

10 Operacje wejścia – wyjścia

10.1 Podstawowe operacje na plikach

Większość programów w trakcie pracy

wykorzystuje pliki do przechowywania danych. Dotychczas przechowywałeś dane w pamięci operacyjnej, tzn., że po zakończeniu programu twoje dane ulegały zniszczeniu, po przeczytaniu tego rozdziału to się zmieni.

Ze względu na typ danych przechowywanych w plikach możemy wyróżnić:

- pliki tekstowe - zawierające tekst ;
- pliki elementowe (zdefiniowane) - elementami tych plików są: tablice, rekordy;
- pliki amorficzne - o nieokreślonej strukturze.

Pierwszą rzeczą w momencie rozpoczęcia pracy na plikach jest stworzenie zmiennej plikowej,

robimy to tak:

```
var
```

```
nazwa_zmiennej : text; plik tekstowy
```

```
nazwa_zmiennej : file of typ: plik zdefiniowany;
```

```
nazwa_zmiennej : file plik o nieokreślonej  
strukturze;
```

Zanim zaczniemy wykonywać jakiegokolwiek operacje na pliku musimy zadeklarowanej zmiennej przypisać "fizyczną" nazwę pod jaką plik będzie widoczny na dysku, robimy to tak:

```
Assign(zmienna_plikowa, sciezka);
```

zmienna_plikowa – wykonując operacje na pliku do niej będziemy się odwoływać,

sciezka - ścieżka do pliku np.: 'C:\TEMP\PLIK.DAT'

Uwaga:

Procedura `Assign` nie sprawdza istnienia, a nawet możliwości dostępu do pliku opisanego ścieżką.

Po skojarzeniu należy plik otworzyć. Mamy do wyboru kilka procedur otwierających:

- **Reset**(zmienna_plikowa) - otwiera tylko plik już **istniejący**, brak pliku powoduje wystąpienie błędu fatalnego wejścia-wyjścia. Plik otwarty do czytania lub czytania i pisania.
- **Rewrite**(zmienna_plikowa) - tworzy plik, jeśli już istnieje to otwiera plik skojarzony ze zm_plikową. Jeśli plik już istnieje to dane w nim zapisane zostaną nadpisane. Plik otwarty do pisania.
- **Append**(zmienna_plikowa) - Procedura ta dotyczy tylko plików tekstowych. Otwiera do pisania istniejący plik, (skojarzony ze zmienną_plikową). Dane będą dopisywane na koniec pliku.
- **Close**(zmienna_plikowa) - zamyka plik skojarzony ze zmienną plikową.
- Function **Eof**(zm_Plik):Boolean; Zwraca TRUE gdy osiągnięty został koniec pliku FALSE w przeciwnym przypadku.

Zapisywanie i odczytywanie danych z pliku odbywa się za pomocą procedur:

Read, Readln, Write, Writeln:

Read(zmienna_plikowa,lista_argumentów);

Write(zmienna_plikowa,lista_argumentów);

Procedury te były już używane wcześniej, pierwszy parametr był wtedy opuszczany. Oznaczało to operacje na specjalnym pliku tekstowym – **konsoli**. Pliku tego można używać jawnie:

```
var plikwe,plikwy;
```

```
...
```

```
Assign(plikwe,'');
```

```
Assign(plikwy,'');
```

```
Reset(plikwe);
```

```
Rewrite(plikwy);
```

```
...
```

```
read(plikwe,n);
```

```
for i:=1 to 10 do
```

```
writeln(plikwy,'Tekst na ekran(',n,',')razy');
```

Składnia procedur czytania i pisania z plików powinna więc być znana .

10.2 Pliki tekstowe

Konsola – plik tekstowy standardowo otwarty do czytania i pisania.

Metody otwarcia:

Reset – tylko do czytania

Rewrite – tylko do pisania

Append –do aktualizacji, czyli pisania na końcu pliku.

10.2.1 Plik tekstowy a plik file of char

```
Program pliki_tekstowe;
var
plik_wejscowy : text;
plik_wyjsciowy : file of char;
znak : char;
begin
Assign(plik_wejscowy, 'plik.in');
Assign(plik_wyjsciowy, 'plik.out');
{ $I- } Reset(plik_wejscowy); { $I+ }
if IOResult <> 0 then exit;
Rewrite(plik_wyjsciowy);
while not EoF(plik_wejscowy) do begin
Read(plik_wejscowy, znak);
Write(plik_wyjsciowy, znak);
Write(znak);
end;
Close(plik_wejscowy);
Close(plik_wyjsciowy);
end.
```

10.3 Pliki elementowe

Metody otwarcia:

Reset – do czytania i pisania

Rewrite – tylko do pisania

Append –NIE JEST DOSTĘPNA, ale w zamian mamy:

Seek, FilePos, FileSize.

