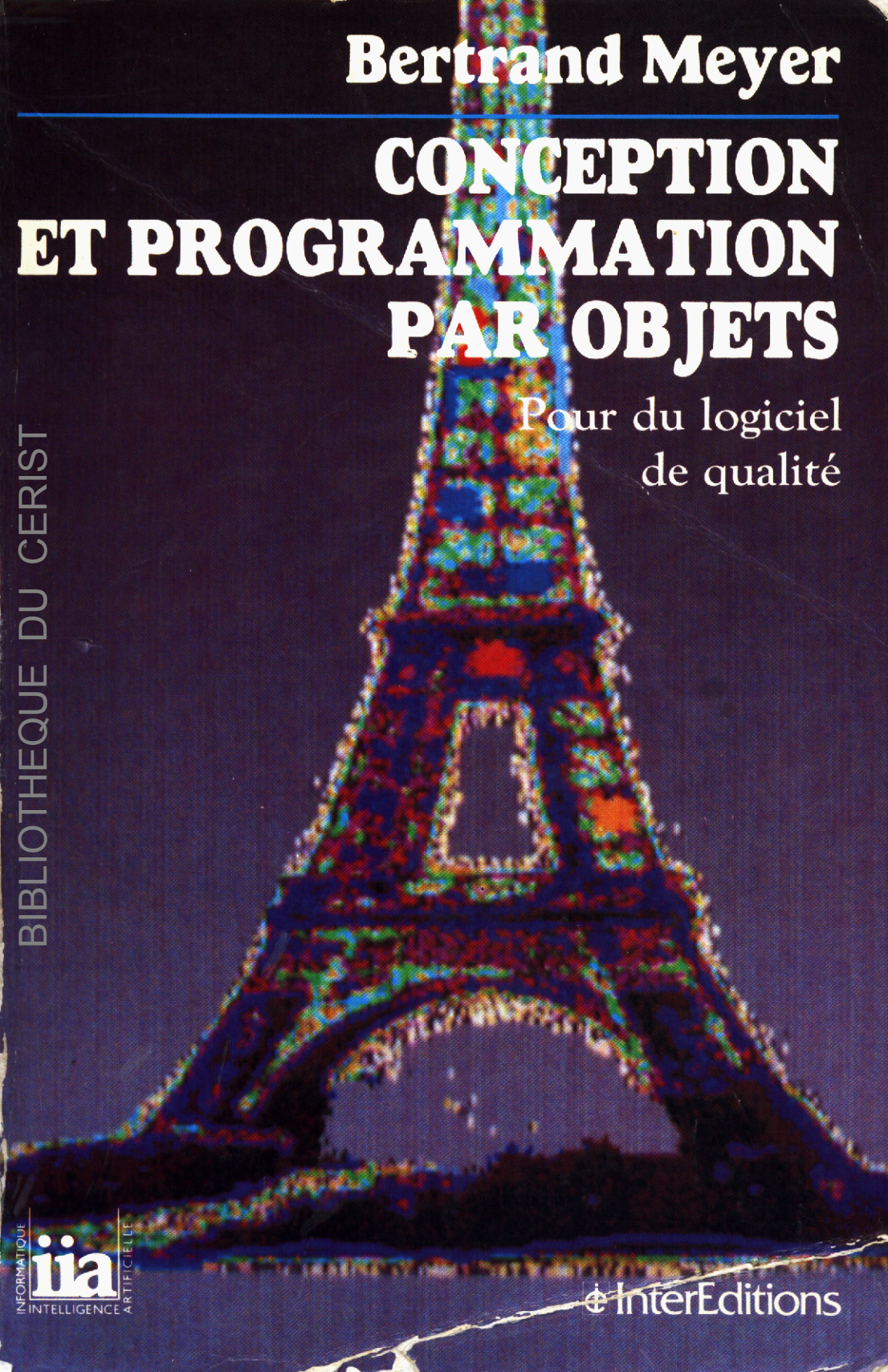




Bertrand Meyer

CONCEPTION ET PROGRAMMATION PAR OBJETS

Pour du logiciel
de qualité

BIBLIOTHEQUE DU CERIST

C2345

Bertrand Meyer

*Interactive Software Engineering, Santa Barbara
Société des Outils du Logiciel, Paris*

