

## Avant-propos

### Chapitre 1 Présentation

- 1. Introduction . . . . . 13
- 2. Historique . . . . . 15
  - 2.1 Pourquoi Java ? . . . . . 15
  - 2.2 Objectifs de la conception de Java . . . . . 16
  - 2.3 Essor de Java . . . . . 17
- 3. Les caractéristiques du langage Java . . . . . 19
- 4. La plateforme Java SE . . . . . 20
  - 4.1 La machine virtuelle Java (JVM) . . . . . 21
  - 4.2 L'API Java . . . . . 22
    - 4.2.1 Les API de base . . . . . 23
    - 4.2.2 Les API d'accès aux données et d'intégration avec l'existant . . . . . 25
    - 4.2.3 Les API de gestion de l'interface des applications avec l'utilisateur . . . . . 25
  - 4.3 Les API d'accès réseau . . . . . 26
  - 4.4 Les outils d'aide au développement . . . . . 26
- 5. Versions et releases . . . . . 27
- 6. Installation du JDK . . . . . 28
  - 6.1 Installation du JDK sur Windows . . . . . 28
    - 6.1.1 Installation de la plateforme OpenJDK . . . . . 28
    - 6.1.2 Installation de la plateforme Oracle JDK . . . . . 30
    - 6.1.3 Configuration des variables d'environnement . . . . . 32
    - 6.1.4 Tester le bon fonctionnement . . . . . 33
  - 6.2 Installation du JDK sur Linux . . . . . 34
    - 6.2.1 Installation de la plateforme OpenJDK . . . . . 34
    - 6.2.2 Installation de la plateforme Oracle JDK . . . . . 35
    - 6.2.3 Configuration de la variable JAVA\_HOME . . . . . 36

6.2.4	Tester le bon fonctionnement . . . . .	37
6.3	Installation du JDK sur macOS . . . . .	37
6.3.1	Installation de la plateforme Open JDK . . . . .	37
6.3.2	Installation de la plateforme Oracle JDK . . . . .	39
6.3.3	Configuration du JAVA_HOME sur macOS . . . . .	42
7.	Environnements de développement intégré (IDE) . . . . .	43
7.1	Installation d'IntelliJ IDEA . . . . .	43
7.1.1	Installation d'IntelliJ IDEA CE sur Mac . . . . .	43
7.1.2	Installation d'IntelliJ IDEA CE sur Linux . . . . .	44
7.1.3	Installation d'IntelliJ IDEA CE sur Windows . . . . .	45
7.1.4	Création d'un projet Java avec IntelliJ IDEA . . . . .	48
7.2	Installation d'Eclipse . . . . .	50
7.2.1	Téléchargement de l'installateur . . . . .	50
7.2.2	Installation d'Eclipse . . . . .	50
7.2.3	Création d'un projet Java avec Eclipse . . . . .	53
8.	Les bases de données . . . . .	56
8.1	Installation de MySQL . . . . .	56
8.1.1	Installation de MySQL Server sur macOS . . . . .	56
8.1.2	Création d'un utilisateur et d'une base de données sur macOS . . . . .	60
8.1.3	Installation de MySQL Server sur Windows . . . . .	60
8.1.4	Création d'un utilisateur et d'une base de données sur Windows . . . . .	65
8.1.5	Installation de MySQL Server sur Linux . . . . .	66
8.1.6	Création d'un utilisateur et d'une base de données . . . . .	66
8.2	Installation de DBeaver . . . . .	67
8.2.1	Utilisation de DBeaver . . . . .	69
9.	La javadoc . . . . .	72
9.1	Consultation depuis un navigateur . . . . .	72
9.2	Consultation depuis un IDE . . . . .	77
9.2.1	Dans IntelliJ IDEA . . . . .	77
9.2.2	Dans Eclipse . . . . .	78

