

0.1. Rekordy

Oprócz omawianych już typów strukturalnych takich jak tablice, zbiory i pliki, można jeszcze wyróżnić takie typy złożone jak rekordy oraz obiekty. Obiektom poświęcimy następny wykład.

```

typ strukturalny      +-----+
----->| array |
|               ^      +-----+
|   +-----+   |   | set   |
+-->| packed |--+   +-----+
      +-----+      | file |
                        +-----+
                        | record |
                        +-----+
                        | object |----->
                        +-----+

```

Definicja syntaktyczna typu rekordowego, który będziemy teraz omawiać, jest następująca.

typ rekordowy

```

----->( record )-----^-->( end )----->
|               |
+-[ Lista pól ]-+

```

Pola rekordu mogą być dowolnego typu. Jest to cecha, która różni zasadniczo, rekord od tablicy.

Lista pól

```

|
+--[ Cześć stała ]----->
|               |               ^               ^   |               ^
|               |               |               |   |               |
|               +---( ; )--->[ Cześć wariantowa ]->+   +-->( ; )--+
|               ^
+----->+

```

Stała część rekordu zbudowana jest z listy nazw (identyfikatorów) pól oraz ich typów. Oddzielają je średniki.

Cześć stała

```

-----[ Lista nazw ]---( : ) [ typ ]----->
^               |
|               |
+<-----<( ; )<-----+

```

Każde pole zawiera informację, którą można zawsze uzyskać w ten sam sposób.
PRZYKŁAD.

type

```

Data=record
    dzien : 1..31;
    miesiac: (St,Lu,Ma,Kw,Ma,Cz,Li,Si,Wr,Pa,Li,Gr);
    rok: integer
end;

```

type

```

zespolona=record
    re, im: real
end;

```

```

Osoba=record
    Nazwisko: string;
    Imie      : string;
    DataUrodzenia: Data;
    plec      : (mezczyzna,kobieta);
    StanCywilny : (wolny,zonaty,owdowialy,rozwiedziony)
end;

```

Za pomocą konstruktora rekordowego można utworzyć wartość danego typu i przypisać ją jakiejś zmiennej tegoż typu. Jeżeli, przykładowo, mamy zmienne

```

z: zespolona;
d: Data;
o: Osoba;

```

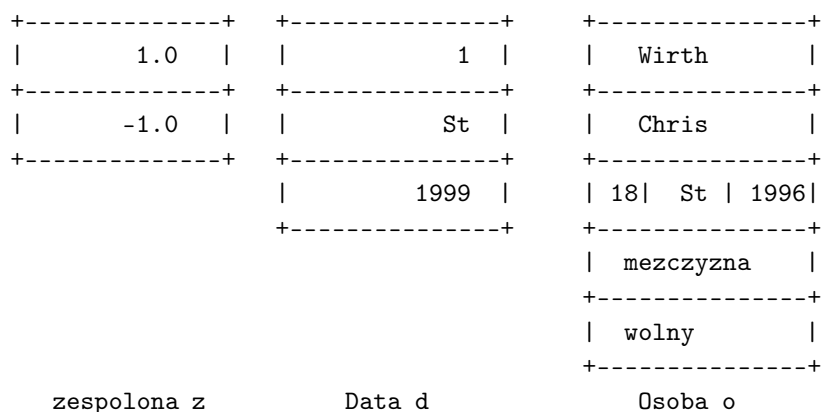
to konkretne wartości można przypisać następująco:

```

z:=zespolona(1.0,-1.0);
d:=Data(1,St,1999);
o:=Osoba('Wirth','Chris',Data(18,St,1966),mezczyzna,wolny);

```

Można to przedstawić na diagramach następująco.



Jak dotrzeć do poszczególnych pól rekordu? Jeśli zmienna x jest typu rekordowego T, a s jest nazwą pola rekordu, to

```

x.s:=xi

```

oznacza przypisanie polu s tegoż rekordu, wartości xi. Typ xi i typ pola x rekordu są takie same.

W przykładach, które były podane wyżej mamy

	typ
z.im	(real)
d.miesiac	(St, ..., Gr)
o.nazwisko	(string)
o.DataUrodzenia	(Data)
o.DataUrodzenia.Dzien	(1..31)
itd.	

