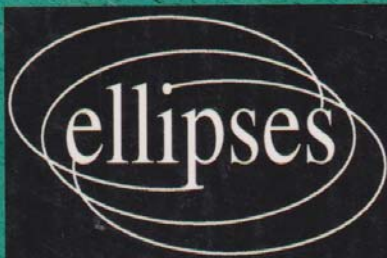


COURS ET EXERCICES

# Principes d'architecture des programmes Java

Valéry Guilhem Frémaux



2-005-550-1

2-005-550-1

COURS & EXERCICES



# Principes d'architecture des programmes Java

Valéry Guilhem FRÉMAUX



# Table des matières

|   |           |
|---|-----------|
| <b>PREMIERE PARTIE CONCEPTS DE BASE DE L'ORGANISATION D'UN PROGRAMME .....</b>            | <b>11</b> |
| <b>I. LES CARACTERISTIQUES SYSTEMIQUES DE LA PROGRAMMATION .....</b>                      | <b>13</b> |
| 1. <i>Le point de vue systémique</i> .....  | 14        |
| a. L'approche systémique .....  | 14        |
| b. La programmation peut-elle être analysée comme un système ? .....                      | 14        |
| c. D'où démarrer ? .....  | 15        |
| 2. <i>Matière et metamatière</i> .....  | 15        |
| a. Données, meta-données, meta-meta-données .....   | 15        |
| b. L'organisation par le besoin d'automatisation .....                                    | 16        |
| c. Les opérations de l'organisation .....   | 17        |
| d. L'organisation comme outil de la complexité .....                                      | 17        |
| e. Les limites de l'organisation .....  | 19        |
| 3. <i>Les grands compromis</i> .....  | 20        |
| a. Mémoire vs. vitesse .....  | 20        |
| b. Organisation vs. productivité .....  | 21        |
| c. Outillage vs. réactivité .....   | 21        |
| <b>II. L'AUTOMATE DE BASE .....</b>   | <b>23</b> |
| 1. <i>Les différents types de projets</i> .....   | 23        |
| a. Trouver son angle de vue .....   | 23        |
| b. Le scripting et l'univers du jetable .....   | 23        |
| c. L'utilitaire ou le programme mono-fonction .....                                       | 24        |
| d. L'application portée ou intégrée .....   | 26        |
| e. L'application autonome .....   | 26        |
| f. Le système .....   | 27        |
| 2. <i>De l'organisation du travail et des grandes pratiques de la programmation</i> ..... | 28        |
| a. Des pratiques opposées. ....   | 28        |
| b. Construire ou récupérer .....  | 29        |
| c. IDE ou programmation de base ? .....   | 30        |

|  |           |
|--|-----------|
| 3. <i>Identification des modes de l'application</i> .....        | 31        |
| a. Du bon usage des méthodologies d'analyse.....                 | 31        |
| b. L'application comme automate.....                             | 32        |
| c. Résolution de la modalité au niveau macroscopique.....        | 35        |
| d. Influences de la modalité sur l'architecture des données..... | 36        |
| 4. <i>Conception des modes</i> .....                             | 37        |
| 5. <i>Les applications complexes</i> .....                       | 38        |
| a. Automates multi-modaux en cascade.....                        | 38        |
| b. Applications multi-planaires.....                             | 39        |
| c. Applications multi-tâches.....                                | 39        |
| III. QUATRE MODELES CLASSIQUES.....                              | 39        |
| 1. <i>Le programme simple</i> .....                              | 40        |
| a. Quand un programme est-il simple ?.....                       | 40        |
| b. Les phases élémentaires d'un programme simple.....            | 41        |
| 2. <i>L'application fenêtrée monodocument</i> .....              | 43        |
| a. La vue.....   | 43        |
| b. Un modèle de données supplémentaire.....                      | 43        |
| c. Le modèle monodocument.....                                   | 45        |
| 3. <i>L'application fenêtrée multi-documents</i> .....           | 45        |
| a. L'arité du problème.....                                      | 45        |
| b. L'application à la problématique multi-documents.....         | 46        |
| c. Les limites pragmatiques.....                                 | 47        |
| d. Le cahier des charges.....                                    | 47        |
| 4. <i>L'architecture client-serveur</i> .....                    | 48        |
| a. Le principe de base.....                                      | 48        |
| b. Serveurs mono-client et multi-clients.....                    | 49        |
| c. Serveurs requête-réponse.....                                 | 50        |
| d. Serveurs transactionnels.....                                 | 51        |
| e. Serveurs protocolaires.....                                   | 52        |
| 5. <i>L'architecture peer-to-peer</i> .....                      | 53        |
| <b>DEUXIEME PARTIE LES PRATIQUES DE BASE EN JAVA.....</b>        | <b>55</b> |
| INTRODUCTION : L'ARCHITECTURE DE BASE DU PROGRAMME.....          | 57        |
| 1. <i>La fonction "main"</i> .....                               | 57        |
| a. La mise en œuvre du contexte d'application.....               | 57        |
| b. Contexte statique et contexte objet.....                      | 57        |
| c. Tactique de démarrage.....                                    | 59        |
| d. Cas de plusieurs fonctions "main".....                        | 60        |
| 2. <i>La boucle principale d'attente</i> .....                   | 60        |
| a. Une boucle pas toujours visible.....                          | 60        |
| b. Cas des programmes simples.....                               | 61        |
| c. Cas des programmes complexes.....                             | 63        |
| d. Programmes à modalité complexe.....                           | 63        |
| e. Multi-modalité multitâche.....                                | 64        |
| 3. <i>Les phases d'initialisation de l'application</i> .....     | 66        |
| a. La création d'une Machine Virtuelle.....                      | 66        |
| b. Le chargement des classes.....                                | 67        |
| c. La recherche d'une référence absolue du système.....          | 67        |
| d. Synthèse sur le fichier de configuration.....                 | 69        |
| e. Le chargement et la distribution des préférences.....         | 70        |
| f. L'amorçage de la boucle principale.....                       | 70        |

