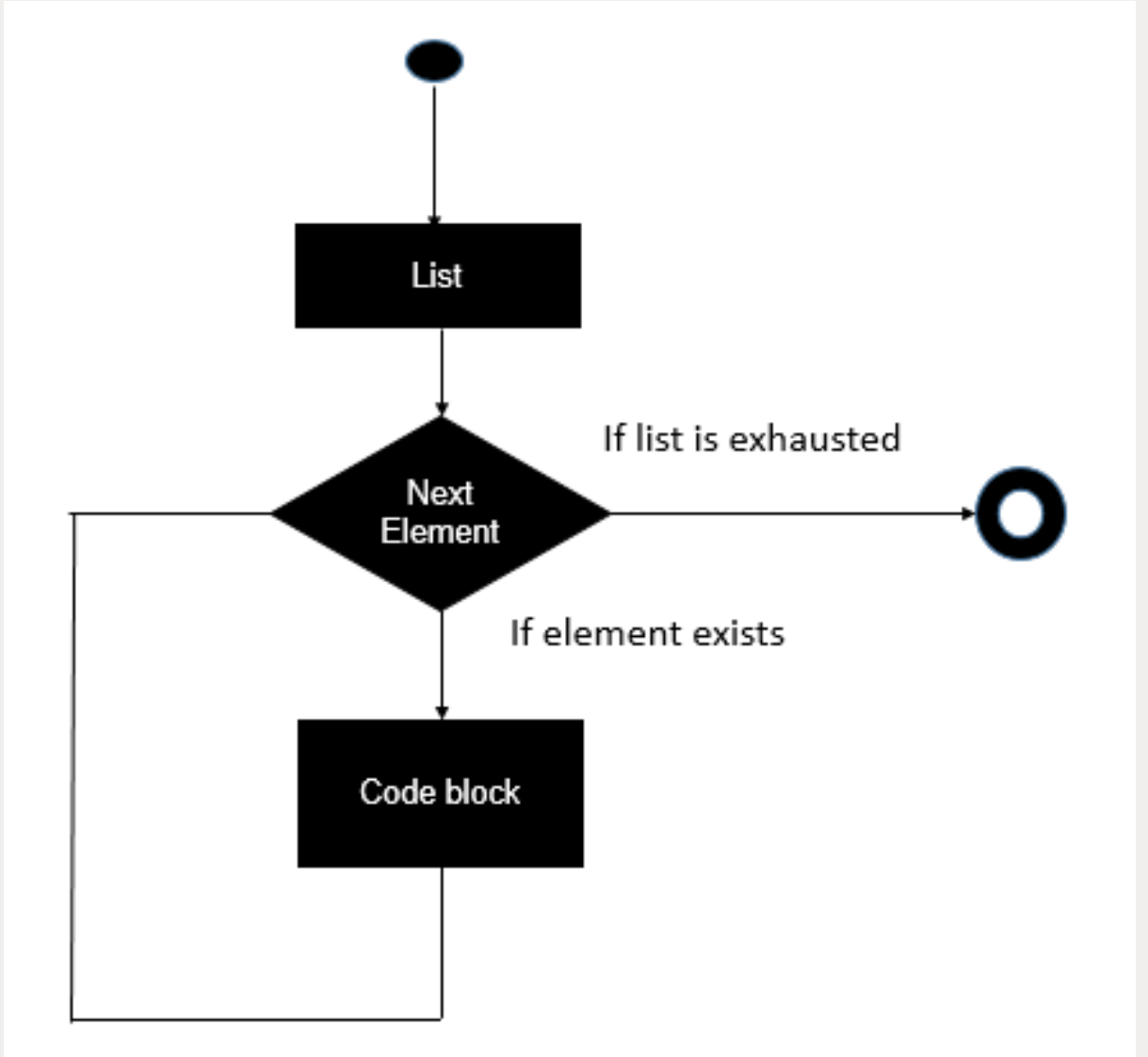
METODE AVANSATE DE GESTIUNE A DOCUMENTELOR ȘI A SISTEMELOR DE CALCUL
- CURS 3 -

Asist. Diana – Florina Șotropa

www.cs.ubbcluj.ro/~diana.sotropa

Cuprins

Sintaxă: FOR


List
Next Element
If list is exhausted
If element exists
Code block

Sintaxă: FOR
Fisiere

```
FOR %%variable IN list DO do_something
```

Executa o comanda pentru toate fisierele din lista de fisiere (list)