- 10. Le premier programme Java : Hello, World ! ..... 78
  - 10.1 Écriture du code ..... 78
  - 10.2 Compilation et exécution du code ..... 80
    - 10.2.1 Théorie ..... 80
    - 10.2.2 Mise en œuvre ..... 80
  - 10.3 Utilisation d'un IDE (IntelliJ ou Eclipse)..... 83
- 11. Conclusion ..... 84

**Chapitre 2**  
**Comprendre un programme**

- 1. Anatomie et point d'entrée d'un programme. .... 85
- 2. Les variables ..... 87
  - 2.1 Introduction ..... 87
  - 2.2 Les emplacements. .... 88
  - 2.3 Le nom. .... 89
  - 2.4 Les types ..... 89
    - 2.4.1 Présentation ..... 89
    - 2.4.2 Les types valeurs. .... 91
    - 2.4.3 Les types références ..... 91
  - 2.5 La déclaration, l'initialisation, l'affectation. .... 93
    - 2.5.1 La déclaration ..... 93
    - 2.5.2 L'initialisation ..... 94
    - 2.5.3 L'affectation ..... 96
  - 2.6 Les types valeurs ..... 97
    - 2.6.1 Les types numériques entiers. .... 97
    - 2.6.2 Les types numériques décimaux ..... 99
    - 2.6.3 L'utilisation de valeurs littérales ..... 100
    - 2.6.4 Le type caractère. .... 102
    - 2.6.5 Le type boolean. .... 103
  - 2.7 Les types références ..... 104
    - 2.7.1 Les tableaux. .... 105
    - 2.7.2 Les chaînes de caractères ..... 116

2.7.3	La notion de Text Blocks . . . . .	126
2.7.4	Les dates et les heures. . . . .	127
2.8	Les valeurs par défaut. . . . .	131
2.9	La portée des variables . . . . .	132
2.10	La durée de vie des variables . . . . .	133
2.11	La conversion de type. . . . .	134
2.11.1	Présentation . . . . .	134
2.11.2	La conversion entre numériques . . . . .	134
2.11.3	La conversion vers une chaîne de caractères. . . . .	135
2.11.4	La conversion depuis une chaîne de caractères. . . . .	137
2.12	L'inférence de type . . . . .	139
3.	Les constantes. . . . .	140
4.	Les énumérations . . . . .	141
5.	Les conventions de nommage . . . . .	144
6.	Les arguments d'un programme . . . . .	146
6.1	Fonctionnement . . . . .	146
6.2	Utilisation dans un IDE . . . . .	147
7.	Les opérateurs . . . . .	150
7.1	Les opérateurs unaires . . . . .	150
7.2	L'opérateur d'affectation . . . . .	151
7.3	Les opérateurs arithmétiques. . . . .	152
7.4	Les opérateurs bit à bit . . . . .	153
7.4.1	La représentation binaire des entiers. . . . .	153
7.4.2	Les opérations logiques . . . . .	156
7.5	Les opérateurs de comparaison . . . . .	159
7.6	L'opérateur de concaténation. . . . .	160
7.7	Les opérateurs logiques. . . . .	161
7.8	Ordre d'évaluation des opérateurs. . . . .	163
8.	Les structures de contrôle . . . . .	164
8.1	Présentation . . . . .	164

8.2	Structures de décision . . . . .	164
8.2.1	Structure if . . . . .	164
8.2.2	Structure ternaire . . . . .	165
8.2.3	Structure switch historique . . . . .	166
8.2.4	Structure switch nouvelle génération . . . . .	168
8.3	Les structures de boucle . . . . .	170
8.3.1	Structure while . . . . .	170
8.3.2	Structure do ... while . . . . .	171
8.3.3	Structure for . . . . .	172
8.4	Interruption d'une structure de boucle . . . . .	175
8.4.1	break . . . . .	175
8.4.2	continue . . . . .	176
8.4.3	return . . . . .	177
9.	Exercices . . . . .	178
9.1	Exercice 1 . . . . .	178
9.2	Exercice 2 . . . . .	178
9.3	Exercice 3 . . . . .	179
9.4	Exercice 4 . . . . .	180
9.5	Exercice 5 . . . . .	181
10.	Corrections . . . . .	181
10.1	Exercice 1 . . . . .	181
10.2	Exercice 2 . . . . .	182
10.3	Exercice 3 . . . . .	184
10.4	Exercice 4 . . . . .	185
10.5	Exercice 5 . . . . .	186
11.	Conclusion . . . . .	187

