

PODPROGRAMY

PODPROGRAMY - jednoduché programové celky, které využíváme při tvorbě složitějších programů (rozdělávání na díly, podoblasty, místní se spakují)

- lze je **POJHENOVAT** a jejich jinou možnost pro jejich aktivaci (spuštění) v programu tam, kde potřebujeme činnost provést
- **POPSÁNY A POJHENOVANÝ** v části definicí a deklarací
- **DRUHY:** PROCEDURY, FUNKCE
- v jazyku Pascalu můžeme **PŘEDEFINOVÁNÝ** (jsou součástí TP) (nemusíme ji definovat) - můžeme ji mít v jiném programu (např. mod.)

1. Procedury

- VÝSLEDKEM procedury je PROVEDENÍ nějaké ČINNOSTI
- DEKLARACE v části bloku na úrovni deklarací proměnných (var) v hlavním programu
- otevření se shledává s HLAVIDKY a BLOKU (TELU) - má stejnou strukturu jako blok programu
- procedura identifikátor (specifikace parametrů);
blok; máme procedury
- DRUHY: PROCEDURY BEZ PARAMETRŮ, PROCEDURY S PARAMETRY

(A) PROCEDURY BEZ PARAMETRŮ

- nejdnodušší typ
- DEKLARACE
 - procedura identifikátor;
 - deklarace lokální;
 - begin
 - postupnost příkazů
 - end;

{ můžeme tu mít, deklarace lokální nemusí být }

UŽITÍ:
program POKUS;
uses CRT;
var ...;
procedure MOJE;
var ...;
begin
...
end;
{ HLAVIDLO PROGRAMU }
begin
...
MOJE;
...
end.
} DEKLARACE PROCEDURY
→ AKTIVACE (VYVOLÁNÍ)
PROCEDURY

Úkoly

① Napište proceduru, která vypíše na řádku 10 hvězdic

```
procedura HVEZDICKY;
begin
    writeln ('*****');
end;
{lokální deklarace není}
```

```
procedure HVEZDICKY;
var I: integer; {lokální deklarace}
begin
    for I := 1 to 10 do
        write ('*');
    writeln;
end;
```

② Využijte proceduru HVEZDICKY v programu pro

- a) výpočet objemu válce,
kterého je polomír a
jeho výška

```
program VALEC (input, output);
uses CRT;
var VYSKA, R, OBJEM: real;
procedure HVEZDICKY;
begin
    writeln ('*****');
end;
{HLAVNÍ PROGRAM}
begin
    clscr;
    HVEZDICKY;
    writeln ('OBJEM VÁLCE');
    HVEZDICKY;
    HVEZDICKY;
    writeln ('Kadív polomír v cm:');
    readln (R);
    writeln ('Kadív výška v cm:');
    readln (VYSKA);
    OBJEM := PI * R * R * VYSKA;
    HVEZDICKY;
    writeln ('Objem válečku je ', V:1:3,
            ' cm krychloviných.');
    HVEZDICKY;
    readln
end.
```

procedura kde využila pro
opakování činnosti slysi

- b) součet přirozených čísel
menších nebo rovních N

1 2 3 4 5	$S := 0$	$N = 5$	$\text{for } I := 1 \text{ do } N$
	$S := S + I$		$\text{dilij } S := S + I$
	$S := S + 2$		(mohou mít i slevy + i)
	!		

program SOUCET (input, output);

uses CRT;

var N, I, S: integer;

procedure HVEZDICKY;

begin

writeln ('*****');

end;

{HLAVNÍ PROGRAM}

begin

clscr;

HVEZDICKY;

writeln ('SOUČET PŘIROZENÝCH
čísel');

HVEZDICKY;

HVEZDICKY;

writeln ('Kadív přirozené
číslo N:');

readln (N);

S := 0;

for I := 1 to N do S := S + I;

HVEZDICKY;

writeln ('Součet přiroz.
čísel menších nebo

rovných ', N, ' je ', S, '!');

HVEZDICKY;

readln

end.

③ Napište program, který dvě zadaná čísla návadí podle velikosti (násoby). Řešení můžete procedurou.

