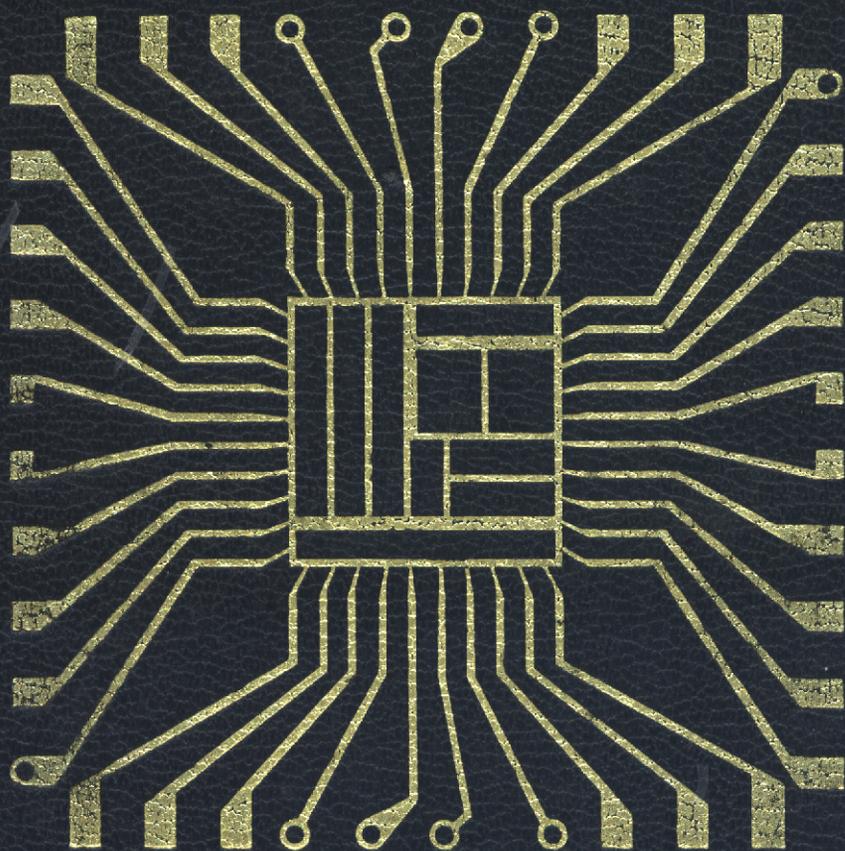


SCIENCES ET PRATIQUES DE L'INFORMATIQUE



Bordas informatique

BIBLIOTHEQUE DU CERIST

**PASCAL
ISO/AFNOR**

BIBLIOTHEQUE DU CERIST



PASCAL ISO / AFNOR

programmation déductive et description de la norme

par

Alain TISSERANT

Ingénieur civil des Mines
Maître de conférences
à l'École des Mines de Nancy
et au C.R.I.N

Bordas informatique

5797

La présente édition a été achevée
d'imprimer en juin 1988

© BORDAS, Paris, 1986

ISBN : 2-04-013355-0

" Toute représentation ou reproduction, intégrale ou partielle, faite sans le consentement de l'auteur, ou de ses ayants-droit, ou ayants-cause, est illicite (loi du 11 mars 1957, alinéa 1^{er} de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal. La loi du 11 mars 1957 n'autorise, aux termes des alinéas 2 et 3 de l'article 41, que les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective d'une part, et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration "

Préface

Penserez-vous que j'ai le goût du paradoxe si j'écris en tête de ce livre que l'informatique évolue lentement et que ce qui y évolue le plus lentement n'est pas ce qui est rigide, « hard », c'est-à-dire le matériel, mais ce qui pourrait apparaître plus flexible, « soft », je veux dire la manière d'exprimer les programmes ? La raison en est sans doute profonde : c'est que les techniques de l'électronique bougent plus vite et plus facilement que les habitudes et les esprits des utilisateurs. Pour les langages de programmation en particulier, le réemploi des programmes déjà écrits chez les gros utilisateurs d'informatique — la sacro-sainte compatibilité — a toujours constitué un frein important à l'évolution; aujourd'hui, les exigences de facilité de réalisation sur des ordinateurs peu coûteux et la simplicité du langage pour des utilisateurs non professionnels sont d'autres freins. De sorte que Pascal, qui date de 15 ans, est encore relativement peu répandu — bien que ce soit sans doute le langage le plus utilisé dans les universités — et qu'il est encore nécessaire de le faire connaître.