**Chapitre 3**  
**Programmation objet**

1.	Introduction à la POO . . . . .	189
2.	Mise en œuvre de la POO avec Java . . . . .	192
2.1	Contexte . . . . .	192
2.2	Création d'une classe . . . . .	193
2.2.1	Déclaration de la classe . . . . .	193
2.2.2	Création des champs . . . . .	195
2.2.3	Création des méthodes. . . . .	196
2.2.4	Création des surcharges de méthode. . . . .	199
2.2.5	Passage de paramètres . . . . .	202
2.2.6	Création des accesseurs . . . . .	203
2.2.7	Création des constructeurs . . . . .	205
2.2.8	Création de champs et méthodes statiques . . . . .	206
2.2.9	Utilisation des annotations . . . . .	209
2.3	Utilisation d'une classe. . . . .	212
2.3.1	Création d'une instance . . . . .	212
2.3.2	Initialisation d'une instance. . . . .	213
2.3.3	Destruction d'une instance . . . . .	215
2.4	Héritage. . . . .	222
2.4.1	Création d'une classe fille. . . . .	222
2.4.2	this et super . . . . .	224
2.4.3	Classes abstraites . . . . .	228
2.4.4	Classes finales . . . . .	229
2.4.5	Classes scellées . . . . .	230
2.4.6	Conversion de type. . . . .	231
2.4.7	La classe Object . . . . .	238
2.5	Interfaces. . . . .	245
2.5.1	Création d'une interface. . . . .	245
2.5.2	Utilisation d'une interface . . . . .	247
2.5.3	Méthode par défaut . . . . .	251
2.5.4	Méthode statique . . . . .	254
2.5.5	Interfaces scellées . . . . .	254

2.6	Classes imbriquées . . . . .	254
2.6.1	Classes imbriquées statiques . . . . .	254
2.6.2	Classes internes (d'instance) . . . . .	256
2.6.3	Classes anonymes . . . . .	257
2.7	Les records . . . . .	263
3.	Les commentaires . . . . .	264
3.1	Présentation . . . . .	264
3.2	Commentaires sur une ligne . . . . .	265
3.3	Commentaires sur plusieurs lignes . . . . .	265
3.4	Commentaires javadoc . . . . .	266
3.5	Bonnes pratiques pour les commentaires . . . . .	268
4.	Les packages . . . . .	268
4.1	Présentation . . . . .	268
4.2	Création d'un package . . . . .	269
4.3	Utilisation et importation d'un package . . . . .	271
4.4	Importation des méthodes statiques . . . . .	273
5.	Les modules . . . . .	274
5.1	Mise en place . . . . .	274
5.2	Présentation . . . . .	275
5.3	Le JDK est modulaire . . . . .	275
5.4	Utilisation des modules . . . . .	276
5.5	Création d'un nouveau module . . . . .	277
6.	La gestion des erreurs . . . . .	279
6.1	Les différents types d'erreurs . . . . .	279
6.1.1	Les erreurs de syntaxe . . . . .	279
6.1.2	Les erreurs d'exécution . . . . .	280
6.1.3	Les erreurs de logique . . . . .	280
6.2	La représentation objet des erreurs . . . . .	281
6.3	Le traitement des exceptions . . . . .	283
6.4	Les exceptions associées à des ressources . . . . .	287
6.5	Utilisation des exceptions . . . . .	288
6.6	Création et déclenchement d'exceptions . . . . .	290

