Inhaltsverzeichnis

1 Einleitung ................................................................. 1
  1.1 Über dieses Buch .................................................. 1
     1.1.1 Motivation ..................................................... 1
     1.1.2 Was leistet dieses Buch und was nicht? ................. 2
     1.1.3 Wie und was soll mithilfe des Buchs gelernt werden? 2
     1.1.4 Wer sollte dieses Buch lesen? ............................. 4
  1.2 Aufbau des Buchs .................................................. 4
  1.3 Konventionen und ausführbare Programme ................... 6

I Java-Grundlagen, Analyse und Design 11

2 Professionelle Arbeitsumgebung ................................. 13
  2.1 Vorteile von IDEs am Beispiel von Eclipse .................. 14
  2.2 Projektorganisation ............................................... 16
     2.2.1 Projektstruktur in Eclipse ................................. 16
     2.2.2 Projektstruktur für Maven und Gradle .................... 18
  2.3 Einsatz von Versionsverwaltungen .......................... 20
     2.3.1 Arbeiten mit zentralen Versionsverwaltungen ......... 23
     2.3.2 Dezentrale Versionsverwaltungen ....................... 28
     2.3.3 VCS und DVCS im Vergleich .............................. 34
  2.4 Einsatz eines Unit-Test-Frameworks .......................... 37
     2.4.1 Das JUnit-Framework ....................................... 37
     2.4.2 Vorteile von Unit Tests .................................... 42
  2.5 Debugging .......................................................... 43
     2.5.1 Fehlersuche mit einem Debugger ......................... 45
     2.5.2 Remote Debugging ......................................... 47
  2.6 Deployment von Java-Applikationen .......................... 52
     2.6.1 Das JAR-Tool im Kurzüberblick ........................... 54
     2.6.2 JAR inspirizieren und ändern, Inhalt extrahieren ....... 55
     2.6.3 Metainformationen und das Manifest .................... 56
     2.6.4 Inspizieren einer JAR-Datei .............................. 59
<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Seitenzahl</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.1 Grundlagen</td>
<td>218</td>
</tr>
<tr>
<td>4.2.2 Konvertierung von Werten</td>
<td>225</td>
</tr>
<tr>
<td>4.2.3 Wissenswertes zu Auto-Boxing und Auto-Unboxing</td>
<td>228</td>
</tr>
<tr>
<td>4.2.4 Ausgabe und Verarbeitung von Zahlen</td>
<td>231</td>
</tr>
<tr>
<td>4.3 Stringverarbeitung</td>
<td>234</td>
</tr>
<tr>
<td>4.3.1 Die Klasse String</td>
<td>235</td>
</tr>
<tr>
<td>4.3.2 Die Klassen StringBuffer und StringBuilder</td>
<td>239</td>
</tr>
<tr>
<td>4.3.3 Ausgaben mit format() und printf()</td>
<td>243</td>
</tr>
<tr>
<td>4.3.4 Die Methode split() und reguläre Ausdrücke</td>
<td>244</td>
</tr>
<tr>
<td>4.4 Datumsverarbeitung</td>