úloha vypadá takto, má 3 podílony:

NACJENI	→ procedura NACJENI
SERAZENI	SERAZENI
TISK	TISK

```

program RAZENI_DVOU_CISEZ (input, output);
uses CRT;
var A,B : integer;
procedure NACJENI;
begin
  writeln('Když zde máme čísla: ')
  readln (A,B)
end;
procedure SERAZENI;
var POM: integer;
begin
  if A > B then
    begin
      POM := A;
      A := B;
      B := POM
    end;
  procedure TISK;
  begin
    writeln('Určitáme'
    ' čísla jsou: ', A, ', ', B);
  end;
{HLAVNÍ PROGRAM}
begin
  choice;
  NACJENI;
  SERAZENI;
  TISK
  readln
end.

```

VÝMĚNA DVOU ČÍSEL

$$\begin{array}{l} A: \boxed{5} \\ B: \boxed{3} \end{array} \rightarrow \begin{array}{l} A: \boxed{3} \\ B: \boxed{5} \end{array}$$

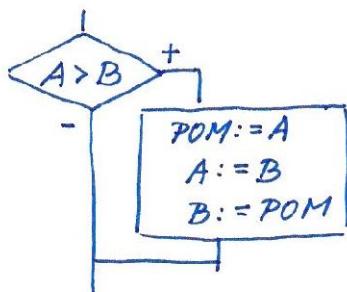
A := B	A	B
B := A	5	3
NEZDE!	A := B	3
	B := A	3
		3

MUSÍME POUŽÍT POMOCNOU
PROHĚNNOU

A	B	POM
5	3	
POM := A	5	5
A := B	3	3
B := POM	3	5

nebo
 $POM := B$
 $B := A$
 $A := POM$

A, B... GLOBALNÍ PROMĚNNÉ (platí v celém programu)
POM... LOKÁLNÍ PROMĚNNÁ (platí jen v proceduře SERAZENI)



algoritmus pro určení čísel A,B
podle velikosti (násobené)

V PROCEDURE má užív dva typy objektů - např. proměnné, konstanty, ...

- LOKÁLNÍ (PROMĚNNÉ)

- definují se v deklarační části procedury - LOKÁLNÍ DEKLARACE

- např. v p. ① - proměnná I je z. npr.

v p. ③ - pomocná proměnná POM

v p. ④ - proměnné X, Y (návaznost); kterou můžete o hodnoty proměnných A, B v hlavním programu

- mají platnost jen v říle procedury (v hlavním programu nejsou dosluhováni)

- GLOBÁLNÍ (PROMĚNNÉ)

- definují se v hlavním programu

- mají platnost v celém programu

- pokud v procedurách proměnná stejná jako v hlavním programu, pak v procedurách platí lokální proměnné (objekty)

- říle můžete mít mnoho lokálních objektů jichž jsou místním označením objektů, mohou v okamžiku realizace procedury existující lokální proměnná činí místosluhovou (nastíněnou) odpovídající globální proměnné stejného jména

- např. A, B, DEZ v p. ④

Úkoly

④ EUKLEIDŮV ALGORITMUS

Pomocí Euklidova algoritmu pro určení největšího společného děliteli 2 kladných cílových čísel vytvořte proceduru NSD a užijte ji v programu.

EUKLEIDŮV ALGORITMUS PRO URČENÍ NSD

Máme dvojici přirozených čísel X, Y. Pokud je dvojice i kladnými čísly, pak vytvoříme dvojici novou tak, že místo většího čísla napíšeme rozdíl menšího čísla a menšího, a menší číslo opíšeme. Nové dvojici vytváříme tak dletoče, ak poslední číslo stojí. Nejdřív vytváříme tak dletoče, ak poslední číslo stojí. Nejdřív vytváříme tak dletoče, ak poslední číslo stojí.

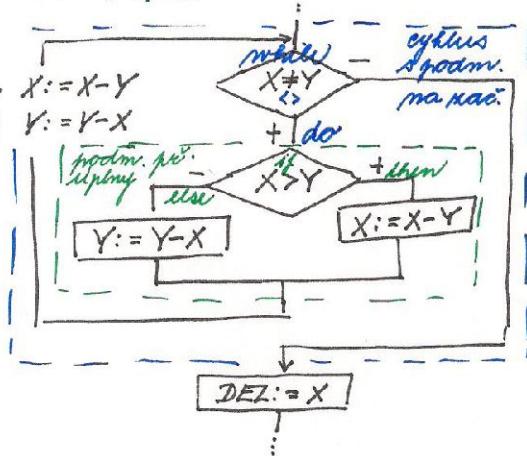
př. $(24, 54) \rightarrow (24, 30) \xrightarrow{54-24} (24, 6) \xrightarrow{30-24} (18, 6) \xrightarrow{18-6} (12, 6) \xrightarrow{12-6} (6, 6) \xrightarrow{6-6} (0, 6)$
KONEC
 $\underline{\underline{NSD(24, 54) = 6}}$

