить.

С помощью операторов Assign и Reset открываем файл phone.dat и устанавливаем указатель в первую позицию. С помощью операторов Assign и Reset создаем временный файл temp.dat и устанавливаем указатель в первую позицию. С помощью операторов Writeln и Readln создаем интерфейс для ввода фамилии (по фамилии мы будем искать в файле нужную нам запись) с клавиатуры, присваиваем введенное значение переменной Nazv. Далее в цикле при помощи условного оператора сравниваем поочередно значение поля SurnameS переменной PhoneF со значением переменной Nazv, если значение не совпадают то данная запись записывается в файл temp.dat, иначе пропускается. В результате имеем два файла с удаленной и не удаленной записью, закрываем оба файла. Удаляем файл phone.dat при помощи оператора Erase, а файл temp.dat переименовываем в phone.dat при помощи оператора Rename.

procedure DeletePhone;

var temp : file of Phone;

Nazv : string[15];

begin

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

Assign(temp,'Phone\temp.dat');

Rewrite(temp);

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

ClrScr;

writeln('Удаление записи:');

writeln;

Write('Введите Фамилию:');

readln(Nazv);

while not Eof(PhoneF) do

begin

Read(PhoneF,PhoneR);

if PhoneR.SurnameS<>Nazv then Write(temp,PhoneR);

end;

Close(temp);

Close(PhoneF);

Erase(PhoneF);

Rename(temp,'Phone\phone.dat');

writeln;

writeln('Запись успешно удалена из базы!!!');

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

Процедура EditPhone – редактирует запись. Работа данной процедуры аналогична работе процедуры DeletePhone. За исключением того, что во временный файл добавляется новая запись, полученная по средствам ввода с клавиатуры фамилии, имени, номера телефона.

procedure EditPhone;

var Nazv : string[15];

temp : file of Phone;

begin

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

Assign(temp,'Phone\temp.dat');

Rewrite(temp);

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

ClrScr;

writeln('Редактирование записи:');

writeln;

Write('Введите Фамилию, которую хотите изменить:');

readln(Nazv);

while not Eof(PhoneF) do

begin

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

Read(PhoneF,PhoneR);

if PhoneR.SurnameS<>Nazv then Write(temp,PhoneR);

end;

writeln;

write('Введите новую Фамилию:');

readln(PhoneR.SurnameS);

write('Введите новое Имя:');

readln(PhoneR.NameS);

write('Введите новый номер телефона:');

readln(PhoneR.PhoneS);

write('Введите новый адрес:');

readln(PhoneR.AddressS);

write(temp,PhoneR);

Close(temp);

Close(PhoneF);

Erase(PhoneF);

Rename(temp,'Phone\phone.dat');

writeln;

writeln('Запись успешно изменена!!!');

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

Процедура FindSurname – поиск по фамилии. Для работы данной процедуры нам потребуются дополнительные переменные:

* k – целого типа, в данной переменной будет хранится количество найденных записей;
* Nazv – строкового типа длинной 15 символов, в этой переменной будет храниться значение поля фамилии записи которую необходимо найти.

Переменной k присваиваем значение равное 0. С помощью операторов Assign и Reset открываем файл phone.dat и устанавливаем указатель в первую позицию. С помощью операторов Writeln и Readln создаем интерфейс для ввода фамилии (по фамилии мы будем искать в файле нужную нам запись) с клавиатуры, присваиваем введенное значение переменной Nazv. В цикле при помощи условного оператора сравниваем значение поля SurnameS со значением переменной Nazv и при совпадение выводим данную запись на экран, при каждом совпадении значение переменной k будет увеличиваться на единицу. Выводим значение переменной k на экран. Закрываем файл.

procedure FindSurname;

var Nazv : string[15];

k : integer;

begin

k:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Поиск записи по Фамилии:');

writeln;

Write('Введите Фамилию:');

readln(Nazv);

while not Eof(PhoneF) do

begin

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

read(PhoneF,PhoneR);

if PhoneR.SurnameS=Nazv then

begin

writeln;

write(PhoneR.SurnameS);

write(' ',PhoneR.NameS);

write('',PhoneR.PhoneS);

write('',PhoneR.AddressS);

k:=k+1;

end;

end;

Close(PhoneF);

writeln;

writeln;

writeln('Поиск завершен, найдено записей: ',k);

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

Процедура FindPhone – поиск по номеру телефона. Работа данной процедуры аналогична работе процедуры FindSurname, за одним лишь исключением: сравнение идет по полю PhoneS, а не по полю SurnameS.

procedure FindPhone;

var Nazv : string[15];

k : integer;

begin

k:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Поиск записи по номеру телефона:');

writeln;