7. Les génériques . . . . .	292
7.1 Présentation . . . . .	292
7.2 Classes génériques . . . . .	294
7.2.1 Définition d'une classe générique . . . . .	294
7.2.2 Utilisation d'une classe générique . . . . .	298
7.2.3 Les méthodes génériques . . . . .	301
7.3 Les génériques et l'héritage. . . . .	302
7.4 Limitations des génériques. . . . .	307
8. Les collections . . . . .	309
8.1 Présentation . . . . .	309
8.2 La classe ArrayList . . . . .	312
8.3 La classe HashSet . . . . .	316
8.4 La classe LinkedList . . . . .	321
8.5 La classe HashMap . . . . .	322
8.6 Streams et pipelines . . . . .	324
9. Les entrées/sorties (I/O). . . . .	324
9.1 Introduction . . . . .	324
9.2 Les flux d'entrée et de sortie en Java . . . . .	325
9.2.1 Les flux de caractères . . . . .	326
9.2.2 Les flux d'octets . . . . .	328
9.3 La lecture et l'écriture de fichiers en Java . . . . .	329
9.3.1 La classe File . . . . .	329
9.3.2 La lecture de fichiers avec FileReader et BufferedReader .	331
9.3.3 L'écriture de fichiers avec FileWriter et BufferedWriter .	333
9.4 La sérialisation d'objets en Java . . . . .	334
9.4.1 L'interface Serializable . . . . .	335
9.4.2 Le mot-clé transient . . . . .	337
9.4.3 L'interface Externalizable. . . . .	338
10. Exercices . . . . .	341
10.1 Exercice 1. . . . .	341
10.2 Exercice 2. . . . .	341
10.3 Exercice 3. . . . .	341

- 10.4 Exercice 4 . . . . . 342
- 11. Corrections . . . . . 342
  - 11.1 Correction de l'exercice 1 . . . . . 342
  - 11.2 Correction de l'exercice 2 . . . . . 344
  - 11.3 Correction de l'exercice 3 . . . . . 348
  - 11.4 Correction de l'exercice 4 . . . . . 353

**Chapitre 4**  
**Les expressions lambda**

- 1. Introduction . . . . . 361
- 2. Fonctionnement . . . . . 361
  - 2.1 Les interfaces fonctionnelles . . . . . 361
  - 2.2 Les méthodes anonymes . . . . . 364
    - 2.2.1 Syntaxe générale. . . . . 364
    - 2.2.2 Déclaration des paramètres . . . . . 364
    - 2.2.3 Déclaration du corps . . . . . 365
    - 2.2.4 Utilisation des variables "externes" . . . . . 366
  - 2.3 Les références de méthodes . . . . . 366
    - 2.3.1 Méthode d'instance . . . . . 366
    - 2.3.2 Méthode de classe. . . . . 367
    - 2.3.3 Constructeur. . . . . 367
  - 2.4 L'API `java.util.function` . . . . . 367
    - 2.4.1 Présentation de l'API. . . . . 367
    - 2.4.2 Utilisation . . . . . 368
- 3. Manipulation des collections . . . . . 372
  - 3.1 L'API `Stream` . . . . . 372
  - 3.2 Théorie . . . . . 372
  - 3.3 Obtenir un `Stream` . . . . . 373
    - 3.3.1 Obtenir un `Stream` générique . . . . . 373
    - 3.3.2 Obtenir un `Stream` de numérique . . . . . 375

3.4	Utiliser un Stream	376
3.4.1	Utiliser un Stream générique	376
3.4.2	Utiliser un Stream numérique	380
3.5	La classe <code>Optional&lt;T&gt;</code>	382
4.	Conclusion	383

