



Expert  
EXPO  
EX

Nouvelle édition

# Java et Eclipse

Développez une application  
avec **Java** et **Eclipse**

Téléchargement  
[www.editions-eni.fr](http://www.editions-eni.fr)



Frédéric DÉLÉCHAMP  
Henri LAUGIÉ

Les éléments à télécharger sont disponibles à l'adresse suivante :  
**<http://www.editions-eni.fr>**  
Saisissez la référence de l'ouvrage **EI8JAV** dans la zone de recherche  
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

**Avant-propos**

- 1. Introduction ..... 11
- 2. Contenu de l'ouvrage ..... 12

**Chapitre 1**  
**Environnement de développement**

- 1. Installation du JDK Java 8 ..... 15
- 2. Installation d'Eclipse Luna ..... 17
- 3. Découverte de l'IDE ..... 21
  - 3.1 Perspective - Vue ..... 22
  - 3.2 Éditeur de code ..... 24
  - 3.3 Débogueur ..... 26

**Chapitre 2**  
**Prise en main d'Eclipse**

- 1. Premier projet ..... 31
- 2. Premier package ..... 36
- 3. Première classe ..... 37
- 4. Première fenêtre ..... 45
  - 4.1 Création de la fenêtre ..... 46
  - 4.2 Ajout de composants ..... 51
- 5. Action d'un bouton ..... 55
- 6. Création d'un menu ..... 58
- 7. Premier exécutable ..... 63

## Chapitre 3 Concepts de base de la POO

1. Introduction . . . . .	67
2. Objet . . . . .	68
3. Classe. . . . .	69
4. Attribut . . . . .	71
5. Type de données. . . . .	72
6. Méthode . . . . .	74
7. Accessibilité. . . . .	77
8. Encapsulation . . . . .	81
9. Constructeur. . . . .	82
10. Héritage. . . . .	83
11. Interfaces. . . . .	89
12. Polymorphisme. . . . .	91
12.1 Par surcharge. . . . .	91
12.2 Par redéfinition . . . . .	93
12.3 Par interface . . . . .	95
13. Principes SOLID . . . . .	97
13.1 Single Responsibility . . . . .	97
13.2 Open/Closed . . . . .	97
13.3 Liskov Substitution . . . . .	98
13.4 Interface Segregation . . . . .	99
13.5 Dependency Inversion . . . . .	100
14. Quelques autres principes utiles . . . . .	101
14.1 DRY (Don't Repeat Yourself) . . . . .	101
14.2 KISS (Keep It Simple, Stupid) . . . . .	101
14.3 YAGNI (You Aren't Gonna Need It). . . . .	102

## Chapitre 4

### La boîte à outils de Java

1. Génériques.....	103
2. Collections .....	104
3. Gestion des erreurs .....	107
4. Boxing/Unboxing .....	109
5. Enums .....	111
6. Gestion du temps et des dates .....	112
7. Événements.....	121
8. Lambdas .....	123
9. Classes graphiques .....	125
9.1 Boutons.....	125
9.1.1 JButton.....	125
9.1.2 JCheckBox .....	126
9.1.3 JRadioButton .....	126
9.1.4 JToggleButton .....	127
9.2 Saisies textuelles .....	127
9.2.1 JTextField.....	127
9.2.2 JPasswordField.....	128
9.2.3 JFormattedTextField.....	128
9.2.4 JTextArea .....	128
9.2.5 JEditorPane et JTextPane .....	129
9.3 Saisie de nombres .....	129
9.3.1 JSpinner .....	129
9.3.2 JSlider .....	130
9.4 Affichage d'informations .....	130
9.4.1 JLabel .....	130
9.4.2 JProgressBar .....	131
9.4.3 JSeparator.....	132

