ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет прикладной математики, информатики и механики

Кафедра программного обеспечения и администрирования

информационных систем

**Типизированные файлы**

Курсовая работа

По специальности 010503 – Математическое обеспечение и администрирования информационных систем

|  |  |
| --- | --- |
| Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | д. ф.-м. н., проф. Артемов М.А. |
| Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | %username% |
| Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | ст. преподаватель Вощинская Г.Э. |

Воронеж 2010

**Аннотация**

Объект исследования – файлы. Цель работы – научиться писать программы, которые используют типизированные и текстовые файлы для вывода результатов и как источник данных.

**Содержание**

[Введение 4](#_Toc263190903)

[1. Постановка задачи 6](#_Toc263190904)

[2. Анализ задачи 7](#_Toc263190905)

[2.1. Структура программы 7](#_Toc263190906)

[2.2. Модель представления данных 7](#_Toc263190907)

[3. Средства реализации 9](#_Toc263190908)

[4. Требования к программному и аппаратному обеспечению 10](#_Toc263190909)

[5. Интерфейс пользователя 11](#_Toc263190910)

[6. Реализация 14](#_Toc263190911)

[6.1. Структуры данных 14](#_Toc263190912)

[6.2. Структура программы 15](#_Toc263190913)

[6.3. Описание функциональных блоков 16](#_Toc263190914)

[7. План тестирования 18](#_Toc263190915)

[Заключение 19](#_Toc263190916)

[Список литературы 20](#_Toc263190917)

[Листинг программы 21](#_Toc263190918)

# Введение

Вся информация в нашей жизни хранится на разного рода носителях. В компьютере для этой цели существуют внешние запоминающие устройства, способные долгое время хранить различные данные в виде файлов. В языке Pascal под файлом понимается область памяти на внешнем запоминающем устройстве, способная хранить некоторую совокупность информации. В эту область внешней памяти можно как поместить определенные данные, так и извлечь их из нее. Эти действия имеют общее название ввод-вывод.

В современной жизни большое значение имеют базы данных, которые тоже являются файлами. Поэтому очень важно создать удобную оболочку для работы с ними. Особенно актуален вопрос создания программ для работы с большими базами данных, содержащими очень много записей и полей, потому что с каждым годом растет объем информации, обрабатываемой компьютерами.

Существуют основные процедуры и функции для работы с файлами:

1. AssignFile(f,FileName) – ассоциирование имени файла на диске с файловой переменной; необходимо выполнить до начала работы с файлом.
2. Rewrite(f) – открытие типизированного файла на запись, при таком открытии все содержимое файла стирается. Если файла с заданным именем не существует, создается пустой файл.
3. Reset(f) – открытие типизированного файла на чтение, не происходит уничтожение или изменение содержимого файла.
4. Seek(f, n) – процедура перемещает курсор на n – позицию.
5. FilePos(f):Longint - возвращает текущую позицию в файле.
6. Read(f,x) – чтение элемента x из файла f.
7. Write(f,x) – запись элемента x в файл f.
8. CloseFile(f) – закрытие файла. Применяется, когда программа завершает обработку файла.

# Постановка задачи

Разработать программу обработки файлов. Для этого необходимо разработать набор инструментальных и алгоритмических средств, позволяющих:

* Создать файл
* Открыть файл
* Добавить запись
* Удалить запись
* Изменить запись
* Отобразить содержимое файла
* Конвертировать типизированного файла в текстовый файл
* Конвертировать текстового файла в типизированный файл
* Занести в отдельный файл записи о студентах, имеющих не более одной четверки за последнюю сессию

# Анализ задачи

## Структура программы

Из главной программы посредством меню должны вызываться подпрограммы, выполняющие основные функции:

* Создание файла
* Открытие файла
* Добавление записи
* Удаление записи
* Изменение записи
* Отображение содержимого файла
* Конвертирование типизированного файла в текстовый файл
* Конвертирование текстового файла в типизированный файл
* Занесение в отдельный файл записи о студентах, имеющих не более одной четверки за последнюю сессию

