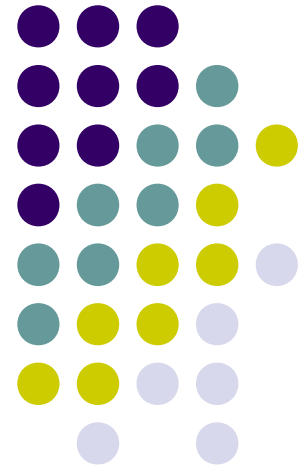


# Outils informatiques 2

## Cours n° 4 : Structures de contrôle répétitives

---

Claire Hanen  
Juliette Arnal



# Structures de contrôle



## *Deux types principaux :*

- Structures de contrôle **alternatives** (choix entre plusieurs séquences d'instructions) (cf CM 3)
- Structures de contrôle **répétitives** (possibilité de répéter, sous conditions, une séquence d'instruction).

Dans un programme, il est souvent nécessaire de répéter une ou plusieurs actions jusqu'à ce que quelque chose se passe, ou tant que quelque chose ne s'est pas passé.

# Petit lexique Anglais/Français :



Loop = Boucle (Faire une boucle, recommencer la boucle)

Do = Faire (Effectuer, faire les instructions qui suivent dans le programme)

While = Tant que (Tant que  $X = 1$ , ou que  $X <> 5$ , ou que le mot de passe est erroné)

Until = Jusqu'à ce que (Jusqu'à ce que  $X = 10$ , ou Jusqu'à ce que le mot de passe soit correct)

For = Pour

Next = Suivant

# Structure répétitive

## Do While...Loop



Syntaxe :

**Do While** expr

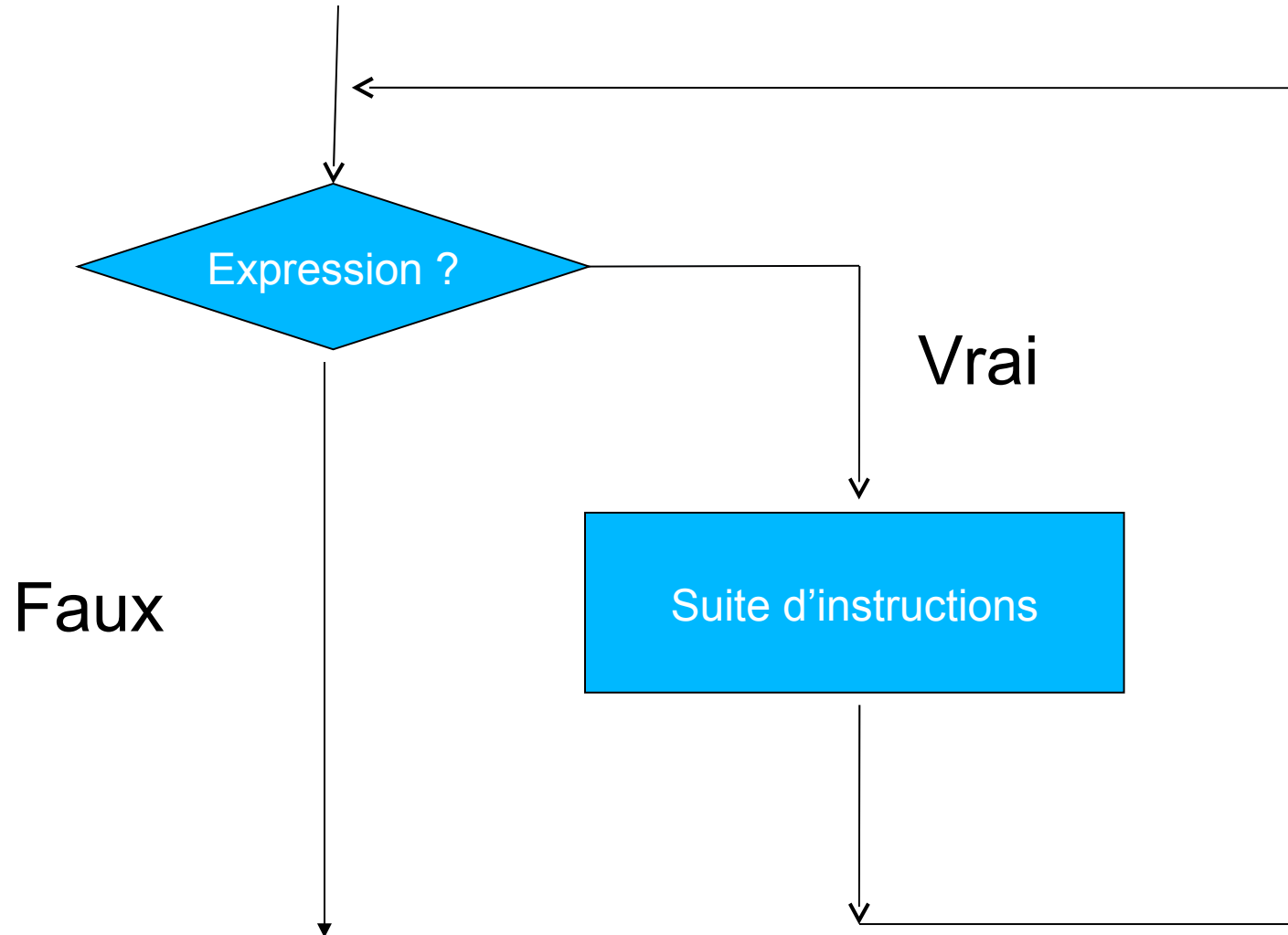
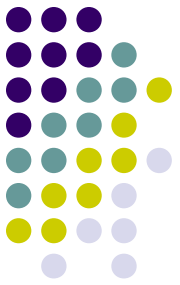
Séquence d'instructions