## Chapitre 5

### Développement d'applications web

1.	Introduction	385
2.	Les bases pour comprendre le Web	386
2.1	Qu'est-ce qu'une API ?	387
2.2	Introduction au protocole HTTP	389
3.	Création d'un serveur web avec le JDK	390
3.1	Présentation de la classe <code>HttpServer</code>	390
3.2	Application serveur	392
3.3	Ajouter des routes et gérer des requêtes	396
3.4	Applications clientes	399
4.	Module de logging	403
4.1	Qu'est-ce que le logging ?	403
4.2	Exemple simple de logging	404
4.3	Configurer les handlers et formatters	405
4.4	Informations générales sur le logging	406

## Chapitre 6

### Accès aux bases de données

1.	Principe de fonctionnement d'une base de données	407
1.1	Terminologie	407
1.2	Le langage SQL	408
1.2.1	Recherche d'informations	409
1.2.2	Ajout d'informations	411

- 1.2.3 Mise à jour d'informations . . . . . 412
- 1.2.4 Suppression d'informations . . . . . 412
- 2. Accès à une base de données à partir de Java . . . . . 413
  - 2.1 Présentation de JDBC . . . . . 415
  - 2.2 Chargement du pilote . . . . . 416
  - 2.3 Établir et manipuler la connexion . . . . . 420
    - 2.3.1 Établir la connexion . . . . . 420
    - 2.3.2 Manipuler la connexion . . . . . 422
  - 2.4 Exécution d'instructions SQL . . . . . 428
    - 2.4.1 Exécution d'instructions de base  
avec le type Statement . . . . . 428
    - 2.4.2 Exécution d'instructions paramétrées  
avec l'objet PreparedStatement . . . . . 437
    - 2.4.3 Exécution de procédures stockées  
avec l'objet CallableStatement . . . . . 443
  - 2.5 Utilisation des jeux d'enregistrements  
avec l'interface ResultSet . . . . . 446
    - 2.5.1 Positionnement dans un ResultSet . . . . . 448
    - 2.5.2 Lecture des données dans un ResultSet . . . . . 452
    - 2.5.3 Modification des données dans un ResultSet . . . . . 456
    - 2.5.4 Suppression de données dans un ResultSet . . . . . 459
    - 2.5.5 Ajout de données dans un ResultSet . . . . . 461
  - 2.6 Gestion des transactions . . . . . 463
    - 2.6.1 Mise en œuvre des transactions . . . . . 465
    - 2.6.2 Points de sauvegarde . . . . . 467
    - 2.6.3 Niveaux d'isolement . . . . . 467

**Chapitre 7**  
**Déploiement d'applications**

- 1. Création d'archives (JAR, WAR) . . . . . 469
  - 1.1 Présentation . . . . . 469

1.2	Manipulation d'une archive . . . . .	470
1.2.1	Création d'une application . . . . .	471
1.2.2	Création d'une archive . . . . .	473
1.2.3	Visualisation du contenu . . . . .	474
1.2.4	Extraction . . . . .	474
1.2.5	Mise à jour . . . . .	475
1.2.6	Exécution . . . . .	475
1.3	Le manifest . . . . .	476
1.3.1	Présentation . . . . .	476
1.3.2	Création . . . . .	476
1.3.3	Utiliser le pilote à l'exécution du JAR . . . . .	478
1.3.4	Embarquer le pilote à l'intérieur même de l'archive (fat JAR) . . . . .	479
2.	Création d'applications natives . . . . .	482
2.1	La création de l'archive Java . . . . .	483
2.2	La création du module de l'application . . . . .	484
2.3	La recherche des dépendances avec jdeps . . . . .	485
2.4	La création de l'arborescence avec jlink . . . . .	486
2.5	L'ajout des dépendances non modulaires . . . . .	488
2.6	Le test de l'application . . . . .	489
3.	Externalisation des paramètres . . . . .	489
3.1	Création du fichier de configuration . . . . .	489
3.2	Utilisation du fichier de configuration . . . . .	491
3.3	Déploiement de l'application . . . . .	492
4.	Création d'un installeur avec jpackage . . . . .	493
5.	Conclusion . . . . .	494
	Index . . . . .	495