Кроме них, должны быть реализованы вспомогательные функции:

* Получение слова из текстового файла с начала строки до двоеточия
* Печать информации о студенте на экран
* Ввод записи с клавиатуры

## Модель представления данных

В типизированных файлах, с которыми будет работать программа, используется следующая структура записи:

* ФИО(40 знаков)
* Номер курса(1 знак)
* Номер группы(1 знак)
* Успеваемость – 5 экзаменов в каждой из 10 сессий

Экзамен:

* Наименование предмета(10 знаков)
* Оценка (1 знак)

Записи упорядочены по номеру курса, внутри курса – по номеру группы, в группе – по ФИО.

В данной задаче в качестве структуры данных выбран типизированный файл. Компонентом файла является запись, так как используются данные различных типов: ФИО является строковым типом, номер курса и номер группы представляются в форме целочисленных данных.

Создание файла может быть реализовано с помощью ввода данных с клавиатуры.

# Средства реализации

При реализации проекта использовались следующие средства:

Интерактивная среда разработки – Borland Delphi 7.0

Язык разработки – Object Pascal

# Требования к программному и аппаратному обеспечению

Приложение предназначено для использования на IBM PC-совместимых компьютерах с операционной системой Windows 2000, XP.

Минимальные системные требования приложения:

* процессор Pentium I;
* 2 Мб оперативной памяти;
* 1 Мб дискового пространства для установки программы.

# Интерфейс пользователя

При запуске приложения пользователь видит главное окно программы, представленное на рис. 5.1.



Рис. 5.1. Главное окно программы

При выборе (1) программа запросит имя файла и создаст пустой файл с заданным именем. Если файл с таким именем уже существует, программа выдаст запрос на замену этого файла (рис. 5.2). Затем программа вновь выведет меню.

Рис. 5.2. Запрос на замену файла

При выборе пункта (2) будет предложено ввести имя файла, который надо открыть, после чего будет выведено сообщение об успешном результате открытия или об ошибке. Пункт (3) позволяет пользователю добавить запись в файл (рис. 5.3), (4) удаляет запись с заданным номером, (5) изменяет запись с заданным номером. Команда (6) выводит на экран содержимое всех записей из открытого файла (рис. 5.4).

Рис. 5.3. Ввод записи с клавиатурыРис 5.4. Отображение содержимого файла на экране

При выборе (7) в отдельный типизированный файл с заданным именем будут занесены те записи, у которых в последней сессии есть не более одной четверки. Команда (8) переносит все данные из открытого типизированного файла в новый текстовый, а (9) из заданного текстового файла переносит все данные в открытый типизированный файл.

# Реализация

## Структуры данных

Структура записей, из которых состоит файл:

student = record // Сведения о студенте

 FIO: string[40]; // фамилия, имя, отчество

 curs: 1..5; // курс

 group: 1..9; // группа

 results: Tsession; // сведения об экзаменах

 end;

Структура записи с результатами сессий:

Tsession = array[1..10,1..5] of record //Сведения об экзаменах

 name: string[10]; //название экзамена

 mark: 1..5; //оценка

 end;

TFile= file of student ; // файловая переменная, содержащая сведения о студентах

## Структура программы

Меню

Создание файла

Открытие файла

Добавление записи

Отображение содержимого файла

Изменение записи

Удаление записи

Занести в отдельный файл записи о студентах, имеющих не более одной четверки за последнюю сессию

Конвертирование типизированного файла в текстовый

Конвертирование текстового файла в типизированный

Чтение записи с клавиатуры

Печать записи на экран

Чтение из текстового файла слова до двоеточия

## Описание функциональных блоков

Read\_student(var st:student) – процедура чтения записи с клавиатуры.

write\_student(var st:student) – процедура печати записи на экран.

New\_file(var f:tfile; var name:string) – процедура создания нового пустого типизированного файла.

Open\_file(var f:tfile; var name:string) – процедура открытия существующего типизированного файла.

View\_file(var f:tfile; var name:string) – процедура отображения содержимого файла на экране.

