

**Hugues Bersini**

*Avec la collaboration d'Ivan Wellesz*

**2<sup>e</sup> édition**

# L'orienté objet

**Cours et exercices en UML 2  
avec Python, Java, C# et C++**

**EYROLLES**

5-529-1

2-005-529-1

Hugues Bersini

# L'orienté objet

2<sup>e</sup> édition

Avec la contribution d'Ivan Wellesz

EYROLLES



# Table des matières

<b>Avant-propos</b> .....	1
L'orientation objet en deux mots .....	2
Objectifs de l'ouvrage .....	3
Plan de l'ouvrage .....	4
À qui s'adresse ce livre ? .....	5
<b>CHAPITRE 1</b>	
<b>Principes de base : quel objet pour l'informatique ?</b> .....	7
Le trio <entité, attribut, valeur> .....	8
Stockage des objets en mémoire .....	9
L'objet dans sa version passive .....	13
L'objet dans sa version active .....	15
Introduction à la notion de classe .....	17
Des objets en interaction .....	19
Des objets soumis à une hiérarchie .....	21
Polymorphisme .....	23
Héritage bien reçu .....	24
Exercices .....	24
<b>CHAPITRE 2</b>	
<b>Un objet sans classe... n'a pas de classe</b> .....	27
Constitution d'une classe d'objets .....	28
La classe comme module fonctionnel .....	31
La classe comme garante de son bon usage .....	33
La classe comme module opérationnel .....	34
Un premier petit programme complet dans les quatre langages .....	35
La classe et la logistique de développement .....	44
Exercices .....	45

CHAPITRE 3

<b>Du faire savoir au savoir-faire...du procédural à l'OO</b> .....	49
Objectif objet : les aventures de l'OO .....	50
Mise en pratique .....	52
Analyse .....	52
Conception .....	54
Impacts de l'orientation objet .....	55

CHAPITRE 4

<b>Ici Londres : les objets parlent aux objets</b> .....	59
Envois de messages .....	60
Association de classes .....	61
Dépendance de classes .....	62
Réaction en chaîne de messages .....	63
Exercices .....	64

CHAPITRE 5

<b>Collaboration entre classes</b> .....	65
Pour en finir avec la lutte des classes .....	66
La compilation Java : effet domino .....	68
En C#, en Python et en C++ .....	69
De l'association unidirectionnelle à l'association bidirectionnelle .....	70
Auto-association .....	72
Package et namespace .....	74
Exercices .....	76

CHAPITRE 6

<b>Méthodes ou messages ?</b> .....	77
Passage d'arguments prédéfinis dans les messages .....	78
Passage d'argument objet dans les messages .....	84
Une méthode est-elle d'office un message ? .....	88
La mondialisation des messages .....	90
Exercices .....	91

CHAPITRE 7

<b>L'encapsulation des attributs</b> .....	95
Accès aux attributs d'un objet .....	96
Encapsulation : pourquoi faire ? .....	101
Exercices .....	105

## CHAPITRE 8

<b>Les classes et leur jardin secret</b> .....	107
Encapsulation des méthodes .....	108
Les niveaux intermédiaires d'encapsulation .....	111
Afin d'éviter l'effet papillon .....	115
Exercices .....	117

## CHAPITRE 9

<b>Vie et mort des objets</b> .....	119
Question de mémoire .....	120
C++ : le programmeur est le seul maître à bord .....	129
En Java, C# et Python : la chasse au gaspi .....	132
Exercices .....	137

## CHAPITRE 10

<b>UML 2</b> .....	143
Diagrammes UML 2 .....	145
Représentation graphique standardisée .....	146
Du tableau noir à l'ordinateur .....	147
Programmer par cycles courts en superposant les diagrammes .....	148
Diagrammes de classe et diagrammes de séquence .....	148
Diagramme de classe .....	149
Les bienfaits d'UML .....	176
Diagramme de séquence .....	179
Exercices .....	184