<td>249</td>
</tr>
<tr>
<td>4.4.1 Fallstricke der Datums-APIs</td>
<td>249</td>
</tr>
<tr>
<td>4.4.2 Das Date-API</td>
<td>251</td>
</tr>
<tr>
<td>4.4.3 Das Calendar-API</td>
<td>253</td>
</tr>
<tr>
<td>4.5 Varianten innerer Klassen</td>
<td>255</td>
</tr>
<tr>
<td>4.6 Ein- und Ausgabe (I/O)</td>
<td>259</td>
</tr>
<tr>
<td>4.6.1 Dateibehandlung und die Klasse File</td>
<td>259</td>
</tr>
<tr>
<td>4.6.2 Ein- und Ausgabestreams im Überblick</td>
<td>264</td>
</tr>
<tr>
<td>4.6.3 Zeichencodierungen bei der Ein- und Ausgabe</td>
<td>267</td>
</tr>
<tr>
<td>4.6.4 Speichern und Laden von Daten und Objekten</td>
<td>269</td>
</tr>
<tr>
<td>4.6.5 Dateiverarbeitung in JDK 7</td>
<td>277</td>
</tr>
<tr>
<td>4.6.6 Erweiterungen im NIO und der Klasse Files in JDK 8</td>
<td>279</td>
</tr>
<tr>
<td>4.7 Fehlerbehandlung</td>
<td>281</td>
</tr>
<tr>
<td>4.7.1 Einstieg in die Fehlerbehandlung</td>
<td>281</td>
</tr>
<tr>
<td>4.7.2 Checked Exceptions und Unchecked Exceptions</td>
<td>287</td>
</tr>
<tr>
<td>4.7.3 Exception Handling und Ressourcenfreigabe</td>
<td>288</td>
</tr>
<tr>
<td>4.7.4 Besonderheiten beim Exception Handling mit JDK 7</td>
<td>294</td>
</tr>
<tr>
<td>4.7.5 Assertions</td>
<td>298</td>
</tr>
<tr>
<td>4.8 Weiterführende Literatur</td>
<td>301</td>
</tr>
<tr>
<td>5 Lambdas, Methodenreferenzen und Defaultmethoden</td>
<td>303</td>
</tr>
<tr>
<td>5.1 Einstieg in Lambdas</td>
<td>303</td>
</tr>
<tr>
<td>5.1.1 Syntax von Lambdas</td>
<td>303</td>
</tr>
<tr>
<td>5.1.2 Functional Interfaces und SAM-Typen</td>
<td>304</td>
</tr>
<tr>
<td>5.1.3 Exceptions in Lambdas</td>
<td>308</td>
</tr>
<tr>
<td>5.2 Syntaxerweiterungen in Interfaces</td>
<td>312</td>
</tr>
<tr>
<td>5.2.1 Defaultmethoden</td>
<td>313</td>
</tr>
<tr>
<td>5.2.2 Statische Methoden in Interfaces</td>
<td>315</td>
</tr>
<tr>
<td>5.3 Methodenreferenzen</td>
<td>317</td>
</tr>
<tr>
<td>5.4 Externe vs. interne Iteration</td>
<td>318</td>
</tr>
<tr>
<td>5.5 Wichtige Functional Interfaces für Collections</td>
<td>319</td>
</tr>
<tr>
<td>5.5.1 Das Interface Predicate&lt;T&gt;</td>
<td>319</td>
</tr>
<tr>
<td>5.5.2 Das Interface UnaryOperator&lt;T&gt;</td>
<td>321</td>
</tr>
<tr>
<td>5.5.3 Fazit</td>
<td>322</td>
</tr>
</tbody>
</table>
7.1.5 Terminal Operations ........................................ 452
7.1.6 Wissenswertes zur Parallelverarbeitung ................. 460
7.2 Filter-Map-Reduce ............................................ 466
  7.2.1 Herkömmliche Realisierung ................................ 466
  7.2.2 Filter-Map-Reduce mit JDK 8 .............................. 468
7.3 Praxisbeispiele .................................................. 471
  7.3.1 Aufbereiten von Gruppierungen und Histogrammen .... 471
  7.3.2 Maps nach Wert sortieren .................................. 472
  7.3.3 Verarbeitung von ZIP-Dateien ............................ 476