**Loop**

- expr est une expression à valeur booléenne (vraie ou fausse)
- Pour utiliser une boucle While, il faut que l'expression devienne fausse à un moment de l'exécution, sinon le programme ne s'arrête pas.
- La séquence d'instruction doit donc être susceptible de modifier la valeur de vérité de expr.

# Instruction répétitive

## Do While...Loop



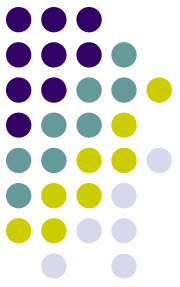


# Un peu de vocabulaire

- **Corps de boucle**: les instructions répétées
- Une **itération**: une exécution du corps de boucle
- Ex: la boucle suivante a 9 itérations
- Condition de **sortie de boucle**: propriété des variables lorsque la condition de continuation (après le while) est fausse.

# Instruction répétitives

## Do While...Loop



Boucle infinie (ce qu'il ne faut pas faire!!) :

Dim n as integer

Dim somme as integer

n=1

somme=0

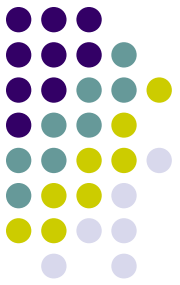
Do while n<10

somme =somme +n

Loop

# Instruction répétitives

## Do While...Loop



Boucle correcte :

calcul de la somme des 9 premiers entiers

Dim n as integer

Dim somme as integer

n=1

somme=0

Do while n<10

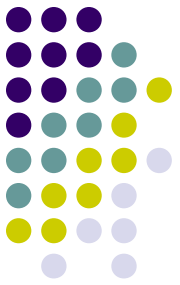
somme =somme +n

n=n+1

Loop



# Pour bien comprendre: déroulage d'une boucle



- Principe: contrôle des valeurs des variables à chaque itération, à un point d'arrêt dans le programme

somme=0

n=1

Do while n<10

Point d'arrêt1

somme =somme +n

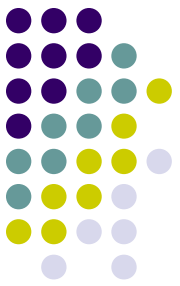
n=n+1

Loop

Point d'arrêt2

n	1	2	3	4	5	6	7	8	9	10
somme	0	1	3	6	10	15	21	28	36	45

On dit que la variable somme est un **accumulateur**



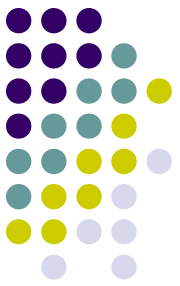
# Exemple: Somme des carrés

- Sub somcarre()
- 'calcule la somme des carrés des entiers de 1 au nombre donné par l'utilisateur
- Dim max As Integer
- Dim n As Integer
- Dim somme As Integer
- max = InputBox("entrez un entier")
- somme = 0
- n = 1
- Do While n <= max
- **Point d'arrêt**
- somme = somme + n \* n
- n = n + 1
- Loop
- **Point d'arrêt**
- MsgBox ("la somme des carrés des entiers de 1 à " & max & " est : " & somme)
- End Sub

