

PLIKI TEKSTOWE

var

```
plik_liczb : file of integer;  
plik_znakow : file of char;  
plik_linii : text; ← deklaracja zmiennej plikowej
```

Przykład sekwencji bajtów:

```
54 55 52 42 4F 20 50 41 53 43 41 4C 20 37 2E 30 0D 0A 42 6F 72 6C 61 6E 64 1A  
T U R B O _ P A S C A L _ 7 . 0           B o r l a n d
```

Interpretacja jako plik tekstowy:

```
TURBO PASCAL 7.0  
Borland
```

Znaki sterujące:

#10	0A	(NL = New Line)	← przejście do nowej linii
#13	0D	(CR = Carriage Return)	← przejście na początek linii
#26	1A		← koniec pliku

Procedury i funkcje dla plików tekstowych

```
assign  
reset                rewrite                append  
Eof = SeekEof      EoLn = SeekEoLn  
read                 readln  
write                writeln  
filesize             filepos                seek        truncate  
close
```

{Przykład programu zliczającego ilości linii w pliku}

```
var
  Plik      : Text;
  Ile_Linii : Word;
Begin
  Ile_Linii:=0;
  Assign(Plik,'opis.txt');
  Reset(Plik);
  While not Eof(Plik) do
    begin
      Inc(Ile_Linii);
      Readln(Plik);
    end;
  Writeln('W pliku jest ', Ile_linii:4,' linijek tekstu');
  Close(Plik);
End.
```

{Przykład programu zliczającego ilości znaków w liniach i ilości linii w pliku}

```
var
  Plik      : Text;
  Ile_Linii : Word;
  Ile_znakow : Word;
  Ile_razem : Word;
  Znak      : Char;
Begin
  Ile_Linii:=0;
  Assign(Plik,ParamStr(1));
  Reset(Plik);
  While not Eof(Plik) do
    begin
      Ile_znakow:=0;
      while not EoLn(Plik) do
        begin
          Read(Plik,Znak);
          Inc(Ile_znakow);
        end;
      Readln(Plik);
      Inc(Ile_linii);
      Writeln('W linii ',Ile_linii:4,' jest ',Ile_znakow:3,
        ' znakow ');
      Inc(Ile_razem,Ile_znakow);
    end;
  Close(Plik);
  Writeln('Liczba linii: ',Ile_Linii);
  Writeln('Liczba znakow: ',Ile_razem);
End.
```

*{Przykład programu przetwarzającego dane liczbowe pobierane z pliku tekstowego
oraz umieszczającego wyniki w innym pliku tekstowym }*

Program Transponowanie_Macierzy_Liczb;

const

MaxRozm = 20; { *Maksymalny rozmiar macierzy* }

type

Tablica = Array[1..MaxRozm,1..MaxRozm] of Real;

var

Macierz : Tablica;

roz_m_w,roz_m_k : Word;

*{W tych zmiennych przechowywany jest rozmiar
wczytywanej macierzy (wiersz,kolumna)}*

procedure Wczytaj_do_pamieci(Nazwa_pliku : String;
var Tab : Tablica
var r_w, r_k : word);

var

Plik : Text;

w,k : Word;

begin

r_w:=0; r_k:=0

Assign(Plik,Nazwa_pliku);

Reset(Plik);

w:=0;

While (not EoF(Plik)) AND (not EoLn(Plik)) **do**

begin

Inc(w); *{Nowy wiersz}*

k:=0;

While not EoLn(Plik) **do**

begin

Inc(k); *{Kolejna kolumna}*

Read(Plik,Tab[w,k]);

if k>r_k **then** r_k:=k; *{Zwieksz rozmiar macierzy}*

end;

ReadLn(Plik);

Inc(r_w); *{Zwieksz rozmiar macierzy}*

end;

