

## TABLICE I REKORDY - złożone struktury danych.

**Tablica** służy do przechowywania dużej ilości danych tego samego typu. Tablica, to jakby pudełko, podzielone na komórki. W każdej komórce tablicy można przechować jakąś wartość (liczbę tekst itp.). Ponieważ tą wartością może być również tablica - można więc tworzyć tablice wielowymiarowe. Podczas deklarowania tablic wielowymiarowych należy uważać na ilość zajmowanej przez nie pamięci. Komórki pamięci można numerować w dowolny, ale kolejny sposób, np. od 1 do 10, ale i od -5 do 4. Wszystko zależy od naszych pomysłów na realizację programu

Przykłady deklaracji zmiennych tablicowych:

tablica jednowymiarowa

```
oceny : ARRAY [1..10] OF byte;
```

tablica o dziesięciu komórkach ponumerowanych od 1 do 10 w których można przechowywać wartości typu byte. W pamięci zajmuje 10 bajtów.

--	--	--	--	--	--	--	--	--	--

tablica dwuwymiarowa

```
klasa : ARRAY [1..5,10..20] OF char;
```

tablica dwuwymiarowa o 5 wierszach ponumerowanych od 1 do 5 i 10 kolumnach ponumerowanych od 10 do 20 o elementach typu char. W pamięci zajmuje  $5 \times 10 = 50$  bajtów.


### Zapis i odczyt z komórek tablicy

Pojedynczy element tablicy wskazujemy za pomocą indeksowania, np.: `oceny[4]` oznacza czwarty element tablicy oceny, `klasa[2,7]` oznacza element tablicy klasa znajdujący się w drugim wierszu i siódmej kolumnie.

Przykłady poleceń:

zapis do tablicy (np. zerowanie tablicy)

```
for i:=1 to 10 do oceny[i]:=0;
```

pętla wyzeruje po kolei wszystkie 10 komórek tablicy oceny

odczyt z tablicy (np. sumowanie elementów tablicy w zmiennej x)

```
x:=0; for i:=1 to 10 do x:=x+oceny[i];
```

jeśli w tablicy oceny znajdują się jakieś konkretne liczby, to w pętli zostaną one po kolei dodane do zmiennej x. Na końcu znajdzie się w niej suma wszystkich komórek tej tablicy

**Rekordy** służą do przechowywania powiązanych ze sobą danych, ale różnych typów. Na przykład opis człowieka może zawierać nazwisko i imię (typ tekstowy o stałej długości 20 i 10 znaków), płeć (np. typ znakowy), wiek (typ liczbowy całkowity), wzrost (typ rzeczywisty) itd. W tablicy nie można przechowywać elementów różnych typów - umożliwia to tylko rekord.

Zdefiniujemy nowy typ danych opisujący człowieka. Rekord ma poła opisane w ramce. Ponieważ jest to nowy typ zmiennej, dlatego posłużymy się nowym słowem TYPE.

TYPE

```
Tosoba = RECORD
    nazwisko:string[20];
    imie:string[10];
    plec:char;
    wiek:byte;
    wzrost:real;
end;
```

Do tak zdefiniowanego typu danych można utworzyć zmienne, w których można przechowywać jednocześnie kilka informacji opisanych w rekordzie.

```
var osoba1,osoba2:Tosoba;
```

Przypisanie danych do takiej zmiennej będzie wyglądało w następujący sposób:

```
osoba1.nazwisko:='Libront';
osoba1.wiek:=35;
osoba2.wiek:=osoba1.wiek+2;
```

## Tablice rekordów

Jeśli chcielibyśmy przechować dane wielu osób, wygodnie jest zadeklarować tablicę złożoną elementów zdefiniowanego typu rekordowego:

```
osoby:array[1..32] of Tosoba;
```

I na przykład wyzerowanie wieku wszystkich osób w takiej tablicy będzie wyglądało tak:

```
for i:=1 to 32 do osoby[i].wiek:=0;
```