Exemple: max=5

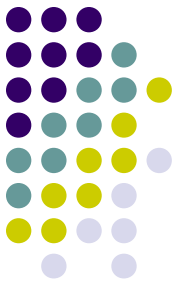
n	1	2	3	4	5	6
somme	0	1	5	14	30	55

# Comment être sûr que ce qu'on calcule est correct?



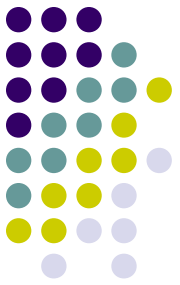
- Propriétés qu'ont les valeurs des variables à chaque itération de la boucle
- Exemple: à chaque début de boucle (au point d'arrêt) somme vaut la somme des carrés des entiers de 1 à la valeur de  $n$  moins 1.
- Raisonnement par **récurrence**: si la propriété est vraie au début d'une itération, elle reste vraie au début de l'itération suivante en fonction des modifications des valeurs des variables qui ont eu lieu durant l'itération.

# Exemple interactif : le jeu de l'addition sans répétition



- Sub jeu1()
- 'Cette macro permet de proposer à un utilisateur d'effectuer une addition de deux nombres au hasard et vérifie s'il a gagné ou perdu
- Dim x As Integer
- Dim y As Integer
- Dim z As Integer
- $x = \text{Int}(\text{Rnd}() * 100) + 1$
- $y = \text{Int}(\text{Rnd}() * 100) + 1$
- $z = \text{InputBox}(\text{"combien font " \& x \& " + " \& y \& " ?"})$
- If  $(z = x + y)$  Then
- MsgBox ("Bravo!")
- Else
- MsgBox ("Perdu!")
- End If
- End Sub

# Jeu de l'addition: plusieurs essais, calcul du score



- Sub jeu2()
- Dim x As Integer, y As Integer, z As Integer, nb As Integer, score As Integer
- $x = \text{Int}(\text{Rnd}() * 100) + 1$
- $y = \text{Int}(\text{Rnd}() * 100) + 1$
- $nb = 0$
- $score = 5$
- $z = \text{InputBox}(\text{"combien font " \& x \& " + " \& y \& " ?"})$
- Do While ( $nb < 5$ ) And ( $z \neq x + y$ )
- $score = score - 1$
- $z = \text{InputBox}(\text{"Erreur, encore un essai: combien font " \& x \& " + " \& y \& " ?"})$
- $nb = nb + 1$
- Loop
- If ( $z = x + y$ ) Then
- $\text{MsgBox}(\text{"Bravo, votre score est de : " \& score \& " points sur 5"})$
- Else
- $\text{MsgBox}(\text{"La bonne réponse était: " \& x + y})$
- End If
- End Sub

Le score est 5-nb d'erreurs:  
A chaque erreur on retire un point  
Et on propose un nouvel essai

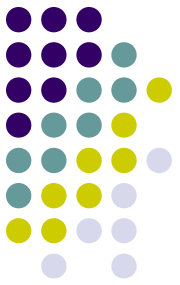
# Boucle pour réaliser un contrôle d'erreur



- Sub moyenne()
  - Dim note1 As Double
  - Dim note2 As Double
  - Dim moyenne As Double
  - note1 = InputBox("Entrez votre première note")
  - Do While (note1 < 0) Or (note1 > 20)
  - note1 = InputBox("Attention, entrez une note entre 0 et 20!")
  - Loop
  - note2 = InputBox("Entrez votre seconde note")
  - Do While (note2 < 0) Or (note2 > 20)
  - note2 = InputBox("Attention, entrez une note entre 0 et 20!")
  - Loop
  - moyenne = (note1 + note2) / 2
  - MsgBox ("votre moyenne est : " & moyenne)
  - End Sub

