

Fiche n°3 : Structures de contrôle et boucles

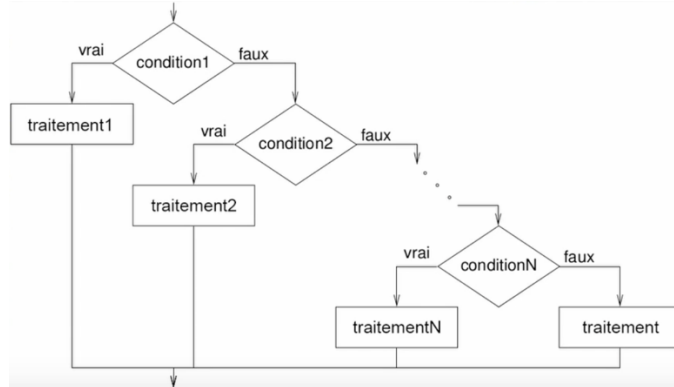
Les structures de contrôle permettent de faire des traitements séquentiels, conditionnels ou itératifs. Les trois types de structures de contrôle sont donc :

- les structures séquentielles
- les structures conditionnelles
- les structures itératives (que l'on appelle aussi les boucles)

Une structure conditionnelle commence par le terme "Si" pour évoquer une condition potentielle, sur laquelle on agit avec le terme "Alors". Si la condition n'est pas remplie le terme "Sinon" propose un traitement alternatif.

Une structure conditionnelle se termine avec le terme "Finsi" qui signifie Fin de Si.

La condition évoquée est toujours une variable booléenne dont le test est donc soit vraie (true) soit faux (false). Dans des cas plus complexes, une variable booléenne peut succéder à une autre pour compléter la description de la condition. Ainsi, si vous modélisez en algorithme un feu tricolore, vous aurez comme premier test booléen le fait de savoir s'il est rouge, comme second celui de savoir s'il est orange, ce qui aboutit à définir comme restant une troisième condition : le feu vert.



Il existe des structures conditionnelles à choix multiples (ou sélectives) qui permettent de traiter des structures conditionnelles imbriquées. Dans ce cas on utilise le terme "Selon" pour faire les tests.

Si on souhaite, par exemple, attribuer une lettre entre A et E des notes qui vont de 1 à 5 on peut procéder de la sorte :

DEBUT

 ECRIRE "Entrer une note entre 0 et 5 : "

 LIRE "nbr"

 Selon nbr Faire

 0 : ECRIRE "F"

 1 : ECRIRE "E"

 2 : ECRIRE "D"

 3 : ECRIRE "C"

 4 : ECRIRE "B"

 5 : ECRIRE "A"

 Sinon : ECRIRE "Vous avez saisi une mauvaise note"

FinSelon

FIN

Une structure itérative (que l'on appelle aussi boucle), permet de refaire faire une action à l'algorithme jusqu'à arriver à obtenir la réponse souhaitée.

Les boucles s'écrivent de la façon suivante :

```
DEBUT
TantQue condition Faire
FinTantQue
FIN
```

Le concept de TantQue permet de délimiter la répétition de l'action exécutée pour éviter qu'elle ne se fasse à l'infini et donc que la réponse ne soit plus pertinente ou qu'elle bloque le système en le saturant.

On pourrait avoir la boucle suivante pour créer une table de multiplications qui va de 1 à 10 :

```
DEBUT
  ECRIRE 'Entrez un chiffre'
  LIRE chiffre
  multiple = 0
  TANTQUEmultiple <=10 FAIRE
    ECRIRE chiffre, "x ", multiple " x ", chiffre * multiple
    multiple = multiple + 1
  FINTANTQUE
FIN
```

Dans cet exemple c'est la ligne : " multiple = multiple + 1" qui reproduit l'opération, et celle-ci est reproduite en core et encore jusqu'à ce que la valeur "multiple + 1" soit inférieure ou égale (<=) à 10.

La boucle "Pour" peut parfois vous aider à simplifier un test en indiquant un rang de valeur limite grâce au terme "Jusqu'à" avant d'initier l'action avec "Faire".

Si vous souhaitez par exemple convertir une unité de mesure anglo-saxonne comme les pouces en centimètre, et le faire de 1 à 12 pouces (ce qui correspond à un pied) vous pouvez procéder de la sorte :

```
DEBUT
  POUR valeur = 1 JUSQU'À 12 FAIRE
    ECRIRE valeur, "Pouces = ", valeur * 2,54, " cm"
  FINPOUR
FIN
```

Cette syntaxe permet de répéter automatique la conversion de la mesure pour toutes les valeurs allant de 1 à 12.