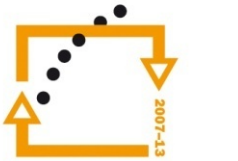

INVESTICE
DO ROZVOJE
VZDĚLÁVÁNÍ

Střední průmyslová škola a Vyšší odborná škola technická Brno, Sokolská 1

Šablona: Inovace a zkvalitnění výuky prostřednictvím ICT

Název: Základy programování a algoritmizace úloh

Téma: Vícerozměrná pole

Autor: Ing. Hodál Jaroslav, Ph.D.

Číslo: VY_32_INOVACE_26–05

Anotace: *Materiál popisuje konstrukci a obsluhu vícerozměrných polí.*
Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum.
Vytvořeno: leden 2013.

Podpora digitalizace a využití ICT na SPŠ CZ.1.07/1.5.00/34.0632

5. Vícerozměrná pole

- definici datového typu pole lze rozšířit i na více rozměrů
- jestliže jednorozměrné pole lze chápat jako **posloupnost** několika hodnot nebo **vektor**, dvourozměrné pole reprezentuje v matematickém smyslu **matici** (nebo také vektor vektorů se stejným počtem prvků)
- analogicky jako je dále popisováno rozšíření na dva rozměry, můžeme postupovat i u vícerozměrných polí

5	-1	2	0	8

2	7	0	-3	4
1	9	7	0	2
-4	1	6	1	8
8	0	2	7	6
-5	6	-1	3	0

Dvourozměrná pole - deklarace

- **deklarace** dvourozměrného pole je velmi podobná deklaraci pole jednorozměrného (viz 26-1 – Pole)
- vzhledem k tomu, že všechny prvky v celém poli musí být stejného druhu, je jedinou změnou deklarace **přidání druhého indexu** (jeho rozsahu)

```
var
  a:array[1..10,1..5] of byte;
```

- oba rozsahy indexů jsou od sebe odděleny **čárkou**
- ta se používá pro oddělení indexů i při práci s prvky pole v příkazech

```
readln(a[3,2]); a[1,1]:=0; write(a[7,5]);
```

Obsluha a typické konstrukce (1)

- pro základní obsluhu dvourozměrného pole je potřeba použít **dva do sebe vnořené cykly** (viz 25-17 – Složitější cykly s proměnnými mezemi a vnořování cyklů)
- typickými konstrukcemi jsou **načtení** a **výpis** pole

```
// načtení hodnot do pole
for i := 1 to 10 do
  for j := 1 to 5 do
    begin
      writeln('Zadej prvek ',i,',',j);
      readln(a[i,j]);
    end;
```

Obsluha a typické konstrukce (2)

```
// výpis hodnot v podobě matice
for i := 1 to 10 do
  begin
    for j := 1 to 5 do write(a[i,j]:4);
    writeln;
  end;
```

- zde je vhodné podotknout, že orientace pole (to který index je „řádkový“ a který „sloupcový“) záleží pouze na přání programátora a jím vytvořeném kódu
- pokud prohodíme červenou a modrou část kódu, dojde k přeorientování matice ve výpisu z matice 10 řádkové a 5 sloupcové na 5 řádkovou a 10 sloupcovou
- Proč **nelze** pouze prohodit indexy `a[i,j]` na `a[j,i]`?

Obsluha a typické konstrukce (3)

- připomeňme, že pole bylo deklarováno jako **`a:array[1..10,1..5] of byte;`**
- kdybychom místo záměny vnoření cyklů pouze prohodili indexy ve výpisu prvků pole, došlo by v uvedeném příkladě k pokusu o přístup k neexistujícím prvkům (např. **`a[1,7]`**)
- na pořadí indexů tedy velmi záleží
- můžeme pracovat i s indexy různých druhů

```
var
  s:array[1..8,'a'..'h'] of byte;
```

- výše uvedenou deklaraci bychom tak mohli chápat např. jako šachovnici zaplněnou čísly a k prvkům by se přistupovalo **`a[2,'c']:=0;`**