Add\_rec(var f:tfile; var name:string) – процедура добавления новой записи в файл.

Del\_rec(var f:tfile; var name:string) – процедура удаления записи с заданным номером из файла.

Change\_rec(var f:tfile; var name:string) – процедура изменения записи с заданным номером.

Type\_text(var f:tfile; var name:string) – процедура конвертирования типизированного файла в текстовый. Формат получившегося текстового файла: на первой строке записаны фамилия, имя, отчество, и далее после двоеточия курс и группа через пробел. Каждая следующая строка, количество которых равно удвоенному номеру курса, содержит 5 фрагментов следующего вида: название экзамена: оценка.

get\_word(var f2:textfile):string – функция, возвращающая слово из текстового файла от текущего положения курсора до ближайшего двоеточия.

Text\_type(var f:tfile; var name:string) – процедура конвертирования текстового файла с заданным форматом в типизированный.

Task(var f:tfile; var name:string) – занесение в отдельный файл записей о студентах, имеющих не более одной четверки за последнюю сессию. Сначала определяется, отсутствуют ли оценки ниже четверки, а потом в случае положительного результата с помощью двух логических переменных определяется, единственная ли четверка или нет. Записи, отобранные таким образом, записываются в новый типизированный файл.

# План тестирования

Тестирование разработанного приложения состоит из следующих этапов:

1. Тестирование функциональности подпрограмм.

Проверяется корректность обработки всех действий, которые может потенциально выполнить пользователь.

1. Проверка пользовательского интерфейса.

Тест всех функций на корректность вывода информации о результате работы подпрограмм, в том числе и при некорректном вводе (рис. 7.1.)

Рис.7.1.Пример некорректного ввода данных

# Заключение

В результате была создана и оттестирована программа, работающая с типизированными файлами и реализующая следующие действия:

* Создание файла
* Открытие файла
* Добавление записи
* Удаление записи
* Изменение записи
* Отображение содержимого файла
* Конвертирование типизированного файла в текстовый файл
* Конвертирование текстового файла в типизированный файл
* Занесение в отдельный файл записи о студентах, имеющих не более одной четверки за последнюю сессию

# Список литературы

1. Алгоритмы и структуры данных./ Н.Вирт, Мир, 1989. – 356с.
2. Введение в язык Паскаль./ Абрамов В.Г. Трифонов Н.П., Трифонова Г.Н. — М.: Наука, 1988. — 320 с.
3. Программирование на языке Паскаль./ Ускова О. Ф.[и др.] ; О. Ф. Усковой – Питер, 2005.– 334 с.

# Приложение 1. Листинг приложения

uses SysUtils,windows;

type

Tsession = array[1..10,1..5] of record //Сведения об экзаменах

 name: string[10]; //название экзамена

 mark: 1..5; //оценка

 end;

 student = record // Сведения о студенте

 FIO: string[40]; // фамилия, имя, отчество

 curs: 1..5; // курс

 group: 1..9; // группа

 results: Tsession; // сведения об экзаменах

 end;

 TFile= file of student ; // файловая переменная, содержащая сведения о студентах

var k:integer;

 fname: string;

 f:TFile;

//--

procedure Read\_student(var st:student); // Ввод записи с клавиатуры

var i,j:integer; // i - номер сессии, j - номер экзамена

begin

with st do

 begin

 write('Введите ФИО: '); readln(FIO); // Ввод ФИО

 write('Введите курс: '); readln(curs); // Ввод курса

 write('Введите группу: '); readln(group); // ввод группы

 writeln('Введите результаты сессий:');

 i:=1; //номер первой сессии=1

 repeat

 writeln('Сейссия №',i,': '); // печать номера сессии

 for j:=1 to 5 do // Ввод всех экзаменов для сессии

 begin

 write('Название ',j,'-го экзамена: ');

 readln(results[i,j].name); //Вывод названия

 write('Оценка за ',j,'-й экзамен: ');

 readln(results[i,j].mark); //Вывод оценки

 end;

 inc(i);

 until (i=curs\*2+1); // условие прекращения ввода - для

 end; // данного курса больше сессий быть не может.