PROGRAM Pliki;

```
USES Crt;
```

```
Const ESC_Key =#27;
```

```
TYPE Osoba = RECORD
```

```
    Imie, Nazw : String[15];
```

```
    Adres      : String[50];
```

```
    Tel       : String[10];
```

```
END;
```

```
    PlikOsob = FILE OF Osoba;
```

```
Function Tworz(Var Plik : PlikOsob):Boolean;
```

```
VAR Nazwa : String;
```

```
Begin
```

```
    Twórz:=False; ClrScr;
```

```
    Write('Podaj nazwe pliku i sciezke : ');
```

```
    ReadLn(Nazwa); Assign(Plik, Nazwa);
```

```
    {$I-} Rewrite(Plik); {$I+}
```

```
    if IOResult <> 0 then exit;
```

```
    Twórz:=True
```

```
End;
```

```
Function Otworz(Var Plik : PlikOsob):Boolean;
```

```
VAR Nazwa : String;
```

```
Begin
```

```
    Otwórz:=False; ClrScr;
```

```
    Write('Podaj nazwe pliku i sciezke : ');
```

```
    ReadLn(Nazwa); Assign(Plik, Nazwa);
```

```
    {$I-} Reset(plik); {$I+}
```

```
    if IOResult <> 0 then exit;
```

```
    Otwórz:=True
```

```
End;
```

```
PROCEDURE CzytajRec (Var x:Osoba);
Begin
  ClrScr;
  Write('Podaj imie      : ');
  ReadLn(x.Imie);
  Write('Podaj nazwisko  : ');
  ReadLn(x.Nazw);
  Write('Podaj adres     : ');
  ReadLn(x.Adres);
  Write('Podaj telefon   : ');
  ReadLn(x.Tel);
End;
PROCEDURE WypiszRec (Var x:Osoba);
Begin
  Writeln('IMIE        : ', x.Imie);
  Writeln('NAZWISKO    : ', x.Nazw);
  Writeln('ADRES        : ', x.Adres);
  Writeln('TELEFON     : ', x.Tel);
End;
PROCEDURE DopiszDoPliku(Var Plik : PlikOsob;
                        x:Osoba);
Begin
  Seek(Plik, FileSize(Plik));
  Write(Plik, x);
End;
PROCEDURE Wypisz(Var Plik : PlikOsob);
Var x:Osoba;
Begin
  Reset(Plik);
  While Not Eof(Plik) Do
    Begin
      ClrScr;
      Read(Plik, x);
      WypiszRec(x)
      If ReadKey=ESC_Key then break;
    End;
End;
```


10.5 Funkcje i procedur operujące na plikach

- `Seek(Plik, n)`
Procedura **Seek** ustawia wskaźnik pliku przed składową o numerze **n**. **Plik** jest plikiem dowolnego typu z wyjątkiem typu `Text`, a **n** jest wyrażeniem typu `LongInt`. Pierwszy składnik pliku ma numer 0. **Seek** może być użyte do rozszerzenia pliku, poprzez ustawienie składnika pliku na pierwszy za końcem pliku, tzn. **Seek(Plik, FileSize(Plik))**.
- `FileSize(Plik)`
Funkcja ta zwraca aktualny wymiar pliku. Rezultat jest typu `LongInt`. Jeżeli **Plik** jest plikiem pustym, to **FileSize(Plik)=0**.
- `FilePos(Plik)`
Funkcja ta podaje aktualną pozycję wskaźnika pliku. Jeżeli wskaźnik pliku znajduje się na początku pliku, to **FilePos(Plik)=0**. Jeżeli wskaźnik pliku znajduje się na końcu pliku, **FilePos(Plik)=FileSize(Plik)**. Funkcja `FilePos` może być użyta tylko wtedy, gdy plik jest otwarty.
- `Erase(Plik)`
Procedura ta wymazuje (usuwa) plik związany ze zmienną plikową **Plik**. Nie jest dopuszczalne stosowanie `Erase` w stosunku do otwartego pliku. Plik musi być zamknięty przed użyciem procedury `Erase`.
- `Rename(Plik, Nowa_nazwa)`
Procedura ta zmienia nazwę pliku zewnętrznego związanego ze zmienną plikową **Plik**.
- `Truncate(Plik)`
Procedura obcina wszystkie składowe pliku poza aktualnym położeniem wskaźnika pliku. Aktualne położenie wskaźnika pliku staje się końcem pliku.