

collection langages et algorithmes de l'informatique

sous la direction de JEAN VIGNES

J.-L. NEBUT

**théorie et pratique
du langage
PASCAL**

**ÉDITIONS
TECHNIP**

collection langages et algorithmes de l'informatique

sous la direction de JEAN VIGNES

Professeur à l'Université Pierre et Marie Curie de Paris (Paris VI)

Conseiller scientifique à l'Institut Français du Pétrole

JEAN-LOUIS NEBUT

Ingénieur ESTP

Maître Assistant à l'Université de Rennes I

**théorie et pratique
du langage
PASCAL**

1980

ÉDITIONS TECHNIP • 27 RUE GINOUX • 75737 PARIS CEDEX 15

technip

© 1980. Éditions Technip - Paris

Toute reproduction, même partielle, de cet ouvrage
par quelque procédé que ce soit est rigoureusement
interdite par les lois en vigueur.

ISBN 2-7108-0389-5

DANS LA MEME COLLECTION :

LANGAGES

- Théorie et pratique de la programmation FORTRAN, par J. VIGNES et M. LA PORTE avec la collaboration de B. HALLOPEAU
- Le langage APL, par B. ROBINET
- Ordinateur CDC 3600. Le langage COMPASS et le système, par M. GINESTE, G. GUIHO, C. PROST et J. VIGNES
- Théorie et pratique de la programmation COBOL, par J. VIGNES, C. COURTY et C. DEBOST

ALGORITHMES

- Algorithmes numériques. Analyse et mise en œuvre.
Tome 1. Arithmétique des ordinateurs. Systèmes linéaires, par M. LA PORTE et J. VIGNES.
Tome 2. Équations et systèmes non linéaires, par J. VIGNES avec la collaboration de R. ALT et M. PICHAT.
- Algorithmes d'accélération de la convergence. Étude numérique, par C. BREZINSKI.
- Algorithmes et pratique de programmation linéaire, par P. CHRÉTIENNE, Y. PESQUEUX et J.-C. GRANDJEAN.

Table des matières

1. INTRODUCTION

1.1. Codification d'un programme PASCAL	2
1.1.1. Vocabulaire de base	2
1.1.2. Règles d'écriture	3
1.1.3. Commentaires	4
1.1.4. Exemple	4
1.2. Mode de présentation du langage	5
1.3. Réalisation d'un programme	7
1.4. Remarques générales	7
EXERCICE	8

2. STRUCTURE DE BASE D'UN PROGRAMME PASCAL

2.1. Structure générale	11
2.2. Définition d'un programme	13
2.3. Les constantes et leur déclaration	14
2.4. Déclaration des variables. Types de variables	15

2.5. Les instructions fondamentales	17
2.5.1. La séquence	17
2.5.2. L'affectation	18
2.5.3. Les itérations : instruction-tantque et instruction-répéter	19
2.5.4. La sélection : instruction-si	24
2.5.5. Introduction aux instructions de lecture-écriture ...	26
2.6. Récapitulation	29
EXERCICES	31

3. LES TYPES - DEFINITION DE TYPE ET TYPES SIMPLES

3.1. Rappels et généralités	33
3.2. Les types de base	33
3.2.1. Le type entier	34
3.2.2. Le type réel	35
3.2.3. Le type booléen	38
3.2.4. Le type caractère	39
3.3. Déclarations de types	42
3.4. Définition de type	43
3.5. Les types simples	44
3.5.1. Les types ordinaux	44
3.5.2. Le type scalaire par énumération	45
3.5.3. Le type intervalle	47
3.5.4. Récapitulation sur les types simples	49
EXERCICES	51

4. LES EXPRESSIONS ARITHMETIQUES ET LOGIQUES

4.1. Priorité des opérateurs	53
4.2. Syntaxe des expressions	56
4.3. Type d'une expression	57
4.4. Les opérateurs arithmétiques	58
4.5. Les opérateurs logiques	60
4.6. Les opérateurs de relation	60
4.7. Compatibilité de type	61
4.8. Résultats d'une expression	62
4.8.1. Avec des entiers	62
4.8.2. Avec des réels	62
4.8.3. Avec des booléens	62
4.9. Exemple	63
EXERCICES	66

5. STRUCTURATION DES DONNEES EN TABLEAU

5.1. Généralités sur la structuration des données	67
5.2. Le type tableau	68
5.3. Variable de type tableau	70
5.4. Variables indicées ou éléments de tableau	70
5.5. L'instruction-pour	73
5.6. Chaînes	77

5.7. Exemples	78
5.7.1. Occurences des lettres dans une phrase	78
5.7.2. Tri de vecteur	78
EXERCICES	80

6. STRUCTURATION DES DONNEES EN ARTICLES

6.1. Le type article	81
6.1.1. Le type article simple	81
6.1.2. Type article avec variante	83
6.2. Variables articles	86
6.3. Désignation des champs	87
6.4. L'instruction-avec	90
EXERCICES	94