Variabila care ia pe rand valoarea fiecarui element din lista se acceseaza:

- In command line: %variable
- In script: %%variable

Exemplu: FOR

```
@echo off
FOR %%F IN (1 2 3 4 5) DO echo %%F
```

Obs. Script-ul va afisa la iesirea standard fiecare element din lista, adica numerele de la 1 la 5

Exemplu: FOR

```
FOR %%G IN (Myfile.txt SecondFile.txt) DO copy %%G
d:\backups\
```

Obs. Script-ul va copia fiecare fisier din lista data (Myfile.txt SecondFile.txt) in destinatia data d:\backups\

Fisierele se regasesc in directorul curent

Exemplu: FOR

```
FOR %%G IN
(a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,
z) DO (md C:\demo\%%G)
```

Obs. Script-ul va crea 26 de foldere in directorul C:\demo cu numele date in lista

Sintaxă: FOR
Fisiere

```
FOR /R [[drive:]path] %%parameter IN (set) DO command
```

Executa o comanda recursiva pentru toate fisierele din arborele indicat prin path

Exemplu: FOR Fisiere

```
For /R C:\temp\ %%G IN (*.bak) do Echo del "%%G"
```

Obs. Script-ul va scrie la iesirea standard toate numele fisierelor cu extensia bak care sunt gasite recursiv incepand cu C:\temp

Sintaxă: FOR
Fisiere

```
FOR /D [/R] %%parameter IN (folder_set) DO command
```

Executa o comanda pentru mai multe directoare sau foldere

- /D – incepe cautarea din directorul current
- /R – recursiv

Se executa comanda pe directoare nu pe fisiere

Exemplu: FOR Fisiere

```
For /R C:\Work\ %%G IN (“User*”) do Echo We found "%%~nxG"
```

```
@Echo Off
CD \Work
FOR /D /R %%G in ("User*") DO Echo We found %%~nxG

=> Se afiseaza toate subfolderele care incep cu user
```

Obs. Primul script va afisa recursiv toate fisierele din directorul Work al caror nume incep cu User

Obs. Al doilea script va afisa recursiv toate subdirectoarele din directorul Work al caror nume incep cu User

Sintaxă: FOR
Lista de numere

```
FOR /L %%variable IN
(lowerlimit,Increment,Upperlimit) DO do_something
```

Exemplu: FOR

```
@ECHO OFF
FOR /L %%X IN (0,2,5) DO ECHO %%X

FOR %%G IN (Sun Mon Tue Wed Thur Fri Sat) DO echo
%%G
```

Obs. Primul script va afisa numerele >= cu 0 si < 5 cu pas 2, incepand cu 0

Obs. Al doilea script va afisa toate elementele din lista, adica zilele saptamanii

Sintaxă: FOR

Continut de fisiere / output-ul unei comenzi

```
FOR /F ["options"] %%parameter IN (filenameset) DO command
```

```
FOR /F ["options"] %%parameter IN ("Text string to process")
DO command
```

```
FOR /F ["options"] %%parameter IN ('command to process') DO
command
```

Sintaxă: FOR

/F - Optiuni

```
- options:
   - delims=xxx (caracterul delimitator; implicit: Space)
   - skip=n (numarul de linii care se ignora; implicit: 0)
   - eol=; (caracterul de la inceputul liniei care indica
     comentariu; implicit: ;)
   - tokens=n (numarul de itemi care se citesc de pe fiecare
     linie; implicit: 1)
   - usebackq (se folosesc “” pentru numele lungi de fisiere,
     ‘’ pentru siruri de caractere lungi si `` pentru comenzi
     care trebuie procesate )
```

Sintaxă: FOR

Continut de fisiere / output-ul unei comenzi – la folosirea optiunii usebackq

```
FOR /F ["options"] %variable IN (“file-set”) DO command
[command-parameters]

FOR /F ["options"] %variable IN ('string') DO command
[command-parameters]

FOR /F ["options"] %variable IN (`command`) DO command
[command-parameters]
```

Exemplu: FOR

/F - Optiuni

```
FOR /f "delims=" %%G in (files.txt) DO copy
“C:\source\folder\%%G" "H:\destination\%%G“
```

Obs. Copiaza fisierele din directorul C:\source\folder
in noua destinatie H:\destinatie

