



Typ rekordowy umożliwia łączenie danych różnych typów w jedną złożoną strukturę zwaną **rekordem**. Poszczególne elementy rekordu nazywamy **polami** danego rekordu.

Deklaracja typu rekordowego ma postać:

```
type nazwa_rekordu = record
  Lista_pola_1;
  Lista_pola_2;
  Lista_pola_n;
end;
var dane: Nazwa_rekordu;
```

Definiowanie typu rekordowego zaczyna się słowem kluczowym `record`, po którym umieszczone są listy poszczególnych pól z opisującymi je typami. Słowo `end` kończy definicję typu rekordowego. Po słowie `var` zadeklarowano zmienną rekordową `dane`.

Na przykładzie danych pracownika przedstawiono poniżej przykład definiowania typu rekordowego:

```
type pracownik = record
  Nazwisko : String[30];
  Imie     : String[20];
  Nr_osoby : Integer;
  Adres    : String[60];
  Telefon  : String[25];
  Plec     : Char;
  Dochody  : Real;
  Wiek     : Byte;
end;
var osoba : pracownik;
```

Każde pole rekordu może zostać wybrane do dalszego przetwarzania za pomocą **wskazania** przez tzw. **deskryptor pola**.

Deskryptor pola ma postać: `Nazwa_rekordu.nazwa_pola`

Dla rekordu zdefiniowanego w poprzednim przykładzie deskryptor `osoba.Imie` wskazuje `Imie` rekordu `osoba`.

Do wprowadzania danych do pól rekordów korzystamy z instrukcji `readln` lub `read`. Do czytania danych z rekordów stosujemy instrukcje `write` lub `writeln`.

Zmienna rekordowa w tym samym czasie może przechowywać dane tylko jednego rekordu. W związku z tym, przed przyjęciem kolejnych danych, dane aktualnie przechowywane muszą być wcześniej zapisane. Wykorzystuje się do tego celu tablicę rekordów (w pamięci operacyjnej) lub plik elementowy (na dysku). W przypadku wykorzystania tablicy rekordów zmienna `osoba` z wcześniejszego przykładu może być zadeklarowana następująco:

```
var osoba : array[1..10] of pracownik;
```

1. Napisz program będący prostą bazą danych osobowych pracowników. Dane jednego pracownika są zapisane w następujących polach rekordu: Nr_ew, Nazwisko, Imie, Plec. Zbiór rekordów jest z kolei zapisany w tablicy rekordów. Po wprowadzeniu danych program wyświetla w układzie kolumnowym zawartość wszystkich rekordów.

```
program Dane_osobowe;
uses crt;
type Dane_personalne = record
  Nr_ew       : Byte;
  Nazwisko    : String[30];
  Imie        : String[20];
  Plec        : Char;
end;
var Dane     : array[1..10] of Dane_personalne;
    i, k      : Byte;
```



```
begin
clrscr;
write('Podaj liczbe rekordow: ');
readln(k);
for i := 1 to k do
begin
writeln;
write ('Podaj Nr_ew ', i, '-ej osoby: ');
readln (Dane[i].Nr_ew);
write ('Podaj Nazwisko ', i, '-ej osoby: ');
readln (Dane[i].Nazwisko);
write ('Podaj Imie ', i, '-ej osoby: ');
readln (Dane[i].Imie);
write ('Podaj Plec (m/k) ', i, '-ej osoby: ');
readln (Dane[i].Plec);
writeln;
end;
writeln;
writeln('Wprowadzone dane:');
writeln;
writeln('Nr_ew', 'Nazwisko':11, 'Imie':15, 'Plec':5);
for i := 1 to k do
begin
write(Dane[i].Nr_ew, Dane[i].Nazwisko:15, Dane[i].Imie:15,
Dane[i].Plec:5);
writeln;
end;
readkey;
end.
```

W Turbo Pascalu istnieje instrukcja wiążąca `with`, umożliwiająca wskazanie całej grupy pól danego rekordu. Instrukcja `with` ma postać: `with nazwa_rekordu do instrukcja;`

Instrukcja po `do` jest najczęściej instrukcją złożoną. Zawarte w niej nazwy pól odnoszą się do rekordu wyszczególnionego w instrukcji `with`.

Zamiast napisać:

```
write(Dane.Nr_ew);
write (Dane.Nazwisko);
```

można zapisać:

```
with Dane do
begin
write(Nr_ew);
write(Nazwisko);
end;
```

Identyfikatory `Nr_ew` i `Nazwisko` będą traktowane jako pola rekordu `Dane`.

2. Zmodyfikuj program `Dane_osobowe` tak, aby podczas wprowadzania danych do pól rekordów oraz wyświetlania tych danych wykorzystać instrukcję wiążącą `with`.

3. Rozbuduj program `Dane_osobowe` w ten sposób, że typ rekordowy `Dane_personalne` zostanie rozszerzony o dodatkowe pole `Data_ur`. Pole `Data_ur`, będące typem rekordowym, powinno składać się z następujących pól:

```
rok      – typ okrojony 1900..2000
miesiac  – typ okrojony 1..12
dzien    – typ okrojony 1..31
```



Zadania

1. Napisz program zapisujący do tablicy rekordów dane dotyczące n studentów ($n \leq 5$ jest wprowadzane z klawiatury). Wprowadzane dane to Numer, Nazwisko, Imię, Oceny. Pole Oceny powinno zawierać oceny z trzech przedmiotów: Ekonometria, Angielski, Programowanie. Po wprowadzeniu danych program powinien wyświetlić zawartość wszystkich rekordów oraz oddzielnie dane studenta, którego średnia ocen będzie najwyższa.
2. Napisz program, który:
 - wprowadzi z klawiatury liczbę całkowitą n (mniejszą od 10) określającą liczbę studentów,
 - dla każdego studenta wprowadzi z klawiatury następujące dane:
numer (liczba całkowita),
nazwisko (do 20 znaków),
sr_ocen (liczba rzeczywista),
 - umieści te dane w tablicy jednowymiarowej (wektorze rekordów),
 - wprowadzi z klawiatury liczbę określającą średnią powyżej której studentom przyznaje się stypendium,
 - wszystkim studentom, którzy osiągnęli tę średnią przyznaje stypendium dowolnej wysokości (wartość stypendium zapisuje do rekordu w polu stypendium),
 - wyświetla tablicę,
 - wyświetla nazwisko studenta, który ma najwyższą średnią.
3. Napisz program, który:
 - wprowadzi z klawiatury pewną liczbę serii danych o studentach, zakończonych liczbą 0, o organizacji:
numer (liczba całkowita),
nazwisko (do 20 znaków),
stypendium (liczba rzeczywista),
 - umieści te dane w tablicy jednowymiarowej (wektorze rekordów),
 - wyznaczy i wyświetli na ekranie wartość średnią stypendiów,
 - wyświetli dane o wszystkich studentach, pokazane w układzie kolumnowym, w którym dane o jednym studentie są pokazane w jednym wierszu,
 - obok wartości stypendium tych studentów, którzy mają stypendia większe od średniej, postawi trzy znaki gwiazdek (***)
4. Napisz program, który pobierze z klawiatury ciąg danych o towarach w magazynie (nazwa, liczba, cena). Pusta nazwa towaru oznacza zakończenie pobierania danych. Dane należy umieścić w wektorze. Program ma wyznaczyć towar o największej wartości.