Iterative constructions

Fundamentals of Computer Science

2010-2011

Ismael Etxeberria Agiriano

18/10/2010

Escuela Universitaria de Ingeniería Vitoria-Gasteiz
Ingeniaritzako Unibertsitate Eskola Vitoria-Gasteiz

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

Index
Iterative constructions

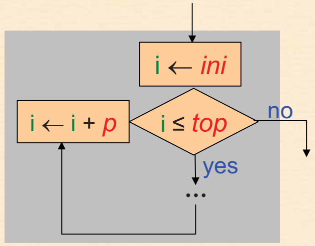
1. Iterative algorithms analysis
2. Ex11: While
3. Ex12: For
4. Ex13: Do - Loop
5. Summary

1.1 Iterative problems

- *Don't stop until you get 145 beats per minute*
- *You are going to take 10 laps to the field*
- *Calculate the average: add up all grades of a subject and divide by the number of students (count)*
- *Find a number that complies certain conditions*
  - *One? All of them?*
  - *In which domain?*
- *Count up all votes in an urn*
  - *While there are votes left*
  - *Until there is no vote left*
- *Operations with strings*
- *You won't leave the house until you find your wallet*

1.2 Analysis: For

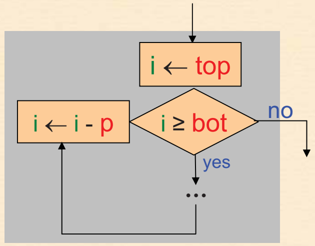
- *Can I determine in advance* ***how many times*** *I need to execute the body ("…")?*
  - Yes: then use a **For** loop
- *It uses a counter i. It's divided into three clauses:*
  1. ***Initialization***: give an initial value ini to the counter i
  2. ***Condition***: verify if the counter i has arrived to top
  3. ***Actualization***: increment the counter i the step p



```
For i = ini To top Step p
  . . .
Next i
```

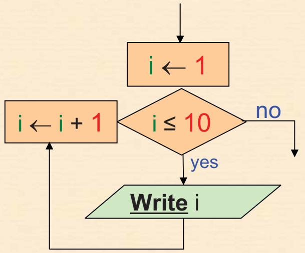
For: decreasing

- *Instead of counting we may set i to discount or decrement:*
  1. ***Initialization*** : give an initial value top to the counter i, the superior value
  2. ***Condition*** : verify if the counter i has arrived to bot (while i ≥ bot)
  3. ***Actualization*** : decrement the counter i the step p, which is negative



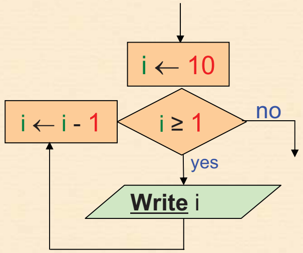
```
For i = top To bot Step -p
   . . .
Next i
```

For: Write numbers from 1 to 10 (increment)



```
Sub Write_Click()
  Dim i As Integer
  For i = 1 To 10 Step 1
    pct1.Print CStr (i)
  Next i
End Sub
```

For: Write numbers from 10 to 1 (decrement)

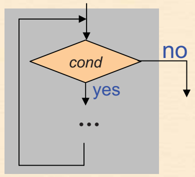


```
For i = 10 To 1 Step -1
   pct1.Print CStr (i)
Next i
```

