

OBSŁUGA PLIKÓW

1. Wstęp.

Program Turbo Pascal umożliwia zapisywanie i odczyt danych, co zapewnia możliwość wielokrotnego ich wykorzystywania. Zapamiętywanie danych odbywa się w tzw. **plikach danych**, które następnie można przeglądać za pomocą innych programów (np. notatnika).

2. Rodzaje plików.

Z punktu widzenia programu pascalowego plik może zawierać określone dane (liczby, teksty, tablice, rekordy itd.).



Ze względu na typ zawartych w nich danych rozróżnia się trzy podstawowe rodzaje plików:

- **pliki tekstowe,**
- **pliki elementowe,**
- **pliki beztypowe.**

1.1. Pliki tekstowe

Plik tekstowy (ang. text file) zawiera on informacje składające się z tekstów. Dane zapisane są w postaci czytelnej dla użytkownika i podzielone na wiersze zakończone znakami końca wiersza.



Deklaracja zmiennej plikowej dla pliku tekstowego ma postać:

```
Var nazwa_pliku : Text;
```

gdzie:

nazwa_pliku - nazwa zmiennej plikowej.



Plik tekstowy reprezentowany jest przez typ Text.

1.2. Pliki elementowe

Plik elementowy składa się z jednakowych elementów tego samego typu zapisanych w takiej samej postaci, w jakiej przechowywane są w pamięci.

Elementy mogą być typu prostego (liczbowego, znakowego) tablicowego lub rekordowego. Jeżeli dany plik zadeklarujemy jako złożony z elementów określonego typu, w programie będzie można zapisywać i odczytywać z niego wyłącznie dane tego typu. Na przykład jeżeli zadeklarujemy plik jako *file od Integer*, to do tego pliku będziemy mogli zapisywać tylko liczby typu integer !!!



Deklaracja zmiennej plikowej dla pliku elementowego ma postać:

Var nazwa_pliku : file of Typ;

gdzie:

nazwa_pliku - nazwa zmiennej plikowej.

Typ – typ prosty (np. integer, char, Real) lub strukturalny (tablica, rekord).

1.3. Pliki beztypowy (amorficzny)

Plik amorficzny jest to plik o nieokreślonej strukturze. Nie posiada także określonego typu.



Deklaracja zmiennej plikowej dla pliku amorficznego ma postać:

Var nazwa_pliku : file;

gdzie:

nazwa_pliku - nazwa zmiennej plikowej.

3. Zmienna plikowa.

Zanim zostaną wykonane jakiegokolwiek operacje na pliku (odczyt, zapis) należy najpierw zadeklarować tzw. **zmienną plikową**, która reprezentować będzie konkretny plik. Ze względu na przechowywane dane w pliku deklaracja zmiennej plikowej będzie miała postać:



Var

PlikTekstowy : text; { plik tekstowy }

PlikElementowy : file of Typ; { plik elementowy }

PlikBezTypu : file; { plik amorficzny }

4. Operacje na plikach tekstowych

4.1. Przyporządkowanie zmiennej plikowej z nazwą pliku – procedura ASSIGN.



Aby móc odwołać się do zawartości konkretnego pliku dyskowego, należy związać go ze zmienną, która będzie go reprezentować. Służy temu procedura **ASSIGN**:

Assign (zmienna_plikowa, nazwa_pliku)

Instrukcja ta wiąże zmienną plikową (reprezentującą plik określonego typu) z plikiem znajdującym się na dysku.

Przykład:

Var f: text; {zmienna plikowa}

...

Assign (f, 'wprow.txt');

- Nazwa pliku może być uzupełniona ścieżką dostępu np.

Assign (f, 'd:\wyniki\pomiar.txt');

4.2. Otwarcie pliku do odczytu lub zapisu.



Aby móc zapisać lub odczytać dane z pliku należy go otworzyć. Służą do tego procedury **RESET**, **REWRITE** lub **APPEND** (tylko dla plików tekstowych - otwarcie do dopisywania). Zapisy tych procedur mają postać:

RESET (zmienna_plikowa)

REWRITE (zmienna_plikowa)

Procedura **RESET** pozwala na otwarcie już **istniejącego pliku**. Służy do **otwierania plików tylko do odczytu**.

Natomiast procedura **REWRITE** otwiera nie tylko nowy plik, ale również plik istniejący. Jeżeli plik nie istniał, to tworzy go, zaś jeśli istniał to usuwa jego zawartość. Procedura **REWRITE służy do otwierania plików tylko do zapisu**.

