



Лекция

**Структура данных - файлы (понятие, свойства, описание, операции, типы).
Типизированные и текстовые файлы**

Файл

это совокупность связанных записей, рассматриваемых как единое целое и хранящихся на внешнем носителе, предназначенный для:


- хранения данных, превосходящих по объему внутреннюю память ПК;
- долговременного хранения данных.

Свойства:

1. Линейная
2. Фиксированного размера
3. С произвольным/последовательным доступом к элементам
4. Однородная

Файл может быть:

- открыт для чтения и при этом закрыт для записи;
- открыт для записи и закрыт для чтения;
- открыт для записи и для чтения;
- закрыт для записи и для чтения.



Указатель на файловую переменную

это указатель на информацию, которая определяет различные стороны файла: имя, статус, текущую позицию

Графическое представление

указатель



| | | | | |
|----------|----------|----------|-------|----------|
| запись 0 | запись 1 | запись 2 | | запись n |
|----------|----------|----------|-------|----------|

Операции над файлами

- установочные и завершающие операции;
- ВВОД / ВЫВОД;
- перемещения по файлу

Установочные и завершающие операции

| Функция | Действие функции |
|--------------------------|---|
| assign(f, 'путь') | связывает логическое имя файла с физическим |
| reset (f) | открывает файл для чтения, при выполнении этой процедуры указатель перемещается в начало |
| rewrite (f) | открывает файл для записи, при выполнении процедуры указатель перемещается в начало, стирая все на своем пути |
| close(f) | закрывает файл |
| IOResult | контролирует ошибку ввода/вывода, если ошибки не было, возвращается ноль, в противном случае запоминает код ошибки (опция проверки ввода/вывода должен быть выключена ({I-})). |

Ввод / вывод

| Функция | Действие функции |
|---|--|
| read(f , <список ввода>) | читает данные из типизированного файла |
| write(f , <список вывода>) | записывает данные в типизированный файла |

Перемещения по файлу

| Функция | Действие функции |
|-------------------|---|
| eof(f) | возвращает значение "истинно", если достигнут конец файла |
| seek(f, n) | перемещает указатель файла на запись с номером N |

Типы файлов:

Типизированный файл

<имя>= file of <тип компонент>;

Текстовый файл

<имя>= text;

Не типизированный файл

<имя>= file;

Типизированный файл

Пусть задано следующее
описание:

f : file of real;

number : real;

Запись данных в файл

```
assign(f, 'real_num');  
rewrite (f); {откроем для записи}  
number:=1;  
while number <>0 do  
    begin  
        readln(number);  
        if number<>0 then write(f, number);  
    end;
```

Чтение данных из файла

```
assign(f, 'real_num');  
reset(f);  
writeln('Чтение данных');  
while NOT EOF(f) do  
    begin  
        read(f, number);  
        writeln('Очередное число: ', number:11:2);  
    end;
```

Задача:

Написать программу создания файла и вывода на печать, содержащего информацию о производительности труда рабочего в течение года.

Информацию представить в виде: название месяца, наименование трех деталей и производительность.

CONST

IZD=3; {количество наименований изделий}

N=12; {количество месяцев}

TYPE

work= record

Month : string[10];

Prod :array [1..IZD] of record

Name: string[20];

proizv: integer

End

End;

file_work=file of work;

VAR

f : file_work;

dbase : string; {имя файла}


```

PROCEDURE vvod(ff: file_work);
VAR   i, j   :integer;
      sved   : work;
BEGIN
for i:=1 to N do
  begin
  with sved do
  begin
    write('Введите название месяца');
    readln(month);
    for j:=1 to IZD do
      with prod[j] do
        begin
          writeln('Введите название изделия и производительность');
          readln(name, proizv)
        end
      end;
    end;
  write(ff, sved)
  end;
END;

