

5200  
Yves TABOURIER · Arnold ROCHFELD · Claude FRANK

307  
C40

BIBLIOTHEQUE DU CERIST

# LA PROGRAMMATION STRUCTUREE

en informatique

les éditions d'organisation - paris

Yves TABOURIER  
Arnold ROCHFELD  
Claude FRANK

LA PROGRAMMATION  
STRUCTURÉE  
EN INFORMATIQUE

1975

**les éditions d'organisation**

5, rue Rousselet - 75007 PARIS

# Table des matières

<b>Introduction</b> .....	11
<b>Chapitre I. — La structuration des programmes</b>	
1. Les buts .....	15
2. Programmes structurés .....	16
2.1 Diagrammes .....	16
2.2 Programme propre .....	17
2.3 Diagrammes privilégiés .....	18
2.4 Programmes structurés .....	21
2.5 Découpage d'un programme : routines, procédures, fonctions, etc. ....	24
3. Comment écrire un programme structuré .....	25
3.1 Considérations préliminaires .....	25
3.2 Les outils de la programmation structurée .....	26
4. La structuration en action .....	29
4.1 Exemple du tri par interclassement : exposé du problème .....	29
4.2 Tri par interclassement : décomposition détaillée .....	30
4.3 Schéma de décomposition .....	35
4.4 Procédure PL/1 de tri par interclassement .....	36
4.5 Procédure COBOL .....	37
4.6 Variante pour le pavé « interclassement » .....	38
4.7 Structuration d'un programme existant .....	39
4.8 Commentaires sur les deux exemples .....	44
<b>Chapitre II. — Aspects liés aux langages de programmation</b>	
1. Généralités .....	47
1.1 De la théorie à la pratique .....	47
1.2 Enrichissement par l'expérience .....	47
2. Procès du GOTO .....	50
2.1 Introduction .....	50
2.2 Raisons qui militent en défaveur de l'emploi du GOTO .....	52
3. Exemple à codifier .....	60
4. PASCAL .....	61
5. PL/1 .....	62
5.1 Caractères favorables à la structuration .....	62
5.2 Caractères dangereux pour la structuration .....	65
5.3 Conclusion .....	65
6. COBOL .....	67
6.1 Les constituants de base .....	67

6.2	Remarques et rappels .....	71
6.3	Exemple de programmation structurée en COBOL .....	73
6.4	Autre exemple de programmation structurée en COBOL : la procédure de tri ..	85
7.	<b>FORTRAN</b> .....	86
7.1	Préliminaires .....	86
7.2	Conventions proposées .....	86
7.3	Suggestion .....	89
7.4	Exemple du tri .....	90
7.5	Exemple du PGCD .....	91
8.	<b>Assembleur</b> .....	91
8.1	Présentation des modules en assembleur .....	91
8.2	Structuration des instructions .....	93
8.3	Macro-instructions de structuration .....	96

### Chapitre III. — Organisation de l'équipe de programmation

1.	L'équipe de programmation .....	103
1.1	Chef de programmation .....	104
1.2	Programmeur de soutien .....	104
1.3	Secrétaire de programmation .....	104
1.4	Equipe d'équipes .....	105
2.	Mise en place de l'équipe .....	105
3.	Conséquences de cette organisation .....	105
4.	Problèmes de documentation et de maintenance des programmes .....	106
4.1	Documentation .....	106
4.2	Mise à jour et maintenance .....	110
5.	La programmation structurée, méthode générale de formation .....	110
5.1	Le compagnonnage habituel en programmation .....	110
5.2	Aspect global de la programmation structurée .....	111
5.3	Aspect universel de la programmation structurée .....	111

### Chapitre IV. — Comparaison avec d'autres méthodes

1.	La méthode LCP .....	
1.1	Généralités .....	115
1.2	Structures LCP .....	116
1.3	Comparaison des structures LCP et PS .....	118
1.4	Les traitements des données en LCP .....	118
1.5	L'emploi des langages de programmation .....	119
1.6	Organisation du travail .....	119
2.	Langages non procéduraux .....	120
3.	<b>ATLAS</b> .....	122
3.1	ATLAS comme super-LCP (ou LCP automatique) .....	122
3.2	ATLAS comme langage sans état .....	122
4.	<b>Conclusion</b> .....	123

