

ADA 95

ORIENTATION OBJET, STRUCTURES DE DONNÉES
ET ALGORITHMES

Philippe GABRINI



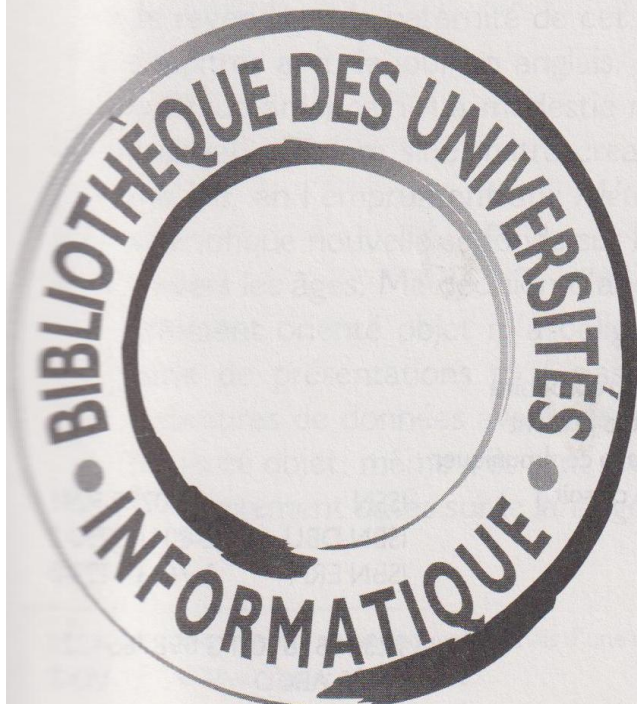
De Boeck
Université

2-005-300-1

ADA 95

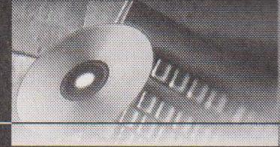
ORIENTATION OBJET, STRUCTURES DE DONNÉES
ET ALGORITHMES

Philippe GABRINI



De Boeck
Université

TABLE DES MATIÈRES



Avant-propos	V
--------------------	---

CHAPITRE 1

Génie logiciel et orientation objet	1
1.1 Développement de logiciels	2
1.2 Génie logiciel	6
Analyse	6
Conception	7
Réalisation	7
Maintenance	8
1.3 Structures de données et algorithmes	8
1.4 Programmation orientée objet	10
Un peu d'histoire	11
1.5 Définitions et caractéristiques de la programmation orientée objet	14
1.6 Abstraction	15
Types de données abstraits	15
1.7 Encapsulation	19
Rappel sur les pointeurs	20
Variables dynamiques	20
Pointeurs	26
Paramètres ACCESS	28
Conversions de pointeurs	28
Réalisation dynamique du type pile	30
1.8 Modularité et généricité	34
1.9 Un exemple de programmation orientée objet	47

CHAPITRE 2

Programmation orientée objet: hiérarchie et polymorphisme	59
2.1 Hiérarchie	60
2.2 Héritage par bibliothèques hiérarchiques	62
2.3 Héritage par extension de types étiquetés	66
2.4 Polymorphisme	74
Polymorphisme statique	75
Polymorphisme dynamique	75
Opérations sur les classes	77
2.5 Types abstraits	82

2.6	Types contrôlés	84
2.7	Analyse et conception orientées objet	86
2.8	Étude de cas: listes généralisées	88
	Conception	88
	Définir le problème	88
	Concevoir une solution	90
	Raffiner la solution	94
	Développer une stratégie de vérification	100
	Réalisation	105
	Coder et tester le programme	105
	Compléter la documentation	114

CHAPITRE 3

Arborescences	121
3.1 Définitions	122
3.2 Abstraction	127
Arbres binaires	128
3.3 Applications	137
Recherche	138
Arbres d'expression	141
Codes de Huffman	144
3.4 Réalisation	151
Réalisation dynamique	151
Complexité des algorithmes	168
La notation grand-O	168
Classes de complexité	169
Exemple	170
Réalisation statique	175
3.5 Étude de cas: un programme jouant au Tic-Tac-Toe	178
Conception	178
Définir le problème	178
Concevoir une solution	179
Raffiner la solution	181
Développer une stratégie de vérification	187
Réalisation	188
Coder et tester le programme	188
Compléter la documentation	202

CHAPITRE 4

Arbres binaires de recherche	209
4.1 Abstraction: l'arbre binaire de recherche	210
4.2 Application: création de l'index d'un texte	213

4.3	Réalisation	225
4.4	Étude de cas: arbres binaires de recherche «filés»	234
	Conception	234
	Définir le problème	235
	Concevoir une solution	237
	Raffiner la solution	238
	Développer une stratégie de vérification	247
	Réalisation	250
	Coder et tester le programme	250
	Compléter la documentation	257

CHAPITRE 5

Arborescences spécialisées

5.1	Arbres équilibrés	262
5.2	Arbres B	279
	Réalisation	284
	Recherche	285
	Insertion	287
	Suppression	291
	Complexité	299
5.3	Arbres de recherche optimaux	299
5.4	Monceaux et files avec priorités	301
5.5	Étude de cas: rééquilibrage global	306
	Conception	306
	Définir le problème	307
	Concevoir une solution	307
	Raffiner la solution	310
	Développer une stratégie de vérification	313
	Réalisation	314
	Coder et tester le programme	314
	Compléter la documentation	319

CHAPITRE 6

Tris internes et externes

6.1	Tris internes	324
	Tri Shell	325
	Tri du monceau («Heapsort»)	328
	Tri radicaire	332
6.2	Tris externes	335
	Tri fusion	336
	Tri polyphasé	343

Arbres binaires de recherche	535
Tables	536
9.4 Étude de cas: ensembles Fusionner-Trouver	538
Conception	538
Définir le problème	538
Concevoir une solution	538
Une réalisation par tableaux	538
Une réalisation par arborescences	540
Raffiner la solution	542
Développer une stratégie de vérification	543
Réalisation	544
Coder et tester le programme	544
Compléter la documentation	546
CHAPITRE 10	
Chaînes de caractères	551
10.1 Abstraction: la chaîne de caractères	553
10.2 Applications	558
Formatage de textes	558
Chiffrement	565
10.3 Réalisation	572
Réalisation statique	572
Réalisation dynamique	579
10.4 Reconnaissance de patrons	586
Recherche simple	586
Algorithme de Knuth-Morris-Pratt	588
Algorithme de Boyer-Moore	591
Algorithme de Rabin-Karp	594
10.5 Étude de cas: performance des algorithmes de recherche de patrons	597
Conception	597
Définir le problème	598
Concevoir une solution	599
Raffiner la solution	599
Développer une stratégie de vérification	601
Réalisation	602
Coder et tester le programme	602
Compléter la documentation	611
ANNEXE A	
Ensemble des caractères du code ASCII étendu	615
ANNEXE B	
Solutions des exercices choisis	621
Bibliographie	663
Index	665