7. LES SOUS-PROGRAMMES (1) PROCEDURES SANS PARAMETRE

7.1. Définition. Utilisation	97
7.2. Déclaration d'un sous-programme sans paramètre	99
7.3. Localité des déclarations et accès aux objets	101
7.3.1. Localité	101
7.3.2. Niveaux	103
7.3.3. Accès aux objets	103
7.3.4. Autres règles d'accès et de visibilité	104
7.3.5. Exemple récapitulatif	105
7.4. Activation de sous-programme sans paramètre	106

EXERCICES	110
-----------------	-----

8. LES SOUS-PROGRAMMES (2) PROCEDURES ET FONCTIONS AVEC PARAMETRES

8.1. Définition. Utilisation	113
8.2. Déclaration de procédure	114
8.3. Les paramètres	115
8.3.1. Paramètre formel	115
8.3.2. Paramètre effectif	115
8.3.3. Déclaration des paramètres formels	115
8.3.4. Modes de transmission des paramètres	116
8.3.4.1. Mode par valeur	116
8.3.4.2. Mode par variable (ou par référence)	117
8.3.4.3. Mode procédure	119
8.3.4.4. Mode fonction	121
8.4. Instruction-appel de procédure	122
8.5. Les fonctions	123
8.5.1. Déclaration de fonction	123
8.5.2. Appel de fonction	124
8.6. Directives	126
8.7. Déclaration des sous-programmes	127
EXERCICES	128

9. AUTRES INSTRUCTIONS OU INDICATIONS SOUS-PROGRAMMES STANDARD

9.1. Compactage des données structurées	129
9.2. Déclarations d'étiquettes	132

9.3. Instruction de branchement inconditionnel	133
9.4. Instruction-cas	135
9.5. Instruction vide	137
9.6. Procédures standard	137
9.6.1. Procédures de manipulation de fichiers	137
9.6.2. Procédures d'allocation dynamique	138
9.6.3. Procédures de compactage et de décompactage	138
9.7. Fonctions standard	139
9.7.1. Fonctions arithmétiques	139
9.7.2. Fonctions de conversion	140
9.7.3. Fonctions d'énumération	140
9.7.4. Prédicats	140
9.8. Déclarations et instructions : récapitulation	141
EXERCICES	143

10. STRUCTURATION DES DONNEES EN ENSEMBLE

10.1. Déclaration de type ensemble	145
10.2. Variables ensemble	146
10.3. Expression d'ensembles	147
10.4. Opérateurs sur les ensembles	149
10.5. Utilisation des ensembles	149
EXERCICES	152

11. STRUCTURATION DES DONNEES EN FICHIER

11.1. Type fichier	153
11.2. Variable fichier	154
11.3. Accès aux fichiers : tampon de fichier	154
11.4. Ouverture et fermeture de fichier	156
11.4.1. Ouverture en écriture : <i>rewrite</i>	157
11.4.2. Ouverture en lecture : <i>reset</i>	158
11.4.3. Fermeture des fichiers	159
11.5. Ecriture sur fichier : <i>put</i>	159
11.6. Lecture d'un fichier : <i>get</i>	161
11.7. Modification d'un fichier	162
11.8. Fin de fichier : <i>eof</i>	164
11.9. Les fichiers et l'environnement	164
EXERCICES	166

12. LES ENTREES-SORTIES DE TEXTE

12.1. Structure des fichiers texte	169
12.2. Prédicat <i>eofln</i>	170
12.3. Généralités sur les procédures <i>read-write</i>	171
12.4. La procédure <i>read</i>	172
12.5. La procédure <i>readln</i>	175

12.6. La procédure *write* 176
12.7. La procédure *writeln* 181
12.8. Le fichier standard *input* 181
12.9. Le fichier standard *output*. Pilotage de l'imprimante 181
EXERCICES 183

**13. POINTEURS ET VARIABLES
DYNAMIQUES**

13.1. Type pointeur et variable pointeur 185
13.2. Création d'une variable dynamique : *new* 187
13.3. Désignation des variables pointées 189
13.4. Première utilisation des pointeurs 190
13.5. Destruction des variables dynamiques : *dispose* 193
13.6. Déclarations de structures de données récursives 194
13.7. Deuxième utilisation des pointeurs : traitement de listes . 196
13.8. Les variables : récapitulation 200
EXERCICES 201

**14. PROCEDURES ET FONCTIONS
RECURSIVES**

14.1. Récursivité simple 203
14.2. Fonctionnement de la récursivité 205
14.3. Récursivité croisée 208
14.4. Premier exemple : parcours ordonné de structure récursive . 209

