

Inhaltsverzeichnis

1	Einleitung	1
1.1	Parallelität	1
1.2	Vorteile	2
1.3	Konzepte der parallelen Programmierung	3
1.4	Probleme der parallelen Programmierung	4
1.5	Paralleles Java-Hamster-Modell	5
1.6	Weitere Parallelitätskonzepte in Java	6
1.7	Anmerkungen zum Buch	7
1.8	Aufbau des Buches	7
2	Das Hamster-Modell	11
2.1	Komponenten des Hamster-Modells	12
2.2	Der Hamster-Simulator	13
2.3	Vordefinierte Klassen des Hamster-Modells	14
2.4	Objektorientierte Hamster-Programme	20
2.5	Die Klasse <code>AllroundHamster</code>	21
3	Selbstständige Hamster	23
3.1	Passive Hamster	23
3.2	Selbstständige Hamster	26
3.3	Parallele objektorientierte Hamster-Programme	33
3.4	Beispielprogramme	34
3.5	Aufgaben	39
4	Threads	43
4.1	Was sind Threads?	43
4.2	Definition von Thread-Klassen	45
4.3	Erzeugung und Start von Threads	47
4.4	Ende von Threads	52
4.5	Runnables	53
4.6	Thread-Namen	56
4.7	Vergleich von Threads und Thread-Objekten	57
4.8	Selbstständige und unselbstständige Hamster	60
4.9	Nutzen und Einsatzbereiche von Threads	63
4.10	Beispielprogramme	65
4.11	Aufgaben	80
5	Scheduling	85
5.1	Scheduling-Strategien in Java	85
5.2	Thread-Zustände	86
5.3	Prioritäten	91

5.4	yield	93
5.5	sleep	93
5.6	join	95
5.7	Unterbrechungen	98
5.8	Dämonen	100
5.9	Zeitschaltuhren	101
5.10	Beispielprogramme	107
5.11	Aufgaben	116
6	Kommunikation zwischen Threads	121
6.1	Zwischenhamsterliche Beziehungen	121
6.2	Kommunikation über gemeinsame Variablen	124
6.3	Kommunikation über gemeinsame Objekte	128
6.4	Kommunikation über Methoden	133
6.5	Beispielprogramme	137
6.6	Aufgaben	147
7	Mehrseitige Synchronisation	149
7.1	Synchronisationsprobleme	150
7.2	Synchronisationsarten	155
7.3	synchronized-Anweisung	157
7.4	synchronized-Methoden	163
7.5	Compileroptimierungen	168
7.6	Granularität der Synchronisation	172
7.7	Allgemein gültige Regeln zur Synchronisation	174
7.8	Beispielprogramme	175
7.9	Aufgaben	181
8	Einseitige Synchronisation	187
8.1	Aktives Warten	187
8.2	Verwendung von sleep	189
8.3	Verwendung von sleep und interrupt	190
8.4	wait und notify	192
8.5	notifyAll	196
8.6	Monitore	206
8.7	Beispielprogramme	207
8.8	Zusammenfassung	217
8.9	Aufgaben	218
9	Verklemmungen	223
9.1	Motivation	223
9.2	Definition	226
9.3	Deadlockverhinderung	228
9.4	Deadlockvermeidung	233
9.5	Deadlockerkennung und -beseitigung	236
9.6	Livelocks	241
9.7	Beispielprogramme	244
9.8	Aufgaben	253

10	Beendigung von Threads	257
10.1	Methode <code>stop</code> der Klasse <code>Thread</code>	258
10.2	Einsatz von Kommunikationsmechanismen	260
10.3	Unterbrechungen	261
10.4	<code>System.exit</code>	263
10.5	Dämonen	264
10.6	Beispielprogramme	267
10.7	Aufgaben	275
11	Klassische Probleme der parallelen Programmierung	279
11.1	Leser-Schreiber-Problem	279
11.2	Erzeuger-Verbraucher-Problem	286
11.3	Philosophen-Problem	290
11.4	Kommunikation über Nachrichten	307
11.5	Aufgaben	319
12	Sperren	327
12.1	Semaphore	327
12.2	Explizite Sperren	338
12.3	Schlossvariablen	343
12.4	Beispielprogramme	361
12.5	Aufgaben	375
13	Zugabe	381
13.1	Atomare Variablen	381
13.2	Queues	384
13.3	Exchanger	391
13.4	Countdown	392
13.5	Sammelpunkte	395
13.6	Thread-Pooling	397
13.7	FutureTasks	402
13.8	Schlussbemerkung	405
13.9	Aufgaben	406
A	Basisklassen des JDK zum Umgang mit Threads	411
A.1	Interface <code>java.lang.Runnable</code>	411
A.2	Klasse <code>java.lang.Thread</code>	411
A.3	Klasse <code>java.lang.Object</code>	418
A.4	Klasse <code>java.lang.InterruptedException</code>	420
A.5	Klasse <code>java.lang.IllegalStateException</code>	420
A.6	Klasse <code>java.lang.IllegalArgumentException</code>	420
A.7	Klasse <code>java.lang.IllegalMonitorStateException</code>	421
B	Klassen des Java-Hamster-Modells	423
B.1	Exception-Klassen	423
B.2	Die Klasse <code>Hamster</code>	426
B.3	Die Klasse <code>Territorium</code>	432
B.4	Klasse <code>AllroundHamster</code>	434

Glossar	441
Literaturverzeichnis	445
Sachverzeichnis	447