

POČÍTAČE A PROGRAMOVÁNÍ

Datová pole, jednorozměrná
statická a dynamická datová pole,
String jako pole

Miroslav Vavroušek

Opakování z minulé přednášky

Cyklus s podmínkou na začátku,

Cyklus s podmínkou na konci

Cykly

- Cykly se používají k opakování zvolené části kódu
- V jazyku Pascal existují tři typy cyklů:
 - Iterační cyklus (**For – to – do**)
 - Cyklus s podmínkou na začátku (**While – do**)
 - Cyklus s podmínkou na konci (**Repeat – until**)
- Vhodný typ cyklu je zvolen na základě vlastností vyžadovaných v konkrétní situaci.

Cyklus s podmínkou na začátku

Cyklus while-do

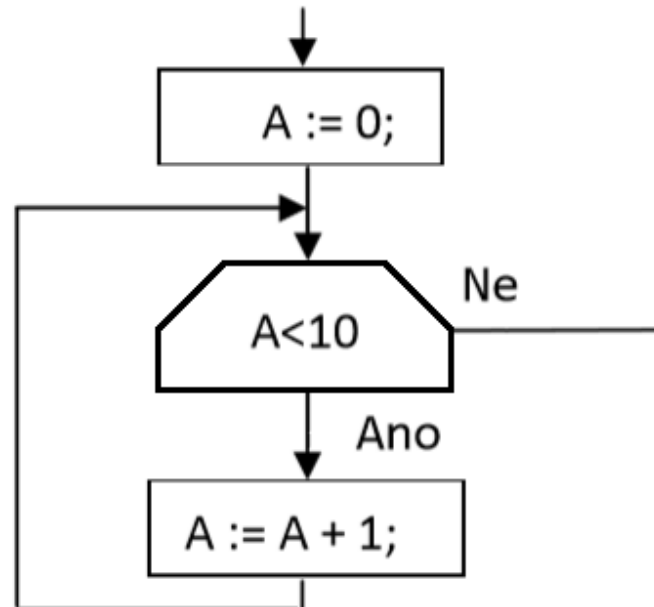
```
while Podmínka do //Podmínka  
begin  
    //Tělo cyklu  
end;
```

```
A := 0; //Iniciace proměnné
```

```
while A < 10 do //Podmínka  
begin  
    //Tělo cyklu
```

```
    A := A + 1; {Změna hodnoty proměnné  
                použité v podmínce.  
                Inkrementace A}
```

```
end;
```



Cyklus s podmínkou na konci

Cyklus repeat-until

```
repeat  
begin
```

```
  //Tělo cyklu
```

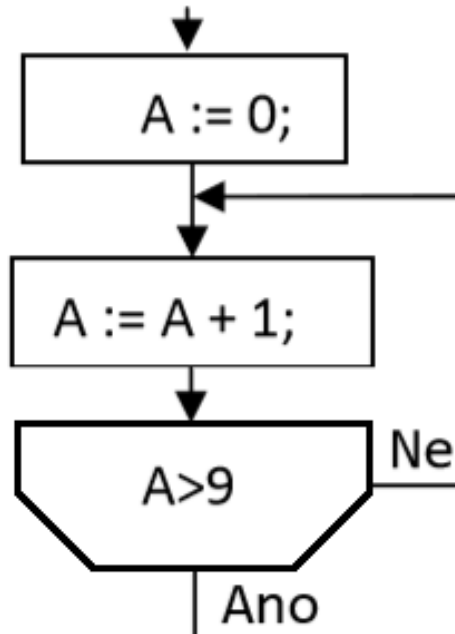
```
end until Podmínka;
```

```
A := 0; //Iniciace proměnné
```

