

CYRILLE HERBY

# APPRENEZ À PROGRAMMER EN **JAVA**

LA PROGRAMMATION PROFESSIONNELLE  
À LA PORTÉE DE TOUS !



Issu du célèbre  
**Site du Zéro**  
[www.siteduzero.com](http://www.siteduzero.com)



[www.siteduzero.com](http://www.siteduzero.com)

**zCorrecteurs.fr**



Cet ouvrage a bénéficié des relectures attentives des zCorrecteurs.



Sauf mention contraire, le contenu de cet ouvrage est publié sous la licence :  
Creative Commons BY-NC-SA 2.0

La copie de cet ouvrage est autorisée sous réserve du respect des conditions de la licence  
Texte complet de la licence disponible sur : <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

Simple IT 2011 - ISBN : 978-2-9535278-3-4

# Sommaire

<b>Avant-propos</b>	<b>i</b>
Et Java dans tout ça? . . . . .	ii
Qu'allez-vous apprendre en lisant ce livre ? . . . . .	iii
Comment lire ce livre? . . . . .	iii
Ce livre est issu du Site du Zéro . . . . .	iv
Remerciements . . . . .	v
<b>I Les bases de Java</b>	<b>1</b>
<b>1 Installer les outils de développement</b>	<b>3</b>
Installer les outils nécessaires . . . . .	4
Votre premier programme . . . . .	14
<b>2 Les variables et les opérateurs</b>	<b>23</b>
Les différents types de variables . . . . .	24
Les opérateurs arithmétiques . . . . .	27
Les conversions, ou « cast » . . . . .	30
<b>3 Lire les entrées clavier</b>	<b>33</b>
La classe Scanner . . . . .	34
Récupérer ce que vous tapez . . . . .	35
<b>4 Les conditions</b>	<b>39</b>

La structure <code>if... else</code> . . . . .	40
La structure <code>switch</code> . . . . .	43
La condition ternaire . . . . .	44
<b>5 Les boucles</b>	<b>47</b>
La boucle <code>while</code> . . . . .	48
La boucle <code>do... while</code> . . . . .	52
La boucle <code>for</code> . . . . .	53
<b>6 TP : conversion Celsius - Fahrenheit</b>	<b>55</b>
Élaboration . . . . .	56
Correction . . . . .	57
<b>7 Les tableaux</b>	<b>61</b>
Tableau à une dimension . . . . .	62
Les tableaux multidimensionnels . . . . .	62
Utiliser et rechercher dans un tableau . . . . .	63
<b>8 Les méthodes de classe</b>	<b>69</b>
Quelques méthodes utiles . . . . .	70
Créer sa propre méthode . . . . .	72
La surcharge de méthode . . . . .	75
<b>II Java et la Programmation Orientée Objet</b>	<b>79</b>
<b>9 Votre première classe</b>	<b>81</b>
Structure de base . . . . .	82
Les constructeurs . . . . .	83
Accesseurs et mutateurs . . . . .	88
Les variables de classes . . . . .	94
Le principe d'encapsulation . . . . .	96
<b>10 L'héritage</b>	<b>99</b>
Le principe de l'héritage . . . . .	100
Le polymorphisme . . . . .	104

---

<b>11 Modéliser ses objets grâce à UML</b>	<b>111</b>
Présentation d'UML . . . . .	112
Modéliser ses objets . . . . .	113
Modéliser les liens entre les objets . . . . .	114
<b>12 Les packages</b>	<b>119</b>
Création d'un package . . . . .	120
Droits d'accès entre les packages . . . . .	121
<b>13 Les classes abstraites et les interfaces</b>	<b>123</b>
Les classes abstraites . . . . .	124
Les interfaces . . . . .	133
Le pattern strategy . . . . .	137
<b>14 Les exceptions</b>	<b>157</b>
Le bloc <code>try{...} catch{...}</code> . . . . .	158
Les exceptions personnalisées . . . . .	160
La gestion de plusieurs exceptions . . . . .	164
<b>15 Les flux d'entrée/sortie</b>	<b>167</b>
Utilisation de <code>java.io</code> . . . . .	168
Utilisation de <code>java.nio</code> . . . . .	187
Le pattern decorator . . . . .	190
<b>16 Les énumérations</b>	<b>197</b>
Avant les énumérations . . . . .	198
Une solution : les <code>enum</code> . . . . .	199
<b>17 Les collections d'objets</b>	<b>203</b>
Les différents types de collections . . . . .	204
Les objets <code>List</code> . . . . .	205
Les objets <code>Map</code> . . . . .	208
Les objets <code>Set</code> . . . . .	209
<b>18 La généricité en Java</b>	<b>213</b>
Principe de base . . . . .	214
Généricité et collections . . . . .	219