## CHAPITRE 11

<b>Héritage</b> .....	189
Comment regrouper les classes dans des superclasses .....	190
Héritage des attributs .....	191
Héritage ou composition ? .....	196
Économiser en rajoutant des classes ? .....	197
Héritage des méthodes .....	197
La recherche des méthodes dans la hiérarchie .....	205
Encapsulation protected .....	205
Héritage et constructeurs .....	207
Héritage public en C++ .....	212
Le multihéritage .....	213
Exercices .....	224

## CHAPITRE 12

<b>Redéfinition des méthodes</b> .....	229
La redéfinition des méthodes .....	230
Beaucoup de verbiage mais peu d'actes véritables .....	231
Un match de football polymorphique .....	232
Exercices .....	259

## CHAPITRE 13

<b>Abstraite, cette classe est sans objet</b> .....	271
De Canaletto à Turner .....	272
Des classes sans objet .....	272
Du principe de l'abstraction à l'abstraction syntaxique .....	273
Un petit supplément de polymorphisme .....	279
Exercices .....	284

## CHAPITRE 14

<b>Clonage, comparaison et assignation d'objets</b> .....	297
Introduction à la classe Object .....	298
Décortiquons la classe Object .....	300
Test d'égalité de deux objets .....	302
Le clonage d'objets .....	306
Égalité et clonage d'objets en Python .....	310
Égalité, clonage et affectation d'objets en C++ .....	312
En C#, un cocktail de Java et de C++ .....	322
Exercices .....	327

## CHAPITRE 15

<b>Interfaces</b> .....	329
Interfaces : favoriser la décomposition et la stabilité .....	331
Java et C# : interface via l'héritage .....	331
Les trois raisons d'être des interfaces .....	332
Les Interfaces dans UML 2 .....	343
En C++ : fichiers .h et fichiers .cpp .....	344
Interfaces : du local à Internet .....	347
Exercices .....	347

## CHAPITRE 16

<b>Distribution gratuite d'objets : pour services rendus sur le réseau</b> .....	351
Objets distribués sur le réseau : pourquoi ? .....	352
RMI (Remote Method Invocation) .....	355
Corba (Common Object Request Broker Architecture) .....	361
Rajoutons un peu de flexibilité à tout cela .....	368
Les services Web sur .Net .....	374
Exercices .....	384

## CHAPITRE 17

<b>Multithreading</b> .....	387
Informatique séquentielle .....	389
Multithreading .....	391
Implémentation en Java .....	392
Implémentation en C# .....	394
Implémentation en Python .....	396
L'impact du multithreading sur les diagrammes de séquence UML .....	397
Du multithreading aux applications distribuées .....	398
Des threads équirépartis .....	399
Synchroniser les threads .....	400
Exercices .....	407

## CHAPITRE 18

<b>Programmation événementielle</b> .....	411
Des objets qui s'observent .....	412
En Java .....	413
En C# : les délégués .....	416
En Python : tout reste à faire .....	423
Exercices .....	425

## CHAPITRE 19

<b>Persistance d'objets</b> .....	427
Sauvegarder l'état entre deux exécutions .....	428
Simple sauvegarde sur fichier .....	429
Sauvegarder les objets sans les dénaturer : la sérialisation .....	435
Les bases de données relationnelles .....	439
Les bases de données relationnelles-objet .....	451
Les bases de données orientées objet .....	454
Exercices .....	456

CHAPITRE 20

<b>Et si on faisait un petit flipper ?</b> .....	459
Généralités sur le flipper et les GUI .....	461
Retour au Flipper .....	466

CHAPITRE 21

<b>Les graphes</b> .....	479
Le monde regorge de réseaux .....	480
Tout d'abord : juste un ensemble d'objets .....	482
Liste liée .....	483
La généricité en C++ .....	490
La généricité en Java .....	493
Passons aux graphes .....	495
Exercices .....	500

CHAPITRE 22

<b>Petite chimie OO amusante</b> .....	505
Pourquoi de la chimie OO ? .....	506
Les diagrammes de classe du réacteur chimique .....	507
Quelques résultats du simulateur .....	521

CHAPITRE 23

<b>Design patterns</b> .....	525
Introduction aux design patterns .....	526
Les patterns « truc et ficelle » .....	528
Les patterns qui se jettent à l'OO .....	535

<b>Index</b> .....	547
--------------------	-----