writeln('Ввод закончен!');

end;

//--

procedure write\_student(var st:student); // Печать записи

var i,j:integer;

begin

with st do

 begin

 writeln(FIO,', ',curs,' курс, ',group,' группа.'); // Печать информации о студенте

 for i:=1 to curs\*2 do // печать сессий

 begin

 write(' Сессия ',i,': ');

 for j:=1 to 5 do

 write(results[i,j].name,': ',results[i,j].Mark,', ');

 writeln;

 end;

 end;

end;

//--

procedure New\_file(var f:tfile; var name:string); // Создание пустого файла

var yn:char;

begin

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 write('Имя нового файла: ');

 readln(name); // Ввод имени

 if name='' then writeln('Имя файла пусто!') else // Проверка на непустоту введенного имени

 if fileexists(name) then // Проверка на существование файла

 begin

 write('Файл с таким именем существует, заменить? (y/n): ');

 readln(yn);

 if yn in ['y','Y','Н','н'] then

 begin // Если файл существует, можно его заменить

 assign(f,name);

 rewrite(f); // путем открытия файла для записи

 writeln('Файл "',name,'" создан!');

 closefile(f);

 end;

 end

 else

 begin

 assign(f,name);

 rewrite(f);

 writeln('Файл "',name,'" создан!');

 closefile(f);

 end;

end;

//--

procedure Open\_file(var f:tfile; var name:string); // Открытие типизированного файла

begin

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 write('Имя файла для открытия: ');

 readln(name); // ввод имени файла для открытия

 if (name='') then writeln('Имя файла пусто, поэтому файл не открыт.') //имя файла не может быть пустым

 else if not fileexists(name) then

 begin

 writeln('Такого файла нет!'); // сообщение в случае отсутствии файла

 name:='';

 end

 else

 begin

 assignfile(f,name);

 reset(f); // открытие для чтения

 closefile(f);

 writeln('Файл "',name,'" открыт!');

 end

end;

//--

procedure View\_file(var f:tfile; var name:string); // просмотр типизировнного файла

var i:integer;

 rec:student;

begin

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 if name='' then writeln('Никакой файл не открыт! Откройте сначала нужный файл!')

 else

 begin

 reset(f);

 if eof(f) then writeln('Файл пуст.') // если уже конец файла, значит, там ничего и не было

 else

 for i:=0 to Filesize(f)-1 do // проходим по всем записям

 begin

 read(f,rec); // считываем текущую запись во временную переменную

 write('№',i+1,' '); // печатаем номера записи

 write\_student(rec); // печать записи из временной переменной

 end;

 writeln('Вывод закончен!');

 end;

end;

//--

procedure Add\_rec(var f:tfile; var name:string); // добавление записи

var rec:student;

begin

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 if name='' then writeln('Никакой файл не открыт! Откройте сначала нужный файл!')

 else

 begin

 reset(f);

 seek(f,Filesize(f)); // установка курсора на конец

 Read\_student(rec); // Чтение записи с клавиатуры в переменную

 write(f,rec); // Добавление этой записи в файл

 writeln('Запись добавлена!'); // Сообщение о добавлении

 closefile(f); // Закрытие файла

 end;

end;

//--

procedure Del\_rec(var f:tfile; var name:string);

var n:integer;

 rec:student;

begin

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 if name='' then writeln('Никакой файл не открыт! Откройте сначала нужный файл!')

 else

 begin

 reset(f); // открытие файла для чтения

 write('Введите номер записи, которую надо удалить: ');

 readln(n);

 if n<1 then writeln('Неверный номер записи.') else // проверка на корректность ввода

 begin

 dec(n); //уменьшение номера на 1, т.к. счет записей начинается с 0

 write('удаляем... ');

 seek(f,FileSize(f)-1); //переход в конец файла

 read(f,rec); //чтение последней записи из файла

 seek(f,n); //переход на позицию удаляемой записи

 write(f,rec); //запись на место удаляемой записи временной переменной rec

 seek(f,Filesize(f)-1); //переход в конец файла

 Truncate(f); //обрезка файла - удаление последней записи

 writeln('удалено!'); //сообщение о результате

 closefile(f); //закрытие файла

 end;

 end;

end;

