

POČÍTAČE A PROGRAMOVÁNÍ

Procedury a funkce – pokračování
Složené datové typy

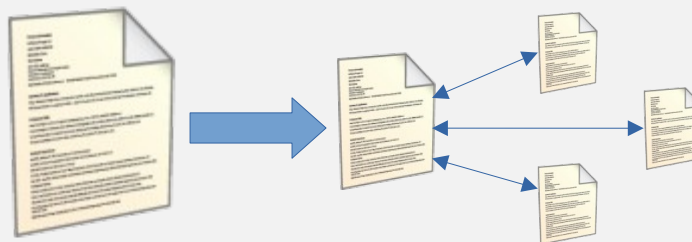
Miroslav Vavroušek

PPI 12

Opakování z minulé přednášky

Procedury a funkce

- Používají se k zapouzdření dílčích částí algoritmu
 - Vytváříme samostatné (menší) funkční celky
- Využití procedur a funkcí pomáhá k rozdělení a strukturování kódu
 - Vytváříme samostatné (menší) funkční celky
- Pomáhají také ke zvýšení přehlednosti a znovupoužitelnosti vytvořeného kódu
 - Vytváříme samostatné (menší) funkční celky :-)



Opakování z minulé přednášky

Procedury a funkce

Procedury **VS.** Funkce

Standardní procedury a funkce

VS.

Uživatelské procedury a funkce

Globální proměnné

VS.

Lokální proměnné

+ Překrývání názvů (proměnných)

Procedury a funkce - pokračování

- Parametry předávané hodnotou a odkazem
- Užitečné standardní procedury a funkce

Je „ReadLN“ procedura
nebo funkce?

Parametry předávané hodnotou a odkazem

- Parametry předávané hodnotou
 - Používáno v předchozích přednáškách
 - Parametry se chovají jako lokální promenné procedury/funkce
 - Změna hodnoty uvnitř procedury/funkce se neprojeví mimo tuto proceduru/ funkci

Parametry předávané **odkazem**

- Parametry se chovají jako odkazy na předané promenné
- Změna hodnoty uvnitř procedury/funkce se projeví změnou hodnoty proměnné na kterou odkaz míří

Parametry předvané odkazem

- Parametry předávané **odkazem**
 - Umožňují v proceduře a funkci nastavit více hodnot
 - Například ReadLN je procedura s parametry předávanými odkazem
 - Přečtená hodnota je uložena do proměnné, kterou předáme odkazem

→ Při deklaraci parametru je použito klíčové slovo **var**

```
procedure Soucet(A: Integer; B: Integer; var C: Integer);  
begin  
  C := A + B;  
end;
```

Volání:
Odkaz MUSÍ být proměnná!
NE konstanta!

```
Soucet( 5, 3, Vysledek);
```

Příklad 1.

Procedura s parametry
předávanými odkazem pro
převod kartézských souřadnic na
polární souřadnice v rovině

`pr12_01_PrevodSouradnic.pas`

Příklad 2.

Procedura s parametry
předávanými hodnotou a
odkazem s ukázkou platnosti a
překrývání proměnných

p

pr12_02_ParametryHodnotouA0dkazem.pas

Užitečné standardní procedury a funkce

- Deklarované a implementované ve standardních knihovna vývojového prostředí
 - Kompletní přehled v referenční příručce:
 - <https://www.freepascal.org/docs-html/rtl/index.html>

Název:	Obsahuje funkce, procedury a konstanty pro:
Crt	Práce s obrazovkou a klávesnicí
Math	Matematické funkce a konstanty
System	Systémové procedury funkce
SysUtils	Další systémové procedury a funkce
StrUtils	Procedury a funkce pro práci s textem

Užitečné standardní procedury a funkce

Knihovna: **Crt**

Název:	Parametry	Výsledek	Popis
ClrScr	Žádné	Žádné	Smaže obrazovku
Delay	Čas pozastavení v milisekundách	Žádné	Pozastaví vykonávání programu
DelLine	Žádné	Žádné	Smaže řádek
GotoXY	Souřadnice znaku X, Y	Žádné	Umístí kurzor na X, Y
TextBackground	Barva (číslo barvy 0 až 255) nebo název konstanty označující barvu	Žádné	Nastaví barvu pozadí
TextColor	Barva (číslo barvy 0 až 255) nebo název konstanty označující barvu	Žádné	Nastaví barvu textu
WhereX	Žádné	Pozice X na obrazovce	Určí pozici kurzoru v ose X
WhereY	Žádné	Pozice Y na obrazovce	Určí pozici kurzoru v ose Y

Složené datové typy (záznamy)

Složené datové typy (záznamy)

- Lze vytvářet složené datové typy
 - Z jednoduchých datových typů
 - Dříve definovaných složených typů

Složený datový typ	
<pre>type Jmeno = record JmenoPromenne: DatovyTyp; //adt. end;</pre>	<pre>type TBod = record X: Integer; Y: Integer; end;</pre>

Jméno: TBod	Vlastnosti:
	X: Integer;
	Y: Integer;

Složené datové typy (záznamy)

- K jednotlivým vlastnostem přistupujeme pomocí „.“

Přístup k jednotlivým vlastnostem

`Jmeno.JmenoAtributu`

Využití dříve definovaných záznamů

type

```
TBod = record //Základní bod
    X: Integer;
    Y: Integer;
end;
```

type

```
TCara = record {Přijetí dříve definovaných záznamů}
    Bod1: TBod;
    Bod2: TBod;
end;
```