C'est l'utilisateur qui permet de sortir de la boucle: s'il continue de se tromper, la boucle ne s'arrête jamais

# Variante 1 de la boucle While



Syntaxe :

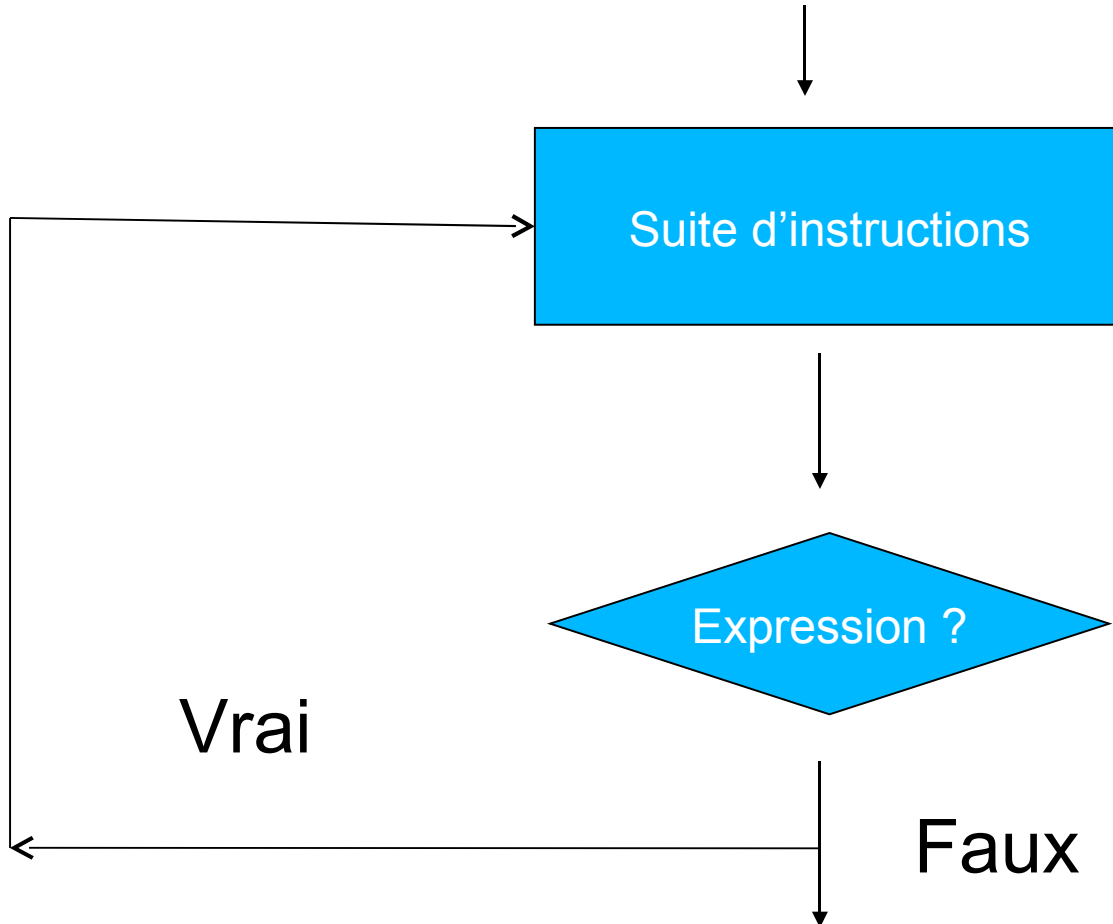
Do

Séquence d'instructions

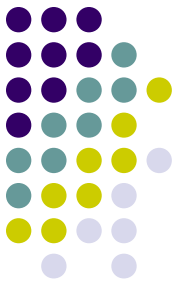
Loop While expr

# Instruction répétitive

## Do ... Loop While



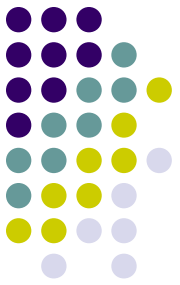




# Exemple Do... Loop While

Ce programme demande un chiffre encore et encore tant que l'utilisateur s'évertue à entrer un autre chiffre que 5

```
Sub TestLoop1()  
    Dim Reponse As Integer  
    Do  
        Reponse = InputBox("Entrez un chiffre")  
    Loop While Reponse <> 5  
End Sub
```