# 4 **Java et Eclipse**

Développez une application avec Java et Eclipse

9.5	Affichage et sélection d'objets . . . . .	133
9.5.1	JComboBox . . . . .	133
9.5.2	JList . . . . .	134
9.6	Données structurées . . . . .	135
9.6.1	JTree . . . . .	136
9.6.2	JTable . . . . .	136
9.7	Conteneurs . . . . .	136
9.7.1	JPanel . . . . .	137
9.7.2	JScrollPane . . . . .	137
9.7.3	JSplitPane . . . . .	138
9.7.4	JTabbedPane . . . . .	139
9.7.5	JToolBar . . . . .	139
9.8	Fenêtres . . . . .	140
9.8.1	JFrame . . . . .	140
9.8.2	JDialog . . . . .	141
9.8.3	JOptionPane . . . . .	142
9.8.4	JFileChooser . . . . .	144
9.8.5	JColorChooser . . . . .	145
9.9	Menus . . . . .	145
9.9.1	JMenuBar . . . . .	146
9.9.2	JMenu . . . . .	147
9.9.3	JMenuItem . . . . .	147
9.9.4	JPopupMenu . . . . .	149
9.10	Layouts . . . . .	149
9.10.1	FlowLayout . . . . .	149
9.10.2	BoxLayout . . . . .	150
9.10.3	BorderLayout . . . . .	150
9.10.4	GridLayout . . . . .	151
9.10.5	GridBagLayout . . . . .	152
9.11	Look and Feel . . . . .	152
10.	Threads . . . . .	154
11.	Annotations . . . . .	161
12.	Autres notions . . . . .	162

**Chapitre 5**  
**La boîte à outils d'Eclipse**

- 1. Propriétés du projet ..... 165
- 2. Tests unitaires. .... 167
- 3. Internationalisation ..... 177
  - 3.1 Traduction des chaînes de caractères ..... 177
  - 3.2 Tests des traductions ..... 182
- 4. Ajout de plug-ins ..... 188
- 5. Ajout sélectif de bibliothèques. .... 191
- 6. Gérer les révisions d'un projet ..... 192

**Chapitre 6**  
**Présentation du projet**

- 1. Application Luna ..... 199
- 2. Cahier des charges ..... 200
- 3. Persistance des données ..... 201
- 4. Utilisation des patrons de conception. .... 202

**Chapitre 7**  
**Analyse**

- 1. Démarche ..... 203
- 2. Ajout du plug-in UML dans Eclipse ..... 207
- 3. Diagramme de cas d'utilisation ..... 209
- 4. Diagramme de séquence. .... 214
- 5. Diagramme de communication. .... 217
- 6. Diagramme de classes. .... 218

# 6 --- Java et Eclipse

Développez une application avec Java et Eclipse

## Chapitre 8 Base de données MySQL

1. Introduction . . . . .	223
2. SQL et les bases de données relationnelles . . . . .	223
2.1 Les bases de données relationnelles . . . . .	223
2.2 SQL . . . . .	225
2.3 Transactions . . . . .	226
3. Installation et configuration du serveur XAMPP . . . . .	228
3.1 Gestion de l'accès au serveur . . . . .	232
3.2 Création de la base de données MySQL . . . . .	233
4. JDBC . . . . .	237
5. JPA . . . . .	239
6. Installation des pilotes . . . . .	240
6.1 Connector/ODBC . . . . .	240
6.2 Connector/J . . . . .	243
6.3 EclipseLink . . . . .	245

## Chapitre 9 Maquettes

1. Création des fenêtres . . . . .	251
1.1 Fenêtre de connexion . . . . .	256
1.2 Fenêtre d'accueil de l'application . . . . .	266
1.3 Fenêtre de gestion des clients . . . . .	275
1.3.1 Panneau principal . . . . .	276
1.3.2 Panneaux d'ajout et de modification . . . . .	277
1.3.3 Panneau de recherche . . . . .	278
1.4 Fenêtre de gestion des articles . . . . .	279
1.5 Fenêtres de gestion des commandes . . . . .	280
1.5.1 Saisie de nouvelles commandes . . . . .	280
1.5.2 Affichage des commandes existantes . . . . .	281

- 2. Gestion avancée des événements.....282
- 3. Activation des fenêtres.....285
  - 3.1 Connexion.....285
  - 3.2 Clients.....291
  - 3.3 Accueil.....292

**Chapitre 10**  
**Connexion**

- 1. Introduction .....297
- 2. Paramètres de connexion .....297
- 3. Mise en service de la fenêtre de connexion.....302
- 4. Lecture des enregistrements.....307

**Chapitre 11**  
**Classes entités**

- 1. Introduction .....315
- 2. Gestion des erreurs .....316
- 3. Classe ModeReglements .....316
- 4. CRUD de la classe ModeReglements .....320
  - 4.1 Créer .....321
  - 4.2 Lire.....322
  - 4.3 Modifier.....323
  - 4.4 Rechercher.....324
  - 4.5 Supprimer .....325
- 5. Classe Client.....326
- 6. CRUD de la classe Client.....331
  - 6.1 Créer .....332
  - 6.2 Lire.....332
  - 6.3 Modifier.....333