př. $(123, 546) \rightarrow (123, 423) \rightarrow (123, 300) \rightarrow (123, 177) \rightarrow (123, 54) \rightarrow (123, 42) \rightarrow (123, 12) \rightarrow (123, 3) \rightarrow (120, 3) \rightarrow (117, 3) \rightarrow (114, 3) \rightarrow (111, 3) \rightarrow (108, 3) \rightarrow (105, 3) \rightarrow (102, 3) \rightarrow (99, 3) \rightarrow (96, 3) \rightarrow (93, 3) \rightarrow (90, 3) \rightarrow (87, 3) \rightarrow (84, 3) \rightarrow (81, 3) \rightarrow (78, 3) \rightarrow (75, 3) \rightarrow (72, 3) \rightarrow (69, 3) \rightarrow (66, 3) \rightarrow (63, 3) \rightarrow (60, 3) \rightarrow (57, 3) \rightarrow (54, 3) \rightarrow (51, 3) \rightarrow (48, 3) \rightarrow (45, 3) \rightarrow (42, 3) \rightarrow (39, 3) \rightarrow (36, 3) \rightarrow (33, 3) \rightarrow (30, 3) \rightarrow (27, 3) \rightarrow (24, 3) \rightarrow (21, 3) \rightarrow (18, 3) \rightarrow (15, 3) \rightarrow (12, 3) \rightarrow (9, 3) \rightarrow (6, 3) \rightarrow (3, 3)$
KONEC
 $\underline{\underline{NSD(123, 546) = 3}}$

odvození pěstání

$$(24, 18) \xrightarrow[X > Y]{=} (6, 18) \xrightarrow[X := X - Y]{X < Y} (6, 12) \xrightarrow[X < Y]{=} (6, 6) \xrightarrow[Y := Y - X]{X = Y \text{ KONEC}} (0, 6) \xrightarrow{DEZ := X}$$

ryvn. cyklus: dokud $X \neq Y$ díly
jistíme $X > Y$, pak
jinak ($\Leftrightarrow X < Y$)



program EUKLEIDES (input, output);

uses CRT; {nežádoucí CRTP23}

var A, B, DEZ: integer;

procedure NSD;

var X, Y: integer;

begin

$X := A$; $Y := B$; {pomocí nichož pev. př. dostaneme hodnoty prom. A, B do procedury}

while $X \neq Y$ do

if $X > Y$ then $X := X - Y$

else $Y := Y - X$;

$DEZ := X$

end;

{HLAVNÍ PROGRAM}

begin

clrscr;

writeln(' * NEJVĚTŠÍ SPOLEČNÝ DELITEL * '');

writeln(' pomocí Euklidova algoritmu');

writeln(' zadaj 2 přirozená čísla A, B:');

readln(A, B);

NSD;

writeln(' největší společný dělitel čísel ', A, ', ', B, ', je ', DEZ, '!');

readln {repeat until keypressed};

end.

(B) PROCEDURY S PARAMETRY

- mají svůj použití
- DEKLARACE (v hlavním programu na deklaraci proměnných)

procedure NAZEV PROCEDURE (návazem specifikace formálních parametrů);
lokalní deklarace;
begin
postupnost příkazů
end;

- VOLÁNÍ (AKTIVACE)

- v hlavním programu v místě, kde činnost potřebujeme provést
{HLAVNÍ PROGRAM}
begin

NAZEV PROCEDURE (návazem skutečných parametrů); → VOLÁNÍ PROCEDURE
end.

(B) UKÁZKA - podíl 2 celých čísel A,B (B≠0)

program PODIL_CISEL (input, output);
uses CRT;
var A,B: integer; P: real;
procedure PODIL (X,Y: integer; var Z: real);
{var ...}
begin
Z := X/Y
parametry
rotání hodnotou parametr rotány
odkazem
end;
{HLAVNÍ PROGRAM}
begin
clsce;
writeln ('Kadiv celá čísla A,B, B≠0');
readln (A,B);
PODIL (A,B,P); → rotání procedure, A,B,P skuteční
writeln (A, ':', B, '=', P); parametry
readln
end.

