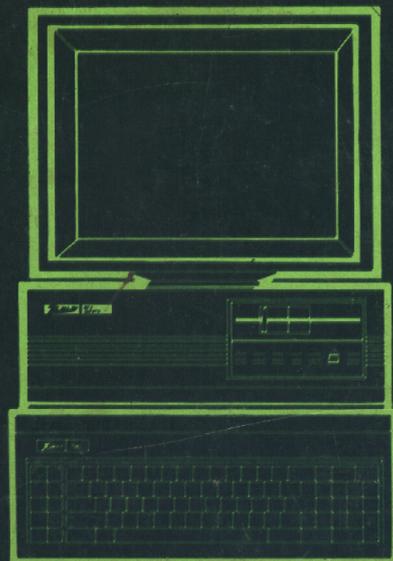


**BAGHDALI LATIFA  
LARIBI SID-AHMED**

# **LANGAGE DE**

# **PROGRAMMATION PASCAL**

**A l'usage des ingénieurs  
techniciens et étudiants**



**ENAL**

Bibliothèque  
CERIST @ 2018

Tst 3024

**BAGHDALI LATIFA  
LARIBI SID-AHMED**

**LANGAGE  
DE  
PROGRAMMATION  
PASCAL**

**A l'usage des ingénieurs  
techniciens et étudiants**

**Entreprise nationale du livre<sup>3</sup>  
3, Bd Zirout Youcef  
Alger**

2564

## AVANT PROPOS

Jusqu'aux années 50 la programmation n'était conçue que comme une codification détaillée sous forme binaire, autrement dit comme l'écriture d'instructions sous forme d'une suite de 0 et de 1. Les insuffisances et l'incommodité manifeste de ce procédé ont amené les programmeurs à abandonner cette technique.

Ecrire un programme en langage machine, seul langage à la disposition des utilisateurs imposait au programmeur :

1. une connaissance approfondie des caractéristiques de la machine utilisée ; l'utilisation d'une autre machine imposait une réécriture complète du programme, l'échange de programmes entre utilisateurs de machines différentes était quasiment prohibé;

2. une pratique très particulière de la machine car il est pratiquement impossible de détecter les erreurs, même les plus simples ou de retrouver la structure du programme;

3. une disponibilité exceptionnelle car le programmeur, ou plutôt le codeur, passait un temps considérable à construire le code optimal de son programme pour utiliser au mieux les performances de sa machine dont les caractéristiques changeaient très rapidement avec le progrès technologique, rendant par la même occasion le code mis au point caduque.

Dès lors on vit apparaître des langages dits de "haut niveau" dont la caractéristique essentielle est l'indépendance par rapport à la structure de la machine. Il fallait une codification comprise à la fois par la machine et par le programmeur pour lui éviter toutes les lourdeurs du langage machine. L'idée de base de ces langages était de rendre inutiles pour le programmeur ces notions de mots mémoire, de registres etc... qui faisaient les joies et les nuits blanches de ses prédécesseurs. Il a donc fallu développer un outil qui permette de traduire le programme écrit dans un langage proche de l'expression écrite humaine en un programme compris par la machine : cet outil joue le rôle de traducteur d'un programme écrit dans un langage dit évolué, c'est à dire proche du langage de l'utilisateur, en langage machine. Cet outil, qui est un programme intermédiaire entre l'homme et la machine porte le nom d'INTERPRETEUR ou de COMPILATEUR. La programmation peut être ramenée à une simple codification de l'algorithme construit par l'utilisateur en un programme compris par la machine. Cette codification est d'autant plus simple que le langage utilisé est proche du langage d'expression de l'algorithme de résolution du problème. La structure d'un grand nombre de langages évolués se veut très proche du langage algorithmique.

Bibliothèque CERIST @ 2018

mique, pour des raisons d'efficacité, aussi les nomme-t-on souvent "langages algorithmiques".

Il existe actuellement plusieurs milliers de langages de programmation qui sont à des stades divers, allant de la proposition à l'utilisation effective. Parmi les langages utilisés par un nombre appréciable d'utilisateurs on peut citer :

- \* FORTRAN (pour les calculs scientifiques),
- \* COBOL (pour la gestion),
- \* ALGOL et PASCAL (langages algorithmiques),
- \* BASIC (langage grand public),
- \* LISP,
- \* PROLOG,
- \* APL ...

Dans cet ouvrage nous étudions le langage PASCAL, l'un des langages les plus répandus dans les milieux de l'enseignement. Tout au long du développement de l'ouvrage nous avons introduit des définitions syntaxiques qui s'adressent plus particulièrement aux informaticiens. Le lecteur pourra "sauter", dans une première lecture, ces paragraphes sans que cela nuise à la compréhension de la plupart des notions.