# 8 \_\_\_\_\_ Java et Eclipse

Développez une application avec Java et Eclipse

6.4	Supprimer .....	334
6.5	Chercher .....	336
7.	Classe Article .....	338
8.	CRUD de la classe Article .....	343
8.1	Créer .....	344
8.2	Lire .....	345
8.3	Modifier .....	346
8.4	Supprimer .....	347
8.5	Rechercher .....	347
9.	Classe Commande .....	349
10.	CRUD de la classe Commande .....	356
10.1	Créer .....	357
10.2	Lire .....	358
10.3	Modifier .....	360
10.4	Supprimer .....	361
10.5	Rechercher .....	362
11.	Prise en compte des entités par JPA .....	362

## **Chapitre 12** **Modèle MVC**

1.	Présentation du concept .....	365
2.	MVC et JTable .....	367
3.	Gestion des clients .....	368
3.1	Modèle métier .....	368
3.2	Modèle graphique .....	368
3.3	Personnalisation de l'affichage de la table .....	376
3.4	Gestion du simple clic .....	383
3.5	Opérations sur le modèle .....	384
3.5.1	Ajout d'un client .....	384
3.5.2	Modification d'un client .....	390
3.5.3	Suppression d'un client .....	395

- 3.5.4 Recherche d'un client ..... 397
- 3.6 Mise à jour de la table..... 399
  - 3.6.1 Notification ..... 399
  - 3.6.2 Événements ..... 403
  - 3.6.3 Écouteur d'événements ..... 403
- 3.7 Nouveau modèle de table..... 407
- 4. Notifications non graphiques ..... 407
- 5. Structure et sources ..... 412
  - 5.1 ModeleClients..... 413
  - 5.2 ControleClient ..... 417
  - 5.3 PClients..... 420
  - 5.4 PClient..... 427

**Chapitre 13**  
**Application multitable**

- 1. Introduction ..... 433
- 2. Classes associées ..... 433
- 3. Gestion des articles..... 434
  - 3.1 Affichage ..... 434
    - 3.1.1 Création du modèle graphique ..... 435
    - 3.1.2 Création du contrôle ..... 437
  - 3.2 Ajout ..... 443
  - 3.3 Modification ..... 446
    - 3.3.1 Sélection d'un article ..... 446
    - 3.3.2 Sauvegarde de la modification ..... 448
  - 3.4 Suppression..... 449
  - 3.5 Recherche ..... 450
- 4. Gestion des commandes..... 453
  - 4.1 Affichage ..... 453
    - 4.1.1 Création du modèle graphique ..... 454
    - 4.1.2 Création du contrôle ..... 459

4.2	Ajout .....	464
4.3	Modification .....	473
4.4	Suppression .....	474
4.5	Recherche .....	477

## **Chapitre 14** **Application finale**

1.	Impressions et exports .....	479
1.1	Fonctionnement de JasperReports .....	480
1.2	Installation de Jaspersoft Studio .....	481
1.3	Créer un état simple .....	483
1.3.1	Préparation .....	483
1.3.2	Création du rapport .....	489
1.4	Créer un état paramétré .....	495
1.5	Intégration de JasperReports dans Eclipse .....	500
1.6	Mise en œuvre dans le projet .....	500
1.6.1	Aperçu .....	505
1.6.2	Impression du rapport .....	506
1.6.3	Export en PDF .....	506
2.	Graphiques .....	517
2.1	Création avec Jaspersoft Studio et JFreeChart .....	517
2.2	Exploitation des graphiques à partir de l'application .....	522
3.	Création de l'exécutable .....	526
4.	Aller plus loin .....	532
	Index .....	539

## Chapitre 5

# La boîte à outils d'Eclipse

### 1. Propriétés du projet

Eclipse permet de personnaliser les propriétés de chaque projet.