PARAMETRY,

- SKUTEČNÉ
- FORMÁLNÍ rotání HODNOTOU
rotání ODKAZEM

PARAMETRY,

- SKUTEČNÉ

- použírají se v hlavním programu až při volání procedury
- jsou tvořeny parametry, se kterými se má skutečně procedura provádět
- psíme je na místem procedury při aktivaci a pořadí, v jakém jsou uvedeny odpovídající formální parametry a procedury

MOJE PROC (A,B,C);
SKUTEČNÉ PARAMETRY

- musí být deklarovány v deklatační části hlavního programu a slyšitelné dekorativní typu jisté odpovídající formální parametry volání hodnotou nbo aktivační

- FORMÁLNÍ

- ustanovují (mávají) v proceduře formální skuteční parametry (v okamžiku deklarace procedury skuteční parametry již nemáme)

- TYPY:

- VOLANÉ HODNOTOU

- používáme pro zadávání vstupních hodnot do procedury
- představují tedy lokální proměnnou, která se má načítat předáním procedury přiřadit hodnotu odpovídajícího skutečného parametru
- hodnota parametru v průběhu vykonání procedury může zmínit, může to být na hodnotu odpovídajícího skutečného parametru
- hodnoty lato do procedury „vnásíme“, ale nevezme „vnásí“ ven do hlavního programu

- SPECIFIKACE : procedură NAZEV (X,Y:integer; W:real); [VAR!]

X,Y,Z...jedna z parametrů

typ - integer, real, ...

- VOLANÉ ODKAZEM

- používáme pro vložení hodnoty proměnných v proceduře jejich hodnoty budou provedením procedury definované (vyrobeny)
- můžeme použít i pro vložení hodnot, které se v průběhu procedury můžou změnit a tím dojde i ke změně odpovídajícího skutečného parametru v hlavním programu

- SPECIFIKACE : procedură NAZEV (var K,L:integer; var M:real);

! mluv (var K,L:integer; M:real);

? NUTNO VAR!

→ M by byl parametr
volání hodnotou

⑤ Jaké hodnoty se vypisují v programech?

a) program VYPIS_1;

var X,Y: integer;

procedure PROC(R,S:integer);

begin

R:=10; S:=20

end;

{HLAVNÍ PROGRAM}

begin

X:=1; Y:=2;

writeln(X,Y)

end.

b) program VYPIS_2;

var X,Y: integer;

procedure PROC(var R,S:integer);

begin

R:=10; S:=20

end;

{HLAVNÍ PROGRAM}

begin

X:=1; Y:=2;

writeln(X,Y)

end.

c) program VYPIS_3;

var X,Y: integer;

procedure PROC(R:integer; var S:integer);

begin

R:=10; S:=20

end;

{HLAVNÍ PROGRAM}

begin

X:=1; Y:=2;

writeln(X,Y);

X:= 2*X+Y;

writeln(X,Y)

end.

d) program VYPIS_4;

var X,Y: integer;

procedure PROC(var R:integer;
S:integer);

begin

R:=10; S:=20

end;

{HLAVNÍ PROGRAM}

begin

X:= 1; Y:= 2;

writeln(X,Y);

Y:= 3*X+Y;

writeln(X,Y)

end.

⑥ Napište program pro náminu z dvou čísel obsahující proceduru s parametry

A:

 A:

 B:

 B:

POM := A
 A := B
 B := POM

a) S PARAMETRY

PROGRAM ZAMENA (input, output);

uses CRT;

var X, Y: integer;

PROCEDURE ZAHEN (var A, B: integer);

var POM: integer; FORMÁLNÍ PARAM.

begin

POM := A;

A := B;

B := POM

end;

{HLAVNÍ PROGRAM}

begin

clrscr;

writeln ('ZAMENA');

writeln ('Zadej 2 cílové čísla:');

readln (X, Y);

ZAHEN (X, Y); X, Y SKUTEČNÉ PAR.

writeln ('AKTIVACE PROCED. S PARAM.'

readln ('X= ', X, ' a Y= ', Y);

readln

end.

b) BEZ PARAMETRŮ

PROGRAM ZAMENA (input, output);

uses CRT;

var X, Y: integer;

PROCEDURE ZAHEN;

var A, B, POM: integer;

begin

A := X, B := Y; hodnoty do procedury

POM := A;

A := B;

B := POM;

X := A; Y := B hodnota původní globální prom. → do

end;

{HLAVNÍ PROGRAM} hlavní program

begin

clrscr;

writeln ('ZAMENA');

writeln ('Zadej 2 cílové

čísla:');

readln (X, Y);

ZAHEN; → AKTIVACE PROC.