## ĆWICZENIA - GRAFIKA 15.PAS

W programie GRAFIKA14.PAS po ekranie poruszał się kwadrat i odbijał się od brzegów. Jeśli chcielibyśmy, aby po ekranie poruszało się kilka niezależnych kwadratów, musielibyśmy każdemu z nich zadeklarować osobny zestaw zmiennych go definiujących (X, Y, BOK, DX, DY, KOLOR itp.). Każda zmienna musiałaby się inaczej nazywać. Gdyby tych kwadratów było np. 100 konieczne byłoby 600 zmiennych. Oczywiście jest to sposób mało praktyczny.

Wygodniej będzie posłużyć się poznaną dzisiaj strukturą tablicową (lub rekordem). Zdefiniujemy tablicę, w której wszystkie potrzebne dane dla 100 różnych kwadratów będą opisane w następujący sposób:

- tablica o 6 kolumnach i 100 wierszach
- każdy wiersz sześciokolumnowy to zestaw zmiennych jednego kwadratu
- każda kolumna to określony rodzaj zmiennej - opisany na rysunku

x	y	bok	dx	dy	kol

definicja takiej tablicy o nazwie np. TK (Tablica Kwadratów).

```
var TK:array[1..100,1..6] of integer;
```

Nie ma znaczenia, czy zdefiniujemy tablicę 100 x 6 czy też 6 x 100 - wszystko zależy od sposobu wykorzystania tej tablicy w programie.

Każdy kwadrat powinien poruszać się w innym kierunku i mieć inne wymiary i położenie początkowe. Najprościej będzie po prostu wylosować te wartości (w programie GRAFIKA15.PAS jest ich 50).

Instrukcja **RANDOMIZE** inicjuje licznik liczb pseudolosowych. Dlaczego nie losowych? Po prostu komputer tak naprawdę nie potrafi wymyślić liczb naprawdę losowych - wszystko sprowadza się do matematycznych obliczeń. Oczywiście dla małych zakresów liczb te liczby mogą wydawać się losowe. Wykonanie instrukcji spowoduje, że po każdym uruchomieniu programu komputer będzie wymyślał inne liczby.

Wykonanie instrukcji **a:=RANDOM(100)** spowoduje przypisanie do zmiennej a wybranej losowo liczby w przedziale od 0 do 99.

```
Randomize;
ile:=50;
for i:=1 to ile do
begin
  TK[i,1]:=random(600);   {x    0..599}
  TK[i,2]:=random(400);   {y    0..399}
  TK[i,3]:=random(20)+10; {bok  10..30}
  TK[i,4]:=random(15)+1;  {kolor 1..15}
  TK[i,5]:=random(9)-4;   {dx   -4..4}
  TK[i,6]:=random(9)-4;   {dy   -4..4}
end;
```

Animacja wszystkich kwadratów odbywa się w identycznej pętli jak w programie GRAFIKA14.PAS. Zamiast wyświetlać, mazać, sprawdzać położenie i przesuwać jeden kwadrat będziemy w pętli FOR robić to dla wszystkich kwadratów w tablicy.

```
repeat
  for i:=1 to ile do kwadrat(TK[i,1],TK[i,2],TK[i,3],TK[i,4]);   {rysowanie}
  delay(10);
  for i:=1 to ile do kwadrat(TK[i,1],TK[i,2],TK[i,3],0);         {mazanie}
```

```

for i:=1 to ile do                                     {nowe położenie}
begin
  TK[i,1]:=TK[i,1]+TK[i,5];
  TK[i,2]:=TK[i,2]+TK[i,6];
end;

for i:=1 to ile do                                     {zmiana kierunku jeśli brzeg}
begin
  if (TK[i,1] >= 640 - TK[i,3]) or (TK[i,1] <= 0) then TK[i,5]:=-TK[i,5];
  if (TK[i,2] >= 480 - TK[i,3]) or (TK[i,2] <= 0) then TK[i,6]:=-TK[i,6];
end;

until keypressed;

```

A teraz ten sam program z użyciem rekordów GRAFIKA16.PAS.