## SOMMAIRE

|  | pages     |
|--|-----------|
| AVANT PROPOS   | 5         |
| <b>CHAPITRE 1. INTRODUCTION AU LANGAGE PASCAL</b>          | <b>13</b> |
| 1.1 Présentation du langage PASCAL                         | 13        |
| 1.2. Notations et vocabulaire                              | 13        |
| 1.2.1. Les caractères de base                              | 13        |
| 1.2.2. Les symboles spéciaux                               | 14        |
| 1.2.3. Les noms dans PASCAL                                | 14        |
| 1.2.3.1. Les mots clés réservés                            | 14        |
| 1.2.3.2. Les mots clés standards (ou prédéfinis)           | 15        |
| 1.2.3.3. Les identificateurs                               | 15        |
| 1.3. Les commentaires                                      | 16        |
| 1.4. Règles d'écriture d'un programme PASCAL               | 16        |
| 1.5. Mode de présentation du langage PASCAL                | 18        |
| Exercices  | 20        |
| <b>CHAPITRE 2. STRUCTURE DE BASE D'UN PROGRAMME PASCAL</b> | <b>21</b> |
| 2.1. Structure générale                                    | 21        |
| 2.2. Définition syntaxique d'un programme                  | 22        |
| 2.3. En-tête de programme PASCAL                           | 23        |
| 2.4. Déclarations d'un programme PASCAL                    | 24        |
| 2.4.1. Déclaration des étiquettes                          | 24        |
| 2.4.2. Définition des constantes                           | 25        |
| 2.4.3. Définition de types                                 | 26        |
| 2.4.4. Déclaration de variables                            | 27        |
| 2.4.5. Déclaration des fonctions et procédures             | 29        |
| Exercices  | 29        |
| <b>CHAPITRE 3. TYPES DE DONNEES</b>                        | <b>31</b> |
| 3.1. Les types scalaires prédéfinis                        | 32        |
| 3.1.1. Le type entier                                      | 32        |
| 3.1.1.1. Les opérateurs arithmétiques                      | 33        |
| 3.1.1.2. Les opérateurs de relation                        | 34        |
| 3.1.1.3. Les fonctions prédéfinies                         | 34        |

|  |           |
|--|-----------|
| 3.1.2. Le type réel                                | 35        |
| 3.1.2.1. Les opérateurs arithmétiques              | 36        |
| 3.1.2.2. Les opérateurs de relation                | 36        |
| 3.1.2.3. Les fonctions prédéfinies                 | 36        |
| 3.1.3. Le type logique                             | 38        |
| 3.1.3.1. Les opérateurs logiques                   | 38        |
| 3.1.3.2. Les opérateurs de relations               | 41        |
| 3.1.4. Le type caractère                           | 41        |
| 3.1.4.1. Les fonctions prédéfinies                 | 42        |
| 3.1.4.2. Les opérateurs de relations               | 42        |
| 3.2. Les types scalaires définis par l'utilisateur | 43        |
| 3.2.1. Le type scalaire par énumération            | 43        |
| 3.2.2. Le type intervalle                          | 45        |
| 3.3. Les types ordinaux                            | 46        |
| 3.3.1. La fonction succ                            | 47        |
| 3.3.2. La fonction pred                            | 48        |
| Exercices  | 49        |
| <br>   |           |
| <b>CHAPITRE 4. LES INSTRUCTIONS DANS PASCAL</b>    | <b>52</b> |
| 4.1. L'instruction d'affectation                   | 52        |
| 4.2. L'instruction composée                        | 53        |
| 4.3. Les instructions répétitives                  | 54        |
| 4.3.1. L'instruction WHILE                         | 54        |
| 4.3.2. L'instruction REPEAT                        | 55        |
| 4.3.3. L'instruction FOR                           | 56        |
| 4.4. L'instruction conditionnelle                  | 58        |
| 4.5. L'instruction de branchement inconditionnel   | 59        |
| 4.6. L'instruction CASE                            | 60        |
| Exercices  | 64        |
| <br>   |           |
| <b>CHAPITRE 5. LE TYPE STRUCTURE : TABLEAU</b>     | <b>70</b> |
| 5.1. Le type tableau                               | 70        |
| 5.1.1. Tableau à une dimension                     | 71        |
| 5.1.2. Tableau à plusieurs dimensions              | 72        |
| 5.1.3. Les opérations permises sur les tableaux    | 75        |
| 5.2. Les chaînes de caractères                     | 76        |
| 5.3. Exemple                                       | 80        |