Car il faut revenir sur la notion de simplicité d'un langage : s'agit-il de la rapidité de son premier apprentissage, ou de la commodité de son utilisation ? Un langage trop simple, aux notions trop rudimentaires, se révèle lourd d'emploi dès qu'on dépasse les applications élémentaires. Un langage trop riche de notions importantes, mais délicates de compréhension, peut faciliter l'écriture de programmes mais se rélever lourd à dominer.

Pascal constitue un bon compromis entre ces difficultés. Il a été construit en réaction aux langages quelque peu monstrueux apparus à la fin des années soixante et qui voulaient intégrer toutes les notions que pouvaient songer à utiliser les programmeurs dans quelque domaine que ce soit. Mais il n'a pas pour autant renoncé aux idées qui permettent une programmation systématique, claire, sûre, concise : structuration du programme en procédures, récursivité, types définis par le programmeur. En ce sens, il est très moderne, et tout utilisateur de l'informatique a intérêt à le connaître.

Ecrire un livre sur un langage demande aussi de trouver un compromis. S'agit-il de donner une pure description du langage ou d'aider à apprendre la programmation ? C'est très différent car connaître un langage en détail n'aide guère à bien programmer. Les idées qui ont présidé à la création de Pascal invitent cependant à lier les deux objectifs.

Et pour ce qui concerne la connaissance même du langage, s'agit-il de la faire acquérir méthodiquement dans un manuel d'apprentissage, ou s'agit-il de permettre de retrouver rapidement les différents éléments en fonction des besoins dans un compagnon d'utilisation ? Le lecteur débutant souhaitera d'abord l'un, mais demandera ensuite l'autre... et préférera pour cela ne pas devoir changer d'ouvrage.

La longue expérience d'enseignement d'Alain Tisserant lui a permis de trouver un équilibre entre tous ces objectifs assez contradictoires et de surmonter, autant qu'il est possible, une difficulté bien connue de ceux qui enseignent la programmation (et bien d'autres choses d'ailleurs), je veux parler de la récursivité de l'apprentissage : on n'apprend pas à programmer à quelqu'un qui ne sait pas du tout ce qu'est programmer; on n'apprend pas à programmer sans langage, mais on n'apprend pas non plus un langage de programmation à quelqu'un qui ne sait pas programmer.

C. PAIR

Avant-Propos

Parmi les langages conçus pour résoudre la crise du logiciel des années 70, c'est Pascal qui a eu le plus de succès. Normalisé, implanté sur pratiquement tous les ordinateurs, du micro au géant, son usage continue encore de s'étendre, des programmes de 10 lignes aux 800 000 lignes d'un central téléphonique.

Plus simple que Cobol ou Fortran, mais aussi plus rapide, il est nettement plus efficace pour la conception, la mise au point et la maintenance des programmes.

Pascal est un langage pour débutant, il a en effet été conçu pour enseigner la programmation en tant que discipline rigoureuse fondée sur quelques concepts fondamentaux apparaissant clairement dans le langage.

Pascal est un langage pour professionnel, il est très efficace à l'exécution, et permet l'écriture de programmes lisibles, donc modifiables par une autre personne, et faciles à mettre au point.

Cet ouvrage a l'ambition de répondre à une demande des praticiens, et aussi de pouvoir servir de support de cours. Progressif, pour être efficace, il comporte des répétitions volontaires, mais emploie un vocabulaire rigoureux et des notations précises. On y trouvera de nombreux exemples, mais peu de détails technologiques.

Le langage décrit correspond aux normes ISO et AFNOR.

