Strukturované příkazy větvení programu a programových cyklů (řídící struktury, řízení běhu programu)

Větvení programu – podmíněný příkaz if

if podmínka **then** příkaz;

Je – li podmínka pravdivá příkaz se provede, je – li nepravdivá příkaz se neprovede.

```
if podmínka then begin
                    Příkaz 1;
                    Příkaz 2;
                    Příkaz n-tý;
                    end;
```

Je – li podmínka pravdivá provede se posloupnost příkazů, je – li nepravdivá posloupnost příkazů se neprovede.

if podmínka **then** Příkaz1 **else** Příkaz 2;

Je – li podmínka pravdivá provede se Příkaz1 a příkaz 2 se neprovádí, je – li nepravdivá Příkaz1 se neprovádí, ale provádí se Příkaz 2.

```
if podmínka then begin
                    Příkaz1;
                    Příkaz2;
                    Příkaz n-tý;
                   end
            else begin
                    Příkaz 1;
                    Příkaz 2;
                    Příkaz n-tý;
                   end;
```

Je – li podmínka pravdivá provede se posloupnost příkazů za klíčovým slovem **then,** posloupnost příkazů za klíčovým slovem **else** se neprovádí. Je – li podmínka nepravdivá neprovede se posloupnost příkazů za klíčovým slovem **then,** posloupnost příkazů za klíčovým slovem **else** se provádí.

Větvení programu – příkaz case (přepínač)

Příkaz **case** vybírá a realizuje jednu z nabízených větví programu.

```
case proměnná of
    hodnota1: příkazy 1;
    hodnota2: příkazy 2;
    hodnotaN: příkazy N;
    else
     kód, který se provede pokud nevyhovuje ani jedna z předchozích možností.
 end;
```

Příkaz **case** se provádí tak, že se vyhodnotí proměnná a provede se ten příkaz, před kterým stojí stejná hodnota jako je hodnota proměnné. Pokud se má provést více jak jeden příkaz, musí se uzavřít mezí klíčová slova **begin** a **end**. Všimněte si že v konstrukci **case** chybí **begin** – není to chyba.

Příklad:
Příkaz case funguje jako přepínač a vybere jednu z možností. Zjistí hodnotu proměnné operator a provede ten příkaz, před kterým stojí hodnota uložená v proměnné operator.

```
case operator of
   opPlus: Vysledek := Cis1 + Cis2;
   opminus: Vysledek := Cis1 - Cis2;
   opkrat: Vysledek := Cis1 * Cis2;
   opDeleno: Vysledek := Cis1 div Cis2;
   opModulo: Vysledek := Cis1 mod Cis2;
 end;
```

Je-li např. v proměnné operator uložena hodnota opPlus, provede se příkaz na prvním řádku a ostatní se neprovádějí.

Příklad:
Case i **of** //Nejdříve se vyhodnotí hodnota selektoru větvení i
1 : Zn := 'A';//Je – li v selektoru větvení i uložena hodnota 1, přiřadí se do Zn hodnota 'A'
2 : Zn := 'B';//Je – li v selektoru větvení i uložena hodnota 2, přiřadí se do Zn hodnota 'B'
3 : Zn := 'C' //Je – li v selektoru větvení i uložena hodnota 3, přiřadí se do Zn hodnota 'C'
else
Zn := 'D';{Není – li v selektoru větvení i uložena hodnota 1,2 nebo 3, přiřadí se do Zn hodnota 'D'}
end;

Cyklus Repeat – until

Pomocí příkazů cyklu se předepisuje opakované provádění příkazu nebo posloupnosti příkazů.

Příkaz cyklu **Repeat – until** má následující tvar:

```
repeat
  příkaz1;
  příkaz2;
  príkaz3;
  příkazN-tý;
until podmínka;
```

Příkaz cyklu **repeat – until** se provádí tak, že se postupně provedou všechny příkazy, potom se vyhodnotí výraz udávající podmínku ukončení. Je – li hodnota podmínky false, cyklus se opakuje. Provádění cyklu repeat končí tehdy, až se

hodnota podmínky změní na true. Vnořená posloupnost příkazů se provede vždy aspoň jednou.

Cyklus While

Příkaz cyklu **While** má následující tvar:

while podmínka **do** příkaz;

Obsahuje – li tělo cyklu více příkazů, musí být vymezeny klíčovými slovy **begin** a **end**.

```
while podmínka do begin
                    příkaz1;
                    příkaz2;
                    příkazn-tý;
                  end;
```

Příkaz cyklu **while** se provádí tak, že se nejprve vyhodnotí výraz udávající podmínku opakování a je – li hodnotou tohoto výrazu true provede se příkaz uvedený za klíčovým slovem **do** a tato činnost se opakuje znovu. Provádění příkazu While končí když výraz udávající podmínku má hodnotu false. Je – li false hodnotou tohoto výrazu již na začátku provádění, pak se příkaz uvedený za **do** neprovede ani jednou. Chceme–li příkazem while předepsat opakované provádění posloupnosti více příkazů, je třeba na rozdíl od příkazu repeat vytvořit z této posloupnosti příkazů složený příkaz vymezený klíčovými slovy begin a end.

Cyklus For

Příkaz cyklu **For** má následující tvar:

```
for proměnná := počáteční hodnota  to koncová hodnota  do
begin
   Příkaz1;
   Příkaz2;
   Příkaz n – tý
end;
```

Cyklus **for** funguje tak, že se při prvním průběhu nastaví hodnota proměnné na počáteční hodnotu, zpravidla na 1. Pak se provede posloupnost příkazů a hodnota proměnné se automaticky zvětší o 1 a opět se skočí na první příkaz cyklu. To se opakuje tak dlouho dokud hodnota proměnné nedosáhne koncové hodnoty. Proměnnou může být libovolný ordinální typ.

Příklad:

```
x:= 0;
for i:= 1 to 5 do begin
                    x:=x+i;
                  end;
```

V tomto případě je možné klíčové slova begin a end vynechat, protože v těle cyklu je pouze jeden příkaz.

For proměnná **:=** počáteční hodnota **downto** koncová hodnota **do**
begin
    Příkaz1;
    Příkaz2;
    Příkaz n – tý
end;

Cyklus **for dowto** funguje tak, že se při prvním průběhu nastaví hodnota proměnné na počáteční hodnotu. Pak se provede posloupnost příkazů a hodnota proměnné se zmenší o 1 a opět se skočí na první příkaz cyklu. To se opakuje tak dlouho dokud hodnota proměnné nedosáhne koncové hodnoty. Proměnnou může být libovolný ordinální typ.

Příklad:
x:=0
for i:= 5 downto 1 do **begin**
    x:=x+1;
end;