

Inhalt

	Vorwort	13
1	Einleitung	21
1.1	Elementare Konzepte und Begriffe	25
1.2	Funktionale Programmierung in Java	31
1.3	Kotlin als Sprache für eine funktionale Programmierung	31
Teil I	Grundlagen der funktionalen Programmierung in Java	33
2	Sprachliche Grundlagen von Java	35
2.1	Generics	35
2.1.1	Typparameter	35
2.1.2	Typconstraints	36
2.1.3	Ko- und Kontravarianz	37
2.1.4	Typinferenz bei Generics	41
2.1.5	Schwachstellen der Generics in Java	43
2.2	Default-Methoden	44
2.3	Lambda-Ausdrücke	46
2.3.1	Funktionale Interfaces	53
2.3.2	Methodenreferenzen	56
2.4	Record-Klassen	58
2.5	Switch-Ausdrücke	60
2.6	Zusammenfassung	62
3	Programmieren ohne Seiteneffekte	63
3.1	Reine Funktionen	63
3.1.1	Iteration vs. Rekursion	66
3.1.2	Referenzielle Transparenz und Ersetzungsprinzip	66
3.2	Funktionale Ausnahmebehandlung mit Optional	70
3.3	Zusammenfassung	72

4	Funktionale Datenstrukturen	73
4.1	Algebraische Datentypen	73
4.2	Pattern Matching	75
4.3	Paare und Tupel	81
4.4	Funktionale Listen	82
4.5	Persistente Collections	89
4.6	Zusammenfassung	93
Teil II	Gestaltungsprinzipien und Entwurfsmuster	95
5	Programmieren mit Funktionsparametern	97
5.1	Höhere Funktionen für Collections	98
5.1.1	Höhere Funktionen bei persistenten Collections	100
5.2	Flexible Programmschnittstellen	101
5.3	Algorithmen	105
5.3.1	Tiefensuche	105
5.3.2	Verallgemeinerung der Suche	108
5.4	Entwurfsmuster	110
5.4.1	Strategie	110
5.4.2	Kommando	111
5.5	Eingebettete und bedingte Ausführung	115
5.5.1	Eingebetteter Code	115
5.5.2	Bedingte Ausführung	116
5.6	Auswertung nach Bedarf	119
5.6.1	Faule Iteratoren	119
5.6.2	Unendliche Folgen	121
5.6.3	Faule Iteration über die Knoten eines Graphen	122
5.7	Zusammenfassung	123
6	Reduktion mit Monoiden	125
6.1	Monoide	125
6.2	Reduktion	126
6.3	Monoide in Java	127
6.4	Reduzierbare Strukturen	129
6.4.1	Reducible	130
6.4.2	Abgeleitete Operationen	133
6.5	Heben von Monoiden auf Container-Strukturen	134
6.5.1	Reduktion von Optional-Werten	134
6.5.2	Reduktion von Map-Einträgen	135
6.6	Parallele Reduktion	136
6.7	Zusammenfassung	140

Inhalt	7
7 Funktionsketten mit Monaden	141
7.1 Flüssige Schnittstellen	141
7.2 Funktoren	142
7.2.1 Funktor Optional	144
7.2.2 Gesetze und Eigenschaften	145
7.3 Monaden	146
7.3.1 Monade Optional	149
7.3.2 Gesetze	150
7.3.3 Bedeutung von Monaden	151
7.3.4 MonadPlus: Monade mit monoider Kombination	159
7.4 Zusammenfassung	159
8 Funktionskomposition	161
8.1 Funktionskomposition bei funktionalen Interfaces	161
8.1.1 Funktionskomposition mit Function	161
8.1.2 Logische Verknüpfungen bei Predicate	163
8.1.3 Bilden von Vergleichsketten mit Comparator	164
8.2 Generatoren für Zufallswerte	167
8.2.1 Monade Gen<A>	168
8.2.2 Erzeugen von Generatoren mit map	169
8.2.3 Generatoren für Listen und Wörter	170
8.2.4 Unendliche Iteratoren für Zufallswerte	171
8.3 Kombinator-Parser	173
8.3.1 Parser und Parser-Ergebnisse	173
8.3.2 Monade Parser	175
8.3.3 Kombinationsoperatoren	178
8.3.4 Parser für arithmetische Ausdrücke	182
8.4 Zusammenfassung	185
Teil III Funktionale Systeme und Bibliotheken	187
9 Streams	189
9.1 Grundlagen von Streams	189
9.1.1 Ein erstes Beispiel	190
9.1.2 Externe vs. interne Iteration	190
9.1.3 Bedarfsauswertung	191
9.2 Klassen von Streams	194
9.3 Stream-Operationen	196
9.3.1 Erzeugeroperationen	197
9.3.2 Zwischenoperationen	200
9.3.3 Terminaloperationen	205

