

TABLE DES MATIÈRES

| | |
|---|-----------|
| INTRODUCTION | 2 |
| LEÇON 1 | 9 |
| <hr/> | |
| 1. Introduction | 9 |
| 2. Description syntaxique et sémantique d'un langage de programmation | 10 |
| 2.1 – Spécification de la syntaxe de PASCAL | 10 |
| 2.2 – Spécification de la sémantique de PASCAL | 12 |
| 3. Programmation en PASCAL | 13 |
| 3.1 – Structure générale des programmes PASCAL | 13 |
| 3.2 – Symboles et séparateurs | 14 |
| 3.3 – Identificateurs | 14 |
| 3.4 – Types et constantes de base | 15 |
| 3.5 – Constantes | 22 |
| 3.6 – Déclarations | 23 |
| 3.7 – Expressions | 32 |
| 3.8 – Instructions | 37 |
| | |
| LEÇON 2 | 53 |
| <hr/> | |
| 1. Introduction | 53 |
| PROGRAMMATION EN PASCAL | |
| 2. Tableaux | 53 |
| 2.1 – Exemple introductif : tri par sélection | 53 |
| 2.2 – Type tableau | 55 |
| 2.3 – Variables tableaux | 55 |
| 2.4 – Types compatibles | 56 |
| 2.5 – Variables indicées | 56 |
| 2.6 – Tableaux à plusieurs dimensions et tableaux de tableaux (*) | 57 |
| 2.7 – Représentation d'un tableau en mémoire (**) | 58 |
| 2.8 – Exemple d'algorithme utilisant des tableaux : carré magique | 59 |
| 3. Ensembles (**) | 59 |
| 3.2 – Représentation d'ensembles en mémoire (**) | 61 |
| 3.3 – Notation d'ensembles (**) | 61 |
| 3.4 – Expressions de type ensemble et opérations sur les ensembles (**) | 61 |
| 3.5 – Affectation à une variable de type ensemble (**) | 62 |
| 3.6 – Exemple : crible d'Eratosthène (**) | 63 |
| 4. Types compacts (**) | 65 |
| 5. Type des chaînes de caractères (*) | 66 |
| 5.1 – Chaînes de caractères de longueur fixe en PASCAL standard (*) | 66 |
| 5.2 – Chaînes de caractères en PASCAL VAX 8600 (*) | 67 |

5.3 - Chaînes de caractères de longueur variable en PASCAL Macintosh (**) 67
 5.4 - Transport d'un programme utilisant des chaînes de caractères entre le Macintosh et le VAX 8600 68

ALGORITHMIQUE ET STRUCTURES D'INFORMATION

6. Recherche en table 69
 6.1 - Définition de la notion de table et critères de comparaison de représentations de tables des machines 70
 6.2 - Recherche directe 70
 6.3 - Recherche séquentielle 74
 6.4 - Recherche dichotomique 76
 6.5 - Recherche par hachage 78
 7. Algorithmes de tri (en mémoire centrale) 82
 7.1 - Tri par remontée des bulles 83
 7.2 - Ordre de grandeur du temps optimum de tri 83
 7.3 - Heapsort 84

ARCHITECTURE DES ORDINATEURS

8. Principe simplifié de fonctionnement d'un ordinateur (**)
 8.1 - Architecture de la machine (**)
 8.2 - Unité de traitement (unité centrale) (**)
 8.3 - Mémoires centrales (**)
 8.4 - Bus de communication (**)
 8.5 - Séquencement de la machine (**)
 8.6 - Langage machine (**)
 8.7 - Exemples de programmes en langage machine (**)

LEÇON 3 97

1. Introduction 97
 2. Procédures sans paramètres 97
 2.1 - Déclaration de procédure 98
 2.2 - Appel de procédure 99
 3. Procédures avec paramètres par valeur ou par variable 101
 3.1 - Paramètres formels 101
 3.2 - Paramètres effectifs 103
 3.3 - Passage de paramètres par valeur 104
 3.4 - Affinement de la notion de variable 105
 3.5 - Passage de paramètres par variable 106
 3.6 - Passage d'un élément de tableau en paramètre par variable (*) 108
 4. Fonctions (*) 109
 4.1 - Déclaration de fonction (*) 109
 4.2 - Appel de fonction (*) 110
 5. Visibilité des identificateurs : structure de bloc 111
 5.1 - Occurrences de définition et d'utilisation d'un identificateur 111
 5.2 - Identificateurs locaux et globaux 112
 5.3 - Visibilité et portée des identificateurs 114
 5.4 - Quelques avantages et inconvénients de la structure de bloc (**)
 116