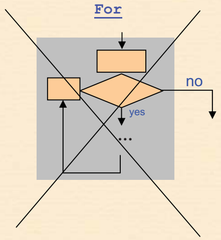
1.3 Analysis: While

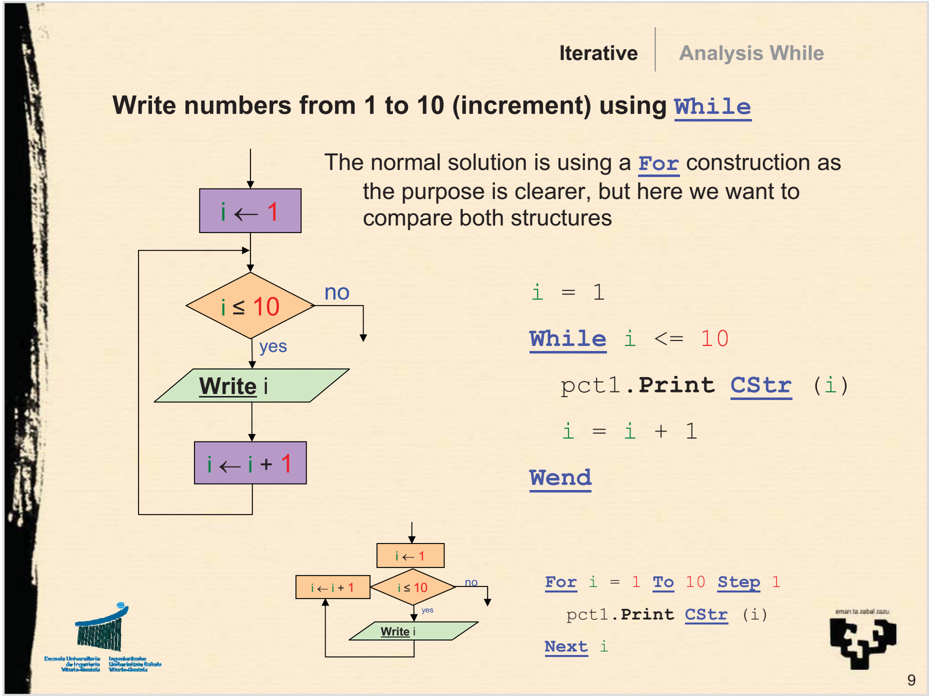
- *Can I determine in advance **how many times** I need to execute the body ("…")? no*
- *Can I just finish?*
- *Do I want the body to be executed zero or more times?*
  - Yes: **While** construction

While



```
While cond
   . . .
Wend
```





Write numbers from 1 to 10 (increment) using While

The normal solution is using a For construction as the purpose is clearer, but here we want to compare both structures