9.4	Collectors	209
9.4.1	Interface Collector	209
9.4.2	Vordefinierte Collectors	211
9.4.3	Downstream Collectors	213
9.4.4	Eine eigene Collector-Implementierung	215
9.5	Anwendungsbeispiele	217
9.5.1	Ergebnisauswertung mit Streams	217
9.5.2	Wortindex zu einem Text	221
9.6	Hinweise	223
9.6.1	Einmal-Iteration	223
9.6.2	Begrenzung von unendlichen Streams	224
9.6.3	Zustandslose und zustandsbehaftete Operationen	224
9.6.4	Reihenfolge von Operationen	225
9.6.5	Kombinationen von Operationen	226
9.7	Interne Implementierung	227
9.7.1	Beispiel	229
9.8	Zusammenfassung	230
10	Parallele Streams	231
10.1	Erzeugen von parallelen Streams	231
10.2	Parallele Ausführung	232
10.2.1	Splitterators	235
10.2.2	Ausführung mit Fork/Join-Framework	237
10.3	Bedingungen bei paralleler Ausführung	239
10.3.1	Parallele Ausführung und Seiteneffekte	239
10.3.2	Eigenschaften der Parameter von reduce	240
10.3.3	Paralleles Sammeln	241
10.4	Laufzeit	243
10.5	Zusammenfassung	245
11	Asynchrone Funktionsketten	247
11.1	Asynchrone Lösung mit Futures	248
11.2	CompletableFuture	251
11.3	Kombination von CompletableFuture	252
11.4	Beispiel Flugsuche	254
11.5	Zusammenfassung	256

12	Reaktive Streams	257
12.1	Grundlagen	258
12.1.1	Kontrakt von Observable	263
12.1.2	Erzeugen von Observables	264
12.1.3	Anmelden und Abmelden von Observer	266
12.2	Varianten	268
12.2.1	Single	268
12.2.2	Completable	270
12.2.3	Maybe	271
12.3	Hot und Cold Observables	273
12.3.1	ConnectableObservable	274
12.3.2	Beispiel Echtzeitdaten	275
12.4	Operationen	277
12.4.1	Abbildungen	279
12.4.2	Filtern und Teilmengen	280
12.4.3	Reduktion und Suche	281
12.4.4	Sammeln	282
12.4.5	Operationen mit Zeit	285
12.4.6	Kombinationen	286
12.4.7	Konvertierungen	287
12.4.8	Seiteneffekte	288
12.5	Asynchrone Ausführung	288
12.5.1	Serialisierung von nebenläufigen Ereignissen	289
12.5.2	subscribeOn und Scheduler	290
12.5.3	observeOn	291
12.6	Fehlerbehandlung	292
12.6.1	Fehlerereignisse auslösen	292
12.6.2	Auf Fehler reagieren	294
12.7	Rückstau und Flusskontrolle	295
12.7.1	Reduktion der Menge der Ereignisse	296
12.7.2	Flowables	297
12.8	Testen reaktiver Streams	299
12.9	Zusammenfassung	301
13	Testen mit und von Funktionen	303
13.1	Funktionsparameter bei JUnit 5	303
13.2	AssertJ: Eine DSL für Unit-Tests	306
13.3	Eigenschaftsbasiertes Testen nach QuickCheck	307
13.3.1	Tests	309
13.3.2	Shrinken der Werte	312
13.4	Zusammenfassung	315