# Exemple Do While ... Loop

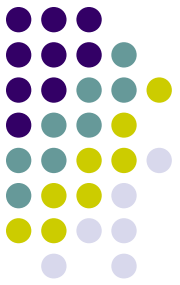
```
Sub TestLoop2()  
  Dim Reponse As String  
  Reponse = InputBox("Entrez le mot de passe")
```

```
Do While Reponse <> "youpi"  
  Reponse = InputBox("Mauvaise réponse.  
  Essayez encore")  
Loop  
End Sub
```

# Différence entre les deux variantes



- Avec Do While, le programme n'entre pas nécessairement dans la boucle. Si on entre le bon mot de passe immédiatement, le programme s'arrête.
- Dans l'exemple précédent on faisait DO quelque chose WHILE une condition, donc on faisait au moins une fois l'instruction
- Ici, c'est DO WHILE une condition, puis l'instruction. Le programme regarde IMMEDIATEMENT si la condition est satisfaite, et si elle ne l'est pas, n'exécute pas le corps de boucle

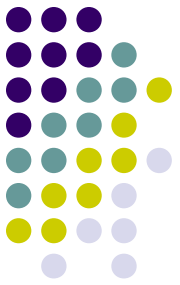


# Exemple Do... Loop While

Ici, par contre, avec le while a la fin de la boucle, le programme fait AU MOINS une fois "Mauvaise réponse", même si l'utilisateur répond tout de suite bien

```
Sub TestLoop3()  
  Dim Reponse As String  
  Reponse = InputBox("Entrez le mot de passe")  
  Do  
    Reponse = InputBox("Mauvaise réponse. Essayez  
encore")  
  Loop While Reponse <> "youpi"  
End Sub
```

# Exemple de la moyenne avec Do...loop While

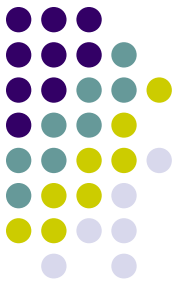


- Sub moyennebis()
- Dim note1 As Double
- Dim note2 As Double
- Dim moyenne As Double
- Do
- note1 = InputBox("Entrez votre première note")
- Loop While (note1 < 0) Or (note1 > 20)
- Do
- note2 = InputBox("Entrez votre seconde note")
- Loop While (note2 < 0) Or (note2 > 20)
- moyenne = (note1 + note2) / 2
- MsgBox ("votre moyenne est : " & moyenne)
- End Sub



# Boucles imbriquées

- Comme toutes les autres structures de contrôle les boucles peuvent être imbriquées les unes dans les autres.
- Exemple: le jeu de l'addition répété tant que l'utilisateur le souhaite.



# Jeu de l'addition répété

- nbadd = 0
- scoreglobal = 0
- Do
  - $x = \text{Int}(\text{Rnd}() * 100) + 1$
  - $y = \text{Int}(\text{Rnd}() * 100) + 1$
  - nbadd = nbadd + 1
  - nb = 0
  - score = 5
  - $z = \text{InputBox}(\text{"combien font " \& x \& " et " \& y \& " ?"})$
  - Do While (nb < 5) And (z <> x + y)
  - score = score - 1
  - $z = \text{InputBox}(\text{"Erreur, encore un essai: combien font " \& x \& " et " \& y \& " ?"})$
  - nb = nb + 1
  - Loop
  - If (z = x + y) Then
  - MsgBox ("Bravo!")
  - Else
  - MsgBox ("La bonne réponse était: " & x + y)
  - End If
- scoreglobal = scoreglobal + score
- continue = InputBox("Voulez-vous une nouvelle addition (oui/non) ?")
- Loop While (continue = "oui")
- $\text{note} = (20 * \text{scoreglobal}) / (5 * \text{nbadd})$
- MsgBox (" le jeu est fini, votre score est de : " & scoreglobal & " sur " & nbadd & "additions" & " et votre note est : " & note & "/20")

Le corps de la boucle Do...While est le traitement d'une seule addition (jeu précédent)

La variable scoreglobal accumule les scores des différentes additions

C'est l'utilisateur qui décide de l'arrêt de la boucle externe.