| | |
|--|------------|
| 6. Effets de bord (*) | 120 |
| 6.1 – Effets de bord des procédures (*) | 120 |
| 6.2 – Effets de bord des fonctions (*) | 120 |
| 7. Alias | 121 |
| 8. Paramètres procéduraux et fonctionnels (**) | 125 |
| 8.1 – Passage de procédures en paramètres (**) | 126 |
| 8.2 – Passage de fonctions en paramètres (**) | 126 |
| 8.3 – Contexte d'exécution d'un sous-programme passé en paramètre (**) | 126 |
| 9. Récursivité | 127 |
| 9.1 – Récursivité simple | 127 |
| 9.2 – Règle de recopie | 130 |
| 9.3 – Preuve de terminaison d'un sous-programme récursif (*) | 131 |
| 9.4 – Coût de la récursivité : exemple du calcul du nombre de combinaisons (*) | 131 |
| 9.5 – Elimination de la récursivité : exemple de la fonction d'Ackermann (**) | 133 |
| 9.6 – Récursivité croisée | 136 |
| 9.7 – Passage de sous-programmes en paramètres de sous-programmes récursifs (**) | 138 |
| 10. Compilation séparée en PASCAL sous UNIX (**) | 140 |
| 10.1 – Décomposition d'un programme en modules (**) | 140 |
| 10.2 – Compilations indépendantes et séparées (**) | 140 |
| 10.3 – Compilation séparée en PASCAL sous UNIX (**) | 140 |
| 10.4 – Automatisation des compilations après modification d'un module (**) | 143 |

LEÇON 4 150

| | |
|--|------------|
| 1. Introduction | 150 |
| 2. Enregistrements | 150 |
| 2.1 – Type enregistrement | 150 |
| 2.2 – Variables de type enregistrement | 152 |
| 2.3 – Champs de variables enregistrements | 152 |
| 2.4 – Enregistrements avec variantes (*) | 154 |
| 2.5 – Instruction "with" (**) | 161 |
| 3. Structures d'informations élémentaires | 162 |
| 3.1 – Piles | 163 |
| 3.2 – Files | 164 |
| 3.3 – Tableaux flexibles | 166 |
| 3.4 – Listes linéaires | 168 |
| 3.5 – Tables | 174 |
| 3.6 – Arbres binaires | 176 |
| 3.7 – Graphes (*) | 183 |

LEÇON 5 192

| | |
|---|------------|
| 1. Introduction | 192 |
| 2. Pointeurs sur des enregistrements alloués dynamiquement | 192 |
| 2.1 – Exemple introductif | 192 |
| 2.2 – Rappels sur les variables statiques | 196 |
| 2.3 – Variables dynamiques et pointeurs | 196 |

| | |
|--|------------|
| 2.4 - Déclaration du type des variables dynamiques et pointeurs | 197 |
| 2.5 - Création d'une variable dynamique repérée par un pointeur | 197 |
| 2.6 - Autres opérations sur les variables dynamiques | 197 |
| 2.7 - Opérations sur les pointeurs | 198 |
| 2.8 - Exemple : inversion destructive d'une liste linéaire | 198 |
| 2.9 - Exemple : inversion destructive d'une liste linéaire | 198 |
| 2.10 - Ramassage de miettes (**) | 201 |
| 3. Structures d'information | 203 |
| 3.1 - Arbres binaires | 203 |
| 3.2 - Représentation d'une table par arbre binaire ordonné aléatoire | 205 |
| 3.3 - Représentation d'une table par un arbre 2-3 ordonné | 213 |
| 3.4 - Comparaison des performances des représentations d'une table par une liste linéaire, par un arbre 2-3 et par hachage (*) | 231 |
| 3.5 - Représentation d'une table par un arbre $\lceil n/2 \rceil$ -n ordonné (**) | 239 |
| LEÇON 6 | 241 |
| 1. Introduction | 241 |
| 2. Notion générale de fichier en PASCAL (et ses extensions) | 242 |
| 2.1 - Fichier à accès direct | 242 |
| 2.2 - Fichier séquentiel | 243 |
| 3. Fichiers à accès direct | 244 |
| 3.1 - Disques magnétiques | 244 |
| 3.2 - Organisation des fichiers sur un disque (**) | 245 |
| 3.3 - Opérations sur les fichiers à accès direct en PASCAL (non standard) | 247 |
| 3.4 - Fichiers à accès direct sous Unix (**) | 253 |
| 4. Fichiers séquentiels (*) | 259 |
| 4.1 - Bandes magnétiques (*) | 259 |
| 4.2 - Analogie entre les bandes magnétiques et les fichiers séquentiels en PASCAL (*) | 260 |
| 4.3 - Fichiers séquentiels en PASCAL standard (*) | 263 |
| 4.4 - Fichiers de texte séquentiels (**) | 265 |
| 5. Différences entre le PASCAL du Macintosh et le PASCAL standard concernant les fichiers (***) | 268 |
| 5.1 - Liaison entre les variables de type fichier internes au programme et les fichiers externes (***) | 268 |
| 5.2 - Entrées-sorties par nécessité (***) | 269 |
| 6. Diagrammes de syntaxe | 272 |
| RÉFÉRENCES ET BIBLIOGRAPHIE | 282 |
| INDEX | 284 |

(*) Les paragraphes marqués de (*) peuvent être lus en deuxième lecture.