//--

procedure Change\_rec(var f:tfile; var name:string);

var n:integer;

 rec:student;

begin

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 if name='' then writeln('Никакой файл не открыт! Откройте сначала нужный файл!')

 else

 begin

 reset(f);

 write('Введите номер записи, которую надо изменить: ');

 readln(n); //считывание с клавиатуры номера изменяемой записи

 if n<1 then writeln('Неверный номер записи.') else

 begin

 dec(n); //уменьшение номера на 1, т.к. счет записей начинается с 0

 seek(f,n); //переход на позицию изменяемой записи

 writeln('Введите новые значения для записи ',n+1);

 read\_student(rec); //ввод новой записи с клавиатуры

 write(f,rec); //добавление новой записи в файл вместо записи N

 writeln('Запись изменена!');

 closefile(f);

 end;

 end;

end;

//--

procedure Type\_text(var f:tfile; var name:string); //Конвертирование типизированного файла в текстовый файл

var n,i,j:integer;

 rec:student;

 f2:textfile;

 name2,str:string;

begin

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 if name='' then writeln('Никакой файл не открыт! Откройте сначала нужный файл!')

 else

 begin

 reset(f);

 write('Введите имя текстового файла, в который нужно записать: ');

 readln(name2); //считывание имени текстового файла

 if name2='' then writeln('Имя файла пусто') else //проверка на пустоту

 begin

 assign(f2,name2);

 rewrite(f2); //открытие для записи текстового файла

 write('Типизированный файл открыт, обработка... ');

 for n:=0 to FileSize(f)-1 do //проходим по всем записям

 begin

 read(f,rec); //чтение записи из типизированного файла

 str:=rec.fio+': '+IntToStr(rec.curs)+' '+IntToStr(rec.group)+' '; //формирование строки м ФИО, курсом и группой

 writeln(f2,str); // запись этой строки ф текстовый файл

 for i:=1 to rec.curs\*2 do //для всех сессий

 begin

 for j:=1 to 5 do //для всех экзаменов в текущей сессии

 begin

 str:=rec.results[i,j].name+': '+IntToStr(rec.results[i,j].Mark)+' '; //формирование строки с названием предмета и оценкой

 write(f2,str); //запись этой строки в текстовый файл

 end;

 writeln(f2); //переход на новую строку для записи следующей сессии

 end;

 end;

 writeln('Выполнено!');

 closefile(f2);

 closefile(f);

 end;

 end;

end;

//--

function get\_word(var f2:textfile):string; //Получение слова из текстового файла с начала строки до двоеточия

var ch:char;

begin

 result:='';

 read(f2,ch); //чтение первого символа

 while ch<>' //пока считанный символ - не двоеточие

 begin

 result:=result+ch; //прибавляем к результату этой символ

 read(f2,ch); //и считываем следующий символ

 end;

end;

//--

procedure Text\_type(var f:tfile; var name:string); //Конвертирование текстового файла в типизированный файл

var i,j:integer;

 rec:student;

 f2:textfile;

 name2:string;

 ch:char;

begin

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 if name='' then writeln('Никакой файл не открыт! Откройте сначала нужный файл!')

