



FICHE MÉTHODE

Organigrammes de traitement (ou de programmation ou ordinoigramme) et logigrammes

Retrouvez toutes les fiches en intégralité avec les liens actifs



► Définition

L'organigramme de programmation est un schéma qui décrit l'enchaînement des opérations élémentaires d'un traitement informatique.

Il permet de visualiser les différentes étapes de traitement et met en évidence les phases d'analyses (losange) et les opérations (rectangle). Il permet de s'assurer que le mode opératoire couvre l'ensemble des situations possibles.

Il est très utilisé pour décrire un processus administratif, une procédure, et à ce moment-là, on l'appelle logigramme et sa représentation est moins stricte.

► Les symboles

Les symboles utilisés sont normalisés mais pas leur dimension (NFZ 67010).

Symboles	Explications	Organigramme	Logigramme
	Début ou fin de processus.	Obligatoire au début et à la fin.	Inutile.
	Sens de lecture et ordre de déroulement des opérations. Quand la ligne de liaison renvoie à une instruction antérieure, il s'agit d'une boucle d'itération .	Boucle d'itération.	Pas de boucle d'itération.
	Entrée ou sortie de données (entrée avant les tests, sortie d'informations en fin de programme).		
	C'est l'acte ou la tâche à réaliser pour faire le traitement. Cela peut être une opération ou un groupement d'opérations .		
	Expression du calcul logique, du test . Il faut choisir entre 2 propositions. Le texte écrit dans ce symbole est rédigé sous la forme d'une question à laquelle on ne peut répondre que par oui ou par non (<, >, =).	Il existe impérativement à la fin d'un traitement.	Inutile à la fin d'un logigramme.
	Indique les renvois quand l'organigramme se développe sur plusieurs feuillets (2 fois: au bas de la première partie et en haut de la seconde). Ils sont numérotés. Bas de 1 ^{re} page 1 1 Haut de 2 ^e page		Rarement utile.

► Règles de présentation

- L'organigramme doit avoir **un titre**.
- L'organigramme se **lit de haut en bas et de gauche à droite**. Si, pour une question de présentation, le sens de lecture est différent, la pointe de flèche placée sur la ligne de liaison précise le sens utilisé.

- Les symboles ou tests, sont de préférence alignés verticalement. Un test (losange) a une entrée et 2 sorties.
- Le symbole « traitement » a une entrée et une sortie.
- Les traits de liaison entre les symboles sont fléchés (horizontales, verticales). Ils se situent aux angles pour le losange et au milieu de la longueur pour les rectangles (voir tableau des symboles).
- Des flèches remontantes permettent d'inverser l'ordre de lecture du bas vers le haut. Il s'agit d'une **boucle d'itération**, c'est-à-dire un traitement qui se répète à l'identique un certain nombre de fois (renvoi à une instruction antérieure); ce type de traitement est nécessaire pour que l'ordinateur puisse effectuer automatiquement l'ensemble des tâches.

► Méthodes de construction

Il faut:

- repérer les conditions (losange) à l'aide de mots ou expressions comme « si », « dans le cas où », et les actions ou opérations de traitement;
- souligner les actions d'une couleur et les conditions (tests) d'une autre couleur;
- classer les conditions dans un ordre logique, d'abord les conditions déterminantes puis celles secondaires;
- suivre l'ordre chronologique de déroulement des opérations en numérotant actions et conditions;
- lister les travaux à effectuer dans un tableau;

Conditions	Actions

- réaliser le schéma en utilisant les symboles et en respectant les règles de présentation;
- s'assurer qu'**aucune branche ne reste en suspens**. Toute branche est reliée à l'organigramme.

<p>« Condition » si</p> <ul style="list-style-type: none"> • Alors oui « Traitement 1 » • Sinon « Traitement 2 » <p>Fin si</p>	<p>Itération</p> <ul style="list-style-type: none"> • Tant que « Non » • Faire « Condition » • Fin tant que « Condition » « Oui » (Traitement 1)
<pre> graph TD A([Début]) --> B{Condition} B -- Oui --> C[Traitement 1] B -- Non --> D[Traitement 2] C --> E([Fin]) D --> E </pre>	<pre> graph TD A{Condition} -- Oui --> B[Traitement 1] B --> A A -- Non --> A </pre>