(**) Les paragraphes marqués de (**) ou les textes en petits caractères peuvent être réservés pour une lecture plus approfondie.

(***) Les paragraphes marqués de (***) sont des compléments d'information qui ne doivent être consultés qu'en cas de problème à caractère technique concernant les entrées-sorties.

LEÇON 1

| | |
|--|-----------|
| 1. Introduction | 9 |
| 2. Description syntaxique et sémantique d'un langage de programmation | 10 |
| 2.1 – Spécification de la syntaxe de PASCAL | 10 |
| 2.1.1 – diagrammes de syntaxe | 10 |
| 2.1.2 – conditions de contexte | 11 |
| 2.2 – Spécification de la sémantique de PASCAL | 12 |
| 3. Programmation en PASCAL | 13 |
| 3.1 – Structure générale des programmes PASCAL | 13 |
| 3.2 – Symboles et séparateurs | 14 |
| 3.3 – Identificateurs | 14 |
| 3.4 – Types et constantes de base | 15 |
| 3.4.1 – Type booléen: <i>boolean</i> | 15 |
| 3.4.1.1 – Valeurs et constantes booléennes | 15 |
| 3.4.1.2 – Opérateurs booléens | 15 |
| 3.4.1.3 – Représentation des booléens en mémoire (*) | 15 |
| 3.4.2 – Type entier: <i>integer</i> | 15 |
| 3.4.2.1 – Valeurs entières | 15 |
| 3.4.2.2 – Constantes entières | 16 |
| 3.4.2.3 – Opérateurs entiers | 16 |
| 3.4.2.4 – Représentation des entiers en mémoire (*) | 16 |
| 3.4.3 – Type réel: <i>real</i> | 17 |
| 3.4.3.1 – Valeurs réelles | 17 |
| 3.4.3.2 – Constantes réelles | 17 |
| 3.4.3.3 – Opérateurs réels | 18 |
| 3.4.3.4 – Représentation des réels en mémoire (**) | 19 |
| 3.4.4 – Type caractère: <i>char</i> | 19 |
| 3.4.4.1 – Valeurs de type caractère | 19 |
| 3.4.4.2 – Constantes de type caractère | 19 |
| 3.4.4.3 – Opérateurs sur les caractères | 20 |
| 3.4.4.4 – Représentation des caractères en mémoire (*) | 21 |
| 3.4.5 – Chaînes de caractères | 22 |
| 3.5 – Constantes | 22 |
| 3.6 – Déclarations | 23 |
| 3.6.1 – Déclarations de constantes | 24 |
| 3.6.2 – Déclarations de types | 25 |
| 3.6.2.1 – Introduction informelle à la notion de type | 25 |
| 3.6.2.2 – Déclaration de types | 29 |
| 3.6.2.3 – Types pré-déclarés | 29 |
| 3.6.2.4 – Type intervalle d'un type ordinal | 29 |
| 3.6.2.5 – Type énuméré | 30 |
| 3.6.2.6 – Types ordinaux et types simples | 31 |
| 3.6.3 – Déclaration de variables | 31 |

| | |
|--|----|
| 3.7 – Expressions | 32 |
| 3.7.1 – Syntaxe des expressions | 33 |
| 3.7.2 – Type des expressions bien formées | 34 |
| 3.7.3 – Règles d'évaluation d'une expression | 36 |
| 3.8 – Instructions | 37 |
| 3.8.1 – Instruction d'affectation | 39 |
| 3.8.2 – Composition séquentielle d'instructions | 39 |
| 3.8.3 – Instruction vide | 40 |
| 3.8.4 – Utilisation du point virgule en PASCAL | 40 |
| 3.8.5 – Composition conditionnelle d'instructions | 40 |
| 3.8.5.1 – Instruction de test booléen: if | 40 |
| 3.8.5.2 – Instruction de sélection de cas: case (*) | 40 |
| 3.8.6 – Composition itérative d'instructions | 42 |
| 3.8.6.1 – Instruction d'itération "tantque": while | 43 |
| 3.8.6.2 – Instruction itérative "répéter": repeat | 43 |
| 3.8.6.3 – Instruction d'itération "pour": for | 44 |
| 3.8.7 – Instructions d'entrées/sorties (*) | 45 |
| 3.8.7.1 – Instruction d'écriture de résultats (*) | 46 |
| 3.8.7.2 – Instruction de lecture de données (*) | 46 |
| 3.8.8 – Etiquettes et instructions de branchement (**) | 48 |
| | 49 |

(*) Les paragraphes marqués de (*) peuvent être lus en deuxième lecture.

(**) Les paragraphes marqués de (**) ou les textes en petits caractères peuvent être réservés pour une lecture plus approfondie.

(***) Les paragraphes marqués de (***) sont des compléments d'information qui ne doivent être consultés qu'en cas de problème à caractère technique concernant les entrées-sorties.