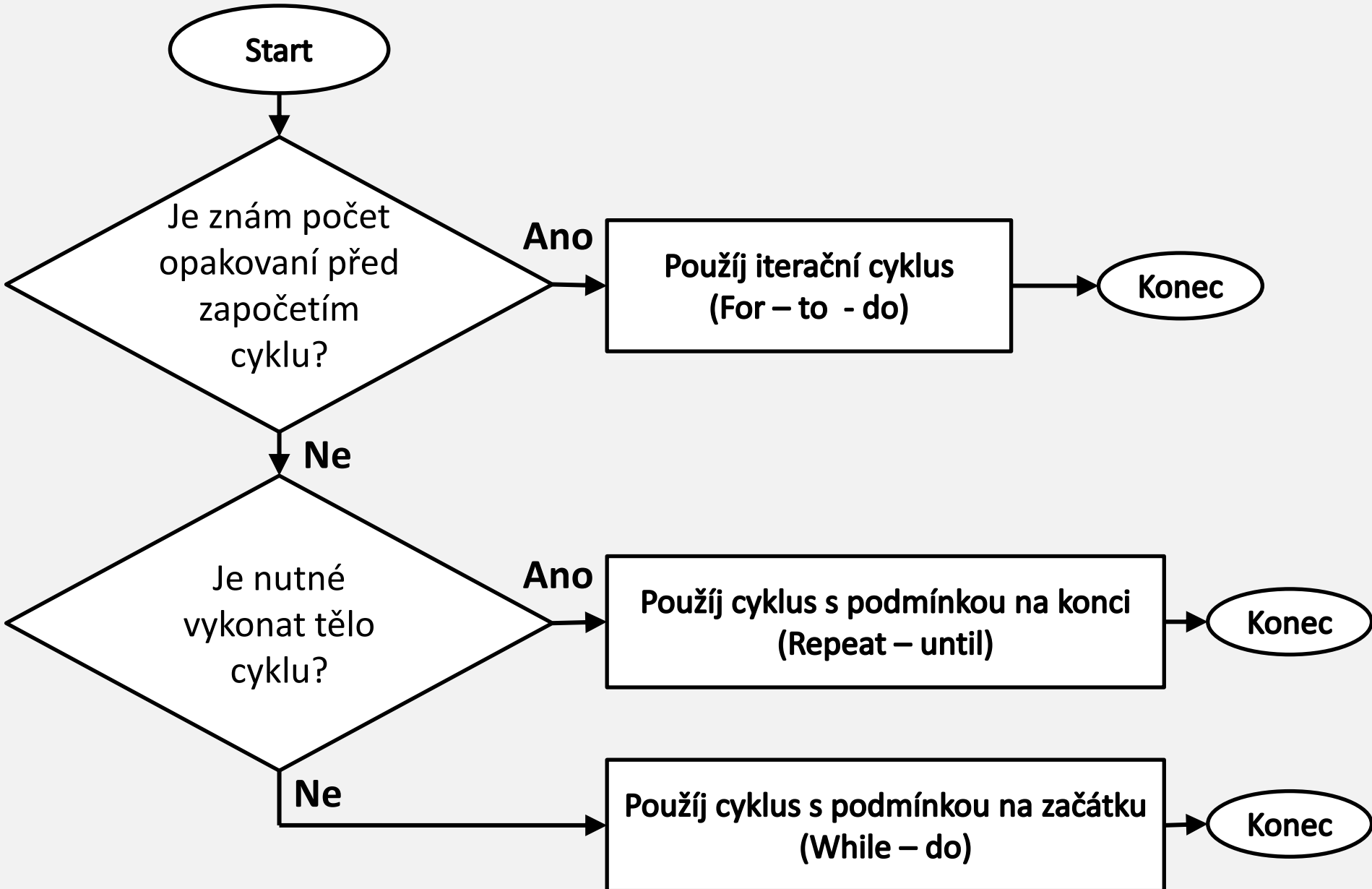
```
repeat  
begin
```

```
  A := A + 1; {Změna hodnoty proměnné  
              použité v podmínce.  
              Inkrementace A}
```

```
end until A > 9; //Podmínka
```



Jak zvolit správný typ cyklu?



Obsah přednášky

Datová pole,
jednorozměrná statická datová pole

Obsah 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'	'!'

Příklad k zamyšlení

- 1) Uživatel zadá řadu 10 čísel. Vypište řadu obráceně.
- 2) Uživatel zadá délku řady čísel a poté zadat řadu. Vypište řadu obráceně.

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.

Deklarace jednorozměrného statického pole

- V deklarační části programu
 - Stejně ostatní proměnné

```
Statická pole - deklarace  
JmenoPole: array [OD..DO] of  
JmenoDatovehoTypu;  
Pole1: array [1..15] of Integer;  
Pole2: array ['A'..'Z'] of String;
```

Jméno pole

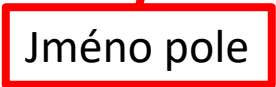

Rozsah indexů

Datový typ

Index	Typ
1	Integer
2	Integer
3	Integer
4	Integer
5	Integer
6	Integer
7	Integer
8	Integer
9	Integer
10	Integer
11	Integer
12	Integer
13	Integer
14	Integer
15	Integer

Použití jednorozměrného statického pole

- Využití pole pomocí jména a indexu!

Statická pole - použití	
JmenoPole[index]	
	
Pole1[5]:= 3; <i>{Do prvku s indexem 5 je uložena hodnota 3}</i>	
A:= Pole2['C']; <i>{Do proměnné A je uložena hodnota prvku s indexem 'C'}</i>	

Příklad k zamyšlení

Nyní víme jak na to!

- 1) Uživatel zadá řadu 10 čísel. Vypište řadu obráceně.
- 2) Uživatel zadá délku řady čísel a poté zadat řadu. Vypište řadu obráceně.

Dynamická jednorozměrná datová pole

- Mohou měnit svou délku
 - Délka není nastavena při kompilaci, ale při použití

Dynamická pole - deklarace
JmenoPole: array of DatovýTyp;
Data: array of String;

- Musím nastavit délku!

- Pomocí funkce

- Indexuje se 0!

Dynamická pole – nastavení délky
SetLength(JmenoPole, Delka);
SetLength(Data, 8); <i>//Pole bude mít indexi 0-7</i>

Dynamická jenorozměrná datová pole

Dynamická pole – ukládání dat do pole

```
SetLength(JmenoPole, Delka); //Musí nastavit délku!
```

```
JmenoPole[Index] := Hodnota  
JmenoPole[Index] := Hodnota  
:  
JmenoPole[Index] := Hodnota
```

```
SetLength(Data, 8); //Musí nastavit délku!
```

```
for I:=0 to 7 do //Projdu všech 8 položek  
begin ReadLN(Data[I]); end; {Uživatel zadá hodnotu do  
                             buňky pole s indexem I. (I  
                             bude postupně 0 až 7)}
```

Jednorozměrná datová pole

- Funkce pro zjištění hranic pole (statická i dynamická)

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

- Použití hodnot v dynamickém poli

– Shodné jako u statických polí

Statická pole - použití

JmenoPole[index]

Jméno pole

Indexů

```
Pole1[5]:= 3; {Do prvku s indexem 5 je uložena  
             hodnota 3}
```

```
A:= Pole2['C']; {Do proměnné A je uložena hodnota  
               prvku s indexem 'C'}
```


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';  
:
```