 else

 begin

 reset(f); //открытие типизированного файла

 write('Введите имя текстового файла, из которого надо загрузить: ');

 readln(name2); //ввод имени текстового файла

 if name2='' then writeln('Имя файла пусто') else

 if not fileexists(name2) then writeln('Такого файла не существует!') else

 begin

 assignfile(f2,name2);

 reset(f2); //открытие текстового файла для чтения

 write('Текстовый файл открыт, обработка... ');

 while not eof(f2) do

 begin

 rec.FIO:=copy(get\_word(f2),1,40); // получение ФИО

 read(f2,rec.curs); readln(f2,rec.group); //Получение курса и группы - следующие два числа

 for i:=1 to rec.curs\*2 do //для каждой строки (то есть сессии)

 begin

 for j:=1 to 5 do //для всех экзаменов

 begin

 rec.results[i,j].name:=get\_word(f2); //получение названия предмета

 read(f2,rec.results[i,j].Mark); //получение оценки

 end;

 readln(f2); //переход к следующей строке, т.е. сессии

 end;

 write(f,rec); //запись временной переменной в типизированный файл

 end;

 writeln('Выполнено!');

 closefile(f2);

 closefile(f);

 end;

 end;

end;

//--

procedure Task(var f:tfile; var name:string); //Занесение в отдельный файл записей о студентах, имеющих не более одной четверки за последнюю сессию

var n,i:integer;

 rec:student;

 f2:tfile;

 b,c,d:boolean;

 name2:string;

begin

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 if name='' then writeln('Никакой файл не открыт! Откройте сначала нужный файл!')

 else

 begin

 write('Введите имя файла для результатов: '); readln(name2); //ввод имени файла, в котором будет результат

 if name2='' then writeln('Ошибка! Имя пусто!')

 else

 begin

 reset(f); //открытие для чтения исходного файла

 assign(f2,name2);

 rewrite(f2); //открытие для добавления результирующего файла

 writeln('Файл "',name2,'" создан!');

 write('Ищем записи, удовлетворяющие условию... ');

 for n:=0 to Filesize(f)-1 do //просматриваем каждую запись в исходном файле

 begin

 b:=true; c:=true; i:=1; d:=true; //инициализация переменных

 seek(f,n); //переходим к текущей записи

 read(f,rec); //считываем ее в переменную

 while (i<=5) and d do //проходим по всем экзаменам из последней сессии

 begin

 if (rec.results[rec.curs\*2,i].mark<4) then d:=false; //d показывает, отсутствуют ли оценки <4 или нет

 inc(i); //переход к следующему экзамену

 end;

 i:=1;

 if d then //если хорошист или отличник, то

 begin

 while c and (i<=5) do //проверяем все экзамены

 begin

 if rec.results[rec.curs\*2,i].mark=4 then //если найдена четверка, то

 if b=true then b:=false else c:=false; //если она первая, то меняем флажок b, иначе помечаем запись как непригодную

 inc(i); //переходим к следующей записи

 end;

 if c then //если запись подходит к условию

 begin

 writeln('Найдена запись ',n+1);

 write(f2,rec); //записываем ее

 end;

 end;

 end;

 closefile(f);

 closefile(f2);

 end;

 end;

end;

//--

begin

SetConsoleCP(1251); //Установка кодировки ввода

SetConsoleOutputCP(1251); //Установка кодировки вывода

fname:='';

 repeat //вывод меню

 writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

 writeln('Меню:');

 writeln('1 - Создать файл');

 writeln('2 - Открыть файл ');

 writeln('3 - Добавить запись');

 writeln('4 - Удалить запись');

 writeln('5 - Изменить запись');

 writeln('6 - Отобразить содержимое файла');

 writeln('7 - Занести в отдельный файл записи о студентах, имеющих не более одной четверки за последнюю сессию');

 writeln('8 - Из типизированного файла в текстовый');

 writeln('9 - Из текстового файла в типизированный');

 writeln('0 - Выход');

 write('Ваш выбор: ');

 repeat

 readln(k); //ввод номера пункта

 if not k in [0..9] then writeln('Некорректный ввод'); //проверка на корректность

 until k in [0..9];

 case k of //вызов соответствующих подпрограмм

 1: New\_file(f,fname);

 2: Open\_file(f,fname);

 3: Add\_rec(f,fname);

 4: Del\_rec(f,fname);

 5: Change\_rec(f,fname);

 6: View\_file(f,fname);

 7: Task(f,fname);

 8: Type\_Text(f,fname);

 9: Text\_Type(f,fname);

 end;

 until k=0; //выход из программы

end.