

POČÍTAČE A PROGRAMOVÁNÍ

Vícerozměrná statická a dynamická pole,
Pole polí

Miroslav Vavroušek

PPI 08

Opakování z minulé přednášky

- Datová pole
 - Jednorozměrná statická datová pole
 - Dynamická datová pole
 - String jako pole

Datový typ Indexu: Integer Datový typ buňky: Integer	
Index	Data
1	8
2	11
:	:
500	82

Datový typ Indexu: Integer Datový typ buňky: Real	
Index	Data
1	0,58
2	14,91
:	:
500	18,01

Datový typ Indexu: Char Datový typ buňky: String	
Index	Data
'a'	'Ahoj'
'b'	'jak se'
:	:
'z'	'!'

Datová pole

- Datová pole představují množiny prvků stejného datového typu.
- K jednotlivým prvkům pole přistupujeme pomocí indexu.
- Pole můžeme rozdělit na statická a dynamická
 - Statická – délka zvolená před kompilací programu
 - Dynamická – délku pole lze měnit za běhu programu
- Každé pole je nějakého konkrétního rozměru
- Jednorozměrná pole si můžeme představit, jako sloupec tabulky kde každá buňka je zvoleného datového typu
- Dvourozměrné pole lze reprezentovat jako tabulku a třírozměrné jako kvádr z krychliček jednotného typu.

Jednorozměrné statická a dynamická pole

- Statické pole

- Deklarace pole

```
Data: array [1..10] of Integer;
```

- Dynamické pole

- Deklarace pole

```
Data: array of Integer;
```

- Nastavení délky pole

```
SetLength(Data, 8);
```

- Statické a dynamické pole

- Uložení hodnoty do pole

```
Data[8] := 16;
```

- Přístup k hodnotě v poli

```
A := Data[8]; //Do A uloží hodnotu z pole Data na indexu 8
```

String jako pole

- S datovým typem String lze pracovat jako s jednorozměrným polem znaků
 - K jednotlivým znakům lze přistupovat pomocí indexu
 - Znaky se Indexují o 1
 - Počet znaků lze zjistit pomocí `Length(JmenoPromenne);`

```
:  
var  
  S: String;  
  C: Char;  
begin  
  S:= 'Ahoj';  
  C:= S[2];    //Uloží do C hodnotu 'h';  
:
```

Obsah přednášky

- Vícerozměrná statická a dynamická pole
- Pole polí

Datový typ Indexu 1 a 2: Integer Datový typ buňky: Integer				
Index 2				
Index 1	1	2	..	200
1	8	5	..	29
2	11	14	..	41
:	:	:	∴	:
500	82	21	..	6

Datový typ Indexu: Integer Datový typ buňky: Pole string	
Index	Data
1	Datový typ Indexu: Integer Datový typ buňky: String
	Index Data
	1 'Ahoj'
	: :
n ₁	'!'
2	Datový typ Indexu: Integer Datový typ buňky: String
	Index Data
	1 'Dalsi pole'
	: :
n ₂	'Stringu!'
:	:
200	Datový typ Indexu: Integer Datový typ buňky: String
	Index Data
	1 'Posledni'
	: :
n ₂₀₀	'v poli!'

Vícerozměrná statická a dynamická pole

- Využívají dvou a více indexů
 - Indexy v hranatých závorkách jsou odděleny čárkami
- Mají dva a více rozměru
 - pro každý rozměr daný index
- Dvourozměrné pole lze reprezentovat tabulkou
 - První index je většinou řádek (Matematická konvence)
 - Volba je na programátorovi

Datový typ Indexu 1 a 2: Integer
Datový typ buňky: Integer

Index 2				
Index 1	1	2	..	200
1	8	5	..	29
2	11	14	..	41
:	:	:	∴	:
500	82	21	..	6

Vícerozměrná statická pole

- Deklarace vícerozměrného statického pole

- Dvourozměrné pole

```
JmenoPole: array  
[DolniIndex1..HorniIndex1,DolniIndex2..HorniIndex2] of DatovyTyp;  
Data: array [1..500,1..200] of Integer;
```

- Vícerozměrné pole deklarujeme analogicky

- Uložení hodnoty do pole


```
JmenoPole[Index1,Index2]:= Hodnota;  
Data[500,2]:= 21;
```

- Pro vícerozměrné pole analogicky

- Přístup k hodnotě v poli

```
JmenoPole[Index1,Index2];  
A:= Data[2,1]; //Do A vloží 11
```

- Pro vícerozměrné pole analogicky



Datový typ Indexu 1 a 2: Integer Datový typ buňky: Integer1				
Index 2				
Index 1	1	2	..	200
1	8	5	..	29
2	11	14	..	41
:	:	:	∴	:
500	82	21	..	6

Vícerozměrná dynamická pole

- Dvourozměrné dynamické pole

INDEXUJE SE OD 0!!!