<b>19 Java et la réflexivité</b>	<b>227</b>
L'objet <code>Class</code> . . . . .	228
Instanciation dynamique . . . . .	232
<b>III Les interfaces graphiques</b>	<b>237</b>
<b>20 Notre première fenêtre</b>	<b>239</b>
L'objet <code>JFrame</code> . . . . .	240
L'objet <code>JPanel</code> . . . . .	245
Les objets <code>Graphics</code> et <code>Graphics2D</code> . . . . .	246
<b>21 Le fil rouge : une animation</b>	<b>259</b>
Création de l'animation . . . . .	260
Améliorations . . . . .	263
<b>22 Positionner des boutons</b>	<b>269</b>
Utiliser la classe <code>JButton</code> . . . . .	270
Positionner son composant : les layout managers . . . . .	272
<b>23 Interagir avec des boutons</b>	<b>289</b>
Une classe <code>Bouton</code> personnalisée . . . . .	290
Interagir avec son bouton . . . . .	298
Être à l'écoute de ses objets : le design pattern <code>Observer</code> . . . . .	318
Cadeau : un bouton personnalisé optimisé . . . . .	327
<b>24 TP : une calculatrice</b>	<b>331</b>
Élaboration . . . . .	332
Conception . . . . .	332
Correction . . . . .	333
Générer un <code>.jar</code> exécutable . . . . .	338
<b>25 Exécuter des tâches simultanément</b>	<b>345</b>
Une classe héritée de <code>Thread</code> . . . . .	346
Utiliser l'interface <code>Runnable</code> . . . . .	350
Synchroniser ses threads . . . . .	354
Contrôler son animation . . . . .	355

---

<b>26 Les champs de formulaire</b>	<b>359</b>
Les listes : l'objet JComboBox . . . . .	360
Les cases à cocher : l'objet JCheckBox . . . . .	370
Les champs de texte : l'objet JTextField . . . . .	381
Contrôle du clavier : l'interface KeyListener . . . . .	385
<b>27 Les menus et boîtes de dialogue</b>	<b>391</b>
Les boîtes de dialogue . . . . .	392
Les menus . . . . .	408
<b>28 TP : l'ardoise magique</b>	<b>439</b>
Cahier des charges . . . . .	440
Prérequis . . . . .	441
Correction . . . . .	442
Améliorations possibles . . . . .	448
<b>29 Conteneurs, sliders et barres de progression</b>	<b>449</b>
Autres conteneurs . . . . .	450
Enjoliver vos IHM . . . . .	467
<b>30 Les arbres et leur structure</b>	<b>471</b>
La composition des arbres . . . . .	472
Des arbres qui vous parlent . . . . .	476
Décorez vos arbres . . . . .	481
Modifier le contenu de nos arbres . . . . .	486
<b>31 Les interfaces de tableaux</b>	<b>495</b>
Premiers pas . . . . .	496
Gestion de l'affichage . . . . .	497
Interaction avec l'objet JTable . . . . .	508
Ajouter des lignes et des colonnes . . . . .	515
<b>32 TP : le pendu</b>	<b>519</b>
Cahier des charges . . . . .	520
Prérequis . . . . .	522
Correction . . . . .	522

<b>33 Mieux structurer son code : le pattern MVC</b>	<b>525</b>
Premiers pas . . . . .	526
Le modèle . . . . .	528
Le contrôleur . . . . .	531
La vue . . . . .	534
<b>34 Le Drag'n Drop</b>	<b>539</b>
Présentation . . . . .	540
Fonctionnement . . . . .	543
Créer son propre <code>TransferHandler</code> . . . . .	547
Activer le drop sur un <code>JTree</code> . . . . .	553
Effet de déplacement . . . . .	558
<b>35 Mieux gérer les interactions avec les composants</b>	<b>565</b>
Présentation des protagonistes . . . . .	566
Utiliser l'EDT . . . . .	567
La classe <code>SwingWorker&lt;T, V&gt;</code> . . . . .	570
<b>IV Interactions avec les bases de données</b>	<b>577</b>
<b>36 JDBC : la porte d'accès aux bases de données</b>	<b>579</b>
Rappels sur les bases de données . . . . .	580
Préparer la base de données . . . . .	584
Se connecter à la base de données . . . . .	591
<b>37 Fouiller dans sa base de données</b>	<b>597</b>
Le couple <code>Statement</code> – <code>ResultSet</code> . . . . .	598
Les requêtes préparées . . . . .	607
Modifier des données . . . . .	613
<code>Statement</code> , toujours plus fort . . . . .	615
Gérer les transactions manuellement . . . . .	617
<b>38 Limiter le nombre de connexions</b>	<b>621</b>
Pourquoi ne se connecter qu'une seule fois ? . . . . .	622
Le pattern singleton . . . . .	622

Le singleton dans tous ses états . . . . .	625
<b>39 TP : un testeur de requêtes</b>	<b>629</b>
Cahier des charges . . . . .	630
Quelques captures d'écran . . . . .	630
Correction . . . . .	630
<b>40 Lier ses tables avec des objets Java : le pattern DAO</b>	<b>633</b>
Avant toute chose . . . . .	634
Le pattern DAO . . . . .	639
Le pattern factory . . . . .	649