i ← 1

i ≤ 10

no

yes

Write i

i ← i + 1

```
i = 1
While i <= 10
  pct1.Print CStr (i)
  i = i + 1
Wend
```

i ← 1

i ← i + 1

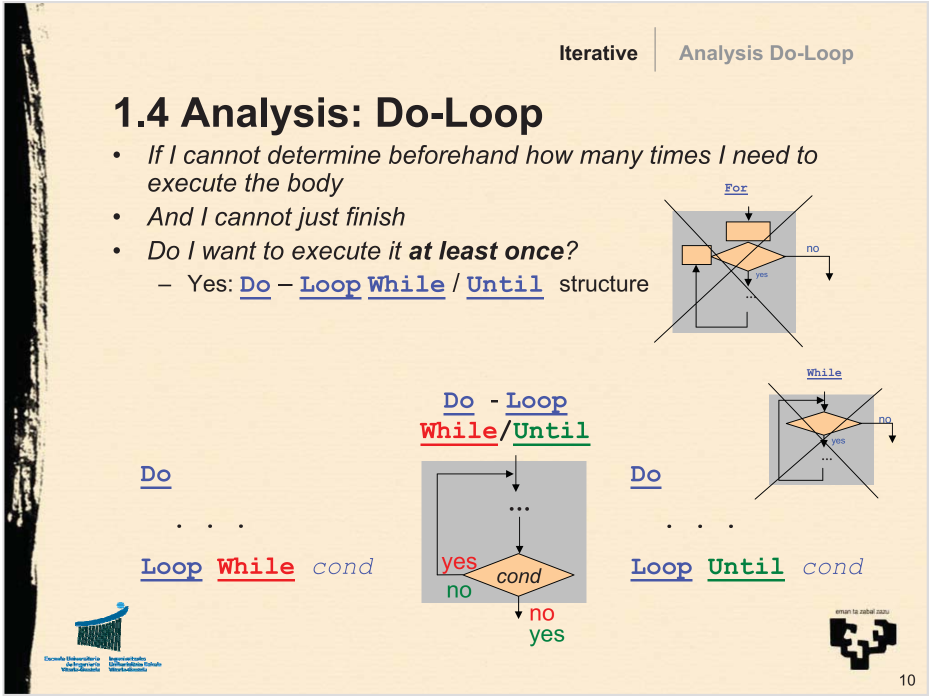
i ≤ 10

no

yes

Write i

```
For i = 1 To 10 Step 1
  pct1.Print CStr (i)
Next i
```



1.4 Analysis: Do-Loop

- *If I cannot determine beforehand how many times I need to execute the body*
- *And I cannot just finish*
- *Do I want to execute it **at least once**?*
  - Yes: Do – Loop While / Until structure

For

no

yes

...

While

no

yes

...

Do - Loop

While/Until

```
Do
  . . .
Loop While cond
```

...

yes

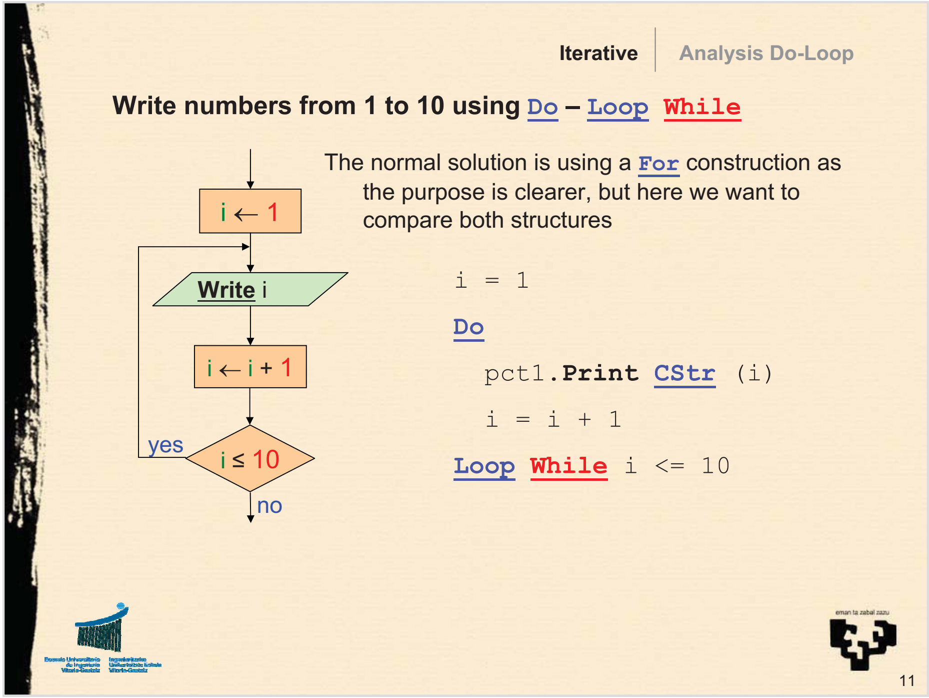
no

cond

no

yes

```
Do
  . . .
Loop Until cond
```



Write numbers from 1 to 10 using Do – Loop While

The normal solution is using a For construction as the purpose is clearer, but here we want to compare both structures

