

## 2. hodina: pokračování práce s poli, cykly, procedury & funkce

### rebin

- /sample
- diskuze opakování posl. prvku při zvětšení (neextrapoluje)

### congrid

- napsaná v IDL (poznámka o tom, co to znamená)
- na rozdíl od „rebin“ nemusí být celočíselný násobek
- na rozdíl od „rebin“ při zmenšování rozměru neprůměruje, jen resampluje
- /center, /interp, /minus\_one

### interpolate

Příklad: čísla od 1 do 5 s krokem 0.2

a) 1. + findgen(21)\*0.2

b) congrid([1.,5.],21,/interp,/minus\_one)

c) interpolate([1.,5.],findgen(21)/20)

Poznámka: IDL skript (@jmeno)

Poznámka: hlavní program (procedura), překlad, spuštění, CTRL+C, stop, breakpoint

### podmínka „if“

•Význam: základní větvící podmínka

•Syntaxe:

if podmínka then příkaz

if podmínka then příkaz1 else příkaz2

•Příklady:

if (x lt 0) then print,'zaporne'

if (x lt 0) then begin

print,'zaporne'

...

endif else begin

print,'kladne nebo nula'

...

endelse

## **podmínka „case“**

- Význam: v závislosti na hodnotě výrazu se provede jeden z možných příkazů

- Syntaxe:

case výraz of

    hodnota1: příkaz1

    hodnota2: příkaz2

    hodnota3: příkaz3

    else: příkaz

endcase

- Poznámka: pokud výraz odpovídá několika různým hodnotám, provede se příkaz odpovídající první z nich.

## **podmínka „switch“**

- Význam: v závislosti na hodnotě výrazu se provede jeden/několik z možných příkazů

- Syntaxe:

switch výraz of

    hodnota1: příkaz1

    hodnota2: příkaz2

    hodnota3: příkaz3

    else: příkaz

endswitch

- Poznámka: je proveden první příkaz, jehož hodnota odpovídá zadanému výrazu a všechny následující příkazy. Pro zamezení provádění následujících příkazů je možno využít příkaz "break".

## **cyklus „for“**

- Význam: provede daný příkaz/blok příkazů tolikrát, kolikrát je stanoveno

- Syntaxe:

for proměnná = od, do (, krok) do příkaz

- Příklady:

for i=0,9 do print,i

for i=9,0,-1 do begin

    print,i

    ...

endfor

- Poznámka: proměnná cyklu má stejný typ jako počáteční hodnota. Pro dlouhé cykly (více než 32767 opakování) je proto nutné zvolit typ long (např. for i=0L,100000 do ...)

## **cyklus „while“**

- Význam: provádí daný příkaz/blok příkazů dokud je splněna podmínka

- Syntaxe:

while podmínka do příkaz

- Příklady:

```
while not eof(lun) do begin  
  (čtení ze souboru "lun")  
endwhile
```

- Poznámka: při nesplnění podmínky není příkaz proveden ani jednou.

## **cyklus „repeat“**

- Význam: provádí daný příkaz/blok příkazů dokud není splněna podmínka

- Syntaxe:

```
repeat příkaz until podmínka
```

- Příklady:

```
repeat begin  
  ...  
  x++  
endrepeat until (x lt 10)
```

- Poznámka: příkaz je vždy proveden alespoň jednou.

Definice struktur a přístup k nim:

```
a = {a:0, b:", c:fltarr(4)}
```

```
print,a.c
```

```
x = replicate(a,5)
```

```
help,x,/struct
```

## **Definice procedur a funkcí v IDL**

Funkce explicitně vrací hodnotu, procedura může předat výsledek jen prostřednictvím parametrů.

## Proměnné uvnitř procedury/funkce

- parametry funkce/procedury (poziční a klíčové)
- systémové proměnné (!. )
- proměnné z bloku COMMON  
`COMMON` jmeno\_bloku,var1,var2,...
- lokální (všechny ostatní proměnné považovány za lokální)

## Typy parametrů

### Poziční

- povinné
- nepovinné (aktuální počet přes `N_PARAMS( )`, parametry lze ubírat zprava)  
`pro` nazev\_procedury, par1,par2,...

### Klíčové

- klíčová slova (nezavislý odkaz přes jméno klíčového parametru)  
`pro` nazev\_procedury, kpar\_jmeno=kpar\_odkaz

## **Příklady**

```
;definice procedury s jedním povinným parametrem
pro FCEX1, x
  print, 'sin(', x, ')=' , sin(x)
end
;volani
FCEX1, !pi
```

```
;definice procedury se dvema povinnými parametry
pro FCEX2, x, y
  print, 'sin(', x, ')=' , sin(x)
  print, 'cos(', y, ')=' , cos(y)
end
;volani
FCEX2, !pi, !pi/4
```

```

;define procedure s dvema nepovinnymi parametry
pro FCEX3, x, y
  if N_PARAMS() eq 0 then return
  print, 'sin(', x, ')=' , sin(x)
  if N_PARAMS() gt 1 then print, 'cos(', y, ')=' , cos(y)
end
;volani
FCEX3, !pi, !pi/4
FCEX3, 0
FCEX3

;define procedure s klicovym parametrem
pro FCEX4, x, SINX=s
  if keyword_set(s) then print, 'sin(', x, ')=' , sin(x)
end
;volani
FCEX4, !pi
FCEX4, !pi, /S
FCEX4, !pi, /SI
FCEX4, !pi, /SIN
FCEX4, !pi, /SINX
FCEX4, !pi, SI=1

;define funkce s klicovymi parametry
function FCEX5, x, SINX=s, COSX=c
  if keyword_set(s) then return, sin(x)
  if keyword_set(c) then return, cos(x)
  return, !values.f_nan
end
;volani
print, FCEX5(!pi)
print, FCEX5(!pi, /SINX)
print, FCEX5(!pi, /COSX)
print, FCEX5(!pi, /C, S=1) ;vrati sin(x)

```

## **Předávání dat hodnotou a odkazem**

```

;procedura menici hodnotu formalniho parametru
pro H1, x
  print, x
  x=0
  print, x
end

H1, 8 ;hodnotou
H1, 2*x ;hodnotou
H1, x ;odkazem

```

### **Příklady:**

test  
 testfunc  
 abshodnota  
 mojesign