

POČÍTAČE A PROGRAMOVÁNÍ

Procedury a funkce

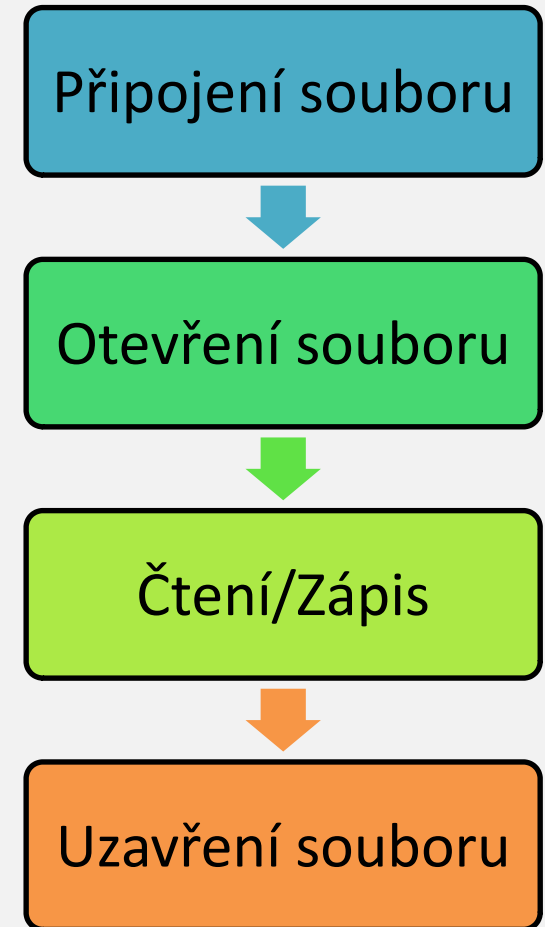
Miroslav Vavroušek

PPI 11

Opakování z minulé přednášky

Práce se soubory v jazyku Pascal

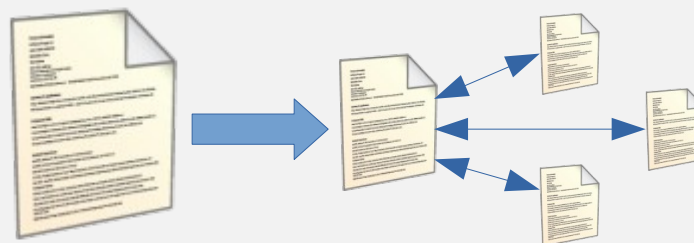
- Binární soubory
 - Soubor reprezentován jako řada bytů
 - Čten/Zapisován po blocích
 - **Struktura souboru je pevně definována**
 - Bez znalosti definice nelze soubor zpracovat
- Textové soubory
 - Soubor reprezentován jako text
 - Soubor lze předčíst v textovém editoru
 - Kódování ASCII
 - Konec řádku CR+LF
 - CR – Carriage Return (ASCII kód 13)
 - LF – Line feed (ASCII kód 10)
 - Čten/zapisován po znacích/řádcích



Procedury a funkce

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 :-)



Procedury a funkce

- Procedury

- Nemají návratovou hodnotu

- Nevrací volajícímu žádnou informaci

- Jedná se o předpis toho, co se má udělat

- Například: Návod, Kuchařka

- Například: Clscr; Randomize; Reset(Soubor);

VS.

- Funkce

- Mají návratovou hodnotu

- Mají nějaký výsledek

- Jedná se o operaci s nějakým závěrem

- Například: Výpočet obvodu kruhu, Sinus

- Například: Random(Max); Sin(Cislo);

Standardní a uživatelské procedury a funkce

- Standardní procedury a funkce
 - Součást vývojového prostředí (prog. jazyka)
 - Uloženy ve standardních knihovnách
 - Popsány v referencích prostředí (prog. Jazyka)
 - Pro Free Pascal: <https://www.freepascal.org/docs-html/ref/ref.html>

VS.