i ← 1

Write i

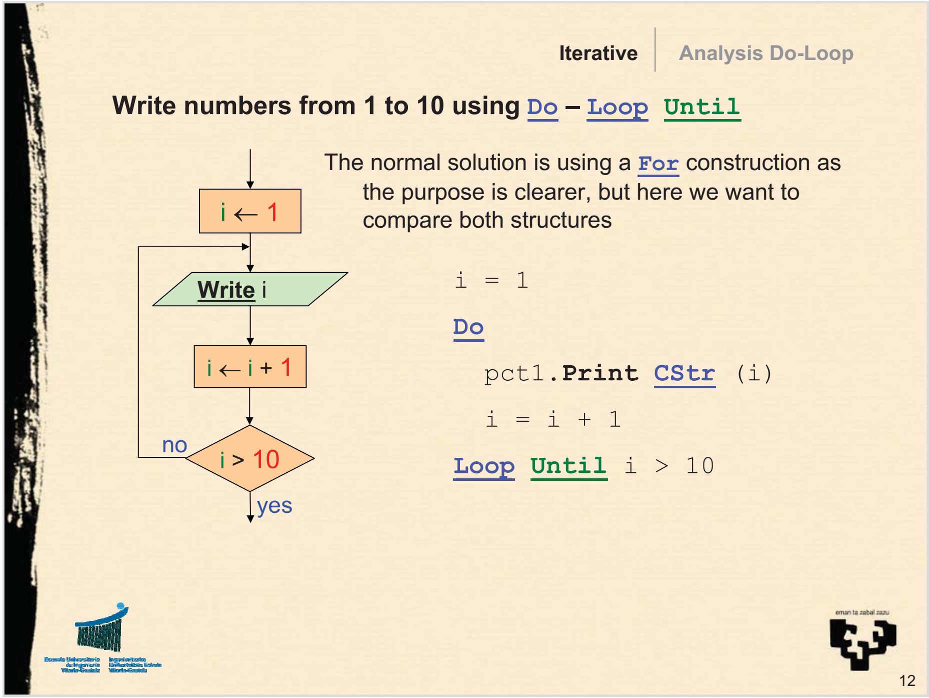
i ← i + 1

yes

i ≤ 10

no

```
i = 1
Do
  pct1.Print CStr (i)
  i = i + 1
Loop While i <= 10
```



Write numbers from 1 to 10 using Do – Loop Until

The normal solution is using a For construction as the purpose is clearer, but here we want to compare both structures

i ← 1

Write i

i ← i + 1

no

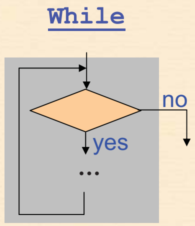
i > 10

yes

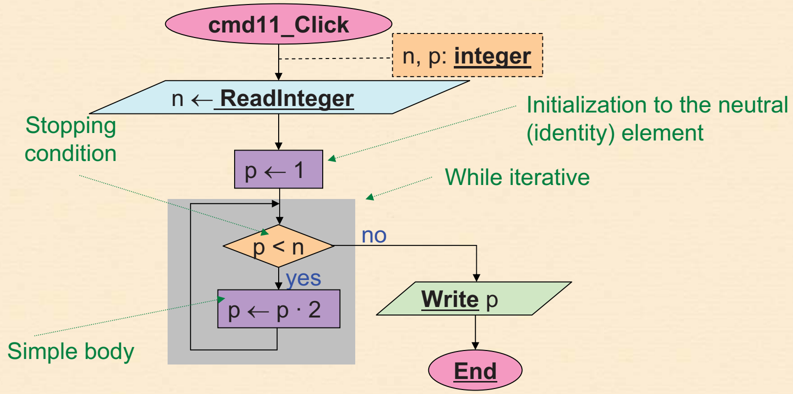
```
i = 1
Do
  pct1.Print CStr (i)
  i = i + 1
Loop Until i > 10
```

2. Example 11

- ***Title***
  - While iterative
- ***Name***
  - cmd11_Click
- ***Description***
  - Calculate the first natural power of 2 greater than or equal to a given number
- ***Observations***
  - Zero or more times: While
  - **Productory** (Capital Pi, Π)



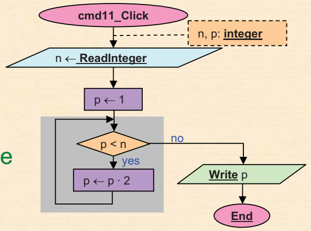
Ex11: Flowchart



Ej11: Codificación VB

```
Sub cmd11_Click()
  Dim s As String
  Dim n As Integer, p As Integer
  s = InputBox ("Number:")
  n = CInt (s)
  p = 1
  While p < n
    p = p * 2
  Wend
  MsgBox "Power: " & p
End Sub
```