**CONCEPTION
ET PROGRAMMATION
PAR OBJETS**

Pour du logiciel de qualité

Traduit de l'américain et mis à jour par Robert Mahl

BIBLIOTHEQUE DU CERIST

 **InterEditions**

L'édition originale de ce livre a été publiée en Grande-Bretagne par Prentice-Hall International (UK) Ltd, Hemel Hempstead, sous le titre *Object-oriented Software Construction*.

© 1988 by Bertrand Meyer

La présente adaptation prend en compte Eiffel version 2.2.

*Publié avec le concours
du Centre National des Lettres*

Deuxième tirage, 1991

6429

© 1990, InterÉditions, Paris.

Tous droits réservés. Aucun extrait de ce livre ne peut être reproduit, sous quelque forme ou par quelque procédé que ce soit (machine électronique, mécanique, à photocopier, à enregistrer ou tout autre) sans autorisation écrite préalable de l'Éditeur.

ISBN 2-7296-0272-0

*Pour Annie, Caroline, Isabelle-Muriel,
Laurent, Raphaël et Sarah*

BIBLIOTHEQUE DU CERIST

Sommaire

Préface	13
Notation syntaxique	17

Première partie Problèmes et principes	19
---	-----------

Chapitre 1 Les facettes de la qualité du logiciel	21
1.1 Facteurs internes et externes	21
1.2 Facteurs externes de qualité	22
1.3 La maintenance du logiciel	26
1.4 Les qualités essentielles	28
1.5 Concepts clés introduits dans ce chapitre	29
1.6 Notes bibliographiques	29

Chapitre 2 La modularité	30
2.1 Cinq critères	31
2.2 Cinq principes	38
2.3 Le principe d'ouverture-fermeture	44
2.4 Concepts clés introduits dans ce chapitre	46
2.5 Notes bibliographiques	47
Exercices	47

Chapitre 3 Objectif : réutilisabilité	49
3.1 La répétition en programmation	49
3.2 Les approches simples de la réutilisabilité	53
3.3 Cinq conditions pour la modularité	54
3.4 Routines	58
3.5 Les paquetages	59
3.6 Surcharge et généricité	61
3.7 Concepts clés introduits dans ce chapitre	64
3.8 Notes bibliographiques	65

Chapitre 4 La route vers les objets	66
4.1 Actions et données	66
4.2 Fonctions, données et continuité	67
4.3 L'approche fonctionnelle descendante	68
4.4 Pourquoi utiliser les données ?	75
4.5 La conception par objets	76
4.6 Trouver les objets	77
4.7 La description des objets : types abstraits de données	78

6 Conception et programmation par objets

4.8	Une définition précise	86
4.9	Sept pas vers le bonheur par les objets	87
4.10	Concepts clés introduits dans ce chapitre	90
4.11	Notes bibliographiques	91
	Exercices	92

Deuxième partie Techniques de conception et de programmation par objets 93

Chapitre 5	Éléments de base de la programmation en Eiffel	95
5.1	Les objets	95
5.2	Une première vision des classes	99
5.3	L'utilisation des classes	101
5.4	Routines	107
5.5	Sémantique des références et des valeurs	114
5.6	Des classes aux systèmes	119
5.7	Classes ou objets ?	123
5.8	Discussion	123
5.9	Concepts clés introduits dans ce chapitre	130
5.10	Résumé syntaxique	131
Chapitre 6	La généricité	134
6.1	Paramétrer les classes	134
6.2	Tableaux	137
6.3	Discussion	139
6.4	Concepts clés introduits dans ce chapitre	139
6.5	Résumé syntaxique	140
6.6	Notes bibliographiques	140
Chapitre 7	Approches systématiques de la construction de logiciel	141
7.1	La notion d'assertion	142
7.2	Préconditions et postconditions	143
7.3	Approche contractuelle et fiabilité du logiciel	146
7.4	Invariants de classes et exactitude des classes	155
7.5	Un peu de théorie	161
7.6	Invariants de représentation	163
7.7	Effets de bord dans les fonctions	165
7.8	Autres constructions mettant en œuvre des assertions	173
7.9	Utilisation des assertions	177

