

Les design patterns en Java

Les 23 modèles
de conception
fondamentaux

Steven John Metsker
William C. Wake

Réseaux
et télécom

Programmation

Génie logiciel

Sécurité

Système
d'exploitation

PEARSON

Table des matières

Préface	1
Conventions de codage	1
Remerciements	2
Chapitre 1. Introduction	3
Qu'est-ce qu'un pattern ?	3
Qu'est-ce qu'un pattern de conception ?	4
Liste des patterns décrits dans l'ouvrage	5
Java	7
UML	7
Exercices	8
Organisation du livre	9
Oozinoz	10
Résumé	11

Partie I

Patterns d'interface

Chapitre 2. Introduction aux interfaces	15
• Interfaces et classes abstraites	16
Interfaces et obligations	17
Résumé	19
Au-delà des interfaces ordinaires	19
Chapitre 3. ADAPTER	21
Adaptation à une interface	21
Adaptateurs de classe et d'objet	25

Adaptation de données pour un widget JTable	29
Identification d'adaptateurs	33
Résumé	34
Chapitre 4. FACADE	35
Façades, utilitaires et démos	36
Refactorisation pour appliquer FACADE	37
Résumé	46
Chapitre 5. COMPOSITE	47
Un composite ordinaire	47
Comportement récursif dans les objets composites	48
Objets composites, arbres et cycles	50
Des composites avec des cycles	55
Conséquences des cycles	59
Résumé	60
Chapitre 6. BRIDGE	61
Une abstraction ordinaire	61
De l'abstraction au pattern BRIDGE	64
Des drivers en tant que BRIDGE	66
Drivers de base de données	67
Résumé	69

Partie II

Patterns de responsabilité

Chapitre 7. Introduction à la responsabilité	73
Responsabilité ordinaire	73
Contrôle de la responsabilité grâce à la visibilité	75
Résumé	77
Au-delà de la responsabilité ordinaire	77

Chapitre 8. SINGLETON	79
Le mécanisme de SINGLETON	79
Singletons et threads	81
Identification de singletons	82
Résumé	84
Chapitre 9. OBSERVER	85
Un exemple classique : OBSERVER dans les interfaces utilisateurs	85
Modèle-Vue-Contrôleur	90
Maintenance d'un objet Observable	96
Résumé	99
Chapitre 10. MEDIATOR	101
Un exemple classique : médiateur de GUI	101
Médiateur d'intégrité relationnelle	106
Résumé	112
Chapitre 11. PROXY	115
Un exemple classique : proxy d'image	115
Reconsidération des proxies d'image	120
Proxy distant	122
Proxy dynamique	128
Résumé	133
Chapitre 12. CHAIN OF RESPONSABILITY	135
Une chaîne de responsabilités ordinaire	135
Refactorisation pour appliquer CHAIN OF RESPONSABILITY	137
Ancrage d'une chaîne de responsabilités	140
CHAIN OF RESPONSABILITY sans COMPOSITE	142
Résumé	142
Chapitre 13. FLYWEIGHT	143
Immuabilité	143
Extraction de la partie immuable d'un flyweight	144
Partage des objets flyweight	146
Résumé	149

Partie III**Patterns de construction**

Chapitre 14. Introduction à la construction	153
Quelques défis de construction	153
Résumé	155
Au-delà de la construction ordinaire	155
Chapitre 15. BUILDER	157
Un objet constructeur ordinaire	157
Construction avec des contraintes	160
Un builder tolérant	163
Résumé	164
Chapitre 16. FACTORY METHOD	165
Un exemple classique : des itérateurs	165
Identification de FACTORY METHOD	166
Garder le contrôle sur le choix de la classe à instancier	167
Application de FACTORY METHOD dans une hiérarchie parallèle	169
Résumé	171
Chapitre 17. ABSTRACT FACTORY	173
Un exemple classique : le kit de GUI	173
Classe FACTORY abstraite et pattern FACTORY METHOD	178
Packages et classes factory abstraites	182
Résumé	182
Chapitre 18. PROTOTYPE	183
Des prototypes en tant qu'objets factory	183
Prototypage avec des clones	185
Résumé	187
Chapitre 19. MEMENTO	189
Un exemple classique : défaire une opération	189
Durée de vie des mementos	196

Persistance des mementos entre les sessions	197
Résumé	200

Partie IV

Patterns d'opération

Chapitre 20. Introduction aux opérations	203
Opérations et méthodes	203
Signatures	205
Exceptions	205
Algorithmes et polymorphisme	206
Résumé	208
Au-delà des opérations ordinaires	209
Chapitre 21. TEMPLATE METHOD	211
Un exemple classique : algorithme de tri	211
Complétion d'un algorithme	215
Hooks	218
Refactorisation pour appliquer TEMPLATE METHOD	219
Résumé	221
Chapitre 22. STATE	223
Modélisation d'états	223
Refactorisation pour appliquer STATE	227
Etats constants	231
Résumé	233
Chapitre 23. STRATEGY	235
Modélisation de stratégies	236
Refactorisation pour appliquer STRATEGY	238
Comparaison de STRATEGY et STATE	242
Comparaison de STRATEGY et TEMPLATE METHOD	243
Résumé	243

Chapitre 24. COMMAND	245
Un exemple classique : commandes de menus	245
Emploi de COMMAND pour fournir un service	248
Hooks	249
COMMAND en relation avec d'autres patterns	251
Résumé	252
Chapitre 25. INTERPRETER	253
Un exemple de INTERPRETER	254
Interpréteurs, langages et analyseurs syntaxiques	265
Résumé	266