<b>Conclusion</b> .....	127
<b>Annexe I. — Aspects mathématiques de la structuration des programmes</b>	
1. Introduction .....	131
2. Les éléments .....	131
2.1 Fonctions .....	131
2.2 Programme, fonction de programme .....	132
2.3 Graphes de contrôle, diagrammes .....	133
2.4 Programme propre, programmes équivalents .....	136
2.5 Décomposition .....	137
2.6 Graphes privilégiés, formules de programme .....	137
2.7 Programmes structurés .....	138
2.8 Arborescence d'un programme structuré .....	139
3. Les résultats .....	141
3.1 Résultats concernant l'existence .....	141
3.2 Résultats concernant la pratique de la programmation structurée.....	150
<b>Annexe II. — Macros pour la structuration en assembleur</b> .....	159
<b>Bibliographie</b> .....	163

Nous remercions J.-P. Vial, M. Csernel, P. Tersinet, A. Petit, J.-L. Armengaud pour l'aide qu'ils nous ont apportée dans l'élaboration de cet ouvrage, ainsi que M. H. N'Guyen pour ses intéressantes remarques concernant notamment l'influence de la programmation structurée sur la formation des programmeurs et le partage des tâches.

## INTRODUCTION (\*)

L'activité de programmation possède un privilège redoutable : c'est l'un des derniers refuges de l'artisanat, mais un artisanat hautement abstrait.

Par une étrange déviation, de nombreux praticiens tendent à en faire une activité sauvage et farouchement individuelle. C'est renier les grandes traditions de l'artisanat : utilité, solidité et communicabilité de l'ouvrage, et compagnonnage; tout artisanat est une activité sociale de production et, loin d'être sauvage, obéit aux « règles de l'art ».

Très vite, des hommes ont senti le danger de prendre l'ésotérique pour génial, le biscornu pour beau et l'acrobatique pour efficace : ils se sont forgé leurs propres règles de l'art et, pour certains, ont codifié et enseigné le savoir-faire qu'ils avaient acquis.

La « programmation structurée » fait partie de ce courant d'efforts. Fondée sur des bases mathématiques dues à Böhm et Jacopini [8] (\*\*), elle en tire les conséquences pratiques pour tous les aspects importants de l'activité de programmation : elle ne se limite pas à la structuration des programmes mais rend compte aussi des aspects liés au langage de programmation et à l'organisation du travail, tant pour la production des programmes que pour leur documentation et leur maintenance.

Les grands noms qui lui sont attachés sont, entre autres, ceux de Dijkstra et de Mills. La réussite la plus marquante est le système d'interrogation en temps réel réalisé par IBM pour le *New York Times* [5], [6] : 83 000 lignes de PL/1 écrites à une moyenne de 35 par homme et par jour, avec une fiabilité telle qu'il n'y a eu qu'une seule perte de contrôle pendant les 13 premiers mois de fonctionnement opérationnel du système.

---

(\*) Le présent ouvrage a pour origine le Rapport de Synthèse et Formation n° 84 de la Direction Scientifique de la SEMA.

(\*\*) Les numéros entre crochets renvoient aux références bibliographiques (p. 163).

Le présent ouvrage est organisé de la façon suivante :

Une première partie comprend les chapitres I et II ainsi que les annexes I et II.

Le chapitre I présente la méthode de structuration des programmes. Le chapitre II, quant à lui, fait apparaître les conséquences de la méthode sur l'outil de l'informaticien que constituent les différents langages de programmation. Un exemple a été traité dans ces différents langages de façon à permettre une comparaison plus concrète des facilités de structuration offertes par ces derniers.

Cette première partie est donc plus spécialement destinée aux professionnels de l'informatique, dans toute sa variété. Ainsi l'annexe I, complément du chapitre I, est-elle destinée aux informaticiens intéressés par les fondements mathématiques de la méthode et qui souhaitent trouver les démonstrations des différents théorèmes utilisés. L'annexe II s'adresse plus particulièrement à ceux, toujours nombreux, qui programment en assembleur. Elle leur fournit des indications sur les possibilités de structurer des programmes écrits dans cette classe de langages.

La deuxième partie, moins technique, comprend les chapitres III et IV.

Le chapitre III intéressera particulièrement, du moins le souhaitons-nous, ceux qui sont amenés à diriger des projets informatiques importants. Il présente le concept clé d'« équipe de programmation », ainsi que les solutions que permet d'apporter la « programmation structurée » aux problèmes de documentation, de partage des tâches et de formation des programmeurs.

Enfin, le chapitre IV présente une comparaison de la « programmation structurée », avec d'autres méthodes et une réflexion sur la place qu'elle occupe dans le développement de l'activité de programmation. Il est donc susceptible d'intéresser des non-praticiens qui cherchent simplement à avoir une vue d'ensemble de cette activité.