Les éléments à télécharger sont disponibles à l'adresse suivante :  
**<http://www.editions-eni.fr>**  
Saisissez la référence ENI de l'ouvrage **RI4JAPOO** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

## Avant-propos

### Chapitre 1

#### Introduction à la POO

- 1. Histoire de la POO ..... 11
- 2. Historique du langage Java ..... 14

### Chapitre 2

#### La conception orientée objet

- 1. Approche procédurale et décomposition fonctionnelle ..... 17
- 2. La transition vers l'approche objet. .... 18
- 3. Les caractéristiques de la POO. .... 19
  - 3.1 L'objet, la classe et la référence ..... 19
    - 3.1.1 L'objet ..... 19
    - 3.1.2 La classe ..... 21
    - 3.1.3 La référence ..... 22
  - 3.2 L'encapsulation ..... 23
  - 3.3 L'héritage. .... 23
  - 3.4 Le polymorphisme ..... 25
  - 3.5 L'abstraction ..... 26
- 4. Le développement objet ..... 27
  - 4.1 Cahier des charges du logiciel ..... 27
  - 4.2 Présentation du cycle en V. .... 28
  - 4.3 Rester AGILE avec le corps en V ..... 32

# 2 \_\_\_\_\_ Apprendre la POO

avec le langage Java

4.4	Modélisation et représentation UML . . . . .	33
4.4.1	Les diagrammes de cas d'utilisation . . . . .	35
4.4.2	Les diagrammes de classes. . . . .	36
4.4.3	Les énumérations. . . . .	42
4.4.4	Les diagrammes de séquences . . . . .	44
4.5	Codage, intégration et mise en production. . . . .	46
5.	Exercices . . . . .	47
5.1	Hiérarchie de classes . . . . .	47
5.2	Relations entre objets. . . . .	49
5.3	Agrégation d'objets. . . . .	49
5.4	Diagramme de cas d'utilisation . . . . .	51
5.5	Diagramme de séquences. . . . .	52

## Chapitre 3

### Introduction à la plate-forme Java

1.	Introduction . . . . .	55
2.	Environnement d'exécution. . . . .	57
3.	Une librairie très complète. . . . .	58
4.	Des outils de développement performants . . . . .	59
5.	Téléchargement et installation du JDK. . . . .	59
6.	Téléchargement et installation d'IntelliJ IDEA . . . . .	60
7.	L'incontournable Hello World . . . . .	65
8.	Un point sur les acronymes . . . . .	76

**Chapitre 4**  
**Les types en Java**

- 1. Introduction ..... 79
- 2. Les types primitifs ..... 80
- 3. Les types référence ..... 82
- 4. Pour nous aider... ..... 84
- 5. La superclasse java.lang.Object ..... 90
  - 5.1 equals..... 91
  - 5.2 hashCode..... 94
  - 5.3 toString..... 96
  - 5.4 finalize..... 98
  - 5.5 getClass, .class et l'opérateur instanceof..... 98
  - 5.6 clone ..... 100
  - 5.7 notify, notifyAll et wait..... 105
- 6. La classe java.lang.String ..... 105
- 7. Exercice ..... 109
  - 7.1 Énoncé..... 109
  - 7.2 Corrigé..... 109

**Chapitre 5**  
**Création de classes**

- 1. Introduction ..... 113
- 2. Package ..... 113
- 3. Déclaration d'une classe ..... 121
  - 3.1 Accessibilité des membres ..... 124
  - 3.2 Attributs ..... 125
  - 3.3 Accesseurs et Mutateurs ..... 128
  - 3.4 Constructeurs ..... 134
    - 3.4.1 Étapes de la construction d'un objet ..... 134
    - 3.4.2 Surcharge de constructeurs..... 136