Se presupune ca files.txt contine un singur nume de
fisier pe linie

Exemplu: FOR

/F - Optiuni

weather.txt

```
----------------------------
January,Snowy,02
February,Rainy,15
March,Sunny,25
```

```
FOR /F "tokens=1,3 delims=," %%G IN (weather.txt) DO @echo %%G %%H
```

Obs. Acest script extrage datele delimitate de “,” dintr-un fisier care contine doar cifre, litere si virgule (nu si alte semne de punctuatii sau spatii)
Din fiecare linie, care contine cuvinte delimitate prin virgule, se extrag cuvintele 1 si 3

```
January,02
February,15
March,25
```

%%H se initializeaza implicit datorita folosirii TOKENS (adica %%G e cuvantul 1, %%H e cuvantul 3)

Exemplu: FOR

```
FOR /F "tokens=1-5" %%A IN ("This is a short
 sentence") DO echo  %%A %%B %%D
```

Obs. Acest script extrage datele delimitate de “ ” din sirul de caractere dat ca parametru

tokens=1-5 => %%A reprezinta token-ul 1, %%B token-ul 2, … %%E token-ul 5

Deoarece delims nu e setat, delimitatorul considerat e cel implicit, adica “ ”

Exemplu: FOR

Variabilele nu se actualizeaza pana cand nu s-a terminat bucla

Obs. tokens=* => fiecare rand se considera un singur token

Se va afisa numarul 1 impreuna cu numele fiecarui fisier din directorul current

Se va afisa numarul total de fisiere din director

```
@echo off
SET count=1
 FOR /f "tokens=*" %%G IN ('dir /b') DO (
 echo %count%:%%G
 set /a count+=1 )
 ECHO %count%
```

Obs. Variabilele definite sunt actualizate doar dupa ce linia a fost parsata. In cazul unor blocuri de instructiuni delimitate de paranteze, intregul bloc este considerat o singura comanda. => variabila %count% este inlocuita inainte de a se finaliza executia blocului de comenzi

Exemplu: FOR

Totusi, folosirea unei subroutine rezolva neajunsul

```
@echo off
SET count=1
FOR /f "tokens=*" %%G IN ('dir /b') DO (call
:subroutine "%%G")
GOTO :eof

:subroutine
 echo %count%:%1
 set /a count+=1
 GOTO :eof
```

Exemplu: FOR

O alta metoda care rezolva neajunsul este reprezentata de folosirea EnableDelayedExpansion

EnableDelayed Expansion

- Permite variabilelor din script-urile CMD sa fie definite doar in faza de executie (de fapt are loc inlocuirea variabilei cu valoarea ei) si nu in faza de parsare

```
Ex. Variabila windir se inlocuieste  cu C:\Windows
```

- Se foloseste: SETLOCAL EnableDelayedExpansion

```
@echo off
SETLOCAL
Set "_var=first"
Set "_var=second" & Echo %_var%
```

⇒first

```
@echo off
SETLOCAL EnableDelayedExpansion
Set "_var=first"
Set "_var=second" & Echo %_var% !_var!
```

⇒ first second

EnableDelayed Expansion

```
@echo off
Setlocal
Set _html=Hello^>World
Echo %_html%
```

```
@echo off
Setlocal EnableDelayedExpansion
Set _html=Hello^>World
Echo !_html!
```

Obs. In primul script se va afisa cuvantul Hello in fisierul World

Obs. In al doilea script se va afisa la iesirea standard Hello>World

EnableDelayed Expansion

```
@echo off
setlocal EnableDelayedExpansion
Set var1=Hello ABC how are you
Set var2=ABC
Set var3=Beautiful
Set result=!var1:%var2%=%var3%!
Echo [!result!]
```

Obs. In acest script se va inlocui in var1, valoarea lui var2 cu valoarea lui var3

=> [Hello Beautiful how are you]