- Uživatelské procedury a funkce
 - Nové procedury a funkce vytvořené programátorem při realizaci projektu
 - Dokumentaci vytváří tvůrce
 - Vlastní knihovny, knihovny třetích stran

Globální a lokální proměnné

- Globální proměnné
 - V deklarační části programu
 - Pokud jsou deklarované dříve než uživatelské funkce a procedury, lze je ve funkcích a procedurách použít
- Lokální proměnné
 - V deklarační části procedury/funkce
 - Existují pouze v proceduře/funkci
- Překrývání názvů (proměnných)
 - Pokud vytvořím lokální proměnnou stejného názvu jako globální → Překryjí ji!
 - Uvnitř procedury/funkce pracuji s lokální proměnnou
 - → nemohu se dostat ke globální proměnné stejného názvu

Uživatelské procedury

Definice uživatelské procedury

- Procedura obsahuje hlavičku, deklarační a implementační část

→ Zapisuje se v deklarační části programu

```
//Žádné parametry
//Žádné lokální proměnné
procedure NavezProcedury1;
begin
    //Implementace procedury 1
end;

//Jeden parametr
//Žádné lokální proměnné
procedure NavezProcedury2(Parametr: DatovyTyp);
begin
    //Implementace procedury 2
end;

//Dva parametry
//Jedna lokální proměnná
procedure NavezProcedury3(Par1: DatTyp; Par2:DatTyp);
var
    LokalniPromenna: DatovyTyp;
begin
    //Implementace procedury 3
end;
```

```
//Žádné parametry
//Žádné lokální proměnné
procedure VypisJmenoAutora;
begin
    WriteLN('Miroslav Vavrousek');
end;

//Jeden parametr
//Žádné lokální proměnné
procedure VypisTextHvezdy( Text: String);
begin
    WriteLN('***' + Text + '***');
end;

//Dva parametry
//Jedna lokální proměnná
procedure VypisObsahObdelniku(a: Integer; b:Integer);
var
    S: Integer;
begin
    S:= a*b;
    WriteLN('Obsah: ', S);
end;
```

Volání (využití) uživatelské procedury

- Vytvořenou proceduru používáme v implementační části programu
 - Používáme je shodně jako standardní procedury
 - Procedury voláme pomocí názvu
 - Pokud procedura využívá parametry musíme definovat jejich hodnotu

```
//Implementační část programu
NazevProcedury1; //Volání procedury NazevProcedury1
NazevProcedury2(Parametr); //Volání procedury NazevProcedury2
NazevProcedury3(Parametr1, Parametr2); //Volání procedury NazevProcedury3

//Implementační část programu
VypisJmenoAutora; //Volání procedury VypisJmenoAutora
VypisTextHvezdy('Ahoj'); //Volání procedury VypisTextHvezdy
VypisObsahObdelniku( 5, 8); //Volání procedury VypisObsahObdelniku
```

Uživatelské funkce

Definice uživatelské funkce

- Funkce obsahuje hlavičku (s návratovým typem), deklarační a implementační část

→ V deklarační části programu (Nazvu přiřadím výsledek)

```
//Žádné parametry
//Žádné lokální proměnné
function NazevFunkce1: NavratovyDatovyTypFunkce;
begin
  NazevProcedury1:= Hodnota; //Vracení výsledku fce
end;

//Jeden parametr
//Žádné lokální proměnné
function NazevFce2(Parametr: DatTyp): NavDatTypFce;
begin
  //Implementace funkce 2 (Musíme vrátit výsledek)
end;

//Dva parametry
//Jedna lokální proměnná
function NazFce3(Par1:DatTyp; Par2:DatTyp): NavDTypF;
var
  LokalniPromenna: DatovyTyp;
begin
  //Implementace funkce 3 (Musíme vrátit výsledek)
end;
```

```
//Žádné parametry
//Žádné lokální proměnné
function PocetPisJmenaAutora: Integer;
begin
  PocetPisJmenaAutora:= Length('Miroslav Vavrousek');
end;

//Jeden parametr
//Žádné lokální proměnné
function NaDruhou(Cislo: Integer): Integer;
begin
  NaDruhou:= Cislo*Cislo;
end;

//Dva parametry
//Jedna lokální proměnná
function Prumer(Soucet:Integer; Pocet:Integer): Real;
var
  P: Real;
begin
  P:= Soucet/Pocet;
  Prumer:= P;
end;
```

Volání (využití) uživatelské funkce

- Vytvořenou funkci používáme v implementační části programu
 - Používáme je shodně jako standardní funkce
 - Funkce voláme pomocí názvu
 - Funkce vrací výsledek
 - Pokud funkce využívá parametry musíme definovat jejich hodnotu

```
//Implementační část programu
```

```
Promenna1:= NazevFunkce1;           //Volání funkce NazevFunkce1  
Promenna2:= NazevFce2(Parametr);    //Volání funkce NazevFce2  
Promenna3:= NazFce3(Parametr1, Parametr2); //Volání funkce NazFce3
```

```
//Implementační část programu
```

```
Vysledek1:= PocetPisJmenaAutora;    //Volání procedury PocetPisJmenaAutora  
Vysledek2:= NaDruhou(6);           //Volání procedury NaDruhou  
Vysledek3:= Prumer( 40, 8);        //Volání procedury Prumer
```