La partie I permet au néophyte de se lancer dans la programmation, en utilisant une méthode qui a fait ses preuves. Dans la partie II, on décrit les objets manipulés puis en partie III leur manipulation; les éléments introduits dans ces deux chapitres correspondent à ce qui est nécessaire à la programmation de tous les jours. La partie IV complète les notions qui la précèdent, pour pouvoir utiliser le langage Pascal en entier. Enfin, la partie V résume le langage en un court aide-mémoire. Des annexes apportent quelques compléments, et facilitent un accès direct dans l'ouvrage.

Pascal, défini en 1969 par Niklaus Wirth, en application d'idées sur la programmation structurée, a connu une rapide diffusion. Aujourd'hui, une norme internationale a fixé le langage; cependant les processeurs de Pascal disponibles n'obéissent pas tous à cette norme; il sera sans doute nécessaire d'apporter des modifications mineures aux programmes donnés en exemple, avant de pouvoir les utiliser sur son ordinateur préféré.

BIBLIOTHEQUE DU CERIST

Table des matières

Préface	page V
Avant-propos	VII

Chapitre 1. **ABORDER L'ECRITURE D'UN PROGRAMME** 1

1.0 Analyser	1
1.1 Partir du résultat	1
1.2 Programmer en Pascal	4
1.3 Itération avec compteur	7
1.4 Conditionnelle	15
1.5 Itération avec arrêt	19
1.6 Tables	24
1.7 Choisir une méthode	34
1.8 Exercices	34

Chapitre 2. **LES BASES DU LANGAGE PASCAL** 37

2.0 Rédiger un programme	37
2.1 Notions élémentaires	38
2.2 Structure globale d'un programme	41
2.3 Les objets manipulés	44
2.3.0 Types	44
2.3.1 Constantes	44
2.3.2 Variables	46
2.3.3 Types simples	47
2.3.4 Définition de type	50
2.3.5 Structure de tableau	51
2.3.6 Structure d'article	53
2.3.7 Règles de compatibilité	54
2.4 Entrées-sorties : input et output	55

Chapitre 3. LES BASES DU LANGAGE : LA MANIPULATION DES OBJETS	63
3.0 Enoncés	63
3.1 Affectation, expression	66
Analyse; Expressions ; Opérateurs arithmétiques, booléens, de relation; fonctions pré-définies arithmétiques, ordinales, de transfert, booléennes	
3.2 Conditionnelles	74
Enoncés si, cas	
3.3 Itérations	77
Enoncés répéter, tant que, pour, de branchement	
3.4 Procédure	85
Exemple; Local/Global; Portée; Paramètres	
3.5 Fonctions	93
Portée; Effets de bord	
3.6 Exercices	95
 Chapitre 4. DES NOTIONS PLUS AVANCÉES	 99
4.0 Introduction	99
4.1 Paramètres	99
Paramètres procédure, fonction, tableau ajustable; Directives	
4.2 Récursivité	105
4.3 Compactage, chaînes de caractères	107
4.4 Ensembles	114
4.5 Articles avec variantes, énoncé avec	116
4.6 Pointeurs et variables dynamiques	119
4.7 Fichiers	125
4.8 Fichiers de texte	130
4.9 Hiérarchie des types	133
4.10 Compléments :	133
Nombre aléatoires; Indentation; Portabilité	
4.11 Exercices	137
 Chapitre 5. AIDE MÉMOIRE	 139
5.0 Programme, bloc, portée, région	139
5.1 Identificateurs, symboles, séparateurs	140
5.2 Variables	140

Table des matières

XI

5.3	Types	141
5.4	Constantes, réels, entiers, chaînes	142
5.5	Intervalle, énumération, booléen, caractère, tableau, article	142
5.6	Affectation	143
5.7	Si, cas, pour, tant que, répéter, avec, read, wriye	144
5.8	Procédure, fonction, ensemble, chaîne, fichier, pointeur ...	145
Chapitre 6. ANNEXES		149
6.0	Index des programmes	149
6.1	Aspects internes, jeu de caractères	151
6.2	Diagrammes de syntaxe	155
6.3	Représentation matérielle, mots-clés	163
6.4	Variante francophone	165
6.5	Bibliographie	167
6.6	Extensions,; Norme étendue	169
6.7	Index	171