Close(Plik);

end; {----- Wczytaj_do_pamieci}

```

procedure Zapisz_na_dysk(Nazwa_pliku : String;
                          Tab          : Tablica;
                          r_w, r_k     : word );

var
  Plik : Text;
  w,k  : Word;
begin
  Assign(Plik,Nazwa);
  ReWrite(Plik);
  for w:=1 to r_w do
    begin
      for k:=1 to r_k do
        Write(Plik,Tab[w,k]:3:3, #32#32#32);
        WriteLn(Plik);
      end;
    Close(Plik);
end; {----- Zapisz_na_dysk}

```

```

procedure Transponuj_macierz( Tab1      : Tablica;
                               var Tab2  : Tablica;
                               r_w, r_k  : word );

var
  w,k : Word;
begin
  for w:=1 to r_w do
    for k:=w to r_k do
      Tab2[k,w]:=Tab1[w,k];
    end;
end; {----- Transponuj_macierz}

```

```

procedure Zeruj_macierz( var Tab : Tablica );
var
  w,k : Word;
begin
  for w:=1 to MaxRozm do
    for k:=w to MaxRozm do
      Macierz[w,k]:=0;
    end;
end; {----- Zeruj_macierz}

```

```

BEGIN
  Zeruj_macierz;
  Wczytaj_do_pamieci('dane.txt', Macierz , rozm_w, rozm_k );
  Transponuj_macierz( Macierz , Macierz , rozm_w, rozm_k );
  Zapisz_na_dysk('wynik.txt', Macierz , rozm_w, rozm_k );
END.

```

{Przykład programu generującego raport do pliku tekstowego}

```
program Baza_Danych_Osobowych;

const
  MAX_IL_OS      = 100;
  DL_NAZWISKO    = 30;
  DL_IMIE        = 15;
  DL_ADRES       = 50;

type
  dane_personalne = record
    nazwisko:string[DL_NAZWISKO];
    imie:string[DL_IMIE];
    adres:string[DL_ADRES];
    il_lat:integer;
    jest_dyplom:boolean;
  end;

  baza = array[1..MAX_IL_OS] of dane_personalne;

procedure GENERUJ_RAPORT( nazwa_pliku:string;
                          spis:baza; il_osob:word );

var
  plik:text;
  i:word;
begin
  assign(plik,nazwa_pliku);
  rewrite(plik); { append(plik) }
  writeln(plik,'Spis mlodych osob ktore ukonczyly studia');
  writeln(plik,'-----');
  for i:=1 to il_osob do
    with spis[i] do
      if (jest_dyplom) AND (il_lat<30) then
        begin
          write(plik, i:3 ,      ' ');
          write(plik, nazwisko , ' ');
          write(plik, imie ,    ' ');
          write(plik, adres  );
          writeln(plik);
        end;
      writeln(plik,'-----');
      writeln(plik,'Koniec zestawienia');
      writeln(plik);
      close(plik);
    end; {----- GENERUJ_RAPORT }
```

{Poprawienie wydruku o wyrównywanie kolumn}

```
function FORMATUJ( tekst:string; dlugosc:byte ):string;
var
  i:byte;
begin
  for i:=length(tekst)+1 to dlugosc do
    tekst[i]:=#32;
  tekst[0]:=char(dlugosc);
  FORMATUJ:=tekst;
end;
```

```
procedure GENERUJ_RAPORT_F( nazwa_pliku:string;
                           spis:baza; il_osob:word );
```

```
var
  plik:text;
  i:word;
```

```
begin
  assign(plik,nazwa_pliku);
  rewrite(plik); { append(plik) }
  writeln(plik,'Spis mlodych osob ktore ukonczyly studia');
  writeln(plik,'-----');
  for i:=1 to il_osob do
    with spis[i] do
      if (jest_dyplom) AND (il_lat<30) then
        begin
          write(plik, i:3 , ' ');
          write(plik, FORMATUJ(nazwisko,DL_NAZWISKO) , ' ');
          write(plik, FORMATUJ(imie, DL_IMIE ) , ' ');
          write(plik, FORMATUJ(adres, DL_ADRES ));
          writeln(plik);
        end;
  writeln(plik,'-----');
  writeln(plik,'Koniec zestawienia');
  writeln(plik);
  close(plik);
end; {----- GENERUJ_RAPORT_F}
```