

# Multimédia

## Programmation structurée

[programmation impérative](#)[langage de programmation](#)[programmation](#)

Source: [Wikipédia](#) sous licence [CC-BY-SA 3.0](#).

La liste des auteurs de cet article est disponible [ici](#).

La **programmation structurée** peut être vue comme un sous-ensemble, ou une branche, de la programmation impérative, un des paradigmes majeurs de la programmation.

Elle est célèbre pour son combat pour la suppression de l'[instruction goto](#) ou du moins pour la réduction de son [usage](#).

Au début des années 1970, les ouvrages de Jean-Dominique WARNIER (1920-1990) ont exprimé les principes de la [Logique](#) de Construction de Programmes (LCP). Selon ces principes, la structure hiérarchique d'un programme se déduit logiquement de la structure des [données](#) utilisées. La construction du programme commence par la structuration des données qu'il doit produire en sortie.

Il est en fait possible de faire de la [programmation structurée](#) dans n'importe quel [langage de programmation](#) procédural, mais depuis que, vers 1970, la [programmation](#) structurée est devenue une technique populaire, la plupart des nouveaux langages de programmation procéduraux ont intégré des mécanismes qui encouragent la programmation structurée (et ont quelquefois abandonné ceux qui facilitaient une programmation dé-structurée). Parmi les langages de programmation structurée les plus connus, on trouve Pascal et Ada.

Pour l'écriture de fragments assez courts, la programmation structurée recommande une [organisation](#) hiérarchique simple du code. On peut le faire dans la plupart des langages de programmation modernes par l'utilisation de structures de contrôles *while*, *repeat*, *for*, *if .. then .. else*. Il est également recommandé de n'avoir qu'un [point](#) d'entrée pour chaque boucle (et un point de sortie unique dans la programmation structurée originelle), et quelques langages l'imposent.

Les programmeurs doivent décomposer leur code en petits sous-programmes (appelés fonctions et procédures dans certains langages), assez petits pour être facilement compris. En général les programmes doivent éviter d'utiliser des variables globales ; au lieu de cela, les sous-programmes doivent utiliser des variables locales et agir sur des arguments fournis explicitement en [paramètre](#), par valeur ou par référence. Ces techniques aident à créer des petits morceaux de code, faciles à comprendre isolément sans avoir à étudier l'[ensemble](#) du programme.

La programmation structurée est souvent confondue avec la méthodologie de développement (*top-down design*). Dans cette approche les programmeurs décomposent la