Dla plików tekstowych istnieje trzecia metoda otwarcia – *otwarcie do dopisywania*. Służy do tego procedura **APPEND**.

APPEND (zmienna_plikowa)

Procedura **APPEND otwiera już istniejący plik wyłącznie do zapisu**. Dane będą dopisywane na koniec pliku. **Procedura APPEND dostępna jest tylko dla plików tekstowych !!!**

4.3. Dostęp do danych w pliku tekstowym – dostęp sekwencyjny.

Pliki tekstowe umożliwiają jedynie **dostęp sekwencyjny** do danych, tj. aby dostać się do żądanego miejsca w pliku należy przeczytać wszystko co jest przed nim. Dlatego każdorazowo należy sprawdzać czy znajdujemy się na końcu wiersza lub na końcu pliku. Służą do tego procedury **EOLN** i **EOF**.

EOLN (zmienna_plikowa)

EOF (zmienna_plikowa)

EOF (end of file – koniec pliku) sprawdza czy osiągnięto koniec pliku. Zawraca wartość *True* jeśli jesteśmy na końcu pliku.

EOLN (end of line – koniec wiersza) określa czy plik tekstowy znajduje się w pozycji końca wiersza. Zwraca wartość *True* jeśli jesteśmy na końcu wiersza.

4.4. Zamknięcie pliku.

Do zamykania pliku stosuje się procedurę **CLOSE** w postaci:

CLOSE (zmienna_plikowa)

Zamknięcie pliku jest niezwykle istotne i nie wolno o tej operacji zapominać !!!



Plik należy zawsze zamykać wywołaniem procedury **CLOSE – w przeciwnym wypadku możemy utracić dane.**

4.5. Zapisywanie i odczytywanie danych z plików tekstowych.



W celu doczytania elementów z pliku używamy instrukcji **read** i **readln**.

READ (zmienna_plikowa, zmienna);

READLN (zmienna_plikowa, zmienna);

Instrukcja **READ** odczytuje z pliku pojedynczą daną. Odczytana wartość umieszczana jest w *zmiennej*. Procedura **read** ma zastosowanie zarówno dla plików tekstowych jak i elementowych.

Natomiast instrukcja **READLN** odczytuje z pliku wartość, (umieszczana jest w *zmiennej*) i przechodzi do kolejnego wiersza – ma ona zastosowanie wyłącznie dla plików tekstowych !!!

Zapisywanie danych do plików tekstowych dokonuje się za pomocą instrukcji **write** i **writeln**.

WRITE (zmienna_plikowa, zmienna)

WRITELN (zmienna_plikowa, zmienna);

Obie instrukcje zapisują do pliku wartość *zmiennej*, przy czym **writeln** dołącza dodatkowo znak nowego wiersza. Instrukcja **write** jest dostępną zarówno dla plików tekstowych jak i elementowych, zaś **writeln** – tylko dla plików tekstowych.



5. Operacje na plikach elementowych

- Podobnie jak w plikach tekstowych przyporządkowanie nazwy pliku do zmiennej plikowej dokonuje się za pomocą procedury **ASSIGN**.
- Do otwierania plików wykorzystuje się procedury **RESET** i **REWRITE** (bez **APPEND**).
- Plik zamyka się procedurą **CLOSE**.
- Do odczytania elementu z pliku wykorzystuje się instrukcję **READ** (bez **readln**).
- Zapisywanie danych do plików dokonuje się za pomocą instrukcji **WRITE** (bez **writeln**).

5.1. Dostęp do danych w pliku elementowym – dostęp swobodny.



Pliki elementowe są plikami o **dostępie swobodnym**. W każdym momencie po otwarciu pliku możemy odwołać się do dowolnego jego elementu. Dostęp do danych zapewnia procedura **SEEK**, której działanie polega na ustawieniu wskaźnika plikowego na żądanej pozycji.

SEEK (zmienna_plikowa, pozycja)

Używając powyższej funkcji należy pamiętać, że elementy pliku liczone są od zera.

Na przykład:

`seek (f, 0)` – ustaw wskaźnik na początku pliku;

`seek (f, 5)` – przenieś wskaźnik plikowy na szósty element w pliku.



Do dopisywania do pliku elementowego używaj instrukcji:

seek (zmienna_plikowa, Filesize(zmienna_plikowa));

Powyższa instrukcja ustawia wskaźnik plikowy za ostatnim elementem pliku, czyli na jego końcu. Jest to odpowiednik procedury **Append** dla plików tekstowych.

np. `seek (f, filesize(f));`

Funkcja **Filesize(zmienna_plikowa)** zwraca rozmiar pliku.