(Proszę zwrócić uwagę na złożoność struktury Osoba, która zawiera Datę, itd.)

```
var a      : array[1..N] of Osoba;
    licznik: integer;

begin
    licznik:=0;
    for i:=1 to N do
        if (a[i].plec=kobieta) and (a[i].stancywilny=wolny) then
            licznik:=licznik+1;
    end;

(*  liczba kobiet w stanie wolnym zapisanych w tablicy a  *)
```

Inny sposób dostępu do pól rekordu polega na użyciu instrukcji wiążącej **with**. Przytoczony wyżej fragment programu ustalającego liczbę kobiet w stanie wolnym, można z jej pomocą zapisać następująco: **PRZYKŁAD**.

```
var a      : array[1..N] of Osoba;
    licznik: integer;

begin
    licznik:=0;
    for i:=1 to N do
        with a[i] do
            if (plec=kobieta) and (stancywilny=wolny) then
                licznik:=licznik+1;
    end;

(*  liczba kobiet w stanie wolnym zapisanych w tablicy a  *)
```

Konstrukcja **with r do** s oznacza, że można używać nazw selektorów pól zmiennej rekordowej r bez poprzedzania ich nazwą zmiennej, ponieważ wiadomo, że będą się do niej odnosiły. Skraca to tekst programu i zapobiega wielokrotnemu wyliczaniu od nowa składowej indeksowanej a[i].

Poniższy program stosuje strukturę rekordu do reprezentowania liczb zespolonych

```
program zespol;
```

```

type
  PostacZespolonej = (kartezjanska, biegunowa);

5
  Zespolona = record
      case Postac: PostacZespolonej of
          kartezjanska: (x, y : real);
          biegunowa: (r, fi : real)
10      end;

procedure CzytajZ(var z: Zespolona);
var
15   Kod: char;
begin

    Writeln('Postac (k\Slash b)');
    Read(Kod);
20   Writeln(Kod);    (* Echo *)

    case Kod of
        'k': begin
            z.postac:=kartezjanska;
25            Read(z.x);
            Read(z.y)
            end;
        'b': begin
            z.postac:=biegunowa;
30            Read(z.r);
            Read(z.fi)
            end;
        else
            Write('Zle dane')
35        end
    end;

end;

Procedure ConvertZ(var z: Zespolona);
40 (*
    zamienia biegunowa -> kartezjanska
    *)
begin
    if (z.postac=biegunowa) then
45        begin
            z.postac:=kartezjanska;
            z.x:=z.r*cos(z.fi);
            z.y:=z.r*sin(z.fi)
        end
50    end;

Procedure SumujZ(z1, z2: Zespolona;
    var SumaZ : Zespolona);
55 (*
    Dodawanie liczb zespolonych, ktore moga byc
    reprezentowane w postaci kartezjanskich x+iy
    lub biegunowej r*e^(i*fi)
60 *)

var
    x1, y1, x2, y2: real;
begin
65     if z1.postac = kartezjanska then
        begin x1:=z1.x; y1:=z1.y end
    else
        begin
            x1:=z1.r * cos(z1.fi);
70            y1:=z1.r * sin(z1.fi)
        end;
end;

```

```

75   if z2.postac = kartezyjanska then
      begin x2:=z2.x; y2:=z2.y end
    else
      begin
        x2:=z2.r * cos(z2.fi);
        y2:=z2.r * sin(z2.fi)
      end;

80   SumaZ.postac:=kartezyjanska;
      SumaZ.x:=x1+x2;
      SumaZ.y:=y1+y2
    end; (* SumujZ *)

85

    var
      z1, z2, z: Zespolona;

90   begin
      z1.postac:=kartezyjanska;
      z2.postac:=kartezyjanska;
      z1.x:=1; z1.y:=2;
      z2.x:=5; z2.y:=7.5;

95   CzytajZ(z1);
      CzytajZ(z2);

      Writeln;
      ConvertZ(z1);
100  Writeln('z1 =', z1.x, '+i(', z1.y, ')');
      ConvertZ(z2);
      Writeln('z2 =', z2.x, '+i(', z2.y, ')');
      SumujZ(z1, z2, z);
105  ConvertZ(z);
      Writeln(z.x:8:4, z.y:8:4);
    end.

```

Listing 1. Liczby zespolone

0.1.1. Rekordy z wariantami

Czasami wygodnie jest korzystać z tzw. rekordów z wariantami. Tak jest np. w przypadku współrzędnych punktu płaszczyzny. Czasami potrzebna jest ich postać biegunowa, z czasami kartezyjska. rekordy wariantowe pozwalają na przechowywanie w nich takich lub innych informacji w zależności od sytuacji. Dla rozpoznania wariantu (rodzaju współrzędnych) potrzebujemy dodatkowego wyróżnika, wielkości, która przyjmuje odpowiednie wartości odróżniające występujące sytuacje (tak jak to jest w przypadku współrzędnych kartezyjskich lub biegunowych). Wyróżnik typu nazywa się też polem wariantowym.

```

[ Czesć wariantowa ]
|
+--( case )-----[ typ pola ]--( of )-->[ wariant ]--->
      |               ^      wyróżnika      ^               |
      |               |               |               |
      +->[ nazwa ]-->( : )--+
                                   |               |
                                   +-( ; )<-----+

```

```

typ pola wyróżnika
|
----->[ Identyfikator typu wyliczeniowego ]----->

```

```

variant
|
+----->[ stała ]----->( : )----( ( )----->( ) )----->
  ^               |               |               |
  |               |               +-->[ Lista ]--+
  +----( , )-----+               pól

```

Na podstawie diagramu widzimy, że każdy wariant identyfikowany jest przez conajmniej jedną stałą. Stałe muszą być różne i muszą być typu wyliczeniowego zgodnego z typem pola wyróżnika.