While iterative



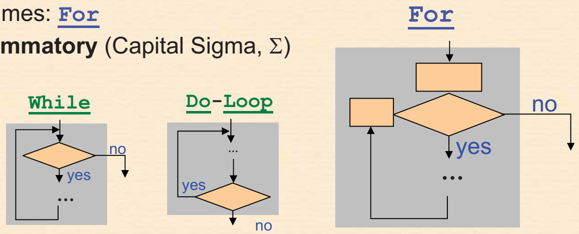
4. Example 12

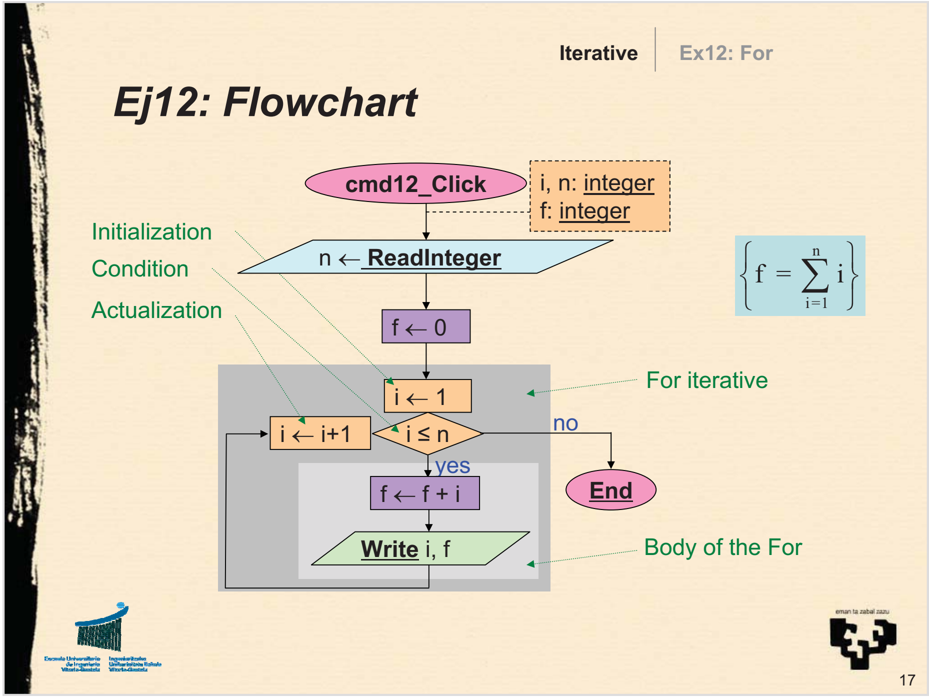
Output for n = 8

```
i  f
-  -
1: 1
2: 3
3: 6
4: 10
5: 15
6: 21
7: 28
8: 36
```

- ***Title***
  - For iterative
- ***Name***
  - cmd12_Click
- ***Description***
  - Write the partial sums of the n first terms of the arithmetic progression with $a_1 = 1$ and $a_i=a_{i-1}+i$ for every i > 1
- ***Observations***
  - n times: For
  - **Summatory** (Capital Sigma, Σ)

