

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>15</b>
1.1	Parallelität, Nebenläufigkeit und Verteilung	15
1.2	Programme, Prozesse und Threads	16
<b>2</b>	<b>Grundlegende Synchronisationskonzepte in Java</b>	<b>20</b>
2.1	Erzeugung und Start von Java-Threads	20
2.1.1	Ableiten der Klasse Thread	20
2.1.2	Implementieren der Schnittstelle Runnable	22
2.1.3	Einige Beispiele	25
2.2	Probleme beim Zugriff auf gemeinsam genutzte Objekte	31
2.2.1	Erster Lösungsversuch	35
2.2.2	Zweiter Lösungsversuch	36
2.3	Synchronized und volatile	38
2.3.1	Synchronized-Methoden	38
2.3.2	Synchronized-Blöcke	39
2.3.3	Wirkung von synchronized	41
2.3.4	Notwendigkeit von synchronized	42
2.3.5	Volatile	43
2.3.6	Regel für die Nutzung von synchronized	44
2.4	Ende von Java-Threads	45
2.4.1	Asynchrone Beauftragung mit Abfragen der Ergebnisse	46
2.4.2	Zwangswises Beenden von Threads	52
2.4.3	Asynchrone Beauftragung mit befristetem Warten	57
2.4.4	Asynchrone Beauftragung mit Rückruf (Callback)	59
2.4.5	Asynchrone Beauftragung mit Rekursion	62
2.5	Wait und notify	65
2.5.1	Erster Lösungsversuch	66
2.5.2	Zweiter Lösungsversuch	67
2.5.3	Dritter Lösungsversuch	68
2.5.4	Korrekte und effiziente Lösung mit wait und notify	69

2.6	NotifyAll .....	77
2.6.1	Erzeuger-Verbraucher-Problem mit wait und notify .....	78
2.6.2	Erzeuger-Verbraucher-Problem mit wait und notifyAll .....	82
2.6.3	Faires Parkhaus mit wait und notifyAll .....	84
2.7	Prioritäten von Threads .....	86
2.8	Thread-Gruppen .....	93
2.9	Vordergrund- und Hintergrund-Threads .....	98
2.10	Weitere „gute“ und „schlechte“ Thread-Methoden .....	99
2.11	Thread-lokale Daten .....	101
2.12	Zusammenfassung .....	103
<b>3</b>	<b>Fortgeschrittene Synchronisationskonzepte in Java .....</b>	<b>108</b>
3.1	Semaphore .....	109
3.1.1	Einfache Semaphore .....	109
3.1.2	Einfache Semaphore für den gegenseitigen Ausschluss .....	110
3.1.3	Einfache Semaphore zur Herstellung vorgegebener Ausführungsreihenfolgen .....	112
3.1.4	Additive Semaphore .....	115
3.1.5	Semaphorgruppen .....	118
3.2	Message Queues .....	121
3.2.1	Verallgemeinerung des Erzeuger-Verbraucher-Problems .....	121
3.2.2	Übertragung des erweiterten Erzeuger-Verbraucher-Problems auf Message Queues .....	123
3.3	Pipes .....	126
3.4	Philosophen-Problem .....	129
3.4.1	Lösung mit synchronized - wait - notifyAll .....	130
3.4.2	Naive Lösung mit einfachen Semaphoren .....	132
3.4.3	Einschränkende Lösung mit gegenseitigem Ausschluss .....	134
3.4.4	Gute Lösung mit einfachen Semaphoren .....	135
3.4.5	Lösung mit Semaphorgruppen .....	138
3.5	Leser-Schreiber-Problem .....	140
3.5.1	Lösung mit synchronized - wait - notifyAll .....	141
3.5.2	Lösung mit additiven Semaphoren .....	145
3.6	Schablonen zur Nutzung der Synchronisationsprimitive und Konsistenzbetrachtungen .....	146
3.7	Concurrent-Klassenbibliothek aus Java 5 .....	150
3.7.1	Executors .....	151
3.7.2	Locks und Conditions .....	157
3.7.3	Atomic-Klassen .....	165
3.7.4	Synchronisationsklassen .....	169
3.7.5	Queues .....	172