- Deklarace dvourozměrného dynamického pole

```
JmenoPole: array of array of DatovyTyp;
```

```
Data: array of array of Integer;
```

- Nastavení rozměrů dvourozměrného dynamického pole

```
SetLength( JmenoPole, Rozmer1, Rozmer2);
```

```
SetLength( Data, 10, 8);
```

- Uložení hodnoty do pole

```
JmenoPole[Index1, Index2] := Hodnota;
```

```
Data[500, 2] := 21;
```

- Přístup k hodnotě v poli

```
JmenoPole[Index1, Index2];
```

```
A := Data[2, 1]; //Do A uloží hodnotu z třetího řádku a druhého sloupce
```

- Pro vícerozměrné pole analogicky

Vícerozměrná dynamická pole

- Funkce pro zjištění hranic vícerozměrného pole
 - Pro první rozměr stejně jako u jednorozměrných polí

```
A:= Low(JmenoPole);      //Do A uloží dolní index pole 1.rozměru  
B:= High(JmenoPole);    //Do B uloží horní index pole 1.rozměru  
C:= Length(JmenoPole);  //Do C uloží délku pole v 1.rozměru
```

- Pro další rozměry a indexy uvedu o jeden méně indexů než rozměr, který požadují

```
A:= Low(JmenoPole[0]);      //Do A uloží dolní index pole 2.rozměru  
B:= High(JmenoPole[0,0]);  //Do B uloží horní index pole 3.rozměru  
C:= Length(JmenoPole[0,0,0]); //Do C uloží délku pole v 4.rozměru
```

- Projití všech prvků dvourozměrného dynamického pole

```
for I:=0 to Length(Data) do      //Projdi všechny prvky v 1. rozměru  
begin  
  for J:=0 to Length(Data[I]) do //Projdi všechny prvky v 2. rozměru  
  begin  
    //  
  end;  
end;  
end;
```

Příklad na procvičení

- Vytvořte program, kterému uživatel zadá rozměr matice. Poté zadá jednotlivé prvky matice. Program vypíše matici a součet prvků na diagonále.



Pole polí

- Lze vytvořit pole, jehož prvky jsou pole
 - Statické i dynamické
 - Jedno i více rozměrné
 - Prvky mohou být pole
 - Statické i dynamické
 - Jedno i více rozměrné

Jednorozměrné pole indexované číslem (Integer), prvky jsou pole textů (Stringu) indexované číslem

Datový typ Indexu: Integer Datový typ buňky: Pole string		
Index	Data	
1	Datový typ Indexu: Integer Datový typ buňky: String	
	Index	Data
	1	'Ahoj'
	:	:
n_1	'!'	
2	Datový typ Indexu: Integer Datový typ buňky: String	
	Index	Data
	1	'Dalsi pole'
	:	:
n_2	'Stringu!'	
:	:	
200	Datový typ Indexu: Integer Datový typ buňky: String	
	Index	Data
	1	'Posledni'
	:	:
n_{200}	'v poli!'	

Jednorozměrné statické pole, jednorozměrných statických polí

- Deklarace

```
JmenoPole: array [DolniIndex1..HorniIndex1] of  
            array [DolniIndex2..HorniIndex2] of DatovyTyp;
```

```
Data: array [1..500] of array [1..200] of Integer;
```

- Uložení hodnoty do pole

```
JmenoPole[Index1][Index2] := Hodnota;
```

```
Data[500][2] := 21;
```

- Přístup k hodnotě v poli

```
JmenoPole[Index1][Index2];
```

```
A := Data[2][3]; //Do A uloží z 2. prvku 3. hodnotu
```

– Pro vícerozměrné pole analogicky

Jednorozměrné dynamické pole jednorozměrných dynamických polí

- Deklarace

```
JmenoPole: array of array of DatovyTyp;
```

```
Data: array of array of Integer;
```

- Nastavení délek

```
SetLength( JmenoPole, Rozmer);
```

```
SetLength( JmenoPole[Index], Rozmer); //Nastavení delky prvku Index
```

```
SetLength( Data, 10);
```

```
SetLength( Data[2], 5); //Pole prvku 2 bude mít 5 prvků
```

- Uložení hodnoty do pole

```
JmenoPole[Index1][Index2]:= Hodnota;
```

```
Data[500][2]:= 21;
```

- Přístup k hodnotě v poli

```
JmenoPole[Index1][Index2];
```

```
A:= Data[2][3]; //Do A uloží z 2. prvku 3. hodnotu
```

– Pro vícerozměrné pole analogicky

Jednorozměrné dynamické pole jednorozměrných statických polí

- Deklarace

```
JmenoPole: array of array [DolniIndex..HorniIndex] of DatovyTyp;
```

```
Data: array of array [1..15] of Integer;
```

- Nastavení délky

```
SetLength( JmenoPole, Rozmer);
```

```
SetLength( Data, 10);
```

- Uložení hodnoty do pole

```
JmenoPole[Index1][Index2] := Hodnota;
```

```
Data[8][2] := 21;
```

- Přístup k hodnotě v poli

```
JmenoPole[Index1][Index2];
```

```
A := Data[2][3]; //Do A uloží z 2. prvku 3. hodnotu
```

– Pro vícerozměrné a další kombinace statických a dynamických polí analogicky