



# Module 2

# Répétitives

*DA & IA – Bloc 1*



# Plan du module

- Qu'est-ce qu'une répétitive?
- Boucle **while**
  - Cas d'utilisation 1
  - Cas d'utilisation 2
  - Exercice
- Boucle while et condition composée
- Boucle **do...while**
- Des boucles dans des boucles: Exercice

# Qu'est-ce qu'une répétitive?

ou **boucle** ou **structure répétitive** ou **structure itérative**

Ensemble d'actions qui peut se dérouler plusieurs fois **tant qu'**une condition reste vérifiée (VRAI).

# boucle *while*

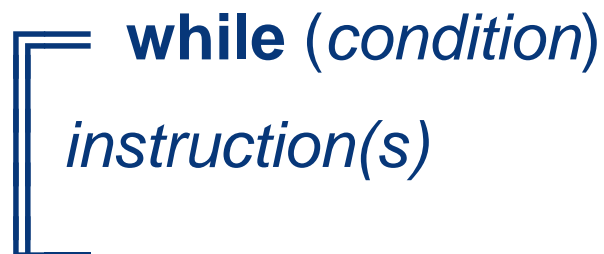
La forme de ce type de répétitive est la suivante :

**TANT QUE** (*condition*)

**FAIRE** *instruction(s)*

**FIN TANTQUE**

Le diagramme d'actions correspondant est le suivant :



# boucle *while*

## Cas d'utilisation 1 : Le nombre d'itérations est connu à l'avance

### Exemple introductif

Ecrire un programme qui permet d'encoder deux nombres et qui affiche leur produit sous la forme **nb1 \* nb2 = produit**

- IN: 2 nombres
- OUT: calcul



```
*  
obtenir nb1, nb2  
produit = nb1 * nb2  
sortir nb1,"*",nb2,"=",produit
```

# boucle *while*

## Cas d'utilisation 1 : Le nombre d'itérations est connu à l'avance

### Exemple 1

Ecrire un programme qui permet d'encoder deux nombres et qui affiche leur produit sous la forme **nb1 \* nb2 = produit et ceci 10 fois**

IN: 2 nombres (10\*)    OUT: calcul (10\*)



\*

```
iCalcul = 1
```

```
while (iCalcul ≤ 10)
```

```
    obtenir nb1, nb2
```

```
    produit = nb1 * nb2
```

```
    sortir nb1, "*", nb2, "=", produit
```

```
    iCalcul ++
```