8  Applikationsbausteine ........................................ 479
  8.1 Einsatz von Bibliotheken ................................... 480
  8.2 Google Guava im Kurzüberblick ............................ 483
    8.2.1 String-Aktionen ........................................ 485
    8.2.2 Stringkonkatenation und -extraktion .................. 486
    8.2.3 Erweiterungen für Collections .......................... 490
    8.2.4 Weitere Utility-Funktionalitäten ...................... 494
  8.3 Wertebereichs- und Parameterprüfungen .................... 500
    8.3.1 Prüfung einfacher Wertebereiche und Wertemengen ... 501
    8.3.2 Prüfung komplexerer Wertebereiche .................... 503
  8.4 Logging-Frameworks ......................................... 507
    8.4.1 Apache log4j2 .......................................... 507
    8.4.2 Tipps und Tricks zum Einsatz von Logging mit log4j2 ... 512
  8.5 Konfigurationsparameter und -dateien ..................... 516
    8.5.1 Einlesen von Kommandozeilenparametern .............. 516
    8.5.2 Verarbeitung von Properties ........................... 524
    8.5.3 Die Klasse Preferences ................................. 531
    8.5.4 Weitere Möglichkeiten zur Konfigurationsverwaltung ... 533

9  Multithreading .................................................. 539
  9.1 Threads und Runnables ..................................... 541
    9.1.1 Definition der auszuführenden Aufgabe ................ 541
    9.1.2 Start, Ausführung und Ende von Threads .............. 542
    9.1.3 Lebenszyklus von Threads und Thread-Zustände ........ 546
    9.1.4 Unterbrechungswünsche durch Aufruf von interrupt() .... 549
  9.2 Zusammenarbeit von Threads ................................ 552
    9.2.1 Konkurrierende Datenzugriffe .......................... 552
    9.2.2 Locks, Monitore und kritische Bereiche .............. 553
    9.2.3 Deadlocks und Starvation .............................. 560
    9.2.4 Kritische Bereiche und das Interface Lock ............ 562
9.3 Kommunikation von Threads ........................................ 564
  9.3.1 Kommunikation mit Synchronisation ......................... 564
  9.3.2 Kommunikation über die Methoden wait(), notify() und notifyAll() ................................................... 567
  9.3.3 Abstimmung von Threads ....................................... 574
  9.3.4 Unerwartete IllegalMonitorStateException .................. 577
9.4 Das Java-Memory-Modell ............................................ 578
  9.4.1 Sichtbarkeit ...................................................... 579
  9.4.2 Atomarität ........................................................ 579
  9.4.3 Reorderings ..................................................... 581
9.5 Besonderheiten bei Threads ......................................... 585
  9.5.1 Verschiedene Arten von Threads ............................... 585
  9.5.2 Exceptions in Threads .......................................... 586
  9.5.3 Sicheres Beenden von Threads .................................. 587
  9.5.4 Zeitgesteuerte Ausführung ..................................... 591
9.6 Die Concurrency Utilities ........................................... 595
  9.6.1 Concurrent Collections ....................................... 596
  9.6.2 Das Executor-Framework ..................................... 602
  9.6.3 Das Fork-Join-Framework .................................... 613
  9.6.4 Erweiterungen im Bereich Concurrency mit JDK 8 .......... 615
9.7 Weiterführende Literatur ............................................ 619

10 Fortgeschrittene Java-Themen ................................. 621
  10.1 Crashkurs Reflection ............................................ 621
      10.1.1 Grundlagen .................................................. 623
      10.1.2 Zugriff auf Methoden und Attribute ....................... 626
      10.1.3 Spezialfälle ................................................ 631
      10.1.4 Type Erasure und Typinformationen bei Generics ........ 634
  10.2Annotations .......................................................... 636
      10.2.1 Einführung in Annotations ................................ 637
      10.2.2 Standard-Annotations des JDKs ............................. 638
      10.2.3 Definition eigener Annotations ............................. 640
      10.2.4 Annotation zur Laufzeit auslesen ........................ 643
  10.3 Serialisierung ..................................................... 644
      10.3.1 Grundlagen der Serialisierung ............................. 645
      10.3.2 Die Serialisierung anpassen ................................ 650
      10.3.3 Versionsverwaltung der Serialisierung ................... 653
      10.3.4 Optimierung der Serialisierung ............................ 657
  10.4 Garbage Collection ............................................... 662
      10.4.1 Grundlagen zur Garbage Collection ....................... 662
      10.4.2 Herkömmliche Algorithmen zur Garbage Collection ....... 666
      10.4.3 Einflussfaktoren auf die Garbage Collection ............ 668
      10.4.4 Der Garbage Collector »G1« ................................. 670
10.4.5 Memory Leaks: Gibt es die auch in Java?!

10.5 Dynamic Proxies

10.5.1 Statischer Proxy

10.5.2 Dynamischer Proxy

10.6 »Nashorn« – die JavaScript-Engine

10.7 Weiterführende Literatur

11 Datumsverarbeitung seit JDK 8

11.1 Überblick über die neu eingeführten Typen

11.1.1 Neue Aufzählungen, Klassen und Interfaces

11.1.2 Die Aufzählungen DayOfWeek und Month

11.1.3 Die Klassen MonthDay, YearMonth und Year

11.1.4 Die Klasse Instant

11.1.5 Die Klasse Duration

11.1.6 Die Aufzählung ChronoUnit

11.1.7 Die Klassen LocalDate, LocalDateTime und LocalDate-Time

11.1.8 Die Klasse Period

11.1.9 Die Klasse ZonedDateTime

11.1.10 Zeitzonen und die Klassen ZoneId und ZoneOffset

11.1.11 Die Klasse Clock

11.1.12 Formatierung und Parsing

11.2 Datumsarithmetik

11.3 Interoperabilität mit Legacy-Code

12 GUIs mit JavaFX

12.1 Einführung – JavaFX im Überblick

12.1.1 Grundsätzliche Konzepte

12.1.2 Layoutmanagement

12.2 Deklarativer Aufbau des GUIs

12.2.1 Deklarative Beschreibung von GUIs

12.2.2 Hello-World-Beispiel mit FXML

12.2.3 Diskussion: Design und Funktionalität strikt trennen

12.3 Rich-Client Experience

12.3.1 Gestaltung mit CSS

12.3.2 Effekte

12.3.3 Animationen

12.3.4 Zeichnen in JavaFX-Komponenten

12.4 Properties, Bindings und Observable Collections

12.4.1 Properties

12.4.2 Bindings

12.4.3 Observable Collections

12.4.4 Dynamisches Filtern von ObservableList

Michael Inden, Der Weg zum Java-Profi, dpunkt.verlag, ISBN 978-3-86490-483-7
## Inhaltsverzeichnis

