

Fișiere Pascal

Fișiere text

Material didactic pentru Informatică
(În corespondență cu curriculum-ul la Informatică)
Clasa a X-a

Obiectivele lecției:

- O1** - să explice noțiunile: fișier text, linie, lungimea liniei, sfârșit de linie, sfârșit de fișier;
- O1** - să enumere procedurile destinate prelucrării fișierelor text;
- O1** - să explice destinația fișierelor standard de intrare și ieșire;
- O1** - să demonstreze înțelegerea metodelor de prelucrare a fișierelor text prin explorarea caracter cu caracter și prin utilizarea specificatorilor de format;
- O1** - să poată crea și prelucra datele din fișierele text.

Fișiere text

Se numește **fișier text** o succesiune de caractere ASCII scrise în una sau mai multe linii și care nu au în mod obligatoriu lungimi egale.

Fișierele text au o importanță deosebită din două motive:

- Sunt direct afișabile;
- Orice editor de texte lucrează cu fișiere de acest tip.

Fișiere text

Fișierele text se declară ca variabilă de tip predefinit:

VAR a, b: text;

Pentru a face legătura între numele fișierului din program și numele său de pe suportul extern pe care se află sau se creează, se folosește procedura **ASSIGN**. Apelul procedurii se realizează prin:

ASSIGN(nume fișier în program, 'calea + nume fișier pe suport extern')

Fișiere text: Example

Var a: text;

.....

Assign(a, 'fisier1.txt');

În acest caz, fișierul va fi creat sau se va găsi în directorul în care ne aflăm și se numește fisier1.txt. O formă echivalentă pentru o asignare de acest tip este:

Var a: text; b: string;

.....

b:= 'fisier1.txt';

Assign(a, b);

Fișiere text: Example

O altă formă de asignare a unui fișier în cazul în care el se află într-un alt directoriu, este:

```
Var a: text;
```

```
.....
```

```
Assign(a, 'D:\clasa10\fisier1.txt');
```

În acest caz, fișierul se găsește pe discul **D** în directorul **clasa10** și se numește **fisier1.txt**

Crearea fișierelor text

Procedura care deschide un fișier pentru a fi creat este **REWRITE**. Forma generală este:

Rewrite(variabila fișier).

Scrierea datelor în fișier cu ajutorul procedurilor **WRITE** și **WRITELN**. Forma generală este:

Writeln(variabila fisier, variabila citită).

După ce a fost creat, fișierul trebuie închis utilizând procedura **CLOSE**. Forma generală este:

Close(ume fisier).

Crearea fișierelor text

```
Program Creare_fisier;  
var f: text; a: string;  
begin  
    assign(f, 'fisier.txt');  
    rewrite(f); {deschide fisierul pentru scriere}  
    repeat  
        readln(a); {Introducerea datelor de la tastatură}  
        if a<>'stop' then  
            writeln(f, a); {datele se scriu in fisierul f }  
    until a='stop';  
    close(f)  
end.
```


Citirea datelor din fișiere text

Pentru a citi un fișier text sunt necesare următoarele proceduri: **RESET**, **READ**, **READLN**.

Procedura **RESET** deschide un fișier pentru citire și are forma generală:

Reset(variabila fisier).

Pentru procedurile **READ** și **READLN** avem forma simplificată:

Read(var de tip fișier, variabilă de tip string);

Readln(var de tip fișier, variabilă de tip string).

Se citește maxim 256 de caractere din linia curentă a fișierului.

Citirea datelor din fișiere text

```
Program Citire_fisier;  
var f: text;  a: string;  
begin  
  assign(f, 'fisier.txt');  
  reset(f); {pregătește fisierul f pentru citire }  
  while not eof(f) do  
    begin  
      readln(f, a);  
      writeln(a);  
    end;  
  close(f)  
end.
```

{cât timp nu este sfârșit de
fisier (**eof(f)**) se execută
instrucțiunile aflate între
begin și end}

Procedura APPEND

Procedura **APPEND** are rolul de a deschide un fișier text care a fost creat pentru extindere (scriere la sfârșit de fișier). Forma generală este:

Append (variabila fișier).

