

Strukturované datové typy

Popisují objekty, které se skládají ze složek (komponent), jednotlivé složky jsou přístupné pomocí speciálních operací nazývaných selektory.

- *homogenní* - všechny složky (komponenty) struktury jsou stejného typu
- *heterogenní* - složky struktury jsou rozdílného typu

Základní strukturované datové typy : *pole*, *záznam*, *soubor*, *množina*

Typ pole

Homogenní strukturovaný datový typ skládající se z pevného počtu položek stejného typu (tzv. *bázového</*

Př.

type POLE = array [1..5] of real;

var A : POLE;

type BARVA = (BILA, ZLUTA, CERVENA, MODRA);

var B: array [BARVA] of boolean;

A[1]:=2.5; A[2]:=3.75; A[3]:=5. ; A[4]:= 7; A[5]:=0

B[BILA]:= true; B[CERVENA]:=false; B[MODRA]:=true;

B[ZLUTA]:=true;

Operace s typem pole : pouze manipulace se složkami

Př: var I: integer;

- přiřazení **A[5]:=3;**
- vstup/výstup **read(A[2]); writeln(A[1]);**
- použití složek ve výrazu **X=A[1] + A[I+2]**

může být libovolný výraz s hodnotou
v intervalu 1 .. 5

Uložení pole v operační paměti:

- souvislá oblast bytů začínající na adrese P (určuje překladač), velikost oblasti závis

Vícerozměrná pole

type T = array[T₁₁] of array [T₁₂] of ... of array[T_{ij}] of T_s ;

T₁₁, T₁₂, ... T_{ij} - typy indexů

T_s - typ složek

Ekvivalentní zkrácený zápis:

type T = array [T₁₁, T₁₂, ... ,T_{ij}] of T_s ;

Operace s vícerozměrným poli: obdobné jako u jednorozměrného pole.

Př: Dvourozměrná pole - matice

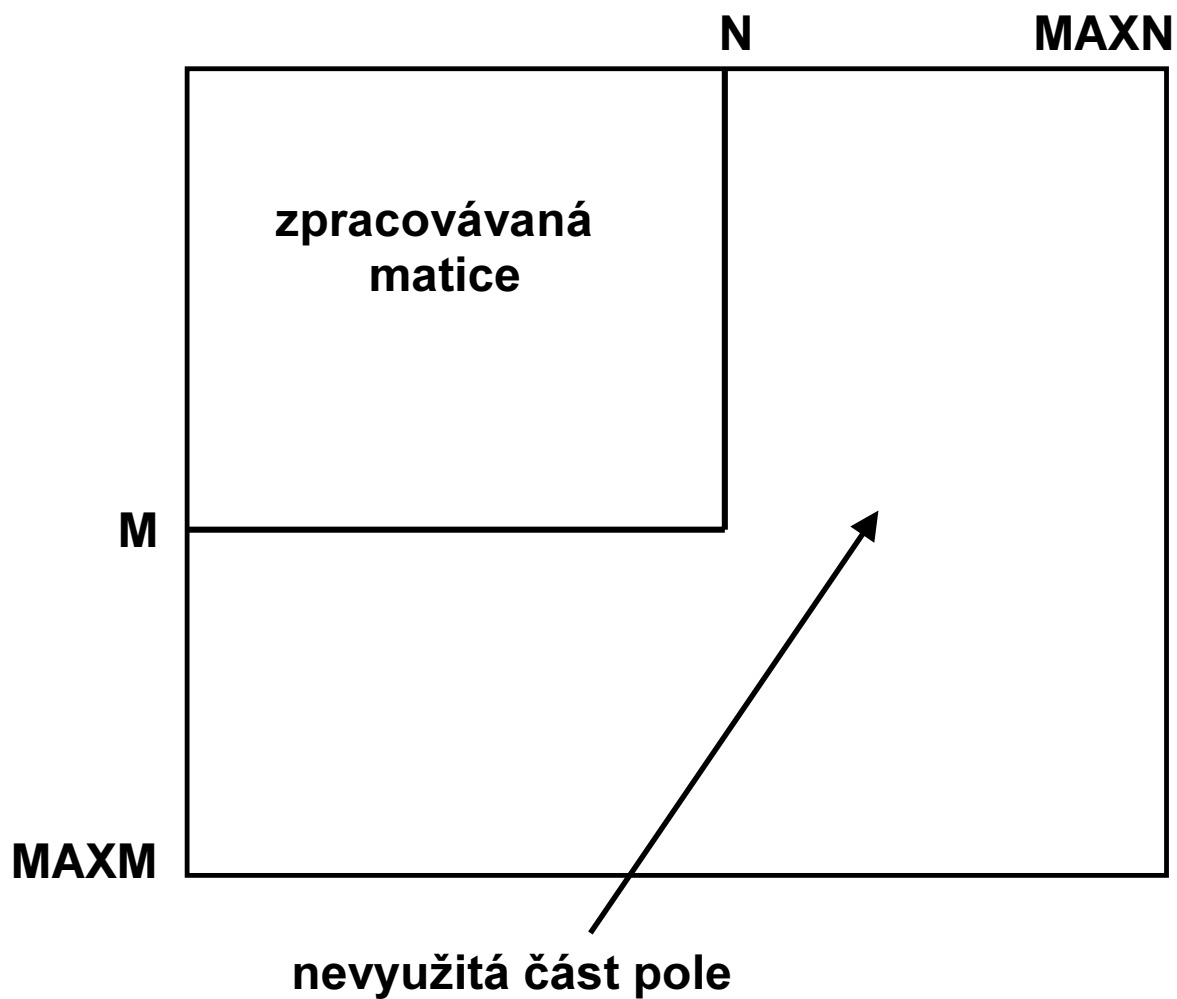
```
const MAXM= 100;  
      MAXN = 100;
```

```
type RADEK  = array [ 1 .. MAXM ] of real;  
      MATICE = array [ 1 .. MAXN ] of RADEK;
```

```
var  A,B: matice;  
      I ,J : integer;  
      M,N: integer
```

```
...
```

```
writeln('zadej rozměr matice);
```