3.8	Das Fork-Join-Framework von Java 7 .....	173
3.8.1	Grenzen von ThreadPoolExecutor .....	173
3.8.2	ForkJoinPool und RecursiveTask .....	175
3.8.3	Beispiel zur Nutzung des Fork-Join-Frameworks .....	177
3.9	Das Data-Streaming-Framework von Java 8 .....	180
3.9.1	Einleitendes Beispiel .....	180
3.9.2	Sequenzielles Data-Streaming .....	182
3.9.3	Paralleles Data-Streaming .....	186
3.10	Die Completable Futures von Java 8 .....	187
3.11	Ursachen für Verklemmungen .....	194
3.11.1	Beispiele für Verklemmungen mit synchronized .....	195
3.11.2	Beispiele für Verklemmungen mit Semaphoren .....	198
3.11.3	Bedingungen für das Eintreten von Verklemmungen .....	199
3.12	Vermeidung von Verklemmungen .....	200
3.12.1	Anforderung von Betriebsmitteln „auf einen Schlag“ .....	203
3.12.2	Anforderung von Betriebsmitteln gemäß einer vorgegebenen Ordnung .....	204
3.12.3	Weitere Verfahren .....	205
3.13	Zusammenfassung .....	207
<b>4</b>	<b>Parallelität und grafische Benutzeroberflächen .....</b>	<b>209</b>
4.1	Einführung in die Programmierung grafischer Benutzeroberflächen mit JavaFX .....	210
4.1.1	Allgemeines zu grafischen Benutzeroberflächen .....	210
4.1.2	Erstes JavaFX-Beispiel .....	211
4.1.3	Ereignisbehandlung .....	212
4.2	Properties, Bindings und JavaFX-Collections .....	216
4.2.1	Properties .....	216
4.2.2	Bindings .....	219
4.2.3	JavaFX-Collections .....	221
4.3	Elemente von JavaFX .....	221
4.3.1	Container .....	221
4.3.2	Interaktionselemente .....	224
4.3.3	Grafikprogrammierung .....	226
4.3.4	Weitere Funktionen von JavaFX .....	232
4.4	MVP .....	233
4.4.1	Prinzip von MVP .....	234
4.4.2	Beispiel zu MVP .....	235
4.5	Threads und JavaFX .....	242
4.5.1	Threads für JavaFX .....	242
4.5.2	Länger dauernde Ereignisbehandlungen .....	243