|  |     |
|--|-----|
| I. LA SYNTHÈSE OBJET .....   | 71  |
| 1. <i>Le Java Programming Style</i> .....  | 71  |
| a. Français ou anglais .....   | 72  |
| b. Nommage des classes, des interfaces et des paquetages .....                             | 72  |
| c. Nommage des variables, membres, instances d'objets, paramètres .....                    | 74  |
| d. Nommage des méthodes .....  | 75  |
| e. L'influence du modèle JavaBeans sur les membres .....                                   | 75  |
| 2. <i>La stratégie objet</i> .....   | 76  |
| a. La vision "comportementaliste" .....  | 77  |
| b. La vision "administrativiste" .....   | 77  |
| c. Démarche d'analyse .....  | 77  |
| d. Coopération ou dépendance .....   | 78  |
| e. Dérivation et composition .....   | 79  |
| f. Influence sur la performance et la qualité .....  | 81  |
| g. Gestion de l'encapsulation .....  | 82  |
| 3. <i>Ensembles d'objets, collections</i> .....  | 84  |
| a. Notions communes aux ensembles et collections .....                                     | 84  |
| b. Collections non ordonnées non étiquetées .....  | 85  |
| c. Collections ordonnées non étiquetées .....  | 86  |
| d. Collections non ordonnées étiquetées .....  | 90  |
| e. Collections optimisées .....  | 92  |
| f. Backreferencing .....   | 93  |
| g. Généralisation .....  | 94  |
| 4. <i>Collections hétérogènes</i> .....  | 96  |
| a. Opportunité des collections hétérogènes .....   | 96  |
| b. Résolution par les interfaces .....   | 97  |
| c. Récupération des propriétés d'objet par transtypage .....                               | 97  |
| d. Appui des implémentations sur les structures de données de <code>java.util</code> ..... | 98  |
| 5. <i>Composition d'objets</i> .....   | 100 |
| a. Problème d'accès aux objets .....   | 101 |
| b. Transmission des ascendances .....  | 102 |
| c. Ascendances multiples .....   | 104 |
| d. Raccourcis d'accès .....  | 105 |
| e. Backreferencing, crossreferencing, références circulaires .....                         | 107 |
| f. Libération des ressources et finalisation .....   | 108 |
| 6. <i>Questions et exercices</i> .....   | 114 |
| II. LA SYNTHÈSE DES TRAITEMENTS .....  | 115 |
| 1. <i>Le traitement comme unité de base de l'action dynamique d'un programme</i> .....     | 115 |
| 2. <i>Traitements localisés</i> .....  | 116 |
| 3. <i>Traitements globaux</i> .....  | 117 |
| a. Définition .....  | 117 |
| b. Distribution de traitement .....  | 117 |
| c. Propagation de traitement .....   | 118 |
| d. Polymorphisme par abstraction .....   | 119 |
| 4. <i>Traitements délégués</i> .....   | 119 |
| a. Délégation de traitement .....  | 119 |
| b. Le modèle d'écouteur/écouté du JavaBeans .....  | 120 |
| c. La méthode de l'adaptateur .....  | 123 |
| 5. <i>Questions et exercices</i> .....   | 123 |
| III. LA SYNTHÈSE MULTITÂCHE .....  | 124 |
| 1. <i>Les ressources multitâches de Java</i> .....   | 124 |
| a. L'implication du temps réel dans la programmation .....                                 | 124 |

