

Bien programmer en **Java 7**

Avec plus de **50 études de cas**
et des comparaisons avec **C++ et C#**

Plus de
10 000 ex.
vendus !
Édition en
couleur

Emmanuel Puybaret



EYROLLES

© Groupe Eyrolles, 2012, ISBN : 978-2-212-12974-8

Table des matières

AVANT-PROPOS	V
Organisation de l'ouvrage • V	
À qui s'adresse cet ouvrage ? • VI	
Remerciements • VII	
1. PRÉSENTATION DES ÉTUDES DE CAS	1
Applications isolées • 2	
Carnet d'adresses • 2	
Calcul des mensualités d'un emprunt • 3	
Forum de discussion • 4	
Principales fonctionnalités • 4	
Architecture technique • 5	
Module de messagerie instantanée (chat) • 6	
En résumé... • 7	
2. PRINCIPES DU LANGAGE ET INSTALLATION DE L'ENVIRONNEMENT	9
Programmer en Java : une démarche objet • 10	
Du binaire à l'objet, 50 ans d'évolution de la programmation • 10	
Ce que fait un objet et comment il le fait... interface et implémentation • 12	
De l'analyse objet à l'écriture des classes Java • 13	
Écriture, compilation, exécution • 13	
À chaque besoin son environnement Java : applets, servlets et applications • 14	
Télécharger et installer les programmes pour développer en Java • 15	
Installation sous Windows • 17	
Installation sous Linux • 18	
Installation sous Mac OS X • 18	
Télécharger les démos et la documentation • 19	
Tester l'installation : votre première application Java • 20	
Compilation de l'application • 21	
Les cinq erreurs de compilation les plus fréquentes • 22	
Exécution de l'application • 23	
Les trois erreurs d'exécution les plus fréquentes • 23	
En résumé... • 25	
3. CRÉATION DE CLASSES	27
Typage : pourquoi et comment ? • 28	
Types de données objet et références • 29	
Écrire une valeur littérale • 29	
Affectation de variable • 30	
Par l'exemple : déclarer et utiliser quelques variables • 31	
Encapsuler pour protéger les données des objets • 32	
Portée d'utilisation et durée de vie • 33	
Manipuler des chaînes avec les méthodes de la classe <code>java.lang.String</code> • 34	
Par l'exemple : construire un texte avec plusieurs chaînes • 36	
Définir une nouvelle classe • 36	
Structure d'un fichier <code>.java</code> • 37	
Commenter une classe • 37	
Déclarer les champs d'une classe • 38	
Déclarer les méthodes d'une classe • 39	
Paramétrage d'une méthode • 40	
Implémenter les méthodes • 40	
Par l'exemple : une classe simulant une télécarte • 40	
Créer des objets • 42	
Par l'exemple : une histoire de télécarte empruntée... • 42	
Initialiser les champs d'un objet • 44	
Initialiser un objet avec un constructeur • 44	
Par l'exemple : une classe simulant un service • 46	

