

PASCAL – TABLICE I REKORDY

1. Zmienne i typy

- Poznaliśmy już zastosowanie zmiennych, jak mówiliśmy – zmienna to takie pudełko w pamięci komputera, w którym można przechowywać pewne wartości podczas wykonywania programu. Aby można było skorzystać ze zmiennej, należy ją zadeklarować.
- Deklaracja zmiennej polega na określeniu jej nazwy oraz typu.

```
var  
a: integer;
```
- Znamy już typy całkowite (np. `integer`), rzeczywiste (np. `real`), logiczne (`boolean`), znakowe (`char`) oraz łańcuchowe (`string`)
- Na razie dla każdego obiektu potrzebowaliśmy zadeklarować osobną zmienną. W praktyce pojedyncze zmienne nie wystarczają, gdyż w większości programów operujemy na wielu zmiennych, należałoby ich zadeklarować wtedy bardzo dużo, co ani nie byłoby wygodne, ani czytelne, ani opłacalne.
- Aby zatem zadeklarować wiele zmiennych tego samego typu, możemy skorzystać z tablic.

2. Tablice

- Tablice można sobie wyobrazić jako uporządkowaną zbiorowość elementów jakiegoś typu, na przykład liczb czy znaków. Wyobraź sobie tablicę jako tabelkę z szeregiem pól, w każdym z tych pól można przechowywać jedną wartość określonego typu. Pola posiadają swoje numery – tzw. indeksy. Do zadeklarowania tablicy używa się słowa kluczowego `array`, zaraz po tym słowie następują nawiasy kwadratowe `[]`, w które ujęte są dwie liczby rozdzielone dwiema kropkami:

```
program prog1;  
var  
tablica1: array[1..10] of integer;  
begin  
tablica1[1]:=100;  
writeln(tablica1[1]);  
readln;  
end.
```

- Zmienna `tablica1` jest zadeklarowana jako tablica 10 liczb całkowitych typu `integer`. Wszystkie elementy tablicy muszą być **tego samego typu**. Typ ten jest określony po słowie kluczowym `of` następującym po zamknięciu nawiasów kwadratowych. Do konkretnej

liczby zapamiętanej w tablicy odwołujemy się tak jak to jest podane w przykładzie przy pomocy indeksu, czyli liczby podanej zaraz po nazwie zmiennej tablicowej w nawiasie kwadratowym, np. `tablica1[1]`.

- W przykładowej tablicy można było używać indeksów od 1 do 10. W praktyce te wartości można dobierać bardziej dowolnie, byleby lewy indeks był mniejszy od prawego. Często jednak programiści zwyczajowo indeksują tablice od 0.
- Przykładowa tablica miała jeden wymiar, co można sobie wyobrazić jako szereg pudełek ustawionych jedno przy drugim, a w każdym znajduje się zmienna określonego typu. W praktyce bardzo wygodne są tablice o większej liczbie wymiarów. Dwuwymiarowa tablica daje się przedstawić w postaci tabelki o określonej liczbie kolumn i wierszy:

```
program prog2;  
var  
tab: array[1..10,1..10] of integer;  
begin  
tab[4,3]:=19;  
writeln(tab[4,3]);  
readln;  
end.
```

- **Zadanie 1 (1 pkt):** Przypomnij sobie z poprzednich zajęć pętlę wypisującą na ekranie tabliczkę mnożenia. Na jej podstawie napisz program wypełniający przy użyciu pętli zadeklarowaną tablicę dwuwymiarową zawierającą tabliczkę mnożenia. Program powinien w wyniku podać wartość pola tablicy o indeksach zadeklarowanych w stałych `tabX` i `tabY`.