Partie V

Patterns d'extension

Chapitre 26. Introduction aux extensions	269
Principes de la conception orientée objet	269
Le principe de substitution de Liskov	270
La loi de Demeter	271
Elimination des erreurs potentielles	273
Au-delà des extensions ordinaires	273
Résumé	274
Chapitre 27. DECORATOR	277
Un exemple classique : flux d'E/S et objets Writer	277
Enveloppeurs de fonctions	285
DECORATOR en relation avec d'autres patterns	292
Résumé	293
Chapitre 28. ITERATOR	295
Itération ordinaire	295
Itération avec sécurité inter-threads	297
Itération sur un objet composite	303
Ajout d'un niveau de profondeur à un énumérateur	310
Enumération des feuilles	311
Résumé	313

Chapitre 29. VISITOR	315
Application de VISITOR	315
Un VISITOR ordinaire	318
Cycles et VISITOR	323
Risques de VISITOR	328
Résumé	330

Partie VI

Annexes

Annexe A. Recommandations	333
Tirer le meilleur parti du livre	333
Connaître ses classiques	334
Appliquer les patterns	334
Continuer d'apprendre	336
Annexe B. Solutions	337
Introduction aux interfaces	337
Solution 2.1	337
Solution 2.2	338
Solution 2.3	338
ADAPTER	338
Solution 3.1	338
Solution 3.2	339
Solution 3.3	340
Solution 3.4	341
Solution 3.5	341
Solution 3.6	342
FACADE	342
Solution 4.1	342
Solution 4.2	343
Solution 4.3	343
Solution 4.4	344

COMPOSITE	345
Solution 5.1	345
Solution 5.2	346
Solution 5.3	346
Solution 5.4	347
Solution 5.5	347
Solution 5.6	348
BRIDGE	348
Solution 6.1	348
Solution 6.2	348
Solution 6.3	349
Solution 6.4	349
Solution 6.5	350
Introduction à la responsabilité	350
Solution 7.1	350
Solution 7.2	351
Solution 7.3	352
Solution 7.4	353
SINGLETON	353
Solution 8.1	353
Solution 8.2	353
Solution 8.3	353
Solution 8.4	354
OBSERVER	354
Solution 9.1	354
Solution 9.2	355
Solution 9.3	356
Solution 9.4	356
Solution 9.5	357
Solution 9.6	357
Solution 9.7	358
MEDIATOR	359
Solution 10.1	359
Solution 10.2	360
Solution 10.3	361
Solution 10.4	361
Solution 10.5	362

PROXY	362
Solution 11.1	362
Solution 11.2	363
Solution 11.3	363
Solution 11.4	363
Solution 11.5	364
CHAIN OF RESPONSABILITY	364
Solution 12.1	364
Solution 12.2	365
Solution 12.3	366
Solution 12.4	366
Solution 12.5	367
FLYWEIGHT	368
Solution 13.1	368
Solution 13.2	369
Solution 13.3	370
Solution 13.4	370
Introduction à la construction	371
Solution 14.1	371
Solution 14.2	372
Solution 14.3	372
BUILDER	373
Solution 15.1	373
Solution 15.2	373
Solution 15.3	374
Solution 15.4	374
FACTORY METHOD	375
Solution 16.1	375
Solution 16.2	376
Solution 16.3	376
Solution 16.4	376
Solution 16.5	377
Solution 16.6	378
Solution 16.7	378
ABSTRACT FACTORY	379
Solution 17.1	379
Solution 17.2	380

Solution 17.3	380
Solution 17.4	381
Solution 17.5	381
PROTOTYPE	382
Solution 18.1	382
Solution 18.2	383
Solution 18.3	383
Solution 18.4	384
MEMENTO	384
Solution 19.1	384
Solution 19.2	385
Solution 19.3	385
Solution 19.4	386
Solution 19.5	386
Introduction aux opérations	387
Solution 20.1	387
Solution 20.2	387
Solution 20.3	388
Solution 20.4	388
Solution 20.5	388
TEMPLATE METHOD	389
Solution 21.1	389
Solution 21.2	389
Solution 21.3	390
Solution 21.4	390
STATE	390
Solution 22.1	390
Solution 22.2	390
Solution 22.3	391
Solution 22.4	391
STRATEGY	392
Solution 23.1	392
Solution 23.2	392
Solution 23.3	392
Solution 23.4	393

COMMAND	393
Solution 24.1	393
Solution 24.2	393
Solution 24.3	395
Solution 24.4	395
Solution 24.5	396
Solution 24.6	396
INTERPRETER	396
Solution 25.1	396
Solution 25.2	397
Solution 25.3	397
Solution 25.4	397
Introduction aux extensions	398
Solution 26.1	398
Solution 26.2	398
Solution 26.3	398
Solution 26.4	399
DECORATOR	399
Solution 27.1	399
Solution 27.2	400
Solution 27.3	401
Solution 27.4	401
ITERATOR	401
Solution 28.1	401
Solution 28.2	402
Solution 28.3	402
Solution 28.4	402
VISITOR	403
Solution 29.1	403
Solution 29.2	403
Solution 29.3	403
Solution 29.4	404
Solution 29.5	404
Annexe C. Code source d'Oozinoz	405
Obtention et utilisation du code source	405
Construction du code d'Oozinoz	406

Test du code avec JUnit	406
Localiser les fichiers	406
Résumé	407
Annexe D. Introduction à UML	409
Classes	409
Relations entre classes	412
Interfaces	414
Objets	414
Etats	416
Glossaire	417
Bibliographie	425
Index	427