Teil IV Funktionale Programmierung in Kotlin	317
14 Einführung in die Sprache Kotlin	319
14.1 Dateien, Packages und Deklarationen	319
14.2 Basisdatentypen und Arrays	320
14.3 Funktionen	322
14.4 Variablen, Ausdrücke und Kontrollstrukturen	324
14.4.1 Variablen	324
14.4.2 Kontrollstrukturen	324
14.5 Klassen, Interfaces und Objekte	327
14.5.1 Klassen	327
14.5.2 Interfaces	334
14.5.3 Objekte	334
14.5.4 Companion-Objekte	335
14.5.5 Spezielle Arten von Klassen	336
14.6 Das Typsystem von Kotlin	337
14.6.1 Verband der Typen und Typinferenz	338
14.6.2 Nullable- und Non-Nullable-Typen	340
14.6.3 Typtest und Typecasts	341
14.7 Generics	343
14.7.1 Typparameter	343
14.7.2 Typconstraints	344
14.7.3 Ko- und Kontravarianz	346
14.7.4 Star-Projektion	349
14.7.5 Reifizierte Typparameter	350
14.8 Erweiterungsfunktionen und Erweiterungsproperties	352
14.9 Delegates	353
14.9.1 Property-Delegates	353
14.9.2 Klassen-Delegates	355
14.10 Zusammenfassung	356
15 Funktionale Datenstrukturen in Kotlin	357
15.1 Datenklassen	357
15.2 Paare und Tripel	358
15.3 Algebraische Datentypen mit Datenklassen	360
15.4 Pattern Matching	361
15.5 Funktionale Liste in Kotlin	363
15.6 Collections in Kotlin	365
15.7 Zusammenfassung	369

16	Lambda-Ausdrücke und höhere Funktionen	371
16.1	Formen von Lambda-Ausdrücken	373
16.2	Funktionsreferenzen und anonyme Funktionen	376
16.3	Closures	377
16.4	Inlining Lambda-Ausdrücke	378
16.5	Java-Interoperabilität und SAM-Interfaces	379
16.6	Höhere Funktionen für FList	381
16.7	Zusammenfassung	383
17	Höhere Funktionen für Collections	385
17.1	Einführung	385
17.2	Abbilden, Filtern, Sortieren, Teilmengen	389
17.3	Reduktion	391
17.4	Suche und Quantoren	392
17.5	Bilden von Maps	393
17.6	Gruppieren und Aufteilen	395
17.7	Weitere Operationen	397
17.8	Sequences	398
17.9	Zusammenfassung	399
18	Lambda-Ausdrücke mit Empfänger	401
18.1	Grundlagen	401
18.2	Varianten und Kombinationen	404
18.3	Scope-Funktionen	406
18.4	Hierarchische Builder	408
	18.4.1 Ein Builder für Trees	409
	18.4.2 Ein Builder für hierarchische Dokumente	410
18.5	Zusammenfassung	414
19	Domänenspezifische Sprachen	415
19.1	Fallbeispiel physikalische Einheiten	416
19.2	Fallbeispiel Parser-DSL	421
	19.2.1 Parser und Result	421
	19.2.2 Funktionen und Kombinationsoperatoren	422
	19.2.3 DSL für Parser-Spezifikationen	424
19.3	Fallbeispiel Zustandsmaschinen	428
19.4	Zusammenfassung	434

20	Gestaltung von Systemen und Frameworks	435
20.1	Koroutinen	435
20.1.1	Suspend-Funktionen und Koroutinen	436
20.1.2	Strukturierte Nebenläufigkeit	440
20.2	Flows	441
20.3	Jetpack Compose	445
20.4	Ktor	450
20.5	Zusammenfassung	453
A	Bibliografie	455
B	Laufzeitexperimente Parallele Streams	461
B.1	Experimentelle Anordnungen	461
B.2	Experimente	462
	Index	473