Surcharger les méthodes et les constructeurs • 47	Limiter la réutilisation avec final • 91
Organiser les fichiers des classes • 49	Déclarer des constantes • 92
Automatiser la compilation avec un fichier de commandes • 50	Typer des constantes avec une énumération • 92
Exécuter une application • 52	Par l'exemple : tester le titre d'un contact • 93
Simplifier l'écriture des classes avec import • 52	En résumé... • 94
Par l'exemple : afficher les unités restantes d'une télécarte • 52	6. LES CLASSES DE BASE DE LA BIBLIOTHÈQUE JAVA 97
En résumé... • 53	La super-classe de toutes les classes : java.lang.Object • 98
4. CONTRÔLE DES TRAITEMENTS AVEC LES OPÉRATEURS, BOUCLES ET BRANCHEMENTS 55	La méthode equals • 98
Opérateurs à connaître • 56	La méthode hashCode • 98
Conversions numériques avec l'opérateur de cast • 58	La méthode toString • 99
Par l'exemple : conversion euro/franc français • 59	Forum : utilisateur du forum de discussion • 99
Priorité des opérateurs • 61	Manipuler les chaînes de caractères (java.lang.String) • 104
Par l'exemple : comparer la somme de montants convertis • 61	Forum : outils de traitement pour les textes du forum • 104
Piloter le programme avec les instructions de contrôle : boucles et branchements • 63	Communiquer avec la machine virtuelle (java.lang.System) • 107
Tester et décider sur condition avec if et switch • 63	Par l'exemple : ce que connaît la JVM de votre système... • 108
Syntaxe des instructions if et if else • 63	Effectuer des calculs mathématiques (java.lang.Math) • 110
Syntaxe de l'instruction switch • 63	Par l'exemple : quelques valeurs mathématiques remarquables • 110
Par l'exemple : convertir un nombre en toutes lettres • 64	Utiliser un type primitif sous forme d'objet avec les classes d'emballage • 111
Répéter un traitement avec les boucles while, do et for • 67	Par l'exemple : calculer les mensualités d'un emprunt • 112
Par l'exemple : quelques calculs de probabilité classiques • 69	Gérer la date et l'heure • 114
Portée des variables locales et des paramètres • 71	Mémoriser la date et l'heure (java.util.Date) • 115
En résumé... • 73	Afficher la date et l'heure (java.text.DateFormat) • 115
5. RÉUTILISATION DES CLASSES 75	Forum : message du forum • 116
Réutiliser en composant : la relation « a un » • 76	Fixer et manipuler la date et l'heure (java.util.GregorianCalendar) • 119
Par l'exemple : une même adresse pour deux personnes • 76	Par l'exemple : bon anniversaire ! • 120
Réutiliser en héritant : la relation « est un » • 78	Les tableaux pour gérer des ensembles d'éléments • 121
Définir une sous-classe • 79	Déclarer et créer un tableau • 122
Initialisation en deux temps pour les objets d'une sous-classe • 80	Utiliser un tableau • 123
Par l'exemple : alcoolisée ou non, choisissez votre boisson • 80	Forum : créer le mot de passe d'un utilisateur • 124
Réutiliser en implémentant différemment : le polymorphisme • 82	Boucle itérative • 124
Relation « est un » et conversion de référence • 82	Par l'exemple : afficher les jours fériés de l'année • 125
Par l'exemple : boisson et boisson alcoolisée, ne mélangez pas les genres... • 83	Tableau multidimensionnel • 126
Modifier l'implémentation d'une méthode avec la redéfinition • 84	Manipuler les tableaux avec java.util.Arrays • 126
Par l'exemple : changer de message • 85	Par l'exemple : trier les paramètres d'une application • 127
Modifier l'implémentation sans oublier la méthode redéfinie • 86	Les collections pour gérer des ensembles d'objets • 128
Par l'exemple : calculer les intérêts d'un compte épargne • 87	Typer les objets d'une collection avec la généricité • 130
Réutiliser sans créer d'objet avec les méthodes de classe • 89	Listes ordonnées d'objets (java.util.ArrayList et java.util.LinkedList) • 130
Par l'exemple : afficher l'état d'un compte • 89	Par l'exemple : casier à bouteilles ou cave à vin ? • 132
	Ensembles d'objets uniques (java.util.HashSet et java.util.TreeSet) • 133