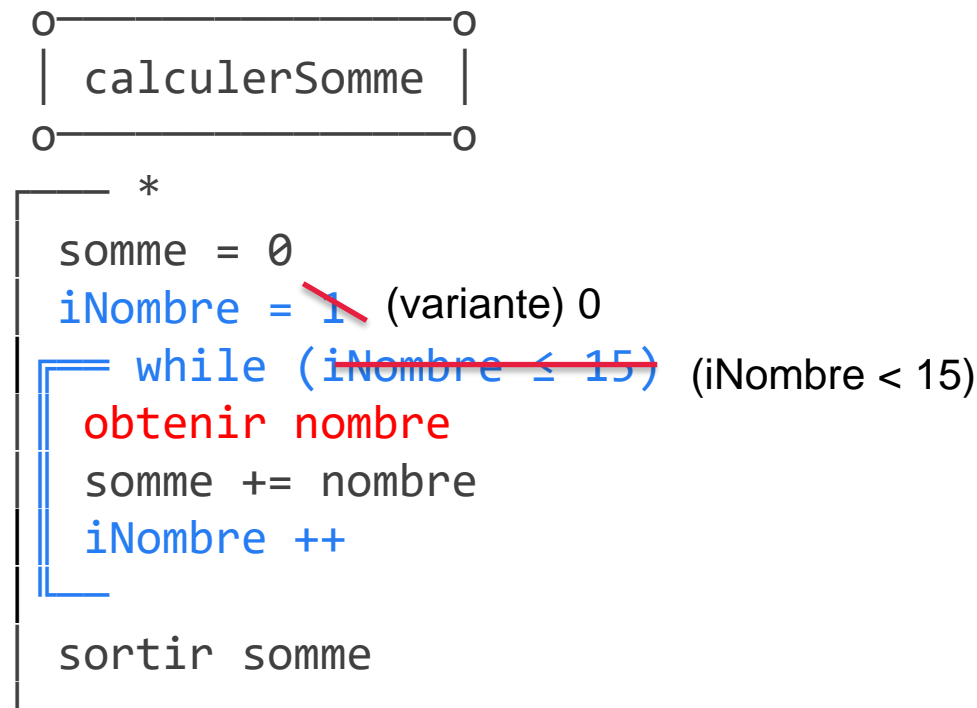
# boucle *while*

## Cas d'utilisation 1 : Le nombre d'itérations est connu à l'avance

### Exemple 2

Ecrire un programme qui permet d'encoder 15 nombres et qui affiche leur somme

IN: 15 nombres    OUT: somme

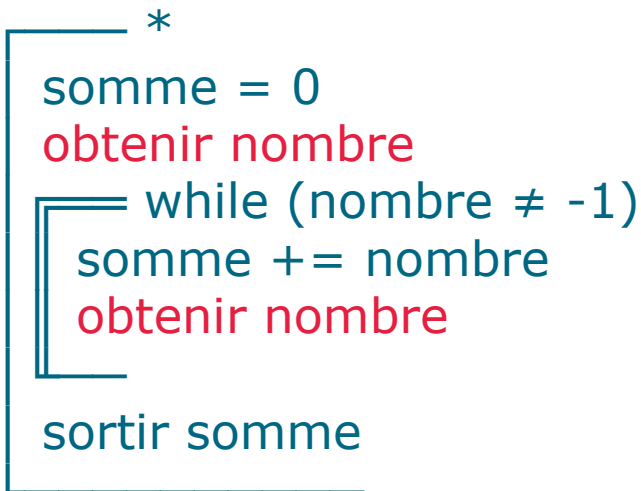


# boucle *while*

## Cas d'utilisation 2 : Le nombre d'itérations n'est pas connu à l'avance.

### Example 3

Ecrire un DA qui permet d'encoder des nombres positifs ( -1 pour terminer) et qui affiche leur somme.





# boucle *while*

## Exercice (résolu au cours)

Ecrire un DA qui permet d'encoder deux nombres naturels différents et qui affiche le carré de chaque naturel compris entre ces deux valeurs (valeurs comprises).

**IN:** nb1, nb2

**OUT:** le carré de chaque naturel compris entre nb1 et nb2 (inclus)

# boucle *while* et condition composée

Dans les exemples ci-dessous, que sait-on à propos des variables de la condition lorsque l'on sort de la boucle?

```
while (a<10 AND trouvé)
...
```

```
while (age>5 AND age ≤ 18)
...
```

```
while (valeur ≤ 0 OR valeur >30)
...
```

```
while (!trouvé OR nombre < min OR nombre > max)
...
```

# boucle do...while

La forme de ce type de répétitive est la suivante :

**FAIRE**

*instruction(s)*

**TANT QUE** *condition* **FIN TANTQUE**

**do**  
*instruction(s)*  
**while** (*condition*)

# boucle do...while

Ce type de boucle va être utilisé dans les contrôles de saisie.

## Exemple 4

On demande à l'utilisateur s'il veut sauvegarder son travail. Il doit répondre par OUI ou par NON et rien d'autre.

```
*  
[  
  Do  
  Sortir "Voulez-vous sauvegarder ?"  
  obtenir choix  
  while (choix ≠ "OUI" AND choix ≠ "NON")  
]
```

# boucle do...while

## Exercice

Complétez la condition dans le while pour obliger l'utilisateur à répondre par une des valeurs possibles.

```
o-----o
| validerChoix |
o-----o
```

```

*
do
  sortir "Encodez votre choix"
  sortir " 1 = français "
  sortir " 2 = anglais "
  sortir " 3 = néerlandais "
  sortir " 4 = espagnol "
  obtenir choix
while(...)
```

# Des boucles dans des boucles: Exercice

On obtient, pour chaque étudiant d'une classe, son nom (XXX pour terminer) ainsi que les cotes d'examen (/20) obtenues par celui-ci dans 5 cours respectivement français, anglais, PP, LP, math.

On vous demande d'afficher :

- pour chaque étudiant :
  - son nom
  - son pourcentage
  - le nombre de cours validés (cote au moins égale à 10)
- pour toute la classe :
  - le nombre total d'échecs
  - le nom de l'étudiant qui a le mieux réussi (meilleur pourcentage et aucun échec)
  - la moyenne en points (/20) pour le cours de math.

# Exercice

IN: Par étudiant  
(...\*)

{

nom (ou XXX )  
par cours  
(5\*)

{ cote (/20)

OUT:

- Par étudiant
- {
- nom  
pourcentage  
nbCoursValidés
- nbEchecs
  - nomMeilleur
  - moyenneMath

On obtient, pour chaque étudiant d'une classe, son nom (XXX pour terminer) ainsi que les cotes d'examen (/20) obtenues par celui-ci dans 5 cours respectivement français, anglais, PP, LP, math.

On vous demande d'afficher :

pour chaque étudiant :

son nom

son pourcentage

le nombre de cours validés (cote au moins égale à 10)

pour toute la classe :

le nombre total d'échecs

le nom de l'étudiant qui a le mieux réussi (meilleur

pourcentage et aucun échec)

la moyenne en points (/20) pour le cours de math.