|   |            |
|---|------------|
| b. Tâches vs. processus .....   | 124        |
| c. Le Thread (fil), élément de base du multitâche Java .....                  | 125        |
| 2. <i>L'analyse multitâche</i> .....  | 127        |
| a. Structure d'un Thread, partage de temps d'exécution, priorités .....       | 127        |
| b. L'identification des Threads .....   | 129        |
| c. Appels synchrones, appels asynchrones .....                                | 137        |
| d. Communication entre Threads .....  | 137        |
| 3. <i>Résolution des problèmes sur les Threads</i> .....                      | 141        |
| a. Conflits sur les variables .....   | 141        |
| b. Conflits sur les traitements .....   | 146        |
| c. Résolution par synchronisation .....                                       | 146        |
| 4. <i>Questions et exercices</i> .....  | 148        |
| IV. LA GESTION PAR UNITES DE COMPILATION .....                                | 148        |
| 1. <i>L'édition de liens de Java</i> .....                                    | 148        |
| a. Qu'est-ce qu'une édition de liens .....                                    | 148        |
| b. Comment la machine virtuelle opère .....                                   | 149        |
| 2. <i>Synthèse d'unités indépendantes au sens de l'édition de liens</i> ..... | 151        |
| a. Les contraintes de la modularité .....                                     | 151        |
| b. Compilation indépendante .....   | 152        |
| c. Structure d'une plate-forme modulaire .....                                | 155        |
| d. Cœur de plate-forme et chargement de "Plug-ins" .....                      | 156        |
| V. LE CONTROLE D'ERREUR .....   | 161        |
| 1. <i>De la bonne utilisation des exceptions</i> .....                        | 162        |
| a. L'exception, une erreur attendue .....                                     | 162        |
| b. Lorsque l'exception n'est pas attendue .....                               | 164        |
| c. Synthèse du jeu d'exceptions .....   | 164        |
| d. Traitement minimal d'exception .....                                       | 166        |
| 2. <i>Les erreurs à l'exécution (les plus courantes)</i> .....                | 167        |
| a. Décoder les exception et erreurs .....                                     | 167        |
| b. Cas particulier : NullPointerException .....                               | 168        |
| c. Cas particulier : ArrayOutOfBoundsException .....                          | 168        |
| d. Cas particulier : ClassCastException .....                                 | 169        |
| e. Cas particulier : InstanciationException .....                             | 170        |
| 3. <i>Questions et exercices</i> .....  | 170        |
| <b>TROISIEME PARTIE LES UTILITAIRES ET API INCONTOURNABLES</b> .....          | <b>171</b> |
| I. ORGANISER LE PARAMETRAGE DE L'APPLICATION .....                            | 173        |
| 1. <i>Introduction au paramétrage</i> .....                                   | 173        |
| 2. <i>Paramétrage simple</i> .....  | 173        |
| a. Paramétrage linéaire .....   | 173        |
| b. Paramétrage multi-utilisateur .....  | 175        |
| c. Paramétrage global et paramétrage local .....                              | 177        |
| d. Globalisation progressive .....  | 178        |
| 3. <i>Paramétrage complexe</i> .....  | 180        |
| a. Paramétrage complexe .....   | 180        |
| b. Accès des instances aux paramètres .....                                   | 182        |
| c. Formats des fichiers des configurations complexes .....                    | 183        |
| 4. <i>Schémas administratifs</i> .....  | 183        |
| a. Obtention du schéma .....  | 183        |
| b. Schéma hiérarchique .....  | 184        |
| c. Schéma multidimensionnel .....   | 184        |
| 5. <i>L'usage du paquetage java.util.prefs</i> .....                          | 186        |
| a. Un principe hiérarchique .....   | 186        |

