

# ALGORITHMIQUE ET PROGRAMMATION EN PASCAL

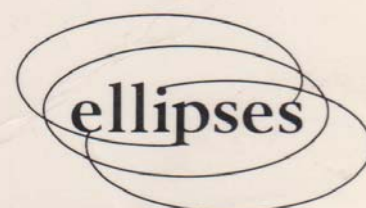
---

Cours

Patrick Cousot



ÉCOLE POLYTECHNIQUE

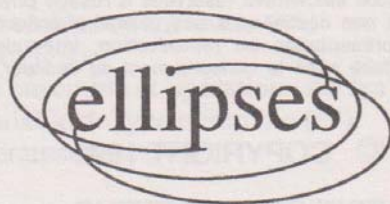


2-005-182-1

ALGORITHMIQUE  
ET  
PROGRAMMATION  
EN  
PASCAL

*COURS*

**Patrick COUSOT**  
Professeur à l'École Polytechnique  
et à l'École Normale Supérieure





# TABLE DES MATIÈRES

<b>INTRODUCTION</b>	<b>2</b>
<b>LEÇON 1</b>	<b>9</b>
<b>1. Introduction</b>	<b>9</b>
<b>2. Description syntaxique et sémantique d'un langage de programmation</b>	<b>10</b>
2.1 – Spécification de la syntaxe de PASCAL	10
2.2 – Spécification de la sémantique de PASCAL	12
<b>3. Programmation en PASCAL</b>	<b>13</b>
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
<b>LEÇON 2</b>	<b>53</b>
<b>1. Introduction</b>	<b>53</b>
PROGRAMMATION EN PASCAL	
<b>2. Tableaux</b>	<b>53</b>
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
<b>3. Ensembles (**)</b>	<b>59</b>
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
<b>4. Types compacts (**)</b>	<b>65</b>
<b>5. Type des chaînes de caractères (*)</b>	<b>66</b>
5.1 – Chaînes de caractères de longueur fixe en PASCAL standard (*)	66
5.2 – Chaînes de caractères en PASCAL VAX 8600 (*)	67

<b>6. Effets de bord (*)</b>	<b>120</b>
6.1 – Effets de bord des procédures (*)	120
6.2 – Effets de bord des fonctions (*)	120
<b>7. Alias</b>	<b>121</b>
<b>8. Paramètres procéduraux et fonctionnels (**)</b>	<b>125</b>
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
<b>9. Récursivité</b>	<b>127</b>
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
<b>10. Compilation séparée en PASCAL sous UNIX (**)</b>	<b>140</b>
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

---

<b>1. Introduction</b>	<b>150</b>
<b>2. Enregistrements</b>	<b>150</b>
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
<b>3. Structures d'informations élémentaires</b>	<b>162</b>
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

---

<b>1. Introduction</b>	<b>192</b>
<b>2. Pointeurs sur des enregistrements alloués dynamiquement</b>	<b>192</b>
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 – Création d'une variable dynamique avec variantes (**)	197
2.7 – Autres opérations sur les variables dynamiques	198
2.8 – Opérations sur les pointeurs	198
2.9 – Exemple : inversion destructive d'une liste linéaire	198
2.10 – Ramassage de miettes (**)	201
<b>3. Structures d'information</b>	<b>203</b>
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
<b>LEÇON 6</b>	<b>241</b>
<hr/>	
<b>1. Introduction</b>	<b>241</b>
<b>2. Notion générale de fichier en PASCAL (et ses extensions)</b>	<b>242</b>
2.1 – Fichier à accès direct	242
2.2 – Fichier séquentiel	243
<b>3. Fichiers à accès direct</b>	<b>244</b>
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
<b>4. Fichiers séquentiels (*)</b>	<b>259</b>
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
<b>5. Différences entre le PASCAL du Macintosh et le PASCAL standard concernant les fichiers (***)</b>	<b>268</b>
5.1 – Liaison entre les variables de type fichier internes au programme et les fichiers externes (***)	268
5.2 – Entrées-sorties par nécessité (***)	269
<b>6. Diagrammes de syntaxe</b>	<b>272</b>
<b>RÉFÉRENCES ET BIBLIOGRAPHIE</b>	<b>282</b>
<b>INDEX</b>	<b>284</b>

(\*) 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

<b>1. Introduction</b>	<b>9</b>
<b>2. Description syntaxique et sémantique d'un langage de programmation</b>	<b>10</b>
2.1 – Spécification de la syntaxe de PASCAL	10
2.1.1 – <i>diagrammes de syntaxe</i>	10
2.1.2 – <i>conditions de contexte</i>	11
2.2 – Spécification de la sémantique de PASCAL	12
<b>3. Programmation en PASCAL</b>	<b>13</b>
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 – <i>Type booléen: 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 – <i>Type entier: 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 – <i>Type réel: 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 – <i>Type caractère: 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 – <i>Chaînes de caractères</i>	22
3.5 – Constantes	22
3.6 – Déclarations	23
3.6.1 – <i>Déclarations de constantes</i>	24
3.6.2 – <i>Déclarations de types</i>	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 – <i>Déclaration de variables</i>	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 (*)	42
3.8.6 – Composition itérative d'instructions	43
3.8.6.1 – Instruction d'itération "tantque": while	43
3.8.6.2 – Instruction itérative "répéter": repeat	44
3.8.6.3 – Instruction d'itération "pour": for	45
3.8.7 – Instructions d'entrées/sorties (*)	46
3.8.7.1 – Instruction d'écriture de résultats (*)	46
3.8.7.2 – Instruction de lecture de données (*)	48
3.8.8 – Etiquettes et instructions de branchement (**)	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.