- Dictionnaires d'objets
(`java.util.HashMap` et `java.util.TreeMap`) • 134
- Par l'exemple : organiser les définitions d'un glossaire • 135
- En résumé... • 137
- 7. ABSTRACTION ET INTERFACE..... 139**
- Créer des classes abstraites pour les concepts abstraits • 140
- Par l'exemple : comparer les surfaces de différentes figures • 140
- Séparer l'interface de l'implémentation • 143
- Définir une interface • 143
- Par l'exemple : donner un prix à un objet • 144
- Implémenter une interface • 144
- Par l'exemple : implémenter le prix d'un objet • 145
- Utilisation des interfaces • 146
- Conversion de référence, suite et fin • 146
- Par l'exemple : boisson ou service, tout se paie • 146
- Par l'exemple : l'addition s'il vous plaît ! • 148
- Implémenter l'interface `java.lang.Comparable` pour comparer deux objets • 150
- Par l'exemple : gérer l'ordre chronologique d'événements • 150
- Énumérer les éléments d'une collection avec l'interface `java.util.Iterator` • 152
- Par l'exemple : trier les événements d'un agenda dans l'ordre chronologique • 153
- Manipuler les collections avec la classe `java.util.Collections` • 154
- Par l'exemple : quels numéros mettre dans ma grille de loto aujourd'hui ? • 155
- En résumé... • 159
- 8. GESTION DES ERREURS AVEC LES EXCEPTIONS 161**
- La pile d'exécution, organisation et fonctionnement • 162
- Par l'exemple : calculer une factorielle • 162
- Gérer les exceptions • 165
- Même un programme simple peut cacher des erreurs • 165
- Intercepter une exception avec `try catch` • 166
- Par l'exemple : vérifier les erreurs de saisie • 167
- Déclencher une exception avec `throw` • 168
- Par l'exemple : surveiller les cas limites • 168
- Décrire un traitement final avec `finally` • 170
- Par l'exemple : `finally`, demander confirmation pour continuer • 171
- Catégories d'exceptions Java • 172
- Exceptions non contrôlées • 172
- Exceptions contrôlées • 173
- Manipuler une classe à l'exécution avec la réflexion • 174
- Créer une classe d'exception • 179
- En résumé... • 179
- 9. LECTURE ET ÉCRITURE DE FICHIERS..... 181**
- Explorer le système de fichiers (`java.io.File`) • 182
- Par l'exemple : rechercher les fichiers dans un dossier et ses sous-dossiers • 183
- Lire et écrire des données sous forme de flux • 184
- Mode d'accès aux données • 185
- Mode d'accès par flux de données • 185
- Mode d'accès aléatoire • 186
- Lecture avec les flux de données • 186
- Contrôler les erreurs sur un flux de données avec les exceptions • 187
- Par l'exemple : compter le nombre d'occurrences d'un caractère dans un fichier • 189
- Écriture avec les flux de données • 190
- Filtrage des données d'un flux • 191
- Par l'exemple : éliminer les commentaires d'un programme Java • 195
- Par l'exemple : compter les lignes de code d'un ensemble de fichiers Java • 198
- Configurer une application • 200
- Fichiers de traduction • 200
- Fichiers de préférences • 201
- En résumé... • 201
- 10. INTERFACES UTILISATEUR AVEC SWING..... 203**
- Composants d'interface utilisateur • 204
- Mise en page des composants avec les layouts • 205
- Agencer les composants les uns à la suite des autres (`java.awt.FlowLayout`) • 205
- Par l'exemple : afficher des champs de saisie et leurs labels • 206
- Disposer les composants dans une grille (`java.awt.GridLayout`) • 207
- Par l'exemple : interface utilisateur d'un clavier de calculatrice • 207
- Placer les composants aux bords du conteneur (`java.awt.BorderLayout`) • 208
- Par l'exemple : interface utilisateur d'un éditeur de textes • 209
- Mise en page évoluée par combinaison de layouts • 212
- Par l'exemple : panneau de saisie des coordonnées d'un contact • 213
- À chaque système son look and feel • 216
- Interagir avec l'utilisateur grâce aux événements • 217