BEZ PAR.

readln ('X= ', X,

' a Y= ', Y);

readln

end.

- ⑦ napište program, v němž vyříšíte proceduru s parametry pro převodní vzdálenosti v metech na kilometry a m.

$$\left[\begin{array}{l} \text{VZD} \\ 5472 \text{ m} : 1000 = 5 \text{ km} \\ 472 \text{ m} \\ M := VZD \bmod 1000 \end{array} \right]$$

```

program PREVOD_VZD (input, output);
uses CRT; {nejdou CRTD23}
var VZDALENOST, KILOMETRY, METRY: integer;
procedure PREVOD (VZD: integer; var KM, M: integer);
begin
  KM := VZD div 1000;
  M := VZD mod 1000;
end;
{HLAVNÝ PROGRAM}
begin
  clc;
  writeln ('* PREVOD VZDÁLENOSTI *');
  writeln ('Kadív zadaj vzdálenost, m mluvčí:');
  readln (VZDALENOST);
  PREVOD (VZDALENOST, KILOMETRY, METRY);
  writeln (VZDALENOST, 'm', KILOMETRY, 'km a', METRY, 'm');
  readln
end.

```

- ⑧ napište procedury převádějící časový údaj T v sekundách (celých) na údaj v hodinách, minutách a sekundách (celá čísla). Připočtěte deklaraci proměnných v hlavním programu a volání procedury (s parametry). Odvodlu pro 8063 s.

$$\left[\begin{array}{l} T := 8063 \text{ s} \\ 8063 : 60 = 134 \text{ min} \\ 206 \\ 263 \\ 23 \text{ s} := T \bmod 60 \quad 14 \text{ min} \quad M := M \bmod 60 \\ M := T \bmod 60 \end{array} \right]$$

```

program PREVOD_CASU (input, output);
var UDAJ, SEK, MIN, HOD: integer;
procedure PREVOD (T: integer; var S, M, H: integer);
begin
  M := T div 60;
  S := T mod 60;
  H := M div 60;
  M := M mod 60
end;

```

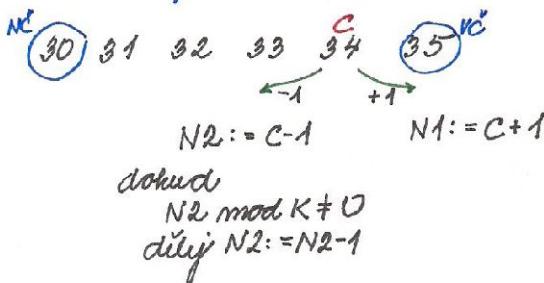
```

{HLAVNÝ PROGRAM}
begin
  PREVOD (UDAJ, SEK, MIN, HOD);
  writeln (UDAJ, 's', HOD, 'h', MIN, 'min', SEK, 's');
end.

```

- ⑨ Napište proceduru s parametry, která nadaným početem řádků C uží výběr nejmenší násobku pravoúhlého čísla N1 a největší ménší pravoúhlé číslo N2 dle kterého číslu K.

Orálek pro $C = 34, K = 5$



```
procedure DELITK (C, K: integer);
var N1, N2: integer;
begin
  N1 := C+1;
  N2 := C-1;
  while N1 mod K > 0 do N1 := N1+1;
  while N2 mod K > 0 do N2 := N2-1;
```

program NEJBL_VYS_NIZ_DELIT_K (input, output);

uses CRT;

var C, K, N1, N2: integer;

procedure DELITK (C, K: integer; var N1, N2: integer);

begin

$N1 := C+1;$

parametry/vratení
podnávání

parametry/vratení
formální

parametry/vratení
odlakování

$N2 := C-1;$

while $N1 \bmod K > 0$ do $N1 := N1 + 1$;

while $N2 \bmod K > 0$ do $N2 := N2 - 1$;

end;

{HLAVNÍ PROGRAM}

begin

clrscr;

writeln ('NEJBLÍŽÍ VÝSOKÉ A NÍZKÉ ČÍSLO');

writeln ('zadej číslo C a číslo K');

readln (C, K);

DELIT (C, K, N1, N2); \rightarrow SKUTEČNÉ PARAMETRY

writeln ('nejblížší ménší číslo je ', N2, '!');

writeln ('nejblížší výšší číslo je ', N1, '!');

readln {repeat until keypressed};

end.

promítni tvořenec rámci

var CISLO, K, X1, X2: integer;

:

DELIT (CISLO, K, X1, X2)

a uprav tvoř. program