### 12.5 Wichtige Bedienelemente
- **12.5.1 Dialoge** .................................................. 754
- **12.5.2 Formatierte Eingabe in TextFields** ..................... 756
- **12.5.3 Die Bedienelemente ComboBox und ListView** ............. 758
- **12.5.4 Tabellen und das Bedienelement TableView** .......... 762
- **12.5.5 Das Bedienelement TreeTableView** ...................... 764
- **12.5.6 Menüs** ...................................................... 769

### 12.6 Multithreading in JavaFX ................................... 770
- **12.6.1 Das Interface Worker** .................................... 770
- **12.6.2 Die Klasse Task<V>** ...................................... 771

### 12.7 Von Swing zu JavaFX ....................................... 774
- **12.7.1 JavaFX in Swing einbinden** .................................. 774
- **12.7.2 Swing in JavaFX einbinden** .................................. 776

### 12.8 Weiterführende Literatur .................................. 778

### 13 Basiswissen Internationalisierung ............................ 779
- **13.1 Internationalisierung im Überblick** ....................... 779
- **13.1.1 Grundlagen und Normen** .................................. 780
- **13.1.2 Die Klasse Locale** .......................................... 781
- **13.1.3 Die Klasse PropertyResourceBundle** .................... 785
- **13.1.4 Formatierte Ein- und Ausgabe** ............................. 788
- **13.1.5 Zahlen und die Klasse NumberFormat** .................. 789
- **13.1.6 Datumswerte und die Klasse DateFormat** ............... 792
- **13.1.7 Textmeldungen und die Klasse MessageFormat** ........ 797
- **13.1.8 Stringvergleiche mit der Klasse Collator** ............ 799

### 13.2 Programmbausteine zur Internationalisierung ............ 804
- **13.2.1 Unterstützung mehrerer Datumsformate** ............... 805
- **13.2.2 Nutzung mehrerer Sprachdateien** ....................... 810

### III Neuerungen in Java 9 ......................................... 821

### 14 Ergänzungen in Java 9 ......................................... 823
- **14.1 Syntaxerweiterungen** ........................................ 823
- **14.1.1 Anonyme innere Klassen und der Diamond Operator** .... 823
- **14.1.2 Erweiterung der @Deprecated-Annotation** ............. 823
- **14.1.3 Private Methoden in Interfaces** ......................... 824

### 14.2 Neues und Änderungen im JDK .............................. 826
- **14.2.1 Das neue Process-API** ................................... 826
- **14.2.2 Neuerungen im Stream-API** ................................ 830
- **14.2.3 Erweiterungen rund um die Klasse Optional** ........... 833
- **14.2.4 Erweiterungen in der Klasse InputStream** ............. 839
- **14.2.5 Erweiterungen in der Klasse Objects** ................. 840
14.2.6 Erweiterungen in der Klasse `CompletableFuture` ..... 841
14.2.7 Collection-Factory-Methoden .................................. 843
14.3 Änderungen in der JVM ........................................... 845
14.3.1 Garbage Collection ............................................. 845
14.3.2 Browser-Plugin ist deprecated .................................. 847
14.3.3 Änderung des Versionsschemas .................................. 847
14.3.4 HTML5 Javadoc .................................................. 848
14.3.5 Java + REPL => `jshell` ........................................ 849
14.4 Fazit ................................................................. 851
15 Modularisierung mit Project Jigsaw .................................. 853
15.1 Grundlagen ............................................................ 854
15.1.1 Begrifflichkeiten .................................................. 854
15.1.2 Ziele von Project Jigsaw ........................................... 855
15.2 Modularisierung im Überblick ........................................ 856
15.2.1 Grundlagen zu Project Jigsaw ...................................... 856
15.2.2 Beispiel mit zwei Modulen ........................................ 863
15.2.3 Packaging .......................................................... 872
15.2.4 Linking ............................................................ 874
15.2.5 Abhängigkeiten und Modulgraphen .............................. 877
15.2.6 Module des JDKs einbinden ........................................ 879
15.2.7 Arten von Modulen ................................................ 886
15.3 Sichtbarkeiten und Zugriffsschutz ................................... 887
15.3.1 Sichtbarkeiten ...................................................... 887
15.3.2 Zugriffsschutz und Reflection .................................... 889
15.4 Kompatibilität und Migration .......................................... 891
15.4.1 Kompatibilitätsmodus .............................................. 892
15.4.2 Migrationsszenarien .............................................. 894
15.4.3 Fallstrick bei der Bottom-up-Migration ......................... 898
15.4.4 Beispiel: Migration mit Automatic Modules .................... 900
15.4.5 Beispiel: Automatic und Unnamed Module ..................... 902
15.4.6 Mögliche Schwierigkeiten bei Migrationen ..................... 904
15.4.7 Fazit ............................................................. 905
15.5 Zusammenfassung .................................................... 905