Événements • 218	Choisir un protocole pour communiquer • 262
Être à l'écoute des événements en implémentant un listener • 218	Adresse IP et port, point de rendez-vous des serveurs Internet • 263
Par l'exemple : quelle heure est-il ? • 219	Requête HTTP vers une URL • 263
Utiliser les classes anonymes pour implémenter un listener • 220	Par l'exemple : afficher le contenu d'une URL
Par l'exemple : calculer des tirages de loto • 220	dans une fenêtre Swing • 264
Par l'exemple : interface utilisateur d'un carnet d'adresses • 222	Programme CGI • 266
Créer vos composants graphiques • 225	Utiliser un formulaire HTML pour paramétrer
Par l'exemple : dessiner le plan d'une maison • 225	un programme CGI • 266
Programmer une applet • 227	Par l'exemple : un formulaire de recherche • 267
Par l'exemple : bienvenue dans le monde des applets ! • 228	Programmation d'une servlet sur le serveur • 268
Créer une interface utilisateur avec une applet • 230	Classe javax.servlet.http.HttpServlet • 268
Par l'exemple : interface utilisateur du calcul de mensualités • 230	Interface javax.servlet.http.HttpServletRequest • 268
En résumé... • 233	Interface javax.servlet.http.HttpServletResponse • 269
11. CONNEXION À LA BASE DE DONNÉES AVEC JDBC 235	Renvoyer du texte HTML avec une servlet • 269
Utilisation d'une base de données en Java • 236	Par l'exemple : Bienvenue dans le monde des servlets ! • 269
Se connecter à une base de données avec un driver JDBC • 237	Installation de Tomcat • 270
Par l'exemple : tester la connexion avec la base de données • 238	Sous Windows • 270
Installation du SGBD MySQL • 239	Sous Linux et Mac OS X • 271
Sous Windows • 239	Lancement de Tomcat • 271
Sous Linux • 239	Sous Windows • 271
Sous Mac OS X • 240	Sous Linux et Mac OS X • 272
Installer le driver JDBC • 240	Organiser les fichiers d'une application web • 272
SQL, le langage des bases de données • 241	Compilation d'une application web • 273
Principaux types de données • 241	Mise en route d'une application web • 273
Mettre à jour les tables et les index • 242	Par l'exemple : exécuter la servlet de bienvenue • 274
Modifier et rechercher les enregistrements d'une table • 242	Cycle d'exécution de la servlet de bienvenue • 274
Programmation SQL avec JDBC • 243	Mise à jour d'une application web • 275
Utiliser une connexion JDBC (java.sql.Connection) • 243	Créer l'interface d'une application web avec les JavaServer Pages • 276
Exécuter des instructions SQL (java.sql.Statement) • 244	Balises JSP pour inclure du contenu dynamique • 277
Exploiter les résultats d'une sélection SQL (java.sql.ResultSet) • 244	Variables JSP prédéfinies • 277
Par l'exemple : enregistrer les factures client • 245	Par l'exemple : bienvenue dans le monde JSP • 278
Obtenir des informations sur la base de données (java.sql.DatabaseMetaData) • 247	Exécuter la page JSP de bienvenue • 279
Forum : gérer la connexion à la base de données • 247	Contrôle des erreurs dans une page JSP • 279
Paramétrer les instructions SQL d'accès à la base du forum	Mise à jour des pages JSP • 280
(java.sql.PreparedStatement) • 250	Utiliser les classes Java dans une page JSP • 280
Forum : stocker utilisateurs et messages dans la base de	Utiliser les composants JavaBeans dans une page JSP • 280
données • 251	Par l'exemple : créer une liste de courses • 282
En résumé... • 258	Faire appel à d'autres pages JSP • 284
12. PROGRAMMATION WEB AVEC LES SERVLETS,	En résumé... • 284
JSP ET JAVABEANS..... 261	13. INTERFACE UTILISATEUR DU FORUM 287
Protocole HTTP et programme CGI • 262	Scénario d'utilisation • 288
Principe de l'architecture client-serveur • 262	Scénario pour un utilisateur non identifié • 288
	Scénario pour un utilisateur identifié • 288

Programmation des pages du forum • 290

Organisation des pages du forum • 290

Utilisation des classes des packages `com.eteks.forum` et `com.eteks.outils` • 290Classe `com.eteks.forum.ConnecteurForum` • 291Classe `com.eteks.forum.UtilisateurForum` • 291Classe `com.eteks.forum.MessageForum` • 292Classe `com.eteks.forum.EnsembleMessagesForum` • 292Classe `com.eteks.outils.OutilsChaîne` • 292Classe `com.eteks.outils.MotDePasse` • 292

Identification de l'utilisateur • 292

Page d'accueil • 296

Inscription d'un utilisateur • 299

Messages d'un sujet • 301

Création de sujet, de message, et modification • 302

Pages de saisie • 302

Pages d'ajout et de modification de message • 305

Quitter l'application • 306

En résumé... • 307

14. ÉCHANGER DES INFORMATIONS AVEC XML..... 309**Premiers contacts avec XML • 310**