# 4 \_\_\_\_\_ Apprendre la POO

avec le langage Java

3.4.3	Chaînage de constructeurs . . . . .	136
3.4.4	L'initialiseur static . . . . .	139
3.4.5	L'initialiseur dynamique . . . . .	140
3.4.6	Les constructeurs de type private . . . . .	142
3.4.7	Le « builder pattern » . . . . .	145
3.5	Destructeurs . . . . .	147
3.6	Le mot-clé this et ses vertus . . . . .	152
3.7	Méthodes . . . . .	155
3.7.1	Déclaration . . . . .	155
3.7.2	Passages de paramètres par valeur . . . . .	159
3.8	Surcharge des méthodes . . . . .	165
3.9	Mécanisme des exceptions . . . . .	167
3.9.1	Présentation . . . . .	167
3.9.2	Principe de fonctionnement des exceptions . . . . .	169
3.9.3	Prise en charge de plusieurs exceptions . . . . .	178
3.10	Exercice . . . . .	180
3.10.1	Énoncé . . . . .	180
3.10.2	Conseils . . . . .	181
3.10.3	Corrigé . . . . .	181
4.	Les interfaces . . . . .	184
4.1	Introduction . . . . .	184
4.2	Le contrat . . . . .	185
4.3	Déclaration d'une interface . . . . .	186
4.4	Implémentation . . . . .	187
4.5	IntelliJ IDEA et les interfaces . . . . .	189
4.6	Représentation UML d'une interface . . . . .	191
4.7	Interfaces et polymorphisme . . . . .	192
4.8	Exercice . . . . .	192
4.8.1	Énoncé . . . . .	192
4.8.2	Conseils . . . . .	193
4.8.3	Corrigé . . . . .	196
4.9	Les interfaces de la machine virtuelle Java . . . . .	200

- 5. Associations, compositions et agrégations . . . . . 203
  - 5.1 Les tableaux. . . . . 211
  - 5.2 Les collections . . . . . 218
    - 5.2.1 ArrayList<E> et LinkedList<E>. . . . . 223
    - 5.2.2 Queue<T> et Stack<T> . . . . . 226
    - 5.2.3 HashMap<K, V> . . . . . 226
    - 5.2.4 Les itérateurs . . . . . 227
  - 5.3 Exercice . . . . . 229
    - 5.3.1 Énoncé . . . . . 229
    - 5.3.2 Corrigé . . . . . 230
- 6. Les classes imbriquées. . . . . 232
- 7. Quelques différences avec le C# . . . . . 236

**Chapitre 6**  
**Héritage et polymorphisme**

- 1. Comprendre l'héritage . . . . . 239
- 2. Codage de la superclasse (classe de base)  
 et de sa sous-classe (classe héritière) . . . . . 240
  - 2.1 Interdire l'héritage . . . . . 240
  - 2.2 Définir les membres héritables . . . . . 241
  - 2.3 Syntaxe de l'héritage. . . . . 241
  - 2.4 Exploitation d'une classe héritée . . . . . 242
- 3. Communication entre classe de base et classe héritière. . . . . 244
  - 3.1 Les constructeurs . . . . . 244
  - 3.2 Accès aux membres de la classe de base depuis l'héritier . . . . 248
  - 3.3 Méthodes virtuelles . . . . . 250
  - 3.4 Méthodes de type final. . . . . 255
- 4. Exercice . . . . . 257
  - 4.1 Énoncé. . . . . 257
  - 4.2 Corrigé. . . . . 258

# 6 --- Apprendre la POO

avec le langage Java

5. Les classes abstraites . . . . .	264
6. Le polymorphisme . . . . .	265
6.1 Comprendre le polymorphisme . . . . .	265
6.2 Exploitation du polymorphisme . . . . .	266
6.3 Les opérateurs instanceof et (). . . . .	267

## Chapitre 7

### Communication entre objets

1. L'événementiel : être à l'écoute . . . . .	269
2. Le pattern Observateur . . . . .	270
2.1 Généralités . . . . .	270
2.2 Implémentation en langage Java . . . . .	271
2.3 Les listeners . . . . .	276
2.4 Utilisation d'un listener dans une application graphique . . . . .	278
3. Exercices . . . . .	285
3.1 Exercice 1 . . . . .	285
3.1.1 Énoncé . . . . .	285
3.1.2 Corrigé . . . . .	286
3.2 Exercice 2 . . . . .	289
3.2.1 Énoncé . . . . .	289
3.2.2 Corrigé . . . . .	289
4. Appels synchrones, appels asynchrones . . . . .	292