$$\left\{ f = \sum_{i=1}^{n} i \right\}$$





Ej12: Flowchart

$$f = \sum_{i=1}^{n} i$$

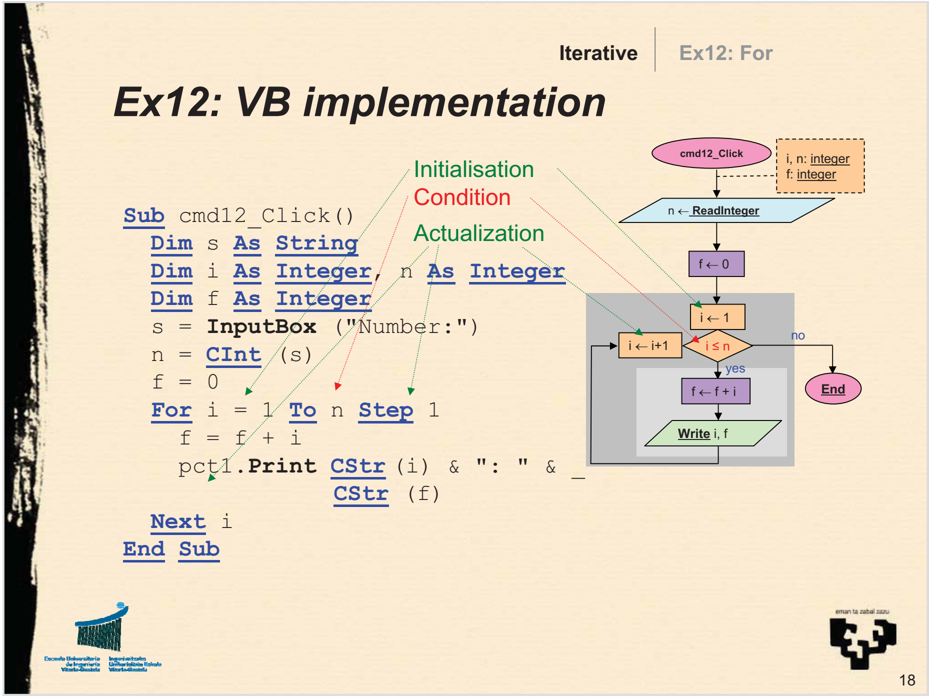
Initialization

Condition

Actualization

For iterative

Body of the For



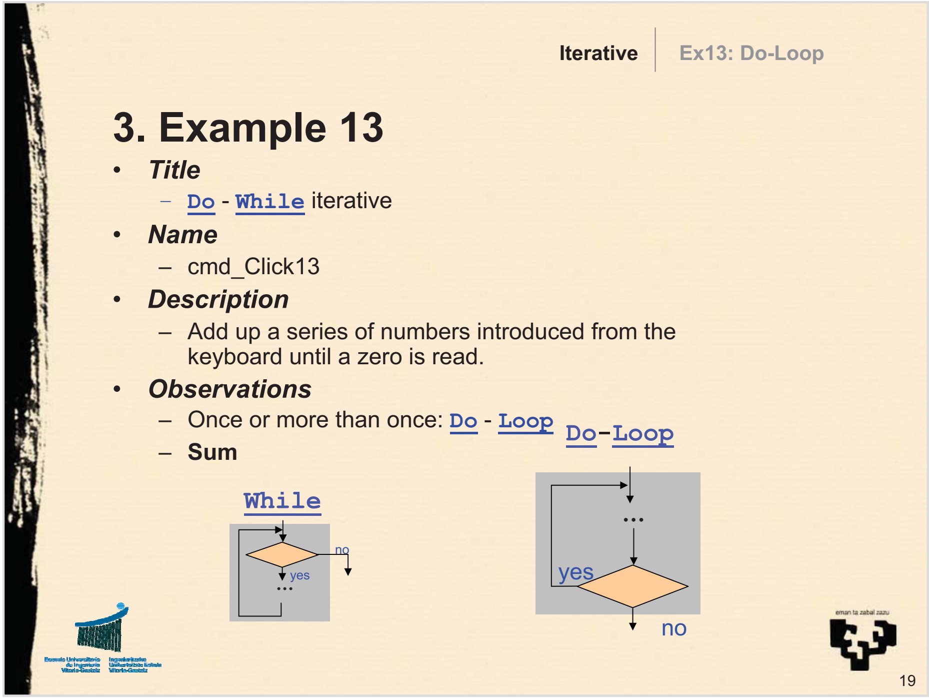
Ex12: VB implementation

```
Sub cmd12_Click()
  Dim s As String
  Dim i As Integer, n As Integer
  Dim f As Integer
  s = InputBox ("Number:")
  n = CInt (s)
  f = 0
  For i = 1 To n Step 1
    f = f + i
    pct1.Print CStr (i) & ": " & _
                CStr (f)
  Next i
End Sub
```