|   |     |
|---|-----|
| b. Import / export des préférences .....                                      | 187 |
| c. Utilisation .....  | 188 |
| d. Applications .....   | 189 |
| 6. <i>Mise en œuvre</i> .....   | 189 |
| II. LA GESTION DES RESSOURCES UTILISEES PAR L'APPLICATION .....               | 190 |
| 1. <i>Le dispositif de gestion de ressources de Java</i> .....                | 190 |
| a. De quelles ressources parle-t-on ? .....                                   | 190 |
| b. Résolution de l'accroissement de matière par l'indirection .....           | 190 |
| c. Les <i>bundles</i> de ressources .....                                     | 191 |
| 2. <i>Gérer l'internationalisation</i> .....                                  | 194 |
| a. L'amplitude de la solution .....   | 194 |
| b. Les Locales .....  | 196 |
| III. GERER UNE FONCTION UNDO/REDO .....                                       | 199 |
| 1. <i>L'usage de la classe java.util.Stack</i> .....                          | 199 |
| 2. <i>Analyse d'une fonction d'historique</i> .....                           | 199 |
| a. La pile undo/redo .....  | 199 |
| b. L'encodage des actions .....   | 200 |
| c. L'annulation de l'annulation : le redo .....                               | 202 |
| d. Les conséquences de la mise en œuvre d'une fonction <i>undo/redo</i> ..... | 202 |
| 3. <i>Mise en œuvre (travaux dirigés)</i> .....                               | 205 |
| a. L'interface Pilotable .....  | 205 |
| b. Le dispatcher .....  | 206 |
| c. La pile d'annulation UndoRedoStack .....                                   | 207 |
| d. La pompe à message .....   | 209 |
| e. La classe d'application de Pilotable .....                                 | 210 |
| f. L'application de test .....  | 212 |
| IV. LA GESTION DES SAUVEGARDES .....  | 213 |
| 1. <i>Approches de la sauvegarde</i> .....                                    | 213 |
| a. Vue d'ensemble .....   | 213 |
| b. Formats de sauvegarde .....  | 214 |
| c. Réduction à un principe .....  | 217 |
| 2. <i>L'infrastructure Java de sérialisation</i> .....                        | 217 |
| a. Présentation générale .....  | 217 |
| b. L'écriture et la lecture d'objets .....                                    | 218 |
| c. Lecture dans un flux d'objets .....  | 219 |
| d. Les interfaces Sérializable et Externalizable .....                        | 219 |
| 3. <i>Organiser les sauvegardes</i> .....                                     | 221 |
| a. Application à sauvegarde unique .....                                      | 221 |
| b. Applications à sauvegarde libre .....                                      | 221 |
| c. Sauvegardes contextuelles .....  | 222 |
| d. Sauvegardes modulaires .....   | 222 |
| 4. <i>Mise en œuvre</i> .....   | 224 |
| V. GESTION DE TRACE .....   | 224 |
| 1. <i>Analyse</i> .....   | 224 |
| a. Intérêt de la mise en place des traces .....                               | 224 |
| b. Traces plates et indentées .....   | 225 |
| c. Traces monotâches et multitâches .....                                     | 226 |
| d. Sélectivité des informations de trace .....                                | 228 |
| 2. <i>Mise en œuvre</i> .....   | 228 |
| VI. LA GESTION DE LA JOURNALISATION .....                                     | 239 |
| 1. <i>Architectures de journalisation</i> .....                               | 239 |
| a. Trace vs. Journalisation .....   | 239 |

|   |            |
|---|------------|
| b. Journalisation simple.....                                     | 239        |
| c. Journalisation transactionnelle .....                          | 241        |
| d. Journalisation multi-thème, localisée .....                    | 242        |
| e. Journalisation hiérarchique.....                               | 244        |
| 2. <i>Constituer une journalisation</i> .....                     | 245        |
| a. Définir les actions journalisables.....                        | 245        |
| b. Constituer les formats d'enregistrement de journalisation..... | 245        |
| 3. <i>La solution Java</i> .....                                  | 246        |
| a. Le paquetage java.util.logging.....                            | 246        |
| b. Le Logger.....   | 247        |
| c. Filtrage et sélection des messages de journalisation.....      | 249        |
| d. Formatage et localisation des messages .....                   | 249        |
| 4. <i>Mise en œuvre</i> .....                                     | 250        |
| VII. CONCLUSION .....   | 254        |
| <br>  |            |
| <b>CORRIGÉS DES QUESTIONS ET EXERCICES .....</b>                  | <b>255</b> |
| <br>  |            |
| <b>LEXIQUE .....</b>  | <b>271</b> |
| <br>  |            |
| <b>TABLE DES ILLUSTRATIONS .....</b>                              | <b>277</b> |
| <br>  |            |
| <b>INDEX.....</b>   | <b>281</b> |