7.10	Traiter les erreurs : exceptions disciplinées	179
7.11	Discussion	192
7.12	Concepts clés introduits dans ce chapitre	198
7.13	Résumé syntaxique	199
7.14	Notes bibliographiques	200
	Exercices	200
Chapitre 8	Autres aspects d'Eiffel	202
8.1	Conventions de style	202
8.2	Conventions lexicales	205
8.3	Routines externes	207
8.4	Passation d'arguments	208
8.5	Instructions	210
8.6	Expressions	214
8.7	Chaînes	218
8.8	Entrées et sorties	219
8.9	Concepts clés introduits dans ce chapitre	219
8.10	Résumé syntaxique	220
Chapitre 9	Concevoir des interfaces entre classes	222
9.1	Listes et éléments de listes	223
9.2	Des objets considérés comme des machines	231
9.3	Traitement des cas anormaux	239
9.4	Exportations sélectives	244
9.5	Documenter une classe	245
9.6	Discussion	251
9.7	Concepts clés introduits dans ce chapitre	256
9.8	Résumé syntaxique	257
9.9	Notes bibliographiques	257
	Exercices	257
Chapitre 10	Introduction à la notion d'héritage	259
10.1	Polygones et rectangles	260
10.2	La signification de l'héritage	272
10.3	Classes différées	279
10.4	Héritage multiple	287
10.5	Discussion	298
10.6	Concepts clés introduits dans ce chapitre	300
10.7	Résumé syntaxique	301
10.8	Notes bibliographiques	301
	Exercices	302

8 Conception et programmation par objets

Chapitre 11 Compléments sur l'héritage	304
11.1 Héritage et assertions	304
11.2 Redéfinition ou renommage ?	309
11.3 Le système de types d'Eiffel	311
11.4 Déclaration par association	317
11.5 Héritage et masquage d'information	324
11.6 Héritage à répétition	327
11.7 Concepts clés introduits dans ce chapitre	333
11.8 Résumé syntaxique	333
11.9 Notes bibliographiques	334
Exercices	334
Chapitre 12 Conception par objets : études de cas	335
12.1 Un système de fenêtrage	335
12.2 Défaire et refaire	340
12.3 Systèmes transactionnels plein écran	346
Exercices	360
Chapitre 13 Constantes et objets partagés	362
13.1 Constantes et types simples	363
13.2 Utilisation des constantes	363
13.3 Constantes de types classes	365
13.4 Constantes de type chaîne	373
13.5 Discussion	374
13.6 Concepts clés introduits dans ce chapitre	381
13.7 Résumé syntaxique	381
13.8 Notes bibliographiques	382
Exercices	382
Chapitre 14 Techniques de conception par objets	383
14.1 Philosophie de la conception	383
14.2 Trouver les classes	387
14.3 Techniques d'interface	390
14.4 Techniques d'héritage	391
14.5 Préférez-vous acheter ou hériter ?	395
14.6 Notes bibliographiques	397
Exercices	397
Chapitre 15 Implémentation : l'environnement d'Eiffel	398
15.1 L'implémentation	398
15.2 Compilation et gestion de configuration	400