EnableDelayed Expansion

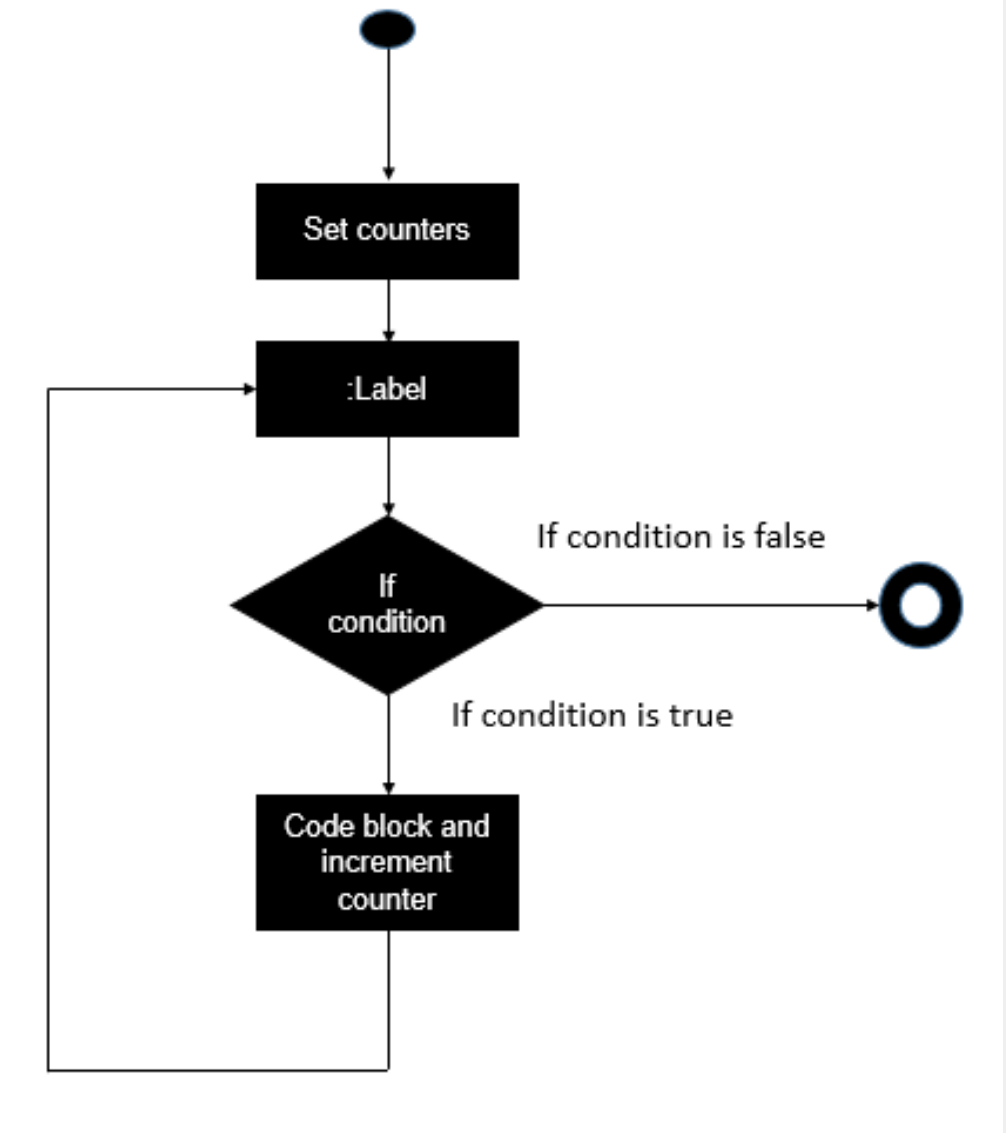
Ex. 1. [0] [0] [0] [0] [0] Total = 5

```
@echo off
setlocal
set _tst=0
FOR /l %%G in (1,1,5) Do (echo [%_tst%]
set /a _tst+=1)
echo Total = %_tst%
```

```
@echo off
setlocal EnableDelayedExpansion
set _tst=0
FOR /l %%G in (1,1,5) Do (echo [!_tst!]
set /a _tst+=1)
echo Total = %_tst%
```

Ex. 2. [0] [1] [2] [3] [4] Total = 5

Sintaxă: FOR clasic



Sintaxă: FOR clasic

```
Set counter
:label
If (expression) exit loop
Do_something
Increment counter
Go back to :label
```

Exemplu: FOR clasic

```
@echo off
SET /A i=1
:loop
IF %i%==5 GOTO END
echo The value of i is %i%
SET /a i=%i%+1
GOTO :LOOP
:END
```

Exemplu: parcurgerea argumentelor

```
@ECHO OFF
:Loop
IF "%1"=="" GOTO completed
FOR %%F IN (%1) DO echo %%F
SHIFT
GOTO Loop
:completed
```