În situația în care fișierul nu există, se generează o eroare de intrare-ieșire. Pentru un fișier deschis cu Append este posibilă numai operația de scriere.

Procedura APPEND: exemplu

```
Program Extindere_fisier;  
var f: text;  a: string;  
begin  
    assign(f, 'fisier.txt');  
    append(f);  
    repeat  
        readln(a);  
        if a<>'stop' then writeln(f, a);  
        until a='stop';  
    close(f)  
end.
```

Funcția EOLN: exemplu

Funcția **EOLN** are ca parametru formal o variabilă fișier și este de tip boolean. La execuție, funcția întoarce valoarea **TRUE** dacă pointer-ul este la sfârșit de linie și **FALSE** în caz contrar.

Funcția EOLN: exemplu

```
Program Functie_EOLN;  
var f: text;  a: string;  
begin  
    assign(f, 'fisier.txt');  
    reset(f);  
    while not eoln(f) do  
        begin  
            readln(f, a); writeln(a);  
        end;  
    close(f)  
end.
```

Fișierele INPUT și OUTPUT

Aceste fișiere se consideră declarate (nu mai este necesar declararea lor ca fișiere text). Ele se asignează în mod automat la tastatură (**INPUT**) și monitor (**OUTPUT**). În situația în care procedurile **READ**, **READLN**, **WRITE**, **WRITELN** nu au precizat la apel variabila fișier, aceasta este considerată automat **INPUT** pentru procedurile de citire și **OUTPUT** pentru procedurile de scriere.

Fișierele INPUT și OUTPUT

Problemă

Scrieți un program care calculează a^n , unde $n \in \mathbb{Z}$ și $a \in \mathbb{R}$. n și a sunt citite din fișierul ***date.in***, iar rezultatul este înscris în fișierul text ***date.out***

Intrare: Fișierul ***date.in***

Ieșire: Fișierul ***date.out***

Fișierele INPUT și OUTPUT

```
Program Calcul_Putere;
```

```
Var fin, fout:Text;
```

```
    a, putere, p:real;
```

```
    n:Integer;
```

```
Begin
```

```
    assign(fin, 'date.in');
```

```
    reset(fin);
```

```
    assign(fout, 'date.out');
```

```
    rewrite(fout);
```

```
    readln(fin, n, a);
```

```
    p:=1;
```

```
    if n<0 then begin
```

```
        for var i:=1 to -n do p:=p*a;
```

```
        putere:=1/p;
```

```
    end;
```

```
    putere:=1;
```

```
    if n>0 then
```

```
        for var i:=1 to n do
```

```
            putere:=putere*a;
```

```
    if n=0 then putere:=1;
```

```
    write(fout, putere);
```

```
    Close(fin); Close(fout);
```

```
End.
```

Concluzii

În concluzie, prezentăm ordinea în care trebuie apelate procedurile destinate prelucrării datelor de tip *fișier*:

1. **Assign(f, s)** – asocierea fișierului **f** cu fișierul extern **s**;
2. **Reset(f)** – deschiderea fișierului **f** pentru citire;
Rewrite(f) – deschiderea fișierului **f** pentru scriere;
Apend(f) – deschiderea fișierului **f** pentru extindere(adăugare date la fișierul existent);
3. **Read(f, v)** – citirea unei componente a fișierului **f**;
Write(f, e) – scrierea unei componente a fișierului **f**;
4. **Close(f)** – închiderea fișierului **f**.

Concluzii

După închiderea fișierului, variabile **f** poate fi asociată cu un alt fișier extern.

Întrucât valorile variabilelor de tip fișier se păstrează pe suporturile externe de informație, **atribuirile de fișiere sunt interzise.**