IV Fallstricke und Lösungen im Praxisalltag ................................ 907
16 Bad Smells ............................................................... 909
16.1 Programmdesign ...................................................... 911
16.1.1 Bad Smell: Verwenden von Magic Numbers .................... 911
16.1.2 Bad Smell: Konstanten in Interfaces definieren ................ 912
16.1.3 Bad Smell: Zusammengehörende Konstanten nicht als Typ definiert ............................................. 914
16.1.4 Bad Smell: Programmcode im Logging-Code ................. 916
16.1.5 Bad Smell: Dominanter Logging-Code .......................... 917
16.1.6 Bad Smell: Unvollständige Betrachtung aller Alternativen . 919
16.1.7 Bad Smell: Unvollständige Änderungen nach Copy-Paste 920
16.1.8 Bad Smell: Casts auf unbekannte Subtypen .................... 922
16.1.9 Bad Smell: Pre-/Post-Increment in komplexeren Statements 923
16.1.10 Bad Smell: Keine Klammern um Blöcke ....................... 925
16.1.11 Bad Smell: Mehrere aufeinanderfolgende Parameter gleicher Typs ............................................. 927
16.1.12 Bad Smell: Grundloser Einsatz von Reflection ................ 928
16.1.13 Bad Smell: System.exit() mitten im Programm .............. 930
16.1.14 Bad Smell: Variablendeclaration nicht im kleinstmöglichen Sichtbarkeitsbereich ............................................. 931
16.2 Klassendesign ............................................. 933
16.2.1 Bad Smell: Unnötigerweise veränderliche Attribute ............ 933
16.2.2 Bad Smell: Herausgabe von this im Konstruktor ............. 935
16.2.3 Bad Smell: Aufruf abstrakter Methoden im Konstruktor ... 937
16.2.4 Bad Smell: Referenzierung von Subklassen in Basisklassen 941
16.2.5 Bad Smell: Mix abstrakter und konkreter Basisklassen ... 943
16.2.6 Bad Smell: Öffentlicher Defaultkonstruktor lediglich zum Zugriff auf Hilfsmethoden ............................................. 945
16.3 Fehlerbehandlung und Exception Handling .......................... 947
16.3.1 Bad Smell: Unbehandelte Exception ............................ 947
16.3.2 Bad Smell: Unpassender Exception-Typ .......................... 948
16.3.3 Bad Smell: Exceptions zur Steuerung des Kontrollflusses 950
16.3.4 Bad Smell: Fangen der allgemeinsten Exception ............. 951
16.3.5 Bad Smell: Rückgabe von null statt Exception im Fehlerfall ............................................. 953
16.3.6 Bad Smell: Unbedachte Rückgabe von null ................... 954
16.3.7 Bad Smell: Sonderbehandlung von Randfällen ............... 957
16.3.8 Bad Smell: Keine Gültigkeitsprüfung von Eingabeparametern ............................................. 958
16.3.9 Bad Smell: Fehlerhafte Fehlerbehandlung .......................... 960
16.3.10 Bad Smell: I/O ohne finally oder ARM ....................... 962
16.3.11 Bad Smell: Resource Leaks durch Exceptions im Konstruktor ............................................. 964
16.4 Häufige Fallstricke ........................................ 968
16.5 Weiterführende Literatur ........................................ 983
## Inhaltsverzeichnis