|   |            |
|---|------------|
| Exercices   | 81         |
| <b>CHAPITRE 6. STRUCTURE DE DONNEES : ARTICLES</b>      | <b>86</b>  |
| 6.1. Le type article simple                             | 87         |
| 6.2. Type article avec variante                         | 88         |
| 6.3. Référence à un champ particulier d'un RECORD       | 93         |
| 6.4. L'instruction WITH                                 | 94         |
| 6.5. Exemple  | 99         |
| Exercices   | 100        |
| <b>CHAPITRE 7. LES ENSEMBLES DANS LE LANGAGE PASCAL</b> | <b>104</b> |
| 7.1. Le type structuré SET                              | 104        |
| 7.2. L'opérateur d'affectation                          | 105        |
| 7.3. Les opérateurs d'ensembles " + - * "               | 106        |
| 7.3.1. L'opérateur "+" ou union                         | 106        |
| 7.3.2. L'opérateur "-" ou différence                    | 106        |
| 7.3.3. L'opérateur "*" ou intersection                  | 106        |
| 7.4. Les opérateurs de relations                        | 107        |
| 7.4.1. L'égalité d'ensembles                            | 107        |
| 7.4.2. L'inégalité d'ensembles                          | 107        |
| 7.4.3. L'inclusion d'ensembles                          | 107        |
| 7.4.4. La contenance d'ensembles                        | 107        |
| 7.4.5. L'appartenance à un ensemble                     | 107        |
| Exercices   | 108        |
| <b>CHAPITRE 8. STRUCTURATION DES DONNEES EN FICHIER</b> | <b>110</b> |
| 8.1. Le type structuré fichier                          | 110        |
| 8.2. Accès aux élément d'un fichier                     | 111        |
| 8.3. Fin de fichier EOF                                 | 112        |
| 8.4. Ouverture et fermeture d'un fichier                | 112        |
| 8.4.1. Ouverture en écriture : REWRITE                  | 112        |
| 8.4.2. Ouverture en lecture : RESET                     | 114        |
| 8.4.3. Fermeture de fichier                             | 115        |
| 8.5. Ecriture sur fichier : PUT                         | 115        |

|  |            |
|--|------------|
| 8.6. Lecture d'un fichier : GET                                | 116        |
| 8.7. Modification d'un fichier                                 | 116        |
| 8.8. Exemple   | 117        |
| Exercices  | 118        |
| <b>CHAPITRE 9. LES FICHIERS TEXTES</b>                         | <b>122</b> |
| 9.1. Structure du fichier texte                                | 122        |
| 9.2. Le prédicat EOLN  | 122        |
| 9.3. La procédure READ   | 124        |
| 9.4. La procédure READLN                                       | 126        |
| 9.5. La procédure WRITE  | 126        |
| 9.6. La procédure WRITELN                                      | 128        |
| 9.7. Exemple d'utilisation des fichiers Textes                 | 129        |
| 9.8. Les fichiers textes prédéfinis                            | 129        |
| 9.9. Exemple d'utilisation des fichiers prédéfinis             | 130        |
| Exercices  | 131        |
| <b>CHAPITRE 10 : TYPE POINTEUR ET LES VARIABLES DYNAMIQUES</b> | <b>134</b> |
| 10.1. Variable dynamique                                       | 134        |
| 10.2. Le type pointeur   | 134        |
| 10.3. Création d'une variable dynamique                        | 136        |
| 10.4. Désignation des variables dynamiques                     | 136        |
| 10.5. Construction de structures de données complexes          | 137        |
| 10.5.1. Construction d'une liste linéaire                      | 139        |
| 10.5.2. Recherche d'un élément dans une liste                  | 140        |
| 10.5.3. Insertion d'un nouvel élément dans une liste           | 141        |
| 10.5.4. Suppression d'un élément dans une liste                | 141        |
| 10.5.5. Extension des listes linéaires                         | 142        |
| 10.6. Destruction d'une variable dynamique                     | 142        |
| 10.7. Exemple  | 145        |
| Exercices  | 146        |
| <b>CHAPITRE 11. LES PROCEDURES ET FONCTIONS</b>                | <b>150</b> |
| 11.1. Structure de bloc dans PASCAL                            | 150        |

|   |            |
|---|------------|
| 11.2. Les procédures                      | 152        |
| 11.3. Activation d'une procédure          | 152        |
| 11.4. Les paramètres                      | 152        |
| 11.4.1. Paramètres transmis par variables | 153        |
| 11.4.2. Paramètres transmis par valeur    | 153        |
| 11.4.3. Paramètre fonction                | 155        |
| 11.4.4. Paramètre procédure               | 157        |
| 11.5. Les fonctions                       | 158        |
| 11.6. Activation d'une fonction           | 159        |
| 11.7. Récursivité des sous-programmes     | 160        |
| Exercices                                 | 163        |
| <b>BIBLIOGRAPHIE</b>                      | <b>170</b> |