```

```

PROCEDURE vivod(ff: file_work);
VAR   i, j       :integer;
      sved       : work;
BEGIN
for i:=1 to N do begin
      read(ff, sved);
      with sved do begin
            write(month, ' ');
            for j:=1 to IZD do
                  with prod[j] do
                        writeln('          ', name, ' ', proizv);
                        writeln;
                        end;
            end;
      end;
END;

```

BEGIN

```
dbase:='a:\\report.db';
```

```
assign(f, dbase);
```

```
{$I-}
```

```
rewrite(f);
```

```
{$I+}
```

```
if IOResult <> 0 then writeln('Ошибка создания файла!')
```

```
    else begin
```

```
        vvod(f);
```

```
        close(f);
```

```
        reset (f)
```

```
        vivod(f);
```

```
        close(f);
```

```
    end;
```

END.

**Текстовый файл - это
совокупность строк различной
длины**

Описание текстового файла:

`<имя>= text;`

Файл может быть:

- открыт для чтения и при этом закрыт для записи;
- открыт для записи и закрыт для чтения.

Установочные и завершающие операции:

- **Assign(f), Reset(f), Rewrite(f), Close(f);**
- **APPEND(f)** - открывает файл для добавления записей в конец файла.

Операции связанные с вводом\выводом

- **READLN(f, < список ввода >)** - читает данные из текстового файла;
- **WRITELN(f, < список ввода >)** - записывает данные в текстовый файла;
- **EoLn(f)** - проверка, достигнут ли конец текущей строки, (возвращает логическое значение true, если текущая строка исчерпана, false в противном случае);
- **SEEKEOF(f)** - осуществляет поиск признака конца файла (возвращает логическое значение true, если файл исчерпан, false в противном случае).

Пример 1:

{фрагмент программного кода}

```
Var a, b : string;  
.....  
read(f, a);  
readln(f);  
read(f, b);  
.....
```

Пусть задан текстовый файл:

Ффффффффф

Аааааааааааа

a:= Ффффффффф

b:=Аааааааааааа

Пример 2:

{фрагмент программного кода}

```
Var a, b:string;  
.....  
read(f, a);  
readln(f, b);  
.....
```

Пусть задан текстовый файл:

Ффффффффф

Аааааааааааа

a:= Ффффффффф

b:=""

Т.к. readln переводит указатель только после того, как отработал.

Пример 3:

```
Var a, b, c : string;
```

```
.....
```

```
begin
```

```
.....
```

```
read(f, a);
```

```
readln(f, b);
```

```
read(f, c);
```

```
.....
```

```
End;
```

Пусть задан текстовый файл:

Ффффффффф

Аааааааааааааа

Nnnnnnnnnnnnnnnnn

a:= Ффффффффф

b:=''

c:= Аааааааааааааа

Чтение данных из файла посимвольно

```
var
```

```
  f : text;
```

```
  ch : char;
```

```
begin
```

```
  assign(f, 'f:\texttt.txt');
```

```
  {$I-}
```

```
  reset(f);
```

```
  {$I+}
```

```
  if IOResult <> 0 then writeln('ошибка открытия')
```

```
    else while not (SEEKEOF(f)) do begin
```

```
      read(f, ch);
```

```
      write(ch, ' ');
```

```
    end;
```

```
end.
```

Чтение данных из файла построчно

```
var
  f : text;
  ch : string;

begin
  assign(f,'f:\texttt.txt');
  {$I-}
  reset(f);
  {$I+}
  if IOResult <> 0 then writeln('ошибка открытия')
    else while not (SEEKEOF(f)) do begin
      readln(f, ch);
      writeln(ch);
    end;
end.
```

Задача № 1. Дан символьный файл f. Напишите программу подсчета количества букв 'т' в заданном файле. Выведите на экран соответствующие сообщения, если таких букв нет, если их количество превышает некоторое заданное число или не превышает.

Входные данные

Текстовый файл – f.

Заданное число – ch (тип - целый).

Выходные данные

Сообщение в соответствии с условием задачи.