4.5.3	Beispiel Stoppuhr .....	248
4.5.4	Tasks und Services in JavaFX .....	254
4.6	Zusammenfassung .....	263
<b>5</b>	<b>Verteilte Anwendungen mit Sockets .....</b>	<b>264</b>
5.1	Einführung in das Themengebiet der Rechnernetze .....	265
5.1.1	Schichtenmodell .....	265
5.1.2	IP-Adressen und DNS-Namen .....	269
5.1.3	Das Transportprotokoll UDP .....	270
5.1.4	Das Transportprotokoll TCP .....	271
5.2	Socket-Schnittstelle .....	272
5.2.1	Socket-Schnittstelle zu UDP .....	272
5.2.2	Socket-Schnittstelle zu TCP .....	274
5.2.3	Socket-Schnittstelle für Java .....	276
5.3	Kommunikation über UDP mit Java-Sockets .....	277
5.4	Multicast-Kommunikation mit Java-Sockets .....	286
5.5	Kommunikation über TCP mit Java-Sockets .....	290
5.6	Sequenzielle und parallele Server .....	300
5.6.1	TCP-Server mit dynamischer Parallelität .....	301
5.6.2	TCP-Server mit statischer Parallelität .....	305
5.6.3	Sequenzieller, „verzahnt“ arbeitender TCP-Server .....	310
5.7	Zusammenfassung .....	314
<b>6</b>	<b>Verteilte Anwendungen mit RMI .....</b>	<b>315</b>
6.1	Prinzip von RMI .....	315
6.2	Einführendes RMI-Beispiel .....	318
6.2.1	Basisprogramm .....	318
6.2.2	RMI-Client mit grafischer Benutzeroberfläche .....	322
6.2.3	RMI-Registry .....	327
6.3	Parallelität bei RMI-Methodenaufrufen .....	331
6.4	Wertübergabe für Parameter und Rückgabewerte .....	335
6.4.1	Serialisierung und Deserialisierung von Objekten .....	336
6.4.2	Serialisierung und Deserialisierung bei RMI .....	341
6.5	Referenzübergabe für Parameter und Rückgabewerte .....	345
6.6	Transformation lokaler in verteilte Anwendungen .....	360
6.6.1	Rechnergrenzen überschreitende Synchronisation mit RMI .....	360
6.6.2	Asynchrone Kommunikation mit RMI .....	363
6.6.3	Verteilte MVP-Anwendungen mit RMI .....	364
6.7	Dynamisches Umschalten zwischen Wert- und Referenzübergabe - Migration von Objekten .....	365
6.7.1	Das Exportieren und „Unexportieren“ von Objekten .....	365

6.7.2	Migration von Objekten .....	368
6.7.3	Eintrag eines Nicht-Stub-Objekts in die RMI-Registry .....	375
6.8	Laden von Klassen über das Netz .....	376
6.9	Realisierung von Stubs und Skeletons .....	377
6.9.1	Realisierung von Skeletons .....	378
6.9.2	Realisierung von Stubs .....	378
6.10	Verschiedenes .....	381
6.11	Zusammenfassung .....	382
<b>7</b>	<b>Webbasierte Anwendungen mit Servlets und JSF .....</b>	<b>383</b>
7.1	HTTP und HTML .....	384
7.1.1	GET .....	384
7.1.2	Formulare in HTML .....	387
7.1.3	POST .....	389
7.1.4	Format von HTTP-Anfragen und -Antworten .....	390
7.2	Einführende Servlet-Beispiele .....	391
7.2.1	Allgemeine Vorgehensweise .....	391
7.2.2	Erstes Servlet-Beispiel .....	392
7.2.3	Zugriff auf Formulardaten .....	394
7.2.4	Zugriff auf die Daten der HTTP-Anfrage und -Antwort .....	396
7.3	Parallelität bei Servlets .....	397
7.3.1	Demonstration der Parallelität von Servlets .....	397
7.3.2	Paralleler Zugriff auf Daten .....	399
7.3.3	Anwendungsglobale Daten .....	403
7.4	Sessions und Cookies .....	406
7.4.1	Sessions .....	407
7.4.2	Realisierung von Sessions mit Cookies .....	411
7.4.3	Direkter Zugriff auf Cookies .....	413
7.4.4	Servlets mit länger dauernden Aufträgen .....	414
7.5	Asynchrone Servlets .....	420
7.6	Filter .....	424
7.7	Übertragung von Dateien mit Servlets .....	425
7.7.1	Herunterladen von Dateien .....	425
7.7.2	Hochladen von Dateien .....	428
7.8	JSF (Java Server Faces) .....	431
7.8.1	Einführendes Beispiel .....	431
7.8.2	Managed Beans und deren Scopes .....	438
7.8.3	MVP-Prinzip mit JSF .....	442
7.8.4	AJAX mit JSF .....	443
7.9	RESTful WebServices .....	447
7.9.1	Definition von RESTful WebServices .....	448

7.9.2 JSON .....	449
7.9.3 Beispiel .....	451
7.10 WebSockets .....	456
7.11 Zusammenfassung .....	460
<b>Literatur .....</b>	<b>463</b>
<b>Index .....</b>	<b>465</b>