

Table des matières

Avant-propos	1
L'orientation objet en deux mots	2
Objectifs de l'ouvrage	5
Plan de l'ouvrage	6
À qui s'adresse ce livre ?	7
 CHAPITRE 1	
Principes de base : quel objet pour l'informatique ?	9
Le trio <entité, attribut, valeur>	10
Stockage des objets en mémoire	11
L'objet dans sa version passive	15
L'objet dans sa version active	17
Introduction à la notion de classe	19
Des objets en interaction	21
Des objets soumis à une hiérarchie	24
Polymorphisme	26
Héritage bien reçu	27
Exercices	27
 CHAPITRE 2	
Un objet sans classe... n'a pas de classe	29
Constitution d'une classe d'objets	30
La classe comme module fonctionnel	33
La classe comme garante de son bon usage	36
La classe comme module opérationnel	37

Un premier petit programme complet dans les cinq langages	39
La classe et la logistique de développement	50
Exercices	52
CHAPITRE 3	
Du faire savoir au savoir-faire... du procédural à l'OO	57
Objectif objet : les aventures de l'OO	58
Mise en pratique	60
Analyse	60
Conception	62
Impacts de l'orientation objet	62
CHAPITRE 4	
Ici Londres : les objets parlent aux objets	65
Envois de messages	66
Association de classes	67
Dépendance de classes	68
Réaction en chaîne de messages	70
Exercices	71
CHAPITRE 5	
Collaboration entre classes	73
Pour en finir avec la lutte des classes	74
La compilation Java : effet domino	76
En C#, en Python, PHP 5 et en C++	77
De l'association unidirectionnelle à l'association bidirectionnelle	79
Auto-association	82
Package et namespace	83
Exercices	86
CHAPITRE 6	
Méthodes ou messages ?	87
Passage d'arguments prédéfinis dans les messages	88
Passage d'argument objet dans les messages	95
Une méthode est-elle d'office un message ?	102
La mondialisation des messages	104
Exercices	105

CHAPITRE 7	
L'encapsulation des attributs	109
Accès aux attributs d'un objet	110
Encapsulation : pourquoi faire ?	115
Exercices	120
CHAPITRE 8	
Les classes et leur jardin secret	123
Encapsulation des méthodes	124
Les niveaux intermédiaires d'encapsulation	127
Afin d'éviter l'effet papillon	131
Exercices	134
CHAPITRE 9	
Vie et mort des objets	135
Question de mémoire	136
C++ : le programmeur est le seul maître à bord	145
En Java, C#, Python et PHP 5 : la chasse au gaspi	148
Exercices	154
CHAPITRE 10	
UML 2	159
Diagrammes UML 2	161
Représentation graphique standardisée	162
Du tableau noir à l'ordinateur	163
Programmer par cycles courts en superposant les diagrammes	164
Diagrammes de classe et diagrammes de séquence	165
Diagramme de classe	165
Les bienfaits d'UML	196
Diagramme de séquence	199
Exercices	205
CHAPITRE 11	
Héritage	211
Comment regrouper les classes dans des superclasses	212
Héritage des attributs	213

Héritage ou composition ?	219
Économiser en rajoutant des classes ?	220
Héritage des méthodes	220
La recherche des méthodes dans la hiérarchie	229
Encapsulation protected	230
Héritage et constructeurs	231
Héritage public en C++	237
Le multihéritage	238
Exercices	249
CHAPITRE 12	
Redéfinition des méthodes	253
La redéfinition des méthodes	254
Beaucoup de verbiage mais peu d'actes véritables	255
Un match de football polymorphique	256
Exercices	288
CHAPITRE 13	
Abstraite, cette classe est sans objet	299
De Canaletto à Turner	300
Des classes sans objet	300
Du principe de l'abstraction à l'abstraction syntaxique	301
Un petit supplément de polymorphisme	309
Exercices	313
CHAPITRE 14	
Clonage, comparaison et assignation d'objets	325
Introduction à la classe Object	326
Décortiquons la classe Object	329
Test d'égalité de deux objets	331
Le clonage d'objets	336
Égalité et clonage d'objets en Python	339
Égalité et clonage d'objets en PHP 5	341
Égalité, clonage et affectation d'objets en C++	343
En C#, un cocktail de Java et de C++	353
Exercices	359

CHAPITRE 15	
Interfaces	361
Interfaces : favoriser la décomposition et la stabilité	363
Java, C# et PHP 5 : interface via l'héritage	363
Les trois raisons d'être des interfaces	364
Les Interfaces dans UML 2	378
En C++ : fichiers .h et fichiers .cpp	379
Interfaces : du local à Internet	382
Exercices	382
CHAPITRE 16	
Distribution gratuite d'objets : pour services rendus sur le réseau	387
Objets distribués sur le réseau : pourquoi ?	388
RMI (Remote Method Invocation)	391
Corba (Common Object Request Broker Architecture)	397
Rajoutons un peu de flexibilité à tout cela	404
Les services Web sur .Net	410
Exercices	420
CHAPITRE 17	
Multithreading	423
Informatique séquentielle	425
Multithreading	427
Implémentation en Java	428
Implémentation en C#	430
Implémentation en Python	433
L'impact du multithreading sur les diagrammes de séquence UML	434
Du multithreading aux applications distribuées	435
Des threads équirépartis	435
Synchroniser les threads	437
Exercices	445
CHAPITRE 18	
Programmation événementielle	449
Des objets qui s'observent	450
En Java	451
En C# : les délégués	454

En Python : tout reste à faire	462
Un feu de signalisation plus réaliste	465
Exercices	467
CHAPITRE 19	
Persistance d'objets	469
Sauvegarder l'état entre deux exécutions	470
Simple sauvegarde sur fichier	471
Sauvegarder les objets sans les dénaturer : la sérialisation	478
Les bases de données relationnelles	483
Réservation de places de spectacles	495
Les bases de données relationnelles-objet	500
Les bases de données orientées objet	504
Linq	505
Exercices	509
CHAPITRE 20	
Et si on faisait un petit flipper ?	511
Généralités sur le flipper et les GUI	513
Retour au Flipper	522
CHAPITRE 21	
Les graphes	535
Le monde regorge de réseaux	536
Tout d'abord : juste un ensemble d'objets	538
Liste liée	539
La généricité en C++	546
La généricité en Java	549
Passons aux graphes	555
Exercices	560
CHAPITRE 22	
Petites chimie et biologie OO amusantes	565
Pourquoi de la chimie OO ?	566
Les diagrammes de classe du réacteur chimique	567
Quelques résultats du simulateur	581
La simulation immunologique en OO ?	583

CHAPITRE 23

Design patterns	589
Introduction aux design patterns	590
Les patterns « truc et ficelle »	592
Les patterns qui se jettent à l'OO	599
Index	613