PP. Laboratorium 4 Gr4

TYP REKORDOWY

POZNANE DOTĄD TYPY DANYCH

- Typ całkowity **INTEGER** (liczby całkowite)
- Typ rzeczywisty **REAL** (liczby zmiennoprzecinkowe)
- Typ znakowy **CHAR** (pojedynczy znak)
- Typ łańcuchowy **STRING** (łańcuch znaków = napis)
- Typ logiczny **BOOLEAN** (przyjmuje wartość **TRUE**/**FALSE**)
- Typy tablicowy zawierający określoną liczbę zmiennych danego typu (definicja w sekcji **TYPE**)

Identyfikator = **array** [index_min..index_max] **of** *TYP*

Dodatkowo można zadeklarować typ 'własny' – czyli pod własną nazwą zapisać istniejący typ sprecyzowany typ danych (definicja w sekcji **TYPE**), np.

Ttab1D = array[0..9] of integer; // Ttab1D – oznacza tablice 10 elementową tablicę 1D

TString25 = String[25]; // TString25 – oznacza napis na max 25 znakach

TYP REKORDOWY

Jest to specjalny typ danych 'agregujący' inne typy.

Innymi słowy jest to złożona **struktura** danych, której elementy (pola) mogą być różnych typów zarówno prostych jak i rekordowych.

Definiuje się w sekcji TYPE

DEFINICJA TYPU REKORDOWEGO

```
type
NAZWA_TYPU = record
 POLE1: TYP1;
 POLE2: TYP2;
 ...
end;
```

PRZYKŁAD UŻYCIA W PROGRAMIE

Definicja typu TAdres, zawierającego: nazweUlicy, numerDomu, Miasto

Definicje najlepiej umieści w module, w którym będą znajdowały się procedury i funkcje służące do obsługi tego typu.

```
Unit Osoby;
interface
type
TAdres = record
 nazwaUL: String;
 numerDomu: Integer;
 Miasto: String;
end;
implementation
end.
```

Po zdefiniowaniu nowego typu możemy go używać w głównym programie.
Do poszczególnych pól danego rekordu możemy odwołać się na dwa sposoby:

1. Poprzez operator kropki– *zmiennaRekordowa.nazwa pola;*
2. Za pomocą konstrukcji **WITH** (daje ona dostęp do wszystkich pól danego rekordu).

```
program Rekordy;
uses Osoby;
var
  adres:TAdres;
begin
 adres.nazwaUL:='Jodlowa';
 adres.numerDomu:=6;
 readln(adres.Miasto);
 writeln();
 writeln(adres.nazwaUL);
 writeln(adres.numerDomu);
 writeln(adres.Miasto);
 readln;
end.
```

```
program Rekordy;
uses Osoby;
var
  adres:TAdres;
begin
 with adres do
 begin
  nazwaUL:='Jodlowa';
  numerDomu:=6;
  readln(Miasto);
 end;
 writeln();
 with adres do
 begin
  writeln(nazwaUL);
  writeln(numerDomu);
  writeln(Miasto);
 end;
 readln;
end.
```

PRZYKŁAD ZAGNIEŻDŻONEGO REKORDU

Definicja typu TOsoba, opisująca osobę za pomocą: Imienia, Nazwiska, roku urodzenia i adresu.
Definicja umieszczona w module OSOBY

```
TOsoba = record
  Imie,Nazwisko: String;
  rokUrodzenia: Integer;
 adres: TAdres;
end;
```

Użycie w programie głównym
Aby dostać się do pól zagnieżdżonego rekordu możemy ponownie wykorzystać operator **.** albo instrukcję **WITH** (w poniższym przykładzie zaprezentowano oba rozwiązania)

```
program Rekordy;
uses Osoby;
var
  osoba:TOsoba;
begin
  osoba.imie:='Jan';
  readln(osoba.nazwisko);
  osoba.rokUrodzenia:=1989;
  with osoba.adres do
  begin
    Miasto:='Katowice';
    NazwaUl:='Norwida';
    NumerDomu:=8;
  end;
  writeln(osoba.imie,' ',osoba.nazwisko,' lat:',2016-osoba.rokUrodzenia);
  writeln('Zamieszkaly/a w: ',osoba.adres.Miasto, ' ul. ',
          osoba.adres.NazwaUl,' ', osoba.adres.NumerDomu);
  readln;
end.
```

TABLICA REKORDÓW

Analogicznie tak jak tablice podstawowych typów możemy tworzyć tablice rekordów – tworząc np. bazę danych osób, adresów, książek, aut itp. Itd.

