Programmation

Processing

FICHE BILAN

STRUCTURES ITERATIVES « FOR »

1. STRUCTURE ITÉRATIVE « FOR » - BOUCLE SIMPLE

SYNTAXE

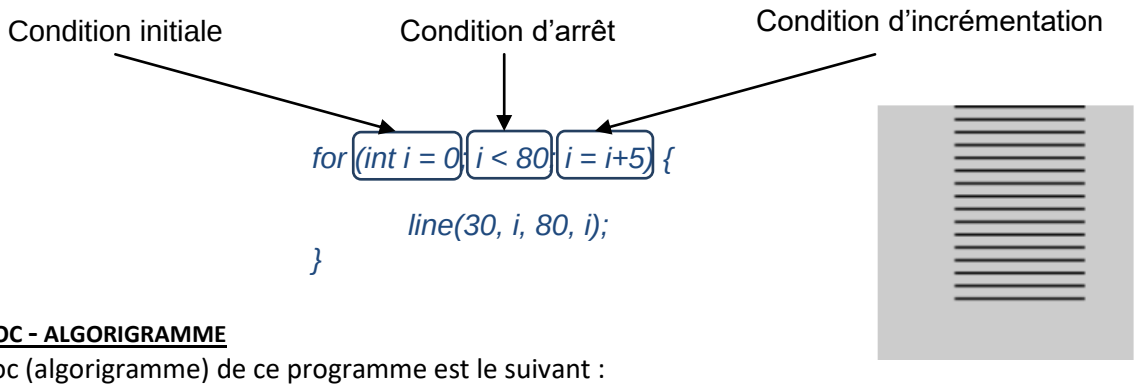
La boucle itérative « for » permet l'exécution d'un bloc d'instructions un certain nombre de fois (un nombre N), en faisant généralement varier un indice entre une valeur initiale et une valeur finale avec un pas d'incrémentation. La syntaxe est la suivante : Pour (i=1 à N) faire « instruction ». Le code est le suivant :

```
for (condition initiale ; condition d'arrêt ; condition d'incrémentation) {
        // code à exécuter tant que la condition d'arrêt n'est pas vérifiée
}
```

Si la condition est vrai, alors exécuter le bloc d'instruction situé entre « { » et « } ». L'expression « *condition* » est nécessairement de type booléen (vrai/faux).

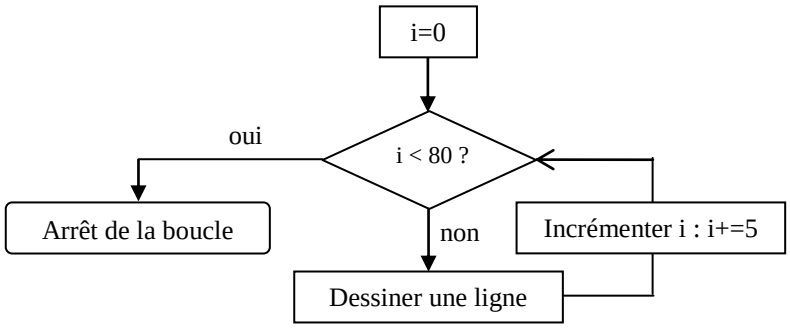
UN EXEMPLE

Un exemple de l'aide en ligne située dans la partie Control > Itération permet de tracer 16 lignes horizontales espacées de 5 pixels. Le code est le suivant :



LE SCHEMA-BLOC - ALGORIGRAMME

Le schéma-bloc (algorigramme) de ce programme est le suivant :



L'algorithme commence avec l'indice i=0. Il teste ensuite la condition d'arrêt : i < 80 ? Initialement la réponse est « Vrai ». L'algorithme exécute ensuite le bloc d'instruction situé entre les accolades, ici « dessiner une ligne à l'ordonnée i=0 ». Ensuite, il incrémente l'indice i : i+=5 est équivalent à i=i+5. L'indice i prenait la valeur 0, il prend maintenant la valeur 5.

Il teste ensuite la condition d'arrêt : i < 80 ? Ici, i=5<80 est vrai. L'algorithme exécute alors le bloc d'instruction situé entre les accolades, ici « dessiner une ligne à l'ordonnée i=5 ». Ensuite, il incrémente l'indice i : i+=5. L'indice i prend alors la valeur 10.