Initialisation

Condition

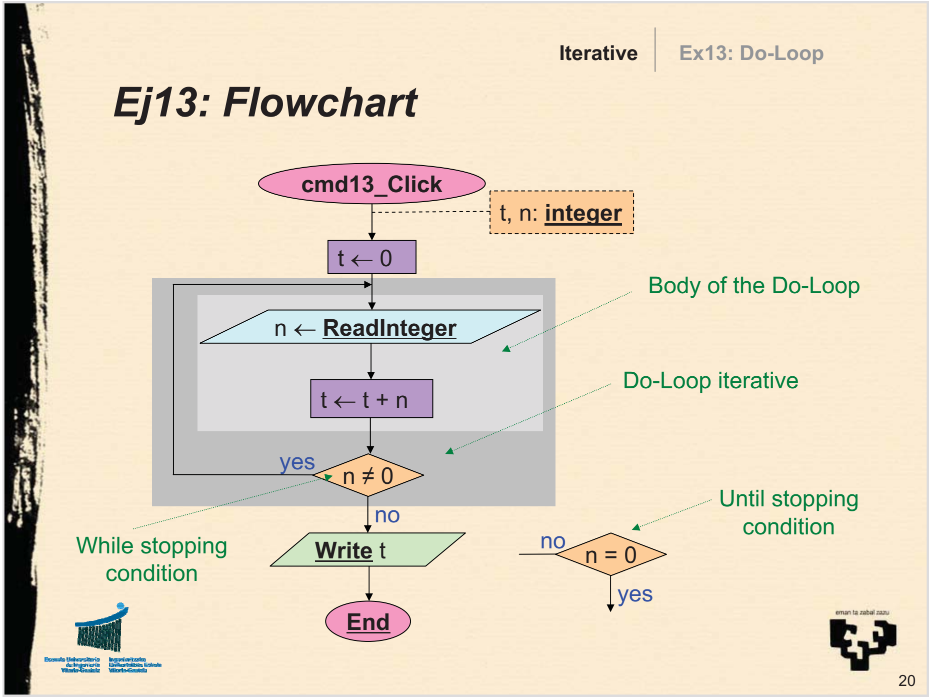
Actualization



3. Example 13

- *Title*
  - Do - While iterative
- *Name*
  - cmd_Click13
- *Description*
  - Add up a series of numbers introduced from the keyboard until a zero is read.
- *Observations*
  - Once or more than once: Do - Loop
  - Sum

While
no
yes
...
Do–Loop
...
yes
no



Ej13: Flowchart

Body of the Do-Loop

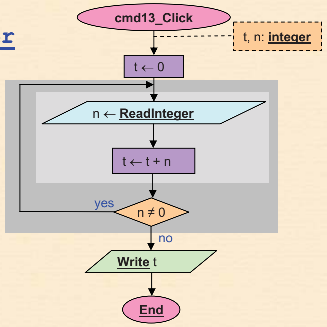
Do-Loop iterative

While stopping condition

Until stopping condition

Ej13: VB implementation

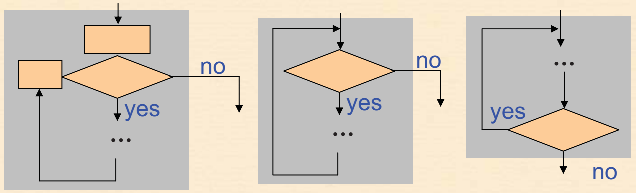
```
Sub cmd13_Click()
  Dim s As String
  Dim n As Integer, t As Integer
  t = 0
  Do
    s = InputBox ("Number:")
    n = CInt (s)
    t = t + n
  Loop While n <> 0
  MsgBox CStr (t)
End Sub
```



5. Summary

- Basic examples
  - *Initialize to the neutral (identity) element*
- Operations patterns
  - ***Sum**. Neutral element: 0*
  - ***Count**. Neutral element: 0*
  - ***Product**. Neutral element: 1*
  - ***Concatenation**. Neutral element: empty string*
  - ***Search**. Neutral element: `False`.*

For | While | Do-Loop


no
yes
...
no
yes
...
...
yes
no