

Table des matières

Introduction	9
Chapitre 1. Introduction aux Threads en Java	11
1.1. Processus <i>versus</i> Thread	11
1.2. Programmation concurrente	12
1.3. Création de Threads	13
1.4. Type de Threads	14
1.5. Monotâche <i>versus</i> multitâche	15
1.6. Différents états d'un Thread	21
1.7. Cycle de vie d'un Thread	21
1.8. Quelques notions sur les Threads	24
1.8.1. Deux Threads sans sleep	24
1.8.2. Répartition de temps entre deux Threads	25
1.8.3. Priorité entre Threads	27
1.9. Programmer une tâche : Timer et TimerTask	29
1.9.1. En précisant un délai initial	29
1.9.2. Avec un délai initial et une périodicité	31
Chapitre 2. Synchronisation des Threads	33
2.1. Synchronisation sur terminaison : méthode join()	33
2.2. Ressource en exclusion mutuelle : modificateur « synchronized »	35
2.3. Variables partagées : classe interne	38
2.4. Problème de l'exclusion mutuelle	40
2.5. Bloc « synchronized »	42

2.6. Méthode d'instance synchronisée.	47
2.7. Variables partagées : variable de classe	49
2.8. Synchronisation entre Threads	51
2.8.1. wait et notifyAll.	51
2.8.2. wait et notify.	53
2.9. Pattern classique producteur-consommateur	57
2.10. Sémaphore en Java.	60
2.10.1. Avant Java 1.5	61
2.10.2. À partir de Java 1.5	63

Chapitre 3. Systèmes temps réel et Java temps réel 67

3.1. Systèmes temps réel.	67
3.1.1. Définition.	67
3.1.2. Exemples de systèmes d'exploitation temps réel	67
3.1.3. Types de temps réel.	68
3.1.3.1. Temps réel souple	68
3.1.3.2. Temps réel ferme	68
3.1.3.3. Temps réel dur	68
3.1.4. Architecture	68
3.1.4.1. Architecture monotâche	68
3.1.4.2. Architecture multitâche	69
3.1.5. Ordonnancement des tâches avec les priorités	69
3.2. Java temps réel.	70
3.2.1. RTSJ (<i>Real-Time Specification for Java</i>)	71
3.2.2. Implémentations.	73

Chapitre 4. Programmation distribuée en Java 77

4.1. Définition d'une application répartie	77
4.2. Communication dans une application répartie	78
4.2.1. Communication bas niveau : socket	78
4.2.1.1. Définition	78
4.2.1.2. Modes de communication.	78
4.2.1.3. Exemple de communication	82
4.2.1.4. Écriture directe dans le stream d'un socket	86
4.2.1.5. Lecture directe du stream d'un socket	86
4.2.1.6. Lecture et écriture de plus haut niveau	87
4.2.1.7. Transmission d'objet par les sockets	89
4.2.1.8. Communications entre un applet Java et un serveur à l'aide des sockets.	91

4.2.2. Communication haut niveau : Middleware	94
4.2.2.1. Introduction à RMI.	96
4.2.2.2. Architecture de RMI.	98
4.2.2.3. Déploiement de RMI.	100
4.2.2.4. Passage des paramètres dans RMI.	108
4.2.2.5. Gestion de la sécurité dans RMI.	117
4.2.2.6. Chargement dynamique des classes.	120
4.2.2.7. Asynchronisme dans RMI : méthodes callbacks	122
4.2.2.8. Ramasse-miettes distribué dans RMI	126
 Annexe	 129
 Liste des acronymes	 155
 Bibliographie	 157
 Index	 159