Najpierw definicja rekordu opisującego jeden kwadrat. Poprzednio był to jeden wiersz tabeli i wszystkie komórki tego samego typu INTEGER - 2 bajty. Cała tablica zajmowała  $100 \times 6 \times 2 = 1200$  bajtów.

W strukturze rekordowej możemy każdą zmienną zdefiniować osobno:

X i Y typu INTEGER lub WORD, bo potrzebne nam są duże liczby dodatnie (2 bajty)

BOK i KOLOR typu BYTE - małe liczby dodatnie od 0 do 255 (1 bajt)

DX i DY - SHORTINT - małe liczby dodatnie i ujemne od -128 do 127 (1 bajt)

Jeśli zdefiniujemy tablicę 100elementową takich rekordów to będzie miała  $100 \times 8 = 800$  bajtów

Oszczędność 400 bajtów - może i niewielka, jeśli weźmiemy pod uwagę dzisiejsze parametry komputerów, ale liczy się sposób, pomysł i realizacja.

```

type
  TKwadrat=record
    x,y:integer;
    bok,kolor:byte;
    dx,dy:shortint;
  end;

```

Definiowanie zmiennej tablicowej -

```

var
  TK:array[1..100] of TKwadrat;

```

Animacja kwadratów - identyczna metoda jak poprzednio. Zmienia się tylko sposób zapisu

```

Randomize;
ile:=50;
for i:=1 to ile do
begin
  TK[i].x:=random(600);
  TK[i].y:=random(400);
  TK[i].bok:=random(20)+10;
  TK[i].kolor:=random(15)+1;
  TK[i].dx:=random(9)-4;
  TK[i].dy:=random(9)-4;
end;

repeat
  for i:=1 to ile do kwadrat(TK[i].x,TK[i].y, TK[i].bok,TK[i].kolor);
  delay(15);
  for i:=1 to ile do kwadrat(TK[i].x, TK[i].y, TK[i].bok, 0);

  for i:=1 to ile do
  begin
    TK[i].x:=TK[i].x+TK[i].dx;
    TK[i].y:=TK[i].y+TK[i].dy;
  end;

  for i:=1 to ile do
  begin
    if (TK[i].x >= 640 - TK[i].bok) or (TK[i].x <= 0) then TK[i].dx:=-TK[i].dx;
    if (TK[i].y >= 480 - TK[i].bok) or (TK[i].y <= 0) then TK[i].dy:=-TK[i].dy;
  end;
until keypressed;

```

---

## ZADANIE

### ODCINKI

Wylosuj położenie (x,y) 100 punktów na ekranie monitora. Narysuj wszystkie odcinki pomiędzy tymi punktami.

- zadeklaruj tablicę 100x2 np. T[1..2,1..100];
- losowo wpisz do tablicy współrzędne X i Y  
FOR i:=1 TO 100 DO  
BEGIN  
    T[1,i]:=random(640);  
    T[2,i]:=random(480);  
END;
- rysuj odcinki każdy z każdym  
FOR i:=1 TO 100 DO  
FOR j:=1 TO 100 DO  
BEGIN  
    LINE(T[1,i], T[2,i], T[1,j], T[2,j]);  
END;

Ile odcinków zostanie narysowanych wielokrotnie?

### TRÓJKĄTY

Po ekranie przesuwa się 10 trójkątów. Rogi każdego trójkąta poruszają się i odbijają osobno.

Rekord ROG z polami X,Y,DX,DY

Rekord TRO z polami R1,R2,R3 typu ROG

Tablica TAB [1..10] o elementach typu TRO

- losowanie współrzędnych i kierunków
- pętla animacyjna
  - \* rysowanie
  - \* wstrzymanie
  - \* wymazanie
  - \* nowe położenie
  - \* odbicie