Przykład (w programie głównym – oczywiście „ładniej" by było zrobić to na procedurach, ale to będzie część zadania domowego).

```
type
 BazaOsob:array[0..9] of TOsoba;
var
 tosoby:BazaOsob;

begin
  // zapis do tablicy
  for i:=low(tosoby) to High(tosoby) do
  begin
    with tosoby[i] do
    begin
      writeln('Podaj dane ',i+1,'-tej osoby');
      write('Imie: ');
      readln(Imie);
      write('Nazwisko: ');
      readln(Nazwisko);
      write('Rok urodzenia: ');
      readln(rokUrodzenia);
      with adres do
      begin
        write('Zamieszkaly/a w: ');
        readln(Miasto);
        write('ulica: ');
        readln(nazwaUL);
        write('Numer domu: ');
        readln(numerDomu);
      end;
    end;
  end;

  // odczyt z tablicy
  for i:=low(tosoby) to High(tosoby) do
  begin
    with tosoby[i] do
    begin
      writeln(Imie,' ', Nazwisko, ' lat: ',2016-rokUrodzenia);
      with adres do
      begin
        writeln('Zamieszkaly/a w: ', Miasto,' ulica:',nazwaUL,
                ' ',numerDomu);
        writeln('--------------------------------');
      end;
    end;
  end;
 readln;
end.
```

PLIKI REKORDOWE (binarne)

Pliki **rekordowe** to specjalny rodzaj plików **binarnych** w którym znajdują się rekordy (mogą to być również wartości pozostałych typów np. Integer). W odróżnieniu od plików tekstowych nie da się ich otworzyć w notatniku.

Zalety plików binarnych – w odróżnieniu od plików tekstowych, gdzie aby odczytać wartość z wiersza 10 trzeba było odczytać wartości wszystkich poprzedzających wierszy tu mamy dostęp **swobodny.**

UWAGI:

- Do zapisu/odczytu z pliku binarnego **NIE** używamy funkcji **writeln**/**readln** (tu nie ma żadnych linii), tylko funkcji **write**/**read**
- Ponieważ mamy do czynienia z danymi binarnymi **musimy** wiedzieć ile dokładnie **bitów** zajmie jeden rekord. Dlatego w rekordach musimy zadeklarować ile **bitów** zajmie zmienna typu string (np. imie: String[25];).

DEKLARACJA ZMIENNEJ PLIKOWEJ SKOJARZONEJ Z PLIKIEM BINARNYM

Jeżeli nie zastosujemy się do wcześniejszych uwag i nie zmienimy w deklaracji rekordu zmiennych typu string poniższy kod nie wykona się – pojawi nam się błąd.

```
Var
p:file of TOsoba;
```

OBSŁUGA PLIKÓW BINARNYCH

Do obsługi plików binarnych wykorzystuje się te same instrukcje co do plików tekstowych.

```
assign(p,'nazwa_pliku');      //przypisuje zmienna z nazwa pliku (jak w TXT)
rewrite(p);
write(p,osoba);              //zapis jednego (CAŁEGO) rekordu do pliku
reset(p);
read(p,osoba);               //odczyt jednego (CAŁEGO) rekordu z pliku
```

FUNKCJA SEEK

Jest to funkcja pozwalająca na ‘umieszczenie’ kursora w wybranej linii pliku binarnego (umożliwia wybranie np. tylko jednego konkretnego rekordu, lub wszystkich począwszy od tego).