15.3	Générer des paquetages C	406
15.4	Problèmes de performances	408
15.5	Autres aspects de l'environnement	411
Chapitre 16	Gestion de la mémoire	420
16.1	Ce qu'il advient des objets	420
16.2	L'approche informelle	425
16.3	Récupérer la mémoire : les problèmes	426
16.4	Désallocation contrôlée par le programmeur	427
16.5	L'approche autogestionnaire	428
16.6	Gestion automatique de la mémoire	434
16.7	L'approche d'Eiffel	436
16.8	Concepts clés introduits dans ce chapitre	439
16.9	Notes bibliographiques	440
	Exercices	441
Troisième partie	Les techniques par objets dans d'autres environnements	443
Chapitre 17	La programmation par objets dans les langages classiques	445
17.1	Niveaux d'aide du langage	445
17.2	Programmer par objets en Pascal ?	446
17.3	Fortran	446
17.4	Programmation par objets et langage C	449
17.5	Notes bibliographiques	453
	Exercices	453
Chapitre 18	Ada et la programmation par objets	454
18.1	Paquetages	454
18.2	Une implémentation de la pile	455
18.3	Masquer la représentation : la vie privée	460
18.4	Exceptions	462
18.5	Tâches	466
18.6	Concepts clés introduits dans ce chapitre	467
18.7	Notes bibliographiques	468
	Exercices	468
Chapitre 19	Généricité ou héritage	470
19.1	Généricité	470
19.2	Héritage	478

10 Conception et programmation par objets

19.3	Simuler l'héritage avec la généricité	481
19.4	Simuler la généricité avec l'héritage	482
19.5	Généricité et héritage en Eiffel	491
19.6	Discussion	494
19.7	Concepts clés introduits dans ce chapitre	495
19.8	Notes bibliographiques	495
	Exercices	496
Chapitre 20	Autres langages à objets	498
20.1	Simula	498
20.2	Smalltalk	513
20.3	Extensions de C	517
20.4	Extensions de Lisp	520
20.5	Autres langages	521
20.6	Notes bibliographiques	521
	Exercices	522
Chapitre 21	Autres questions	523
21.1	Implémenter la réutilisabilité	523
21.2	Persistance	524
21.3	Simultanéité	525
21.4	Notes bibliographiques	526
Quatrième partie	Annexes	527
A	Extraits de la bibliothèque d'Eiffel	529
A.1	Tableaux	529
A.2	Listes générales	531
A.3	Listes dans des tableaux	537
A.4	Éléments chaînables	540
A.5	Listes chaînées	542
A.6	Listes bidirectionnelles	548
A.7	Arbres et nœuds	551
B	Eiffel : un survol rapide	555
B.1	Principes de conception	555
B.2	Classes	556
B.3	Assertions	559
B.4	Exceptions	561
B.5	Classes génériques	562
B.6	Héritage multiple	563

B.7	Polymorphisme	565
B.8	Classes différées	567
B.9	L'implémentation	569
B.10	L'environnement	570
C	La grammaire d'Eiffel	572
C.1	Conventions lexicales	572
C.2	Spécification syntaxique	573
C.3	Précédence d'opérateurs	577
D	Mots réservés et symboles spéciaux d'Eiffel	578
D.1	Mots réservés	578
D.2	Symboles spéciaux	578
E	Entrées, sorties et chaînes	579
E.1	Entrées et sorties standard	579
E.2	Fichiers	580
E.3	Chaînes	582
F	Diagrammes syntaxiques d'Eiffel	587
G	Glossaire anglais-français¹	593
G.1	L'univers des objets	593
G.2	Mots clés et termes réservés	597
G.3	Termes de la bibliothèque d'Eiffel	598
H	Eiffel version 2.2¹	602
H.1	Mots réservés, symboles spéciaux	602
H.2	Adjonctions syntaxiques et sémantiques	602
H.3	L'environnement d'Eiffel 2.2	605
H.4	La bibliothèque de classes d'Eiffel 2.2	606
	Bibliographie	607
	Index	617

¹ Ces deux annexes ont été élaborées par le traducteur dans un souci de mise à jour.