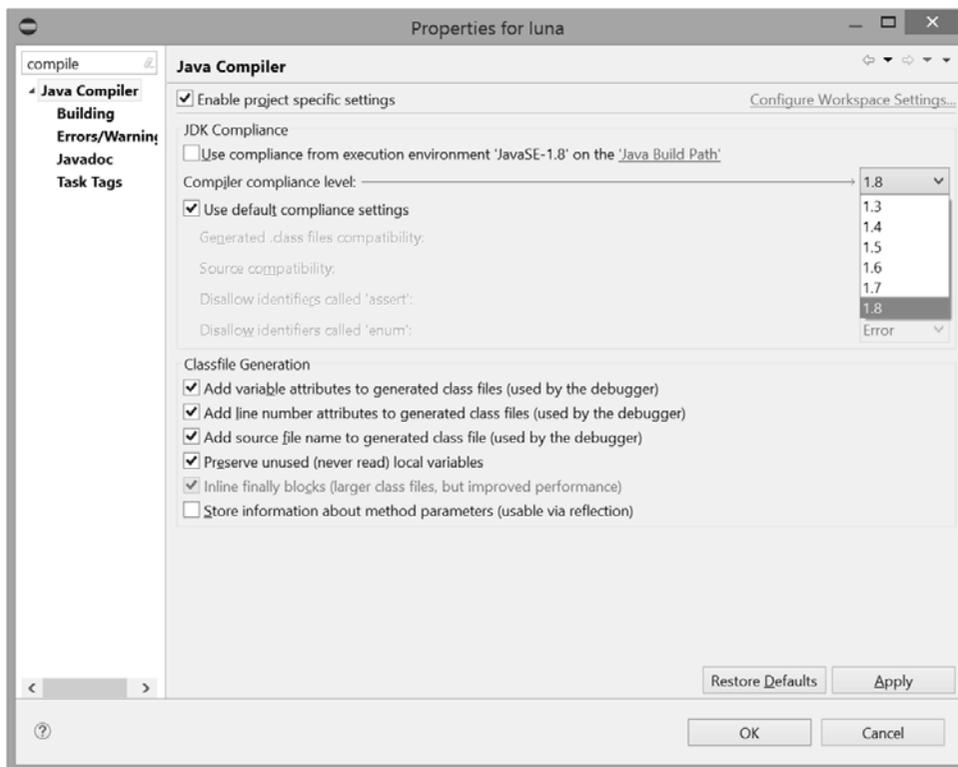
► Faites un clic droit sur le nom du projet puis choisissez l'option **Properties**.

Dans la fenêtre de propriétés, de nombreuses options de personnalisation sont disponibles. Cette fenêtre est particulièrement utile lorsqu'on désire assurer le niveau de compatibilité avec une version du JDK.

#### ■ Remarque

*Pour accélérer la recherche de l'option voulue, et pourvu que vous connaissiez son nom, une saisie rapide vous permet de filtrer les propriétés. Elle se trouve tout en haut à gauche de la boîte de dialogue de propriétés, avec un petit message **type filter text**.*

► Pour changer le compilateur, sélectionnez **Java Compiler**, cochez la case **Enable project specific settings**, décochez la case **Use compliance from execution environment**, puis choisissez la version voulue dans la liste déroulante.



- ▣ Pour personnaliser le code, accédez à l'option **Java Code Style**.

Si vous avez par exemple l'habitude de préfixer l'accès aux propriétés d'une classe avec le mot-clé **this**, cochez l'option **Qualify all generated field accesses with 'this'**.

- ▣ Pour effectuer des actions automatiques à chaque sauvegarde de fichier, accédez à l'option **Java Editor - Save Actions**, et cochez les cases qui vous intéressent.



Il est hors de question de repasser manuellement tous les tests d'une application. Au-delà d'un certain volume de code, cela est tout simplement impossible.

Il est largement préférable de confier ces tests à un système automatique qui les passera tous les jours.

Toute une culture s'est même créée autour de cette manière de développer qui est de placer les tests au centre de l'activité de création logicielle, ou dit autrement le développement piloté par les tests (*Test-Driven Development* ou TDD en anglais).

Ce livre n'est pas une introduction à cette culture, mais quelques principes de bases peuvent d'ores et déjà en être empruntés : par exemple le **Red/Green/Refactor** ou en français : Rouge, Vert, Refaire.

L'idée est de commencer tout d'abord par créer des tests, qui ne passeront pas, et à travailler l'implémentation et le codage jusqu'à ce que ces tests soient corrects. Une fois arrivé à ce point, il faut ensuite retravailler le code produit jusqu'à ce qu'il ait une forme correcte.