UWAGA:

Należy kontrolować, czy nie chcemy ustawić się poza zakresem (czy liczba nie jest większa od ilości rekordów w wybranym pliku).

```
Seek(p,0);                   //ustawi się na początku pliku
read(p,osoba);               // odczyt pierwszej osoby z pliku
Seek(p,9);                   //ustawi się przy 10 rekordzie
read(p,osoba);               // odczyt 10 osoby z pliku
```

SPRAWDZANIE PLIKOW

W celu zabezpieczenia przed próbą otwarcia pliku który nie istnieje (albo próbą zapisu w miejscu do którego nie mamy dostępu) możemy, przed zapisaniem/otworzeniem pliku sprawdzić czy dana operacja się wykona.

W tym celu wykorzystamy znaczniki:

- {$I-} Wyłącza na chwilę obsługę błędów wejścia/wyjścia
- {$I+} Włącza na nowo obsługę błędów

Pomiędzy tymi znacznikami wpisujemy kod który chcemy przetestować. To czy kod się wykonał bezbłędnie czy nie zapisywane jest w **IOResult**.

Fragment przykładowej funkcji (CzyIstnieje) sprawdzającej czy da się otworzyć plik (zwraca TRUE jak się da i FALSE jak nie):

```
{$I-}
Assign(p,nazwaPliku);        // próba przypisania zmiennej p do pliku - nazwaPliku
Reset(p);                     // próba otwarcia pliku w trybie do odczytu
close(p);                     // jak otwieramy to musimy zamknąć
{$I+}
CzyIstnieje:=(IOresult=0);    // w momencie gdy powyższe operacje się powiodły
                              // IOresult przyjmuje wartość 0
          // możecie poczytać jakie jeszcze wartości przyjmuje i w jakich sytuacjach
```

Wykorzystanie w programie

```
if((CzyIstnieje(nazwaPliku))then
begin
  // możemy otworzyć tu plik i wykonywać na nim opreacje
end else
   begin
    writeln('Plik nie istnieje');
end;
```

ZADANIE DOMOWE

* –> dla osób które nie zaliczyły kartkówki

**–> dla osób z którymi się „umówiłam".

Oczywiście jak ktoś chce to może wykonać wszystkie punkty – będzie to z korzyścią dla tej osoby.

UWAGA

Radzę nie przerazić się zadaniem – jest dużo prostsze niż się początkowo wydaje (większość macie w niniejszym dokumencie :)

TREŚĆ

Napisać program który:

- Wczyta z pliku rekordowego dane do tablicy rekordów [TOsoba – zadeklarowana wcześniej]
- Wypisze całą zawartość tablicy rekordów pod podstacją:

Imie	Nazwisko	Lat	Zamieszkaly	ulica	nrDomu
Jan	Nowak	26	Gliwice	Akademicka	25
Anna	Kowalska	32	Katowice	Lompy	6
….					

- Zapisze rekord wybranej osoby do pliku tekstowego (nazwę pliku i numer osoby podaje użytkownik; sprawdzić czy nie podaje liczby spoza zakresu tablicy);
- Zamieni rekord o podanym indeksie na nowy (podany przez użytkownika)
- Zapisze zmienione dane do nowego pliku binarnego (podanego przez użytkownika)

Lista procedur i funkcji które należy zaimplementować:

- Procedura pobierająca dane JEDNEJ osoby od użytkownika
- Procedura pobierająca od użytkownika WSZYTSKIE osoby do tablicy rekordów
- Procedura wypisująca wszystkie osoby z tablicy rekordów
- Procedura zapisująca do pliku TXT wybrany rekord pod postacią:
  Jan Nowak, lat 26 Zamieszkaly/a w Gliwice, ul Akademicka 25
- Procedura zapisująca do pliku DAT
- Procedura odczytująca z pliku DAT (pamiętać o sprawdzeniu czy plik istnieje)
- Procedura zamieniająca wybrany rekord z tablicy nowym podanym przez użytkownika
- Funkcja sprawdzająca czy plik istnieje
- *Procedura wyświetlająca z PLIKU DAT jeden rekord podany przez użytkownika (SEEK)
- **Procedura zapisująca do pliku TXT tablicę rekordów (każda wartość w nowej linii)
  Jan
  Nowak
  26
  Gliwice
  Akademicka
  25
  Anna
  Kowalska
  …
- **Procedura odczytująca z pliku TXT rekordy do tablicy