Write('Введите номер телефона:');

readln(Nazv);

while not Eof(PhoneF) do

begin

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

read(PhoneF,PhoneR);

if PhoneR.PhoneS=Nazv then

begin

writeln;

write(PhoneR.SurnameS);

write(' ',PhoneR.NameS);

write('',PhoneR.PhoneS);

write('',PhoneR.AddressS);

k:=k+1;

end;

end;

Close(PhoneF);

writeln;

writeln;

writeln('Поиск завершен, найдено записей: ',k);

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

Процедура SortSurname – сортировка по фамилии. Данная процедура открывает файл phone.dat, далее в цикле переписывает все записи в массив а. Затем при помощи цикла и условного оператора сравнивает и сортирует записи в алфавитном порядке по полю Фамилия. Далее при помощи цикла выводит на экран уже отсортированные записи из массива.

procedure SortSurname;

var i,j,c: integer;

tr: Phone;

begin

c:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Сортировка по Фамилии:');

writeln;

while not Eof(PhoneF) do begin

inc(c,1);

read(PhoneF,a[c]);

end;

for i:=1 to c do

for j:=1 to c-1 do begin

if a[j].SurnameS>a[j+1].SurnameS then begin

tr:=a[j+1];

a[j+1]:=a[j];

a[j]:=tr;

end;

end;

for i:=1 to c do begin

writeln;

write(a[i].SurnameS);

write(' ',a[i].NameS);

write('',a[i].PhoneS);

write('',a[i].AddressS);

end;

Close(PhoneF);

writeln;

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

Процедуры SortAddress и SortPhone – работают аналогично процедуре SortSurname, только за одним исключением сравниваются и сортируются записи по полям адрес и номер телефона соответственно.

procedure SortAddress;

var i,j,c: integer;

tr: Phone;

begin

c:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Сортировка по адресу:');

writeln;

while not Eof(PhoneF) do begin

inc(c,1);

read(PhoneF,a[c]);

end;

for i:=1 to c do

for j:=1 to c-1 do begin

if a[j].AddressS>a[j+1].AddressS then begin

tr:=a[j+1];

a[j+1]:=a[j];

a[j]:=tr;

end;

end;

for i:=1 to c do begin

writeln;

write(a[i].SurnameS);

write(' ',a[i].NameS);

write('',a[i].PhoneS);

write('',a[i].AddressS);

end;

Close(PhoneF);

writeln;

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure SortPhone;

var i,j,c: integer;

tr: Phone;

begin

c:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Сортировка по адресу:');

writeln;

while not Eof(PhoneF) do begin

inc(c,1);

read(PhoneF,a[c]);

end;

for i:=1 to c do

for j:=1 to c-1 do begin

if a[j].PhoneS>a[j+1].PhoneS then begin

tr:=a[j+1];

a[j+1]:=a[j];

a[j]:=tr;

end;

end;

for i:=1 to c do begin

writeln;

write(a[i].SurnameS);

write(' ',a[i].NameS);

write('',a[i].PhoneS);

write('',a[i].AddressS);

end;

Close(PhoneF);

writeln;

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

Процедура SortType – подменю сортировки. Данная процедура при помощи операторов Writeln и Readln создает подменю сортировки, для удобства работы с программой конечного пользователя. В меню присутствуют следующие пункты:

* 1. Не сортировать – вызывает процедуру WritePhone;
* 2. Сортировать по Фамилии – вызывает процедуру SortSurname;
* 3. Сортировать по адресу – вызывает процедуру SortAddress;
* 4. Сортировать по номеру телефона – вызывает процедуру SortPhone;
* 5. Возврат в главное меню – возвращается к главному меню;

procedure SortType;

var b: integer;

begin

repeat

begin

ClrScr;

writeln('"Телефонный справочник" - меню сортировки');

writeln;

writeln('1. Не сортировать');

writeln('2. Сортировать по Фамилии');

writeln('3. Сортировать по адресу');

writeln('4. Сортировать по номеру телефона');

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

writeln('5. Возврат в главное меню');

writeln;

write('Выберите пункт меню: ');

readln(b);

case b of

1 : WritePhone;

2 : SortSurname;

3 : SortAddress;

4 : SortPhone;

end;

end;

until b=5;

end;

Процедура PhoneMenu – меню. Данная процедура при помощи операторов Writeln и Readln создает меню, для удобства работы с программой конечного пользователя. В меню присутствуют следующие пункты:

