

TYP POLE

Jednorozměrné a dvourozměrné pole

Pole je datová struktura skládající se z prvků stejného typu, které se vzájemně rozlišují pomocí indexu. Použití pole je mnohem efektivnější než načítání dat do mnoha proměnných, což je nepraktické, protože bychom si museli pamatovat velké množství jmen a nemohli bychom například použít cyklus pro jejich zpracování. Jednotlivé prvky pole se rozlišují pomocí indexů. Každému indexu (nebo kombinaci indexů u vícerozměrných polí) odpovídá jeden prvek pole.

Např. **jednorozměrné pole** je tvořeno posloupností prvků, kde polohu prvku určuje hodnota indexu.

	prvek1	prvek2	prvek3	...	prvekN
index	1.	2.	3.	...	N.

Typ pole patří mezi tzv. **strukturované typy**, které specifikují množiny strukturovaných hodnot, tj. takových hodnot, které se skládají s dílčích složek

Popis typu pole má základní tvar

array [interval indexů] of typ prvku

Velikost pole určuje interval indexů. Interval (meze) musí být konstanty (nemohou to být proměnné).

Př.:

```
type
  radaCisel = array [1..3] of real;
  veta = array [0..79] of char;
var
  U, V : vektor;
  otazka : veta;
  W : array [1..3] of real;
```

Typ `radaCisel` je typ pole o třech prvcích typu `real` a indexy jsou z intervalu `1..3`; `veta` je typ pole o 80 prvcích typu `char`.

Pole je možné deklarovat přímo jako proměnnou (`w`) nebo si nejprve pro něj můžeme nejprve vytvořit typ jako `v` v našem příkladu (`radaCisel`, `veta`). Druhý způsob má výhodu v tom, že je možné obsah jednoho pole přiřadit poli druhému (pokud je stejného typu) jednoduchým přiřazovacím příkazem. Pole je pak také možno použít jako parametr v procedurách a funkcích.

Hodnoty proměnných typu pole se obvykle definují a mění postupně tak, že se přiřazují hodnoty jednotlivým složkám. Přiřazovacím příkazem je však možno definovat i celou hodnotu typu pole (na pravé straně musí být proměnná téhož typu).

Např.:

```
U := V;    je ekvivalentní s
U[1] := V[1]; U[2] := V[2]; U[3] := V[3];
```

Nelze pokud

```
var
  W : array [1..3] of real;
```

Pak příkaz

```
U := W;
```

není dovolen, protože proměnná `w` není typu `radaCisel`.

Př.:

Načtěte do proměnné typu pole 10 teplot měřených během dne. Zjistěte maximální teplotu, minimální teplotu a kolikáté byly v pořadí a průměrnou teplotu.

```
program MereniTeploty;
const
  N=10;
var
  Teploty : array [1..N] of real;
  i, iMax, iMin : integer;
  MaxTeplota, MinTeplota, Prumerna, Soucet : real;
begin
  writeln ('Zadej jednotlivé teploty');
  for i:= 1 to N do
    begin
      write ('Zadej ', i, '. teplotu: ');
      readln (Teploty[i]);
    end;
  MaxTeplota := Teploty[1];
  iMax := 1;
  for i:= 2 to N do
    if Teploty[i] > MaxTeplota then
      begin
        MaxTeplota := Teploty[i];
        iMax := i;
      end;
  MinTeplota := Teploty[1];
  iMin := 1;
  for i:= 2 to N do
    if Teploty[i] < MinTeplota then
      begin
        MinTeplota := Teploty[i];
        iMin := i;
      end;
  Soucet := 0;
  for i:= 1 to N do
    Soucet := Soucet + Teploty[i];
  Prumerna := Soucet/N;
  writeln ('Posloupnost teplot');
  for i:= 1 to N do
    write (Teploty[i]:5:1, ' ');
  writeln;
  writeln;
  writeln ('Maximalni teplota je ', MaxTeplota:5:1, ' a byla ', iMax, '.');
  writeln ('Minimalni teplota: ', MinTeplota:5:1, ' a byla ', iMin, '.');
  writeln ('Prumerna teplota je ', Prumerna:5:1);
end.
```

Na typ prvků se neklade žádné omezení, mohou být rovněž typu pole.

Zavádějí se typy *vícerozměrných polí*. Popis *dvourozměrného pole* (matice) má základní tvar

array [interval indexů] **of array** [interval indexů] **of** typ složek

Např. můžeme definovat typ maticeA jako typ dvourozměrného pole obsahujícího M krát N prvků typu real, kde M a N jsou opět konstanty.

```
type
  maticeA = array [1..M] of array [1..N] of real;
```

nebo zkráceně

```
type
  maticeA = array [1..M, 1..N] of real;
```

indexy	1.	2.	3.	...	N.
1.	prvek [1,1]	prvek [1,2]	prvek [1,3]	...	prvek [1,N]
2.	prvek [2,1]	prvek [2,2]	prvek [2,3]	...	prvek [2,N]
3.	prvek [3,1]	prvek [3,2]	prvek [3,3]	...	prvek [3,N]
...
M.	prvek [M,1]	prvek [M,2]	prvek [M,3]	...	prvek [M,N]

Př.: Napište program, který vynásobí všechny prvky matice zadaným číslem.

```

program VynasobMaticiciCislem;
const
  M=3;
  N=3;
var
  maticeA, maticeB : array [1..M, 1..N] of real;
  i, j : integer;
  x : real;
begin
  writeln ('Zadej cislo pro vynasobeni matice: ');
  readln (x);
  write ('Zadej jednotlivé prvky matice: ');
  for i:= 1 to M do
    for j := 1 to N do
      begin
        write ('Zadej prvek ', i, ', ', j);
        readln (maticeA[i,j]);
      end;
  for i:= 1 to M do
    for j := 1 to N do
      maticeB [i,j] := x * maticeA [i,j];
  writeln ('Puvodni matice:');
  for i:= 1 to M do
    begin
      for j := 1 to N do
        write (maticeA[i,j], ' ');
      writeln;
    end;
  writeln ('Vynasobena matice:');
  for i:= 1 to M do
    begin
      for j := 1 to N do
        write (maticeB[i,j], ' ');
      writeln;
    end;
end.

```

Zpracovala: Ing. Simona Martínková, březen 2007