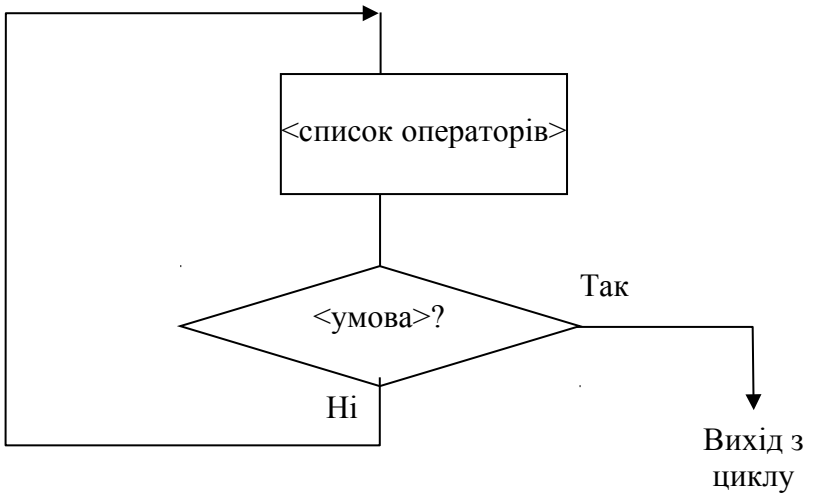
Лекція №17
Генерація коду для конструкції циклу з післяумовою repeat

Синтаксис:

1. <цикл repeat> → repeat <список операторів> until <умова>
2. <умова> → <вираз><операція порівняння><вираз>
3. <список операторов> → <оператор>
4. <список операторов> → <список операторов>;<оператор>
5. < операція порівняння > → >
6. < операція порівняння > → >=
7. < операція порівняння > → <
8. < операція порівняння > → <=
9. < операція порівняння > → =
10. < операція порівняння > → <>

Неформальна семантика:



Приклад.

Розглянемо семантичне визначення для конструкції циклу з післяумовою repeat.
repeat x:=y; b:=b+1 until a>b

```
?L1:  NOP
      MOV  AX,y   ┐
      MOV  x,AX   │
      MOV  AX,b   } x:=y; b:=b+1
      ADD  AX,1   │
      MOV  b,AX   ┘
      MOV  AX,a   ┐
      MOV  BX,b   } a>b
      CMP  AX,BX  ┘
      JLE  ?L1
```

Розглянемо порядок обходу дерева розбору і генерації коду для конструкції циклу з післяумовою repeat:

1. Генерація внутрішньої мітки, наприклад ?L1 .
2. Генерація команди NOP з внутрішньою міткою початку циклу ?L1: ?L1: NOP

3. Спуск по піддереву <список операторів> (* SPR(k2) *), під час якого генеруються команди реалізації цього списку операторів:

```
MOV  AX,y
MOV  x,AX
MOV  AX,b
ADD  AX,1
MOV  b,AX
```

4. Спуск по піддереву <умова> (* SPR(k1) *), під час якого генеруються команди реалізації цієї умови:

```
MOV  AX,a
MOV  BX,b
CMP  AX,BX
```

5. Генерація команди умовного переходу з внутрішньою міткою ?L1 на початок циклу:

```
JLE   ?L1
```

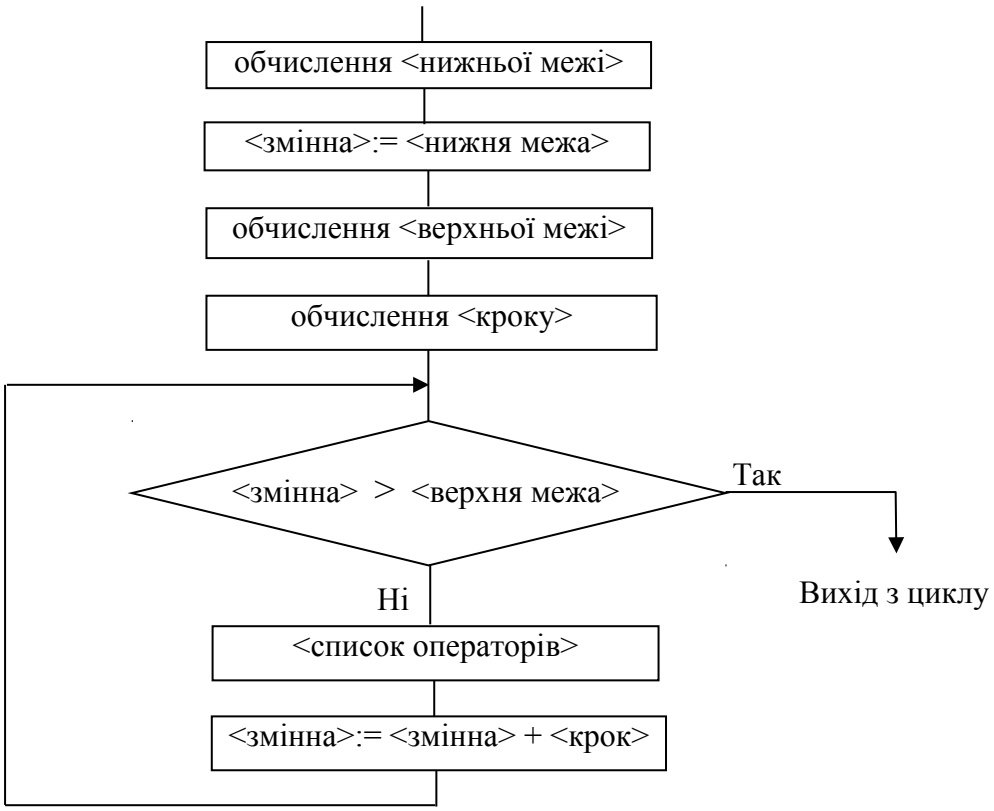
Генерація коду для конструкції циклу з лічильником (параметром) for