* 1. Показать все записи – вызывает процедуру WritePhone;
* 2. Добавить запись – вызывает процедуру AddPhone;
* 3. Удалить запись – вызывает процедуру DeletePhone;
* 4. Редактировать запись – вызывает процедуру EditPhone;
* 5. Поиск по Фамилии – вызывает процедуру FindSurname;
* 6. Поиск по номеру телефона – вызывает процедуру FindPhone;
* 7. Выход – завершает работу программы;

procedure PhoneMenu;

var b: integer;

begin

repeat

begin

ClrScr;

writeln('"Телефонный справочник" - Главное меню');

writeln;

writeln('1. Показать все записи');

writeln('2. Добавить запись');

writeln('3. Удалить запись');

writeln('4. Редактировать запись');

writeln('5. Поиск по Фамилии');

writeln('6. Поиск по номеру телефона');

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

writeln('7. Выход');

writeln;

write('Выберите пункт меню: ');

readln(b);

case b of

1 : SortType;

2 : AddPhone;

3 : DeletePhone;

4 : EditPhone;

5 : FindSurname;

6 : FindPhone;

end;

end;

until b=7;

end;

В основной программе происходит очистка экрана при помощи оператора ClrScr и вызывается процедура PhoneMenu.

begin

ClrScr;

PhoneMenu;

end.

**Заключение**

Данная программа продемонстрировала хорошую работоспособность. Имеет интуитивно понятный интерфейс и не должна вызывать затруднений в работе конечного пользователя. Для корректной работы программы файл phone.dat должен находиться в каталоге Phone, который в свою очередь должен находится в каталоге где установлен язык программирования Turbo Pascal.

**Список литературы.**

1. «Основы алгоритмизации и программирование», Е. Н. Сафьянова, учебное пособие, Томск, 2000 г.

**Приложение 1. Листинг программы.**

{Для корректной работы программы файл phone.dat

должен находиться в каталоге Phone, который в

свою очередь должен находится в каталоге где

установлен язык программирования Turbo Pascal}

program Phonebook;

uses Crt;

type Phone = record

SurnameS : string[15];

NameS : string[15];

PhoneS : string[15];

AddressS: string[50];

end;

var PhoneF : file of Phone;

PhoneR : Phone;

a : array[1..600] of Phone;

procedure WritePhone;

begin

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Все записи:');

writeln;

while not Eof(PhoneF) do

begin

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

read(PhoneF,PhoneR);

writeln;

write(PhoneR.SurnameS);

write(' ',PhoneR.NameS);

write('',PhoneR.PhoneS);

write('',PhoneR.AddressS);

end;

Close(PhoneF);

writeln;

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure AddPhone;

begin

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

while not Eof(PhoneF) do Read(PhoneF,PhoneR);

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

ClrScr;

writeln('Добавление записи:');

writeln;

Write('Введите Фамилию:');

readln(PhoneR.SurnameS);

write('Введите Имя:');

readln(PhoneR.NameS);

write('Введите номер телефона:');

readln(PhoneR.PhoneS);

write('Введите адрес:');

readln(PhoneR.AddressS);

write(PhoneF,PhoneR);

Close(PhoneF);

writeln;

writeln('Запись успешно добавлена в базу!!!');

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure DeletePhone;

var temp : file of Phone;

Nazv : string[15];

begin

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

Assign(temp,'Phone\temp.dat');

Rewrite(temp);

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

ClrScr;

writeln('Удаление записи:');

writeln;

Write('Введите Фамилию:');

readln(Nazv);

while not Eof(PhoneF) do

begin

Read(PhoneF,PhoneR);

if PhoneR.SurnameS<>Nazv then Write(temp,PhoneR);

end;

Close(temp);

Close(PhoneF);

Erase(PhoneF);

Rename(temp,'Phone\phone.dat');

writeln;

writeln('Запись успешно удалена из базы!!!');

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure EditPhone;

var Nazv : string[15];

temp : file of Phone;

begin

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

Assign(temp,'Phone\temp.dat');

Rewrite(temp);

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

ClrScr;

writeln('Редактирование записи:');

writeln;

Write('Введите Фамилию, которую хотите изменить:');

readln(Nazv);

while not Eof(PhoneF) do

begin

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

Read(PhoneF,PhoneR);

if PhoneR.SurnameS<>Nazv then Write(temp,PhoneR);

end;

writeln;

write('Введите новую Фамилию:');

readln(PhoneR.SurnameS);

write('Введите новое Имя:');

readln(PhoneR.NameS);

write('Введите новый номер телефона:');

readln(PhoneR.PhoneS);

write('Введите новый адрес:');

readln(PhoneR.AddressS);

write(temp,PhoneR);

Close(temp);

Close(PhoneF);

Erase(PhoneF);

Rename(temp,'Phone\phone.dat');

writeln;

writeln('Запись успешно изменена!!!');

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure FindSurname;

var Nazv : string[15];

k : integer;

begin

k:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Поиск записи по Фамилии:');

writeln;