Identyfikator pola wyróżnika jest opcjonalny.

PRZYKŁAD.

```

type
  Wspolrzedne=record
    case rodzaj:(kartezjanskie,biegunowe) of
      kartezjanskie:(x,y :real);
      biegunowe      :(r,fi:real)
    end;

```

Nazwą pola znacznikowego jest w tym wypadku rodzaj, a współrzędne mają nazwy x, y lub r, fi w zależności od rodzaju.

Typ Osoba, który został zdefiniowany poprzednio, zawiera informacje dotyczące płci osób itp. Można tam również umieścić informację o zarobkach, w przypadku mężczyzn oraz o trzech wymiarach sylwetki kobiecej (Przykład z Wirtha). Dobrze jest zredefiniować ten rekord tak by rozróżniany był ten fakt. Zapiszmy: PRZYKŁAD.

```

Osoba=record
  Nazwisko, Imie      : string;
  DataUrodzenia: Data;
  StanCywilny      : (wolny,zonaty,owdowialy,rozwieziony);
  case plec:(meczczyna,kobieta) of
    mezczyzna: (waga: real;
                brodaty: boolean);
    kobieta:   (wymiary: array[1..3] of real)
  end;

```

Operacje na wariantach najlepiej jest pogrupować w instrukcję wybiórczą (case), której struktura odzwierciedla strukturę rekordu z wariantami.

```

case x.sn of
  c1: s1;
  c2: s2;
  ...
  cm: sm
end;

```

Podam teraz przykład obliczania odległości na płaszczyźnie przy zadanych kartezjańskich lub biegunowych współrzędnych punktów.

PRZYKŁAD.

```
(*
  podprogram OdlAB oblicza odległość między dwoma punktami
  A i B zadanymi w postaci zmiennych a i b typu rekordowego
  Wspolrzedne, będącego typem rekordowym z wariantami
  (patrz: Iglewski,...)
*)

function OdlAB(a, b: Wspolrzedne): real;
var d: real;
begin
  case a.rodzaj of
    kartezjanskie: case b.rodzaj of
      kartezjanskie: d:=sqrt(sqr(a.x-b.x)+sqr(a.y-b.y));
      biegunowe      : d:=sqrt(sqr(a.x-b.r*cos(b.fi))+sqr(a.y-b.r*sin(b.fi)))
    end
    biegunowe: case b.rodzaj of
      kartezjanskie: d:=sqrt(sqr(b.x-a.r*cos(a.fi))+sqr(b.y-a.r*sin(a.fi)));
      biegunowe      : d:=sqrt(sqr(a.r)+sqr(b.r)-2*a.r*b.r*cos(a.fi-b.fi))
    end;
  OdlAB:=d;
end;
```

Następny przykład został wzięty z podręcznika Turbo Pascal 7, Language Guide, str. 33.

PRZYKŁAD.

```
TWielobok=record
  X, Y : Real;
case Rodzaj : Figura of
  TProstokat: (Wysokosc,Szerokosc: real);
  TTrojkat:   (bok1,bok2,kat: real);
  TOKrag:     (promien: real)
end;
```

0.1.2. Stałe typu rekordowego

stała rekordowa

```
|
+-->( ( )--->[ Nazwa pola ]--->( : )---[ stała określonego ]-->( ) )-->
      |                                     typy                                |
      +----->( ; )<----->+
```

PRZYKŁAD.

```
const a: integer=3;
type TPunkt=record X,Y: real end;
   TWektor=array[0..1] of TPunkt;
   TMiesiac=(Sty,Lut,Mar,Kwi,Maj,Cze,Lip,Sie,Wrz,Paz,Lis,Gru);
   TData=record D:1..31; M: TMiesiac; R: 1900..1999 end;
const Poczatek: TPunkt=(X: 0.0, Y: 0.0);
```

```
Linia: TWektor=((X:-3.1; Y:1.5), (X:5.8; Y:3.0));  
Dzien: TData=(D:2; M:Gru; R:1998);
```

Pola musimy podawać w takim porządku w jakim pojawiły się w definicji typu rekordowego.

Jeśli rekord zawiera pola typu plikowego to stałe tego typu nie mogą być definiowane.

Jeśli rekord posiada warianty, to tylko pola wybranego wariantu można ustalić.

W przypadku gdy wariant zawiera pole wyróżnika musimy to pole wyspecyfikować.