### 17 Refactorings

<table>
<thead>
<tr>
<th>Unterkapitel</th>
<th>Seitenzahl</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.1 Refactorings am Beispiel</td>
<td>986</td>
</tr>
<tr>
<td>17.2 Das Standardvorgehen</td>
<td>994</td>
</tr>
<tr>
<td>17.3 Kombination von Basis-Refactorings</td>
<td>997</td>
</tr>
<tr>
<td>17.3.1 Refactoring-Beispiel: Ausgangslage und Ziel</td>
<td>997</td>
</tr>
<tr>
<td>17.3.2 Auflösen der Abhängigkeiten</td>
<td>999</td>
</tr>
<tr>
<td>17.3.3 Vereinfachungen</td>
<td>1006</td>
</tr>
<tr>
<td>17.3.4 Verlagern von Funktionalität</td>
<td>1010</td>
</tr>
<tr>
<td>17.4 Der Refactoring-Katalog</td>
<td>1011</td>
</tr>
<tr>
<td>17.4.1 Reduziere die Sichtbarkeit von Attributen</td>
<td>1011</td>
</tr>
<tr>
<td>17.4.2 Minimiere veränderliche Attribute</td>
<td>1014</td>
</tr>
<tr>
<td>17.4.3 Reduziere die Sichtbarkeit von Methoden</td>
<td>1018</td>
</tr>
<tr>
<td>17.4.4 Ersetze Mutator- durch Business-Methode</td>
<td>1020</td>
</tr>
<tr>
<td>17.4.5 Minimiere Zustandsänderungen</td>
<td>1021</td>
</tr>
<tr>
<td>17.4.6 Führe ein Interface ein</td>
<td>1021</td>
</tr>
<tr>
<td>17.4.7 Spalte ein Interface auf</td>
<td>1022</td>
</tr>
<tr>
<td>17.4.8 Führe ein Read-only-Interface ein</td>
<td>1023</td>
</tr>
<tr>
<td>17.4.9 Führe ein Read-Write-Interface ein</td>
<td>1023</td>
</tr>
<tr>
<td>17.4.10 Lagere Funktionalität in Hilfsmethoden aus</td>
<td>1024</td>
</tr>
<tr>
<td>17.4.11 Trenne Informationsbeschaffung und -verarbeitung</td>
<td>1026</td>
</tr>
<tr>
<td>17.4.12 Wandle Konstantensammlung in enum um</td>
<td>1033</td>
</tr>
<tr>
<td>17.4.13 Entferne Exceptions zur Steuerung des Kontrollflusses</td>
<td>1036</td>
</tr>
<tr>
<td>17.4.14 Wandle in Utility-Klasse mit statischen Hilfsmethoden um</td>
<td>1038</td>
</tr>
<tr>
<td>17.4.15 Löse if-else/instanceof durch Polymorphie auf</td>
<td>1042</td>
</tr>
<tr>
<td>17.5 Defensives Programmieren</td>
<td>1045</td>
</tr>
<tr>
<td>17.5.1 Führe eine Zustandsprüfung ein</td>
<td>1045</td>
</tr>
<tr>
<td>17.5.2 Überprüfe Eingabeparameter</td>
<td>1046</td>
</tr>
<tr>
<td>17.6 Fallstricke bei Refactorings</td>
<td>1051</td>
</tr>
<tr>
<td>17.7 Weiterführende Literatur</td>
<td>1053</td>
</tr>
</tbody>
</table>

### 18 Entwurfsmuster

<table>
<thead>
<tr>
<th>Unterkapitel</th>
<th>Seitenzahl</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.1 Erzeugungsmuster</td>
<td>1058</td>
</tr>
<tr>
<td>18.1.1 Erzeugungsmethode</td>
<td>1058</td>
</tr>
<tr>
<td>18.1.2 Fabrikmethode (Factory method)</td>
<td>1061</td>
</tr>
<tr>
<td>18.1.3 Erbauer (Builder)</td>
<td>1064</td>
</tr>
<tr>
<td>18.1.4 Singleton</td>
<td>1067</td>
</tr>
<tr>
<td>18.1.5 Prototyp (Prototype)</td>
<td>1072</td>
</tr>
<tr>
<td>18.2 Struktur muster</td>
<td>1076</td>
</tr>
<tr>
<td>18.2.1 Fassade (Façade)</td>
<td>1076</td>
</tr>
<tr>
<td>18.2.2 Adapter</td>
<td>1078</td>
</tr>
<tr>
<td>18.2.3 Dekorierer (Decorator)</td>
<td>1080</td>
</tr>
<tr>
<td>18.2.4 Kompositum (Composite)</td>
<td>1084</td>
</tr>
<tr>
<td>Kapitel</td>
<td>Seite</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>18.3 Verhaltensmuster</td>
<td>1088</td>
</tr>
<tr>
<td>18.3.1 Iterator</td>
<td>1088</td>
</tr>
<tr>
<td>18.3.2 Null-Objekt (Null Object)</td>
<td>1090</td>
</tr>
<tr>
<td>18.3.3 Schablonenmethode (Template method)</td>
<td>1093</td>
</tr>
<tr>
<td>18.3.4 Strategie (Strategy)</td>
<td>1097</td>
</tr>
<tr>
<td>18.3.5 Befehl (Command)</td>
<td>1109</td>
</tr>
<tr>
<td>18.3.6 Proxy</td>
<td>1116</td>
</tr>
<tr>
<td>18.3.7 Beobachter (Observer)</td>
<td>1118</td>
</tr>
<tr>
<td>18.3.8 MVC-Architektur</td>
<td>1127</td>
</tr>
<tr>
<td>18.4 Weiterführende Literatur</td>
<td>1129</td>
</tr>
</tbody>
</table>