3. Rekordy

- Tablice przechowują w pamięci zmienne tego samego typu. Jednak często napotykamy na sytuację, gdy chcemy zapisać w jednej zmiennej reprezentację jakiegoś obiektu rzeczywistego posiadającego atrybuty różnego typu. Załóżmy, że zmienna ma zawierać informacje dotyczące osoby. Te informacje to jej imię, nazwisko, płeć i rok urodzenia. Przykładowy program mógłby wyglądać tak:

```
program osoba;  
var  
imie: string[20];  
nazw: string[30];  
plec: char;           {k lub m}  
rok: word;  
begin  
end.
```

- W powyższym programie nie użyliśmy zmiennych, a tylko je zadeklarowaliśmy, ponieważ chodziło tylko o danie przykładu. Przy okazji zwróć uwagę, jaki był dobór typów dla konkretnych zmiennych (maksymalne długości napisów, zakres liczb przeznaczonych na płeć i rok urodzenia).

- Powyższy program jest zupełnie poprawny, jednak w praktyce mało przydatny. Już dla kilku osób nagromadziłoby się wiele zmiennych, które trzeba by jakoś ze sobą wiązać (wiek z nazwiskiem dla każdej osoby etc) Tym razem z pomocą przychodzą rekordy.
- Można sobie zatem wyobrazić rekord jako zmienną (zbiór pól), która zawiera kilka pól o **dowolnie wybranych typach**.
- W praktyce wykorzystuje się jednak rekord jako pojedynczą zmienną o zadeklarowanym przez programistę typie (gdzie deklaracja typu oznacza opisanie jakie pola znajdują się w danym typie rekordu).
- Aby posłużyć się zmienną rekordową (zadeklarować zmienną typu rekordowego) musimy najpierw zadeklarować taki typ. Typ deklarujemy po słowie kluczowym `type` po którym nie piszemy średnika, podobnie jak po słowie `var`. Po zadeklarowaniu typu i jego zdefiniowaniu możemy przystąpić do deklaracji zmiennej. W strukturze programu ważne jest, że najpierw definiujemy nowy typ, a dopiero potem go używamy deklarując zmienne tego typu. Poniżej przedstawiam przykładowy program z typem rekordowym `czlowiek` i zmienną `osoba` tego typu.

```

program rekordy;
type
  czlowiek = record
    imie: string[20];
    nazw: string[30];
    plec: char;           {k lub m}
    rok: word;
  end;
var
  dane_osoby: czlowiek;
begin
end.

```

- W powyższym programie zadeklarowaliśmy rekord posiadający 4 pola. Deklarację typu zakończył napis `end;` (zwróć uwagę na średnik)
- Kiedy już zdefiniowaliśmy typ rekordowy, czas go wykorzystać w praktyce. W tym celu należy zadeklarować zmienne tego typu. Robimy to tradycyjnie, po słowie kluczowym `var`. Nazwy typu rekordowego używamy tak jak nazwy każdego innego typu. w naszym programie zadeklarowaliśmy zmienną `osoba`. Od tego momentu ma ona w sobie kilka części - pól rekordu. Jeśli chcemy się do nich dostać, czyli je zapisać lub odczytać, robimy to pisząc nazwę zmiennej, następnie kropkę (bez spacji!) i następnie nazwę pola które chcemy odczytać/zapisać. Nazwa pola pochodzi z definicji typu, a nie z deklaracji zmiennej. Na przykład możemy stworzyć taki program:

```

program rekordy_moje;
type
  czlowiek = record
    imie: string[20];
    nazwisko: string[30];
    plec: char;           {k lub m}
  end;

```

```

    rok: word;
end;
var
    dane_osoby: czlowiek;
begin
    writeln('Podaj swoje imie:');
    readln(osoba.imie);
    writeln('Podaj swoje nazwisko:');
    readln(osoba.nazwisko);
    writeln('Podaj plec [k/m]:');
    readln(osoba.plec);
    writeln('W którym roku sie urodziles:');
    readln(osoba.rok);
    writeln('Teraz wiem cos o Tobie! Nazywasz sie ',osoba.imie,'
',osoba.nazwisko);
    writeln('Twoja plec to "',osoba.plec,'" . Urodziles sie w ',osoba.rok,'
roku!');
    readln;
end.