Description d'un document XML • 310

Par l'exemple : représenter une facture en XML • 311

Document XML bien formé • 312

Espace de noms • 312

Par l'exemple : associer un espace de noms aux éléments d'une facture • 313

Document XML valide et DTD • 313

Créer une DTD • 314

Par l'exemple : définir la DTD des factures • 315

Utiliser une DTD dans un document XML • 316

Par l'exemple : utiliser la DTD d'une facture dans un document XML • 316

Typier les informations XML avec un schéma XML • 317

Types prédéfinis XML Schema • 317

Déclarer la syntaxe d'un document avec les éléments XML Schema • 317

Par l'exemple : définir le schéma XML des factures • 321

Utiliser un schéma XML dans un document XML • 322

Par l'exemple : utiliser le schéma XML d'une facture dans un document • 322

Transformer un document XML en un autre document • 322

Retrouver des éléments et des attributs avec XPath • 323

Transformer un document XML en un autre document XML avec XSLT • 323

Par l'exemple : extraire la liste des articles d'une facture • 324

Analyser un document XML avec JAXP • 325

Obtenir une instance d'un analyseur • 326

Analyser un document avec SAX • 326

Par l'exemple : rechercher les articles d'une facture • 327

Vérifier la validité d'un document avec SAX • 329

Par l'exemple : rechercher les erreurs dans un document XML • 329

Analyser un document avec DOM • 332

Par l'exemple : rechercher le client d'une facture • 332

Forum : rechercher les utilisateurs ou les messages d'un document XML • 333

Transformer un document XML • 338

Par l'exemple : transformer une facture par programme • 338

Par l'exemple : créer le document XML d'un modèle DOM • 339

Gérer la correspondance entre objets et éléments XML avec JAXB • 340

Définir la syntaxe d'un document XML avec les annotations JAXB • 341

Par l'exemple : définir une société et ses employés avec JAXB • 341

Marshalling et unmarshalling • 344

Par l'exemple : créer le document XML d'une société • 345

Par l'exemple : lire les objets correspondant au document XML d'une société • 345

En résumé... • 346

15. MESSAGERIE INSTANTANÉE AVEC LA PROGRAMMATION MULTITÂCHE 349**Gestion d'animations avec la classe `javax.swing.Timer` • 350**

Par l'exemple : afficher les nouvelles • 350

Programmation d'un thread avec la classe `java.lang.Thread` • 352Implémenter la méthode `run` • 353**Ajout d'un module de chat au forum de discussion • 354**

Interaction entre l'applet de chat et les pages JSP • 355

Composants JavaBeans du serveur pour le chat • 356

Ensemble des messages du chat • 356

Message du chat • 357

Ensemble des participants au chat • 357

Date de la dernière lecture des messages • 357

Pages JSP de gestion du chat • 357

Arrivée d'un utilisateur dans le chat • 357	Installation • 387
Lecture des participants au chat • 358	Démarrage • 387
Lecture des messages du chat • 359	Création des classes • 387
Ajout d'un message dans le chat • 360	Édition des classes • 388
Départ d'un participant du chat • 361	Compilation et exécution • 388
Interface utilisateur du chat • 361	Eclipse • 388
Threads nécessaires au chat • 366	Installation • 389
Gestion de l'accès aux pages JSP du serveur • 366	Démarrage • 389
Page de lancement de l'applet • 367	Création d'un projet • 389
Intégration du chat au forum de discussion • 368	Création des classes • 389
Synchronisation du module de chat • 369	Édition des classes • 390
États d'un thread • 369	Compilation et exécution • 390
Synchroniser les traitements sur les données partagées • 370	E. Erreurs de compilation les plus fréquentes • 390
De la nécessité de synchroniser... • 370	Symbole introuvable • 391
Synchroniser avec synchronized • 371	Déclaration de classe incorrecte • 392
Chat : synchroniser l'accès à la liste des participants • 372	Déclaration de méthode incorrecte • 392
Synchroniser les traitements dans un ordre déterminé • 375	Modificateur d'accès incorrect • 392
Synchroniser avec wait et notify • 375	Déclaration de variable locale incorrecte • 393
Chat : synchroniser l'envoi des nouveaux messages aux applets • 377	Utilisation de variable incorrecte • 393
En résumé... • 381	Erreur avec return • 393
ANNEXES 383	Erreur dans les conditions des instructions if, for ou while • 394
A. Types de licences logicielles • 383	Équilibre incorrect entre accolades ouvrantes et fermantes • 394
B. Fichiers du forum de discussion • 384	Chaîne littérale non fermée • 394
C. Précisions sur les commentaires javadoc • 386	Commentaire non fermé • 394
D. Mise en route de ConTEXT et d'Eclipse • 386	F. Bibliographie • 395
ConTEXT • 387	G. Glossaire • 396
	INDEX 399