V Qualitätssicherungsmaßnahmen

19 Programmierstil und Coding Conventions

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Seite</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.1 Grundregeln eines guten Programmierstils</td>
<td>1133</td>
</tr>
<tr>
<td>19.1.1 Keep It Human-Readable</td>
<td>1134</td>
</tr>
<tr>
<td>19.1.2 Keep It Simple And Short (KISS)</td>
<td>1134</td>
</tr>
<tr>
<td>19.1.3 Keep It Natural</td>
<td>1134</td>
</tr>
<tr>
<td>19.1.4 Keep It Clean</td>
<td>1135</td>
</tr>
<tr>
<td>19.2 Die Psychologie beim Sourcecode-Layout</td>
<td>1135</td>
</tr>
<tr>
<td>19.2.1 Gesetz der Ähnlichkeit</td>
<td>1135</td>
</tr>
<tr>
<td>19.2.2 Gesetz der Nähe</td>
<td>1137</td>
</tr>
<tr>
<td>19.3 Coding Conventions</td>
<td>1138</td>
</tr>
<tr>
<td>19.3.1 Grundlegende Namens- und Formatierungsregeln</td>
<td>1139</td>
</tr>
<tr>
<td>19.3.2 Namensgebung</td>
<td>1142</td>
</tr>
<tr>
<td>19.3.3 Dokumentation</td>
<td>1145</td>
</tr>
<tr>
<td>19.3.4 Programmdesign</td>
<td>1147</td>
</tr>
<tr>
<td>19.3.5 Klassendesign</td>
<td>1152</td>
</tr>
<tr>
<td>19.3.6 Parameterlisten</td>
<td>1155</td>
</tr>
<tr>
<td>19.3.7 Logik und Kontrollfluss</td>
<td>1157</td>
</tr>
<tr>
<td>19.4 Sourcecode-Prüfung mit Tools</td>
<td>1159</td>
</tr>
<tr>
<td>19.4.1 Metriken</td>
<td>1161</td>
</tr>
<tr>
<td>19.4.2 Sourcecode-Prüfung im Build-Prozess</td>
<td>1165</td>
</tr>
</tbody>
</table>

20 Unit Tests

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Seite</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.1 Testen im Überblick</td>
<td>1175</td>
</tr>
<tr>
<td>20.1.1 Was versteht man unter Testen?</td>
<td>1175</td>
</tr>
<tr>
<td>20.1.2 Testarten im Überblick</td>
<td>1176</td>
</tr>
<tr>
<td>20.1.3 Zuständigkeiten beim Testen</td>
<td>1179</td>
</tr>
<tr>
<td>20.1.4 Testen und Qualität</td>
<td>1181</td>
</tr>
</tbody>
</table>
20.2 Wissenswertes zu Testfällen ........................................ 1185
  20.2.1 Testfälle mit JUnit 4 definieren ............................. 1185
  20.2.2 Problem der Kombinatorik ................................. 1193
20.3 Motivation für Unit Tests aus der Praxis ......................... 1197
20.4 JUnit Rules und parametrierte Tests ............................. 1206
  20.4.1 JUnit Rules im Überblick .................................. 1206
  20.4.2 Parametrierte Tests ......................................... 1210
20.5 Fortgeschrittene Unit-Test-Techniken .............................. 1215
  20.5.1 Stellvertreterobjekte / Test-Doubles ......................... 1218
  20.5.2 Vorarbeiten für das Testen mit Stubs und Mocks .......... 1222
  20.5.3 Die Technik EXTRACT AND OVERRIDE ................. 1224
  20.5.4 Einstieg in das Testen mit Mocks und Mockito .......... 1231
  20.5.5 Abhängigkeiten mit Mockito auflösen .................. 1239
  20.5.6 Unit Tests von privaten Methoden ....................... 1242
20.6 Unit Tests mit Threads und Timing .............................. 1244
  20.6.1 Funktionale Erweiterung: Aggregation und Versand .... 1244
  20.6.2 Test der Aggregation und des Versands ................ 1246
  20.6.3 Test des nebenläufigen Versands ......................... 1249
20.7 Test Smells ...................................................... 1251
20.8 Nützliche Tools für Unit Tests ................................... 1256
  20.8.1 Hamcrest .................................................. 1256
  20.8.2 MoreUnit .................................................. 1262
  20.8.3 Infinitest .................................................. 1262
  20.8.4 JaCoCo ...................................................... 1263
  20.8.5 EclEmma ................................................... 1266
20.9 Ausblick auf JUnit 5 .............................................. 1268
  20.9.1 Einführendes Beispiel ...................................... 1268
  20.9.2 Wichtige Neuerungen in JUnit 5 im Überblick ........ 1269
20.10 Weiterführende Literatur ......................................... 1273