```

- Rekordy są "fundamentem" tzw. programowania obiektowego – to właśnie od nich się ono rozwinęło. Programowanie obiektowe jest podstawą obecnego programowania, w szczególności języka C++ i innych języków na nim bazujących. Obecnie prawie wszystkie programy pisane w językach wysokiego poziomu są pisane w technice obiektowej.
- **Zadanie 2 (1 pkt):** Napisz program, który pobierze od użytkownika szereg danych, a następnie uzupełni pola w **dwóch** rekordach. Każdy rekord ma opisywać dane pewnej książki, powinien posiadać następujące pola: *tytul*, *autor*, *rok_wydania*. W wyniku program powinien wypisać wszystkie dane z obydwóch rekordów.

4. Tablice rekordów

- Uzasadniając celowość zastosowania rekordów posłużyliśmy się przykładem zbierania danych na temat osób. W przypadku kilku osób pojedyncze zmienne nie byłyby wygodnym narzędziem. W przypadku kilkunastu osób nawet rekordy deklarowane osobno dla każdej osoby stałyby się niewygodne.
- Znamy już sposób na przechowywanie wielu zmiennych tego samego typu. Tablice pozwalają też na przechowywanie wielu zmiennych typu rekordowego. Zadeklarujmy tablicę przechowującą pięć rekordów typu *czlowiek*:

```

program baza1;
type
    czlowiek = record
        imie: string[20];
        nazwisko: string[30];
        plec: char;
        rok: word;
    end;
var
    baza_ludzi: array[1..5] of czlowiek;
begin
end.

```

- Aby odwołać się do konkretnego pola rekordu dla konkretnej osoby w zmiennej `baza_ludzi` należy użyć odwołania (podobnie jak w tablicach i rekordach) np:
- `baza_ludzi[1].nazwisko` lub `baza_ludzi[3].plec`
- Przykład:

```
program baza1;
type
  czlowiek = record
    imie: string[20];
    nazwisko: string[30];
    plec: char;
    rok: word;
end;
var
  baza_ludzi: array[1..5] of czlowiek;
begin
  writeln('Podaj imie:');
  readln(baza_ludzi[1].imie);      {do tablicy zostanie zapisane imie}
  writeln('Podaj nazwisko:');
  readln(baza_ludzi[1].nazwisko); {do tablicy zostanie zapisane nazwisko}
end.
```

- **Zadanie 3 (1 pkt):** Napisz program pobierający od użytkownika dane (Imię, nazwisko, płeć, rok urodzenia) dotyczące 3 osób. Zrób to przy użyciu pętli `FOR..TO..DO`. Następnie program powinien wypisać te dane dla każdej osoby w osobnej linii (z odpowiednimi napisami co jest czym), również przy użyciu pętli.
- W miejscu, w którym wypisywana jest na ekranie płeć, program zamiast literki `k` lub `m` powinien napisać pełny wyraz: `kobieta` lub `mezczyzna`.
- Zapisz efekty swojej pracy, przydadzą się w dalszych ćwiczeniach.

5. Zadanie 4 (1 pkt):

- Napisz program, który najpierw buduje tablicę 10-elementową wypełnioną liczbami losowymi z przedziału (0..1) o 2 miejscach po przecinku (granulacja do setnej części), a następnie sumuje poszczególne pola z tablicy i podaje wynik (wynik skumulowany dla poszczególnych wartości tablicy oraz wynik dla sumy całkowitej z tablicy).
- Podpowiedź: aby uzyskać liczby losowe wykorzystaj funkcję `random(x)`, losującą liczbę całkowitą z przedziału `(0..x)`. Aby liczby były losowe, przed wykonaniem funkcji należy wcześniej uruchomić generator liczb losowych procedurą `randomize`.
- Przykład:


```
program randomizacja;
var
  x: integer;
begin
  randomize;
  x:=random(100)+10;
end.
```
- Powyższy program losuje liczbę całkowitą losową z przedziału (10..110)