## Chapitre 8

### Le multithreading

1. Introduction . . . . .	295
2. Comprendre le multithreading . . . . .	295
3. Multithreading et Java . . . . .	298

- 4. Implémentation des threads en Java . . . . . 299
  - 4.1 Étendre la classe Thread . . . . . 299
  - 4.2 Implémenter l'interface Runnable . . . . . 302
  - 4.3 S'endormir et S'attendre . . . . . 304
  - 4.4 Abandon depuis le thread primaire . . . . . 307
  - 4.5 Threads et classes anonymes . . . . . 310
    - 4.5.1 Avec l'interface Runnable . . . . . 310
    - 4.5.2 Avec la classe Thread . . . . . 311
    - 4.5.3 Accès simplifié aux variables  
et aux données membres . . . . . 312
- 5. Synchronisation entre threads . . . . . 316
  - 5.1 Nécessité de la synchronisation . . . . . 316
  - 5.2 Les méthodes « synchronized » . . . . . 318
  - 5.3 Les traitements « synchronized » . . . . . 320
  - 5.4 La classe Semaphore . . . . . 322
- 6. Communication inter-threads . . . . . 325
  - 6.1 La méthode join . . . . . 325
  - 6.2 Les objets de synchronisation . . . . . 328
- 7. Exercice . . . . . 337
  - 7.1 Énoncé . . . . . 337
  - 7.2 Corrigé . . . . . 339

**Chapitre 9**  
**Les tests**

- 1. Introduction . . . . . 347
- 2. Environnement d'exécution des tests unitaires . . . . . 349
- 3. Le projet avec tests unitaires . . . . . 350
- 4. La classe de tests . . . . . 352
- 5. Contenu d'une méthode de test . . . . . 354
- 6. Traitements de préparation et de nettoyage . . . . . 358

# 8 \_\_\_\_\_ Apprendre la POO

avec le langage Java

7. Les tests avec paramètres externes . . . . .	361
8. Les suites de tests . . . . .	367
9. Exercice . . . . .	368
9.1 Énoncé . . . . .	368
9.2 Corrigé . . . . .	368

## Chapitre 10 La réflexion

1. Introduction . . . . .	371
2. Mais pour quoi faire ? . . . . .	372
3. Introspection d'une classe Java . . . . .	374
4. Chargement dynamique et utilisation d'une classe découverte . . .	376
5. Exercice . . . . .	385
6. Privé, mais pas tant . . . . .	386
7. Décompilation et obfuscation . . . . .	387

## Chapitre 11 Anonymat et lambda

1. Introduction . . . . .	393
2. Les classes anonymes . . . . .	394
2.1 D'une pierre deux coups . . . . .	394
2.2 Syntaxe particulière . . . . .	394
2.3 Exemple d'extension d'une superclasse . . . . .	395
2.4 Exemple d'implémentation d'une interface . . . . .	398
2.5 Échange d'informations . . . . .	399
2.6 Exercice . . . . .	402
2.6.1 Énoncé . . . . .	402
2.6.2 Corrigé . . . . .	402

3. Les expressions lambda .....	404
3.1 Le concept .....	404
3.2 Les interfaces « fonctionnelles » comme modèles .....	405
3.3 Les syntaxes lambda .....	407
3.4 Échange d'informations .....	408
3.5 Exercice .....	408
3.5.1 Énoncé .....	408
3.5.2 Correction .....	408
3.6 Paramètre type Lambda .....	411
3.7 java.util.function, un package « vivier » .....	413
3.8 Application sur des collections .....	414
3.9 Conclusion .....	416
Index .....	417