Write('Введите Фамилию:');

readln(Nazv);

while not Eof(PhoneF) do

begin

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

read(PhoneF,PhoneR);

if PhoneR.SurnameS=Nazv then

begin

writeln;

write(PhoneR.SurnameS);

write(' ',PhoneR.NameS);

write('',PhoneR.PhoneS);

write('',PhoneR.AddressS);

k:=k+1;

end;

end;

Close(PhoneF);

writeln;

writeln;

writeln('Поиск завершен, найдено записей: ',k);

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure FindPhone;

var Nazv : string[15];

k : integer;

begin

k:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Поиск записи по номеру телефона:');

writeln;

Write('Введите номер телефона:');

readln(Nazv);

while not Eof(PhoneF) do

begin

PhoneR.SurnameS:='';

PhoneR.NameS:='';

PhoneR.PhoneS:='';

PhoneR.AddressS:='';

read(PhoneF,PhoneR);

if PhoneR.PhoneS=Nazv then

begin

writeln;

write(PhoneR.SurnameS);

write(' ',PhoneR.NameS);

write('',PhoneR.PhoneS);

write('',PhoneR.AddressS);

k:=k+1;

end;

end;

Close(PhoneF);

writeln;

writeln;

writeln('Поиск завершен, найдено записей: ',k);

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure SortSurname;

var i,j,c: integer;

tr: Phone;

begin

c:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Сортировка по Фамилии:');

writeln;

while not Eof(PhoneF) do begin

inc(c,1);

read(PhoneF,a[c]);

end;

for i:=1 to c do

for j:=1 to c-1 do begin

if a[j].SurnameS>a[j+1].SurnameS then begin

tr:=a[j+1];

a[j+1]:=a[j];

a[j]:=tr;

end;

end;

for i:=1 to c do begin

writeln;

write(a[i].SurnameS);

write(' ',a[i].NameS);

write('',a[i].PhoneS);

write('',a[i].AddressS);

end;

Close(PhoneF);

writeln;

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure SortAddress;

var i,j,c: integer;

tr: Phone;

begin

c:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Сортировка по адресу:');

writeln;

while not Eof(PhoneF) do begin

inc(c,1);

read(PhoneF,a[c]);

end;

for i:=1 to c do

for j:=1 to c-1 do begin

if a[j].AddressS>a[j+1].AddressS then begin

tr:=a[j+1];

a[j+1]:=a[j];

a[j]:=tr;

end;

end;

for i:=1 to c do begin

writeln;

write(a[i].SurnameS);

write(' ',a[i].NameS);

write('',a[i].PhoneS);

write('',a[i].AddressS);

end;

Close(PhoneF);

writeln;

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure SortPhone;

var i,j,c: integer;

tr: Phone;

begin

c:=0;

Assign(PhoneF,'Phone\phone.dat');

Reset(PhoneF);

ClrScr;

writeln('Сортировка по адресу:');

writeln;

while not Eof(PhoneF) do begin

inc(c,1);

read(PhoneF,a[c]);

end;

for i:=1 to c do

for j:=1 to c-1 do begin

if a[j].PhoneS>a[j+1].PhoneS then begin

tr:=a[j+1];

a[j+1]:=a[j];

a[j]:=tr;

end;

end;

for i:=1 to c do begin

writeln;

write(a[i].SurnameS);

write(' ',a[i].NameS);

write('',a[i].PhoneS);

write('',a[i].AddressS);

end;

Close(PhoneF);

writeln;

writeln;

write('Нажмите любую клавишу...');

ReadKey;

end;

procedure SortType;

var b: integer;

begin

repeat

begin

ClrScr;

writeln('"Телефонный справочник" - меню сортировки');

writeln;

writeln('1. Не сортировать');

writeln('2. Сортировать по Фамилии');

writeln('3. Сортировать по адресу');

writeln('4. Сортировать по номеру телефона');

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

writeln('5. Возврат в главное меню');

writeln;

write('Выберите пункт меню: ');

readln(b);

case b of

1 : WritePhone;

2 : SortSurname;

3 : SortAddress;

4 : SortPhone;

end;

end;

until b=5;

end;

procedure PhoneMenu;

var b: integer;

begin

repeat

begin

ClrScr;

writeln('"Телефонный справочник" - Главное меню');

writeln;

writeln('1. Показать все записи');

writeln('2. Добавить запись');

writeln('3. Удалить запись');

writeln('4. Редактировать запись');

writeln('5. Поиск по Фамилии');

writeln('6. Поиск по номеру телефона');

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

writeln('7. Выход');

writeln;

write('Выберите пункт меню: ');

readln(b);

case b of

1 : SortType;

2 : AddPhone;

3 : DeletePhone;

4 : EditPhone;

5 : FindSurname;

6 : FindPhone;

end;

end;

until b=7;

end;

begin

ClrScr;

PhoneMenu;

end.