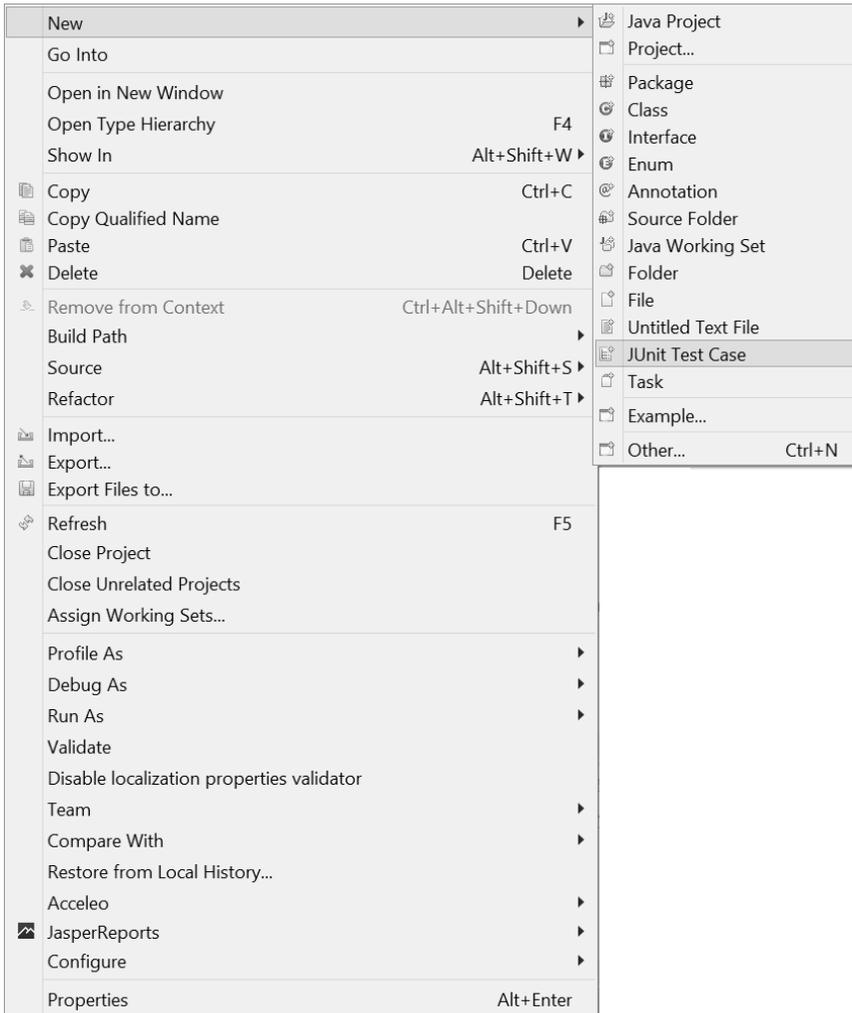
Eclipse propose des outils pour créer des tests de bas niveau, dits **unitaires**.

▣ Revenez au projet **MonPremierProgramme** en faisant un clic droit dessus dans l'explorateur de packages, et choisissez l'option **New - Source Folder**.

▣ Nommez le nouveau dossier **src/test/java**. Cliquez sur **Finish**.

Les classes de tests seront destinées à aller dans ce dossier. Créer ce dossier permet de faire une séparation claire entre les classes de tests unitaires et les classes qui seront utilisées lors du déploiement de l'application.

▣ Refaites un clic droit sur le projet dans l'explorateur de packages, et choisissez l'option **New - JUnit Test Case**.



Une boîte de dialogue s'ouvre permettant de configurer le test.

▣ Choisissez les options suivantes :

**JUnit Test Case**

Select the name of the new JUnit test case. You have the options to specify the class under test and on the next page, to select methods to be tested.

New JUnit 3 test  New JUnit 4 test

Source folder:

Package:

Name:

Superclass:

Which method stubs would you like to create?

setUpBeforeClass()  tearDownAfterClass()  
 setUp()  tearDown()  
 constructor

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

Class under test:

Cette classe teste la classe **MonPremierProgramme**. Elle est située dans le dossier **src/test/java**, et elle a comme package celui de la classe à tester, c'est-à-dire **priseEnMain.premier**.

### ■ Remarque

Une bonne pratique est de terminer le nom des classes de test par **Test**, comme **MonPremierProgrammeTest**.

▣ Cliquez sur **Next**.