Алгоритм

1. Ввести число.
2. Подсчитать количество букв 'т' в файле.
3. Сравнить количество букв 'т' с заданным числом и выдать соответствующее сообщение.

Основная программа

```
uses obr_stroki;  
type  
    tx = text;  
var  
    f          : tx;  
    stkol, ch  : word;  
begin  
    writeln('введите число');  
    readln(ch);  
    stkol:=koll(f);  
    rezul(stkol, ch);  
end.
```

```

function koll (var st : tx):word;
var
    k, chl, i   : word;
    txtt       : string;
begin
    {$I-}
    reset(st);
    {$I+}
    if IOResult <> 0 then writeln('ошибка открытия файла')
        else begin
            k:=0;
            while NOT (SEEKEOF(st)) do begin
                readln(st, txtt);
                chl:=length(txtt);
                i:=1;
                while i<=chl do begin
                    if txtt[i]='т' then k:=k+1;
                    i:=i+1;
                end;
            end;
            koll:=k;
            close(st);
        end;
end;

```

procedure rezul (stkoll, chh : word);

begin

if stkoll=0 then writeln('в файле нет букв 'Т' ')

else if stkoll>chh then writeln('букв 'Т' в файле больше, чем заданное число')

else if stkoll=chh then writeln('равное количество')

else writeln('букв 'Т' в файле меньше, чем
заданное число')

end;

begin

assign(st, 'F:\fttt.txt');

end.

Задача № 2. Дан символьный файл f. Преобразовать его следующим образом: если нет символа '*', то оставить его без изменений, иначе заменить каждый символ '*', встречающийся после первого его вхождения, на символ '-'. Результат записать в файл g.

Входные данные.

Текстовый файл – f.

Выходные данные.

Текстовый файл – g.

Алгоритм.

1. Поиск первого символа '*' в исходном файле (f), алгоритм поиска по ключу.
2. Замена каждого символа '*', после первого его вхождения, на символ '-'.
 - 2.1. считать строку из исходного файла;
 - 2.2. каждый символ считанной строки проверить, если символа '*', удалить и на его место вставить символ '-', перейти к следующему элементу строки;
 - 2.3. преобразованную строку записать в новый файл (g).

Описание данных

type

```
tx = text;
```

var

```
f, g      : tx;
```

```
fl       : boolean;
```

```
num      : word;
```

Основная программа

```
begin {main}
  assign(f, 'F:\fttt.txt');
  {$I-}
  reset(f);
  {$I+}
  if IOResult <> 0 then writeln('ошибка открытия файла')
    else begin
      poisk(fl, num);
      if fl then writeln('нет *')
        else begin
          assign(g, 'F:\gttt.txt');
          {$I-}
          rewrite(g);
          {$I+}
          if IOResult <> 0 then writeln('ошибка
открытия файла')
            else
              zamena(num)
        end;
      end;
    end;
end.
```

```

procedure poisk(var flag : boolean; var numm : word);
var  txtt  :string;
      i, j   :word;
begin
    j:=1;
    flag:=true;
    while NOT (SEEKEOF(f)) and flag do
      begin
        readln(f, txtt);
        i:=1;
        while flag and (i<=length(txtt)) do
          if txtt[i]= '*' then begin
            flag:=false;
            numm:=j;
          end
          else begin
            i:=i+1;
            j:=j+1
          end
        end;
      end;
    close(f);
end;

```

```

procedure zamena(numm : word);
var
    i, j      : word;
    txtt     : string;
begin
    reset(f);
    j:=1;
    while NOT (SEEKEOF(f)) do
        begin
            readln(f, txtt);
            i:=1;
            while i<=length(txtt) do begin
                if (txtt[i]='*') and (j<>numm) then
                    begin
                        delete(txtt,i,1);
                        insert('-', txtt, i);
                    end;
                i:=i+1;
                j:=j+1
            end;
            writeln(g, txtt);
        end;
    close(f);
    close(g);
end;

```