Sintaxa: nested FOR

```
FOR %%G... DO (for %%U... do ...)
```


Sintaxă: Funcții

- Declararea Funcției
- Definirea Funcției

Sintaxă: Definirea Funcției

```
:function_name
Do_something
EXIT /B 0
```

Exemplu: Definirea Funcției

```
:Display
SET /A index=2
echo The value of index is %index%
EXIT /B 0
```

Sintaxă: Apelarea Funcției

```
call :function_name
```

Exemplu: Apelarea Funcției

```
@echo off
SETLOCAL
CALL :Display
EXIT /B

:Display
SET /A index=2
echo The value of index is %index%
EXIT /B 0
```

Sintaxă: Apelarea funcției cu parametri

```
Call :function_name
parameter1, parameter2…
parametern
```

```
~1 = parameter1
```

Exemplu: Apelarea funcției cu parametri

```
@echo off
SETLOCAL
CALL :Display 5 , 10
EXIT /B

:Display
echo The value of parameter 1 is %~1
echo The value of parameter 2 is %~2
EXIT /B 0
```

Obs. %~1 – este primul parametru transmis la apelul functiei, adica 5
Obs. %~2 – este al doilea parametru transmis la apelul functiei, adica 10

Sintaxă: Returnare valori prin funcție

```
Call :function_name value1, value2… valuen
```

Exemplu: Returnare valori prin funcție

```
@echo off
SETLOCAL
CALL :SetValue value1,value2
echo %value1%
echo %value2%
EXIT /B

:SetValue
set "%~1=5"
set "%~2=10"
EXIT /B 0
```

IDEI FINALE

Idei finale

- Parsarea unei comenzi:
  - Substituirea variabilelor – variabilele sunt inlocuite cu continutul lor
  - Ghilimelele – pentru a elimina semnificatii speciale, se pot incapsula caracterele speciale intre ghilimele (in afara de %)
  - ^ - se foloseste tot pentru eliminarea unei semnificatii special; pentru evitarea unui caracter dintr-o comanda localizata dupa pipe (|) trebuie folosit ^^^
- Caractere speciale: <, >, |, &, ^ (uneori si ! si \)
- Variabile:
  - %varname% - pentru numele de variabile care contin litere
  - %n, 0<=n<=9 – pentru parametrii din linia de comanda
  - %* - toate valorile din linia de comanda (in afara de %0)

Idei finale

echo “Johnson & son” => “Johnson & son”
echo Johnson ^& son => Johnson & son
echo Johnson & son => son se interpreteaza ca o comanda diferite (eroare)

echo A ^^ B => A ^ B

if 1 equ 1 ^
echo Equal &^
echo Indeed, equal => ^ face escape la New Line (cele 3 linii sunt tratate ca una)

echo %temp% => C:\Users\Diana\AppData\Local\Temp
echo ^%temp^% => %temp%
echo %%temp%% => %C:\Users\Diana\AppData\Local\Temp%

Idei finale

Procesarea sirurilor de caractere:

set a=abcd

echo %a:~0,1% => a
echo %a:~1,1% => b
echo %a:~0,2% => ab
echo %a:~1,2% => bc
%a:~m,n% retuneaza n caractere incepand cu pozitia m

echo %a:~1% => bcd
%a:~m% retuneaza toate caracterele incepand cu pozitia m

echo %a:~-1% => d
echo %a:~-2% => cd
%a:~-n% retuneaza n caractere de la final

echo %a:~0,-2% => ab
echo %a:~0,-1% => abc
echo %a:~1,-1% => bc
%a:~m,-n% retuneaza toate caracterele incepand cu pozitia m in afara de ultimele n

Idei finale

Procesarea sirurilor de caractere:

if not "%a:bc=%"=="%a%" echo YES
Daca variabila a contine bc ca si subsir se va afisa YES

if %a:~0,1%==a echo yes
Daca variabila a incepe cu a se va afisa YES
if %a:~0,2%==ab echo yes
Daca variabila a incepe cu ab se va afisa YES

set a=abcd & echo %a:c=%
Se sterge din sirul de caractere din variabila a caracterul c
set a=abcd & echo %a:c=e%
Se inlocuieste in sirul de caractere din variabila a caracterul c cu caracterul e
set a=abcd & echo %a:*c=%
Se sterge din sirul de caractere totul pana la c, inclusiv