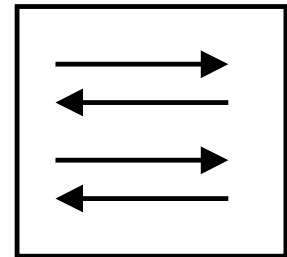
Průchod maticí :

- po řádcích
 for l:= 1 to M do
 for J:= 1 to N do " zpracuj A[l

- speciální případ

```

for I:= 1 to M do
  for J:= 1 to N do
    begin
      if odd(I) then K:=J else K:=N-J+1;
      " zpracuj A[I,K] " ;
    end
  
```



V případě čtvercové matice řádu N

- po hlavní diagonále

```

for I := 1
```

Pole znaků, řetězce znaků

V Turbo (Borland Paccsalu) – dva způsoby reprezentace řetězců:

1. typ `array[1..K] of char`

Deklarace : *type T = array[T_i] of char;*

var A : T;

Tento způsob definuje řetězce pevné délky !!!!

Povolené operace:

+ spojí dva řetězce

2. Datový typ *string*

Deklarace :

i. var retezec: string;

ii. var retezec: string[K];

Tento způsob definuje posloupnost znaků dynamické délky s maximální konstantní velikostí 255 znaků (*i*), popř. K znaků (*ii*).

Operátory :

+ spojení dvou řetězců

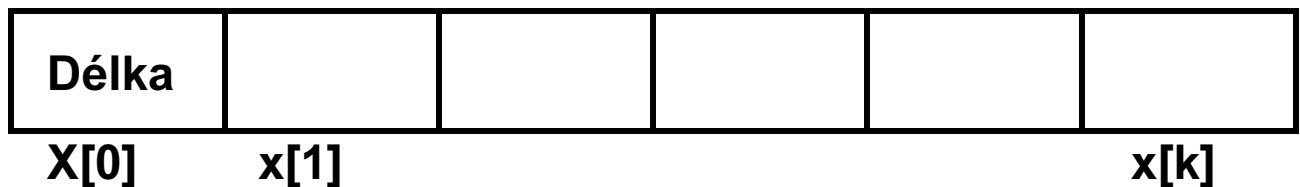
:= přiřazení - přiřazovat se mohou i řetězce roz

Write(x) - vytiskne řetězec *x* na obrazovce

Podobně pro Readln(x) a Writeln(x).

S proměnnou *x* typu *string[K]* lze zacházet také jako s jednorozměrným polem deklarovaným jako

X: array [0 .. K] of char;



tj. jednotlivé složky řetězce je možné indexovat.

<

Typ záznam

Nehomogenní (heterogenní) statická datová struktura, ve které mohou být složky obecně různého typu. Složky záznamu jsou též nazývány položkami.

Záznamy se používají k uložení údajů, které spolu logicky souvisí, ale nejsou stejného typu.

Definice typu záznam :

type T = record

$S_1: T_1;$

$S_2: T_2;$

.....

Př.

```
type DATUM = record
    DEN      : 1 .. 31;
    MESIC    : 1 .. 12;
    ROK       : 1900 .. 2010;
end;
```

```
type OSOBA = record
    JMENO, PRIJMENI : string[20];
    DAT_NAROZ       : DATUM;
End
```

Operace se záznamem

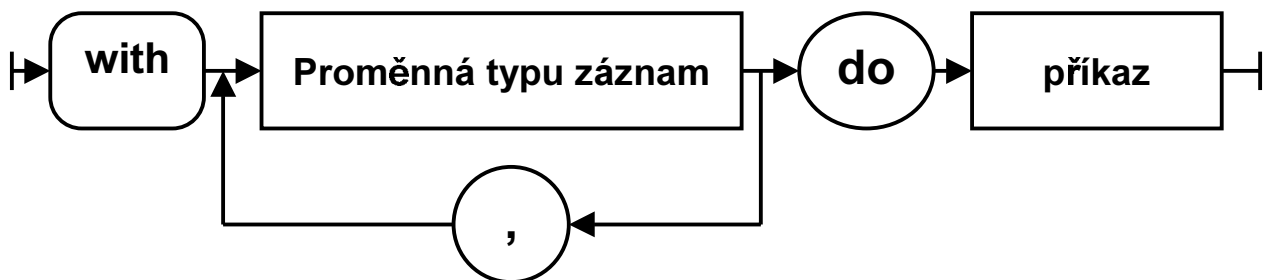
Přístup k jednotlivým položkám :

- pomocí selektoru složky

$Z.S_i$ kde Z - proměnná typu

- pomocí příkazu *with*

with Z do P ; kde *Z* - proměnná typu záznam
P- libovolný příkaz obsahující práci se složkou *S_i*



Př.

```
    Type T1 = record
        JMENO ,PRIJMENI : string[20] ;
        OS_CISLO : integer;
    end
```

```
type  T2 = record
    A : integer ;
    B : real;
end ;
```

```
type  T3 = record
    A : integer ;
    B : real;
    OS_CISLO : integer;
end ;
```

```
var X,Y : T1;
    Z   : T2;
    K   : T3;
```

with X,Y do ; - Chyba - shodná jména identifikátorů

```
with Z, X do begin
    A :=
```