Синтаксис:

1. <цикл for> → for <змінна>:= <нижня межа> to <верхня межа> by <крок> do <оператор>
2. <нижня межа> → <вираз>
3. <верхня межа> → <вираз>
4. <крок> → <вираз>

При реалізації компілятора приймається ряд внутрішніх угод (погоджень) для нього. Однією з внутрішніх угод для конструкції циклу for приймемо таку: кількість повторень циклу зберігатиметься в регістрі CX.

Неформальна семантика (якщо крок додатній):



Приклад.
Розглянемо семантичне визначення для конструкції циклу з лічильником (параметром) for наступного вигляду:

for i:=LowBound to HighBound by Step do <оператор>

```
        MOV     AX,LowBound     ; <нижня межа> → у AX
        MOV     i,AX            ; <нижня межа> через AX → в i
        MOV     AX,HighBound    ; <верхня межа> → у AX
        CMP     AX,i            ; порівняння (віднімання) верхньої та нижньої меж
        JL      ?L1             ; якщо <верхня межа> < <нижня межа>, то вихід з
                                ; циклу
        MOV     DX,Step         ; <крок> → у DX
        MOV     .S+0,DX         ; <крок> → у робочу комірку.
; обчислити кількість ітерацій циклу (<верхня межа> – <нижня межа>) / <крок> + 1
        SUB     AX,i        ; <верхня межа> – <нижня межа> → у AX
        CWD                 ; розширити AX до DX:AX
        DIV     .S+0        ; (<верхня межа> – <нижня межа>) / <крок> → у AX
        INC     AX          ; (<верхня межа> – <нижня межа>) / крок> +1 → у AX
        MOV     CX,AX       ; кількість повторень → у CX
?L2:    NOP
        <оператор>          ; команди, що були згенеровані при обробці <оператора>
        MOV     AX,i
        ADD     AX,.S+0     } нарощування змінної на величину  <кроку>
        MOV     i,AX
        LOOP    ?L2
?L1:    NOP
```

Розглянемо порядок обходу дерева розбору і генерації коду для конструкції циклу з лічильником (параметром) for:

1. Спуск по піддереву <нижня межа> (* SPR(k4) *)
2. Спуск по піддереву <змінна> (* SPR(k5) *)
3. Генерація команди пересилки <нижня межа> → <змінна>
   ```
   MOV  AX,LowBound
   MOV  i,AX
   ```
4. Спуск по піддереву <верхня межа>, (* SPR(k3) *)
5. Генерація команди завантаження <верхньої межі>
   ```
   MOV  AX,HighBound
   ```
6. Генерація команди порівняння і команди умовного переходу з внутрішньою міткою ?L1 для виходу з циклу
   ```
   CMP  AX,i
   JL   ?L1
   ```
7. Спуск по піддереву <крок> (* SPR(k2) *)
8. Генерація команд пересилки <кроку> в робочу комірку
   ```
   MOV  DX,2
   MOV  .S+0,DX
   ```
9. Генерація команд розрахунку числа повторень циклу
   ```
   SUB  AX,i
   CWD
   DIV  .S+0
   INC  AX
   ```
10. Пересилка числа повторень в CX
    ```
    MOV  CX,AX
    ```

11. Генерація команди NOP з внутрішньою міткою ?L2 (початок операторів циклу)

```
?L2:    NOP
```

12. Спуск по піддереву <оператор>, де генеруються команди його реалізації (* SPR(k1) *)
13. Генерація команд зміни значення змінної циклу на величину <кроку>

```
MOV  AX,i
ADD  AX,.S+0
MOV  i,AX
```

14. Генерація команди LOOP з переходом на початок циклу

```
LOOP ?L2
```

15. Генерація команди NOP з міткою виходу з циклу.

```
?L1:    NOP
```