Etc…

L'algorithme s'arrête lorsque la condition d'arrêt i<80 n'est plus vérifiée. La dernière ligne horizontale est donc dessinée pour i=75. Les lignes sont donc dessinées pour les indices i∈{0, 5, 10, …, 75}, c'est-à-dire 16 itérations, soit 16 lignes.

2. DOUBLE BOUCLE « FOR »

SYNTAXE

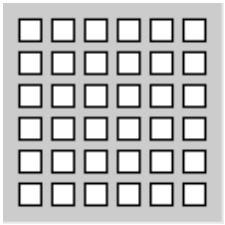
Généralement la double boucle est utilisée pour répéter des actions dans deux dimensions (dans le plan). La deuxième boucle est exécutée autant de fois que la condition d'arrêt de la première boucle n'est pas vérifiée. La syntaxe est la suivante :

```
for (condition initiale 1 ; condition d'arrêt 1; condition d'incrémentation 1) {
        for (condition initiale 2 ; condition d'arrêt 2; condition d'incrémentation 2) {
                // code à exécuter tant que la condition d'arrêt 2 n'est pas vérifiée
}
```

UN EXEMPLE

L'objectif est de dessiner une grille de 6×6 carrés de 10 pixels de largeur par 10 pixels de hauteur. Le programme est le suivant :

```
translate(7, 7);
//Première boucle (hauteur)
  for (int j = 0; j < 6; j ++) {
    //Seconde boucle (largeur) : dessiner une ligne
      for (int i = 0; i < 6; i++) {
          rect(i * 15, j * 15, 10, 10);
      }
   }
```

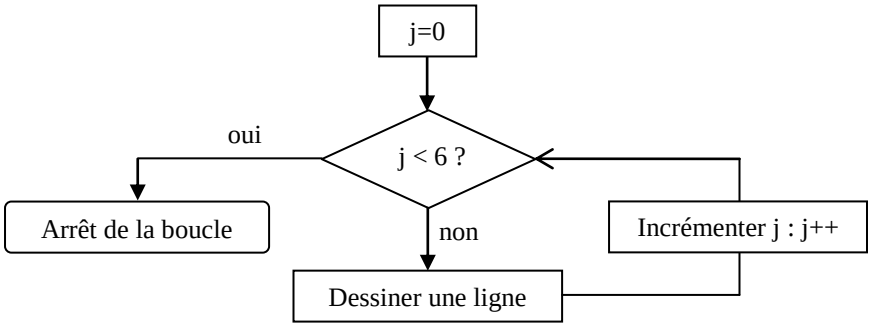


L'instruction *translate(7,7)* permet de déplacer l'origine des axes du point (0,0) vers le point (7,7), de sorte que les différents carrés ne soient pas collés aux bords de la fenêtre d'affichage.

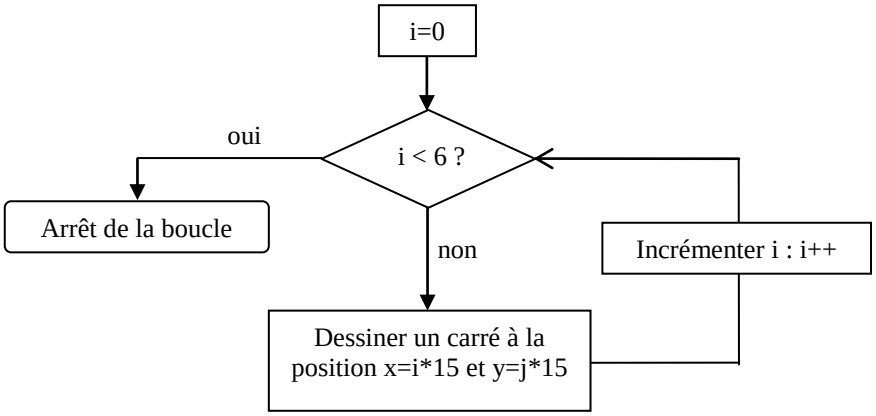
La deuxième boucle permet de dessiner une ligne horizontale de 6 carrés. La première boucle répète 6 fois l'opération la deuxième boucle permettant de dessiner la ligne de 6 carrés.
Dans les deux boucles précédentes, l'incrémentation i++ et j++ signifient « incrémenter i et j de 1 ». i++ est équivalent à i=i+1 et j++ est équivalent à j=j+1.

