

# DO\_LOOP, EXIT\_LOOP, END\_LOOP

## Akcia DO\_LOOP, EXIT\_LOOP, END\_LOOP

### Funkcia

Pomocou akcií je možné implementovať cyklus.

### Deklarácia

```
[FOR _ctrlVar [RANGE struct | = lBoundExpr TO uBoundExpr] ] DO_LOOP

...; akcie 1

[EXIT_LOOP [expression]]

...; akcie 2

[EXIT_LOOP [expression]]

...; akcie 3

END_LOOP
```

### Parametre

expression	in	Výraz typu Bool.
------------	----	------------------

### Popis

Cyklus má dva varianty:

#### 1. Cyklus s riadiacou premennou.

Na začiatku cyklu je riadiaca premenná nastavená na hodnotu 1 (variant s kľúčovým slovom **RANGE**) alebo na hodnotu danú výrazom *lBoundExpr*. Hodnota výrazu musí byť platná. Taktiež sa jedenkrát vyhodnotí horné obmedzenie pre hodnotu riadiacej premennej.

V prvom prípade (variant **RANGE**) je to rozmer danej štruktúry (*structDIM*).

V druhom prípade je horné obmedzenie získané vyhodnotením výrazu *uBoundExpr*. Hodnota výrazu musí byť platná. Riadiaca premenná cyklu bude pri jednotlivých iteráciách automaticky postupne zvyšovaná až po horné obmedzenie. Riadiaca premenná musí byť deklarovaná typu **INT**. V tele cyklu je možné ju meniť. Prípadné zneplatnenie jej hodnoty spôsobí chybu za behu pri vykonávaní akcie **END\_LOOP**. Cyklus je možné predčasne ukončiť akciou **EXIT\_LOOP**. Po ukončení cyklu má riadiaca premenná hodnotu o 1 väčšiu ako horné obmedzenie.

```
INT _i
INT _uBound

_uBound := 10
FOR _i=2 TO _uBound DO_LOOP
    _uBound := _uBound + 1 ; zmena hodnoty nemá vplyv na počet iterácií
END_LOOP
; hodnota premennej _i je 11
```

```
INT _i
RECORD (SD.ArchVal) _struct

REDIM _struct[10]
FOR _i RANGE _struct DO_LOOP
    REDIM _struct[2] ; zmena rozmeru nemá vplyv na počet iterácií
END_LOOP
; hodnota premennej _i je 11
```

#### 2. Cyklus bez riadiacej premennej.

Akcie uzavreté medzi akcie **DO\_LOOP** a **END\_LOOP** budú cyklicky vykonávané. Ukoní cyklus je možné akciou **EXIT\_LOOP**. Ak je akcia s parametrom, cyklus bude ukončený ak *expression*

nadobudne hodnotu @TRUE.

```
INT _i

_i := 1
DO_LOOP
    EXIT_LOOP _i = 10
    _i := _i + 1
END_LOOP
```

Ekvivalentný zápis bez podmieneného ukonenia cyklu:

```
INT _i

_i := 1
DO_LOOP
    IF _i = 10 THEN
        EXIT_LOOP
    ENDIF
    _i := _i + 1
END_LOOP
```

#### Poznámka

Chybný zápis cyklu nastane, ak je jeho zaiatok a koniec "prekřížený" s akciou **IF THEN**.

Napríklad:

```
INT _i
_i := 1

IF Sec = 1 THEN
    DO_LOOP
ENDIF
    EXIT_LOOP _i = 10
    _i := _i + 1
END_LOOP
```

alebo

```
INT _i

_i := 1
DO_LOOP
    IF _i = 10 THEN
        EXIT_LOOP
    ENDIF
    _i := _i + 1
IF Sec = 1 THEN
    _i := _i + 1
END_LOOP
ELSE
    END_LOOP
ENDIF
```



**Súvisiace stránky:**

[Akcie v skriptoch](#)