21 Codereviews .......................................................... 1275
  21.1 Definition ...................................................... 1275
  21.2 Probleme und Tipps zur Durchführung ....................... 1277
  21.3 Vorteile von Codereviews .................................... 1279
  21.4 Codereview-Checkliste ........................................ 1282

22 Optimierungen ....................................................... 1283
  22.1 Grundlagen ..................................................... 1284
    22.1.1 Optimierungsebenen und Einflussfaktoren .............. 1285
    22.1.2 Optimierungstechniken ................................ 1286
    22.1.3 CPU-bound-Optimierungsebenen am Beispiel ........ 1288
    22.1.4 Messungen – Erkennen kritischer Bereiche .......... 1292
    22.1.5 Abschätzungen mit der O-Notation .................. 1299
<table>
<thead>
<tr>
<th>Kapitel</th>
<th>Titel</th>
<th>Seite</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.2</td>
<td>Einsatz geeigneter Datenstrukturen</td>
<td>1302</td>
</tr>
<tr>
<td>22.2.1</td>
<td>Einfluss von Arrays und Listen</td>
<td>1303</td>
</tr>
<tr>
<td>22.2.2</td>
<td>Optimierungen für Set und Map</td>
<td>1307</td>
</tr>
<tr>
<td>22.2.3</td>
<td>Design eines Zugriffssinterface</td>
<td>1309</td>
</tr>
<tr>
<td>22.3</td>
<td>Lazy Initialization</td>
<td>1312</td>
</tr>
<tr>
<td>22.3.1</td>
<td>Lazy Initialization am Beispiel</td>
<td>1312</td>
</tr>
<tr>
<td>22.3.2</td>
<td>Konsequenzen des Einsatzes der Lazy Initialization</td>
<td>1315</td>
</tr>
<tr>
<td>22.3.3</td>
<td>Lazy Initialization mithilfe des PROXY-Musters</td>
<td>1317</td>
</tr>
<tr>
<td>22.4</td>
<td>Optimierungen am Beispiel</td>
<td>1320</td>
</tr>
<tr>
<td>22.5</td>
<td>I/O-bound-Optimierungen</td>
<td>1327</td>
</tr>
<tr>
<td>22.5.1</td>
<td>Technik – Wahl passender Strategien</td>
<td>1327</td>
</tr>
<tr>
<td>22.5.2</td>
<td>Technik – Caching und Pooling</td>
<td>1331</td>
</tr>
<tr>
<td>22.5.3</td>
<td>Technik – Vermeidung unnötiger Aktionen</td>
<td>1331</td>
</tr>
<tr>
<td>22.6</td>
<td>Memory-bound-Optimierungen</td>
<td>1334</td>
</tr>
<tr>
<td>22.6.1</td>
<td>Technik – Wahl passender Strategien</td>
<td>1334</td>
</tr>
<tr>
<td>22.6.2</td>
<td>Technik – Caching und Pooling</td>
<td>1337</td>
</tr>
<tr>
<td>22.6.3</td>
<td>Optimierungen der Stringverarbeitung</td>
<td>1343</td>
</tr>
<tr>
<td>22.6.4</td>
<td>Technik – Vermeidung unnötiger Aktionen</td>
<td>1345</td>
</tr>
<tr>
<td>22.7</td>
<td>CPU-bound-Optimierungen</td>
<td>1348</td>
</tr>
<tr>
<td>22.7.1</td>
<td>Technik – Wahl passender Strategien</td>
<td>1348</td>
</tr>
<tr>
<td>22.7.2</td>
<td>Technik – Caching und Pooling</td>
<td>1350</td>
</tr>
<tr>
<td>22.7.3</td>
<td>Technik – Vermeidung unnötiger Aktionen</td>
<td>1351</td>
</tr>
<tr>
<td>22.8</td>
<td>Weiterführende Literatur</td>
<td>1354</td>
</tr>
<tr>
<td>23</td>
<td>Schlussgedanken</td>
<td>1355</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VI</th>
<th>Anhang</th>
<th>1357</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Grundlagen zur Java Virtual Machine</td>
<td>1359</td>
</tr>
<tr>
<td>A.1</td>
<td>Wissenswertes rund um die Java Virtual Machine</td>
<td>1359</td>
</tr>
<tr>
<td>A.1.1</td>
<td>Ausführung eines Java-Programms</td>
<td>1359</td>
</tr>
<tr>
<td>A.1.2</td>
<td>Sicherheit und Speicherverwaltung</td>
<td>1360</td>
</tr>
<tr>
<td>A.1.3</td>
<td>Sicherheit und Classloading</td>
<td>1361</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Literaturverzeichnis</th>
<th>1363</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>1367</td>
</tr>
</tbody>
</table>