LE SCHEMA-BLOC - ALGORIGRAMME

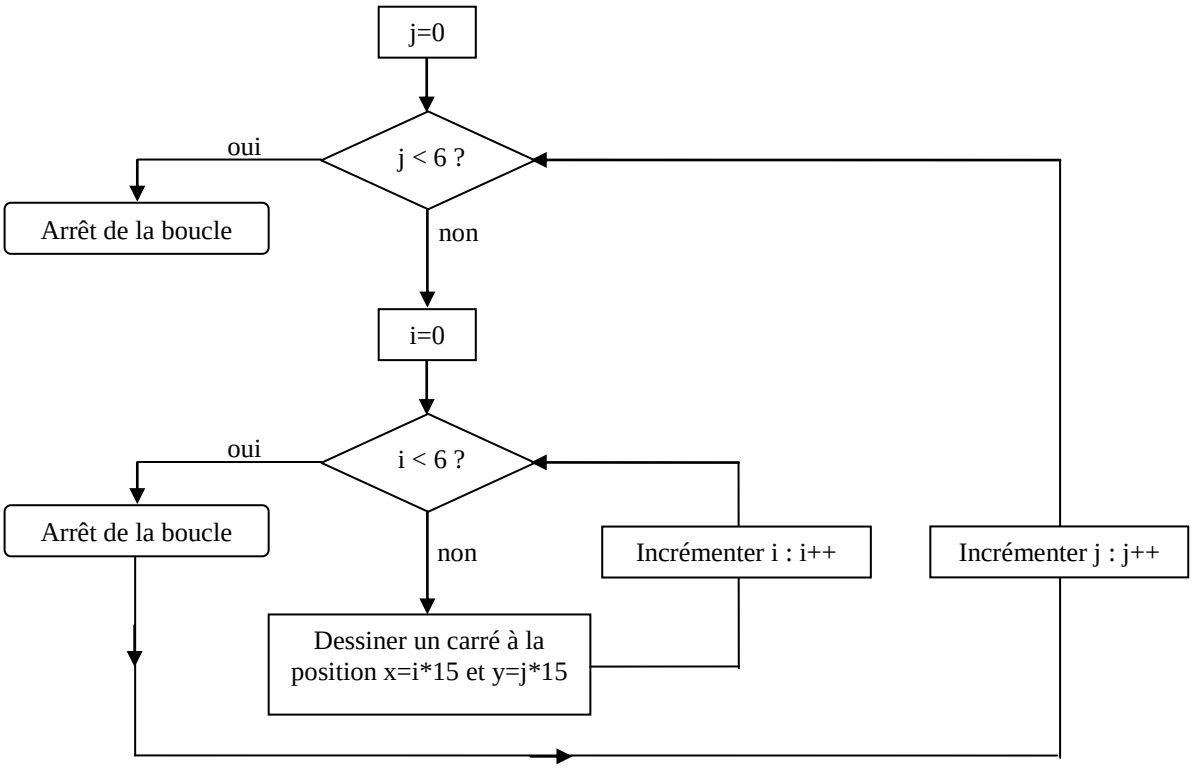
Le schéma-bloc simplifié du programme est le suivant :



Dans le schéma-block précédent, l'action « dessiner une ligne » est décrite par l'algorigramme suivant :



L'algorigramme global de la double boucle est donc le suivant :



L'algorithme commence avec l'indice **j=0**.
Il teste ensuite la condition d'arrêt : j < 6 ? Initialement la réponse est « Vrai ».
L'algorithme exécute ensuite le bloc d'instruction situé entre les accolades de cette première boucle, c'est-à-dire la deuxième boucle :

> Pour i ∈{0,1,…,5} dessiner un carré à la position x=i*15 ∈{0, 15, 30,…, 75}, y = j*15=0 (car j=0). Ceci permet de dessiner la **première ligne** de carrés.

Il incrémente l'indice j de 1 : **j=1**.
Il teste ensuite la condition d'arrêt : j < 6 ? qui est vrai.
L'algorithme exécute ensuite le bloc d'instruction situé entre les accolades de cette première boucle, c'est-à-dire la deuxième boucle :

> Pour i ∈{0,1,…,5} dessiner un carré à la position x=i*15 ∈{0, 15, 30,…, 75}, y = j*15=15 (car j=1). Ceci permet de dessiner la **deuxième ligne** de carrés.

Etc…

L'algorithme s'arrête lorsque j=6, c'est-à-dire lorsque toutes la grille de carrés est dessinée.