14.5. Deuxième exemple : solution par essais successifs 211

EXERCICES 213

Corrigé des exercices

Corrigé de l'exercice du chapitre 1 215

Corrigé des exercices du chapitre 2 216

Corrigé des exercices du chapitre 3 218

Corrigé des exercices du chapitre 4 220

Corrigé des exercices du chapitre 5 222

Corrigé des exercices du chapitre 6 224

Corrigé des exercices du chapitre 7 227

Corrigé des exercices du chapitre 8 230

Corrigé des exercices du chapitre 9 233

Corrigé des exercices du chapitre 10 234

Corrigé des exercices du chapitre 11 235

Corrigé des exercices du chapitre 12 242

Corrigé des exercices du chapitre 13 244

Corrigé des exercices du chapitre 14 248

Annexe 1. Restrictions et extensions courantes 251

Annexe 2 261

Annexe 3. Grammaire Pascal 263

Index 271

Introduction

La première version du langage PASCAL a été définie il y a dix ans, à Zürich. Son auteur, Niklaus Wirth, poursuivait deux buts : il voulait construire un outil pour enseigner la programmation d'une manière systématique, à partir de concepts simples bien reflétés par le langage; d'autre part, la mise en oeuvre de cet outil devait produire des programmes sûrs et efficaces sur les ordinateurs classiques. Ce langage s'est développé dans les milieux universitaires sur tous les continents, et il commence sa percée dans le monde industriel et commercial. Les fabricants de micro processeurs 16 bits proposent PASCAL avant FORTRAN pour leurs petites machines.

Le relativement faible intérêt des grands constructeurs d'ordinateurs pour PASCAL provient du fait qu'il n'y a pas de définition "officielle" du langage, et que les multiples équipes qui ont construit les traducteurs ont souvent ajouté quelque chose au langage pour leurs propres besoins. Cela explique aussi la difficulté de réaliser un ouvrage de présentation de PASCAL : il y a plusieurs définitions du langage de référence, et la norme internationale définissant PASCAL est encore en cours d'élaboration.

Le langage présenté ici se réfère aux ouvrages de N. Wirth ⁽¹⁾ et aux travaux des divers groupes de normalisation de PASCAL. Il contient en annexe une liste non exhaustive des plus fréquentes extensions que le programmeur peut trouver dans son manuel d'utilisation.

Cette introduction présente la manière de coder un programme PASCAL et la façon dont sera exposé le langage dans les chapitres 2 à 14.

⁽¹⁾ Kathleen Jensen et Niklaus Wirth. "Pascal User Manual and Report". Second Edition (1978). Springer-Verlag.

REMARQUES.

1. On ne confondra pas "symbole" et "représentation du symbole" lors de la codification. Ainsi, := est un seul symbole, et sa représentation est faite à l'aide de deux caractères typographiques, : et =, qui représentent aussi deux symboles.

2. L'écriture des mots-clés et des identificateurs avec minuscules ou majuscules est indifférente. Dans cet ouvrage, les mots-clés et les identificateurs prédéfinis seront écrits en minuscules, et les mots-clés seront soulignés; les identificateurs définis par le programmeur seront écrits en majuscules.

3. Lorsque la représentation typographique des symboles spéciaux n'est pas possible, des alternatives sont utilisées : elles dépendent d'une implantation particulière sur un matériel donné. Le tableau suivant donne les alternatives les plus courantes :

références	[]	{ }	↑	≠	≤	≥
alternatives	(. .)	(* *)	@	<>	<=	>=
	(/ /)		^	#	<u>le</u>	<u>ge</u>

1.1.2. Règles d'écriture.

Il n'y a pas de format ou cadrage particuliers pour un programme PASCAL : l'écriture sur un bordereau de perforation est libre. Par contre, on respectera les règles suivantes :

a) Tout symbole spécial dont la représentation s'exprime par plus d'un caractère est écrit sur autant de cases qu'il y a de caractères dans la représentation du symbole.

b) Les blancs (ou espaces), les fins de ligne (ou fins de carte) et les commentaires (voir plus loin) sont des séparateurs.

c) Il doit y avoir au moins un séparateur ou un symbole spécial entre deux entités telles que : identificateur, mot-clé, nombre.

d) Le nombre de séparateurs n'est pas significatif : on peut par exemple intercaler des cartes vierges dans un programme.

1.1.3. Commentaires.

Un commentaire est une séquence de caractères encadrée par les deux symboles spéciaux { et }. Les commentaires apportent quelques explications sur le programme; on indiquera en général en commentaire la signification des variables et des constantes, les conditions de validité des programmes, le rôle des sous-programmes.

Il résulte des règles précédentes qu'on peut placer un commentaire n'importe où (sauf à l'intérieur d'un identificateur, d'un nombre, d'un mot-clé ou d'un symbole spécial dont la représentation nécessite plus d'un caractère).

Dans les programmes donnés en exemple, le texte des commentaires sera écrit en minuscules.

1.1.4. Exemple.

Le texte suivant est la codification du premier programme du chapitre 2. On remarquera les décalages du texte à chaque ligne : ils servent à mettre en valeur les différentes sections et la structure du programme.

```

PROGRAM TABULATION;
  CONST
    M=10;
  TYPE
    INDICE=0..M;
  VAR
    N:INDICE;
    NP2:INTEGER; {2 PUISSANCE N}
  BEGIN
    N:=0; NP2:=1;
    REPEAT
      N:=N+1;
      WRITE(N,N*N,SQRT(N),1/N);
      NP2:=NP2*2;
      WRITELN(NP2)
    UNTIL N=M
  END.

```