



Elisabeth
Jung

Band I

Java 7

Das Übungsbuch

Über 200 Aufgaben mit vollständigen Lösungen

Stichwortverzeichnis

Symbole

...-Operator 56
<>-Operator siehe Diamond-Operator
==-Operator 51, 154, 156

A

Abbildungen siehe Map
AbstractAction 391, 392
AbstractBorder 313, 317
AbstractButton 302, 310, 311, 313, 319, 555
AbstractMap 445, 549
AbstractTableModel 595
acceptDrag() 604
acceptDrop() 604
AccessibleObject 453
Action 391, 392
ActionEvent 209, 211, 212, 266, 389, 599
ActionListener 211, 257, 264, 266, 388, 391, 393,
468, 552, 593, 599, 696
actionPerformed() 211, 257, 265, 391, 596, 696
add() 255, 264, 303, 307, 391, 674, 677, 697
addActionListener() 211
addTab() 597
addTreeModelListener() 463
addWindowListener() 210, 248
Adressraum 47
andFilter() 594
Ansicht *siehe* Viewer
Applet 188
Appletviewer 34, 188
Arc2D.Float 319
Argumentenliste 56
Array 52, 53, 442, 443, 459
ArrayIndexOutOfBoundsException 691
Array-Initialisierer 33, 52
ArrayList 446
Array-Objekt 52, 53, 108, 158, 198, 442, 447
Arrays 52, 53, 587, 686
ArrayStoreException 686
Arraytyp 31, 662, 685, 689
Arrayvariable 33
AtomicInteger 145
AtomicLong 145
Aufzählungen *siehe* Enumerationen
Aufzählungskonstanten 449
Ausnahme *siehe* Exception
Auto(Un)Boxing 146

Autoboxing 51
Auto-unboxing 51
AWT
 Abstract Windowing Toolkit 187
AWT-Container 187, 205
AWTEvent 209, 708
AWT-Komponente 187, 189, 201, 208, 260, 303

B

BasicArrowButton 438
BasicButtonUI 312
BasicLookAndFeel 321
BasicTextAreaUI 314
BasicTextFieldUI 322
Behälter 434, 435
Benutzer-Defaults 315
Beschriftung 276, 310, 312, 319, 392
Betriebssystem 47, 187, 188
BigDecimal 145, 146
BigInteger 145
Bild *siehe* Image
Bilddatei *siehe* Image
Bildschirmausgaben 34
Boolean 50
Border 313
BorderFactory 200
BorderLayout 199, 202, 211, 244, 247, 318
Box 189, 203
BoxLayout 200, 202
bridge methodes *siehe* Brückenmethoden
Brückenmethoden III, 440, 441
BufferedImage 695
Button 211, 212, 598, 699
ButtonModel 302
ButtonUI 315, 321
Byte 50, 145
Bytecode 33, 159
byteValue() 51, 678

C

Calendar 588, 589
call by reference
 Referenzübergabe 40
call by value 39
 Wertübergabe 40
canImport() 605, 608, 610
Canvas 208, 244, 261, 262, 264, 391

catch-Block 47, 549, 587, 691
 ChangeEvent 209, 389
 Character 50
 Class 46, 153, 452, 456, 458, 547, 598, 691
 ClassCastException 660, 691
 Class-Literal 666
 ClassNotFoundException 554, 666, 690
 Class-Token siehe Class-Literal
 Clipboard 599, 602
 Clipping Region 197
 Clip-Rectangle 196, 197
 clip-rectangle-Parameter 192
 clone() 153, 154, 157, 158
 Cloneable 153, 155, 158
 CloneNotSupportedException 154
 Collection 266, 445, 446, 549
 Collections 683
 Color 194, 198, 201, 202, 265, 317, 383, 392, 696
 color-Parameter 192
 ColorUIResource 316, 554
 ComboBoxModel 303
 Comparable 153, 156, 679, 683
 Comparator 683
 compareTo() 154, 157
 Component 187, 191, 194, 205, 208, 224, 244,
 253, 598, 709
 ComponentAdapter 698
 ComponentEvent 209
 componentResized() 698
 ComponentUI 309, 315
 concat() 54, 546
 Constraints
 Einschränkungen 200
 Constructor 452, 457, 547, 692, 693
 Container 188, 196, 201, 205, 208, 244, 253, 438
 contains() 319
 ContentPane 253, 254, 257, 258, 318, 382
 Controller 189, 309, 321
 Copy-Konstruktor 39, 157, 158
 copyOf() 587, 588
 copyOfRange() 587, 588
 createGraphics() 591
 createTransferable() 605, 607
 createUI() 309, 707

D

DataFlavor 599, 603, 604
 DataFlavor.stringFlavor 599, 603, 605
 DataFormatSymbols 589
 Date 589
 dateFilter() 594
 DateFormat 589
 Datenkapselung 38
 Datentyp 31, 147, 150, 152
 deep cloning
 tiefes Klonen 157
 DefaultButtonModel 302, 306

DefaultCaret 314
 DefaultComboBoxModel 302
 DefaultListModel 608
 DefaultMutableTreeNode 307, 308, 464, 607
 DefaultRowSorter 594
 DefaultTableModel 594
 DefaultTreeModel 307, 308, 462, 464, 465, 609
 DefaultTreeSelectionModel 307
 Dekoration 695, 705
 descendingKeySet() 590
 Destruktor 159
 Dialog 188, 197, 390
 Dialog.ModalExclusionType 592
 Dialog.ModalityType 592, 593
 Diamond-Operator 672, 676, 677
 digit() 51
 Dimension 195, 392
 Document 301
 Double 50, 145, 437
 doubleValue() 146
 down-casting
 verkleinernde Konvertierung 114, 117, 139
 drag-and-drop 599
 dragEvent() 604
 draw() 263, 264, 386
 drawImage() 258
 drawString() 258
 drop() 604, 605
 DropMode 607
 DropTarget 603, 604
 DropTargetDragEvent 604
 DropTargetDropEvent 604
 DropTargetEvent 604
 DropTargetListener 604, 605

E

Eigenschaften

-background 199
 -bounds 205
 -clip-rectangle 193
 -color 701
 -contentAreaFilled 306, 311
 -focus 317
 -font 192, 451
 -font-ascent 194
 -font-descent 194
 -foreground 192, 600, 706
 -opaque 256, 271, 306
 -optimizedDrawingEnabled 193
 -size 701
 -String 464
 -text 600
 elements() 450
 Ellipse2D.Float 386
 EmptyStackException 552
 Entry 596, 597
 Enum 448

enum siehe Enumerationen
 Enumeration 450, 463
 Enumerationen 445, 448, 449, 549, 590, 609
 -Tag 449, 493, 494, 564
 -WTag 449, 492, 494, 564
 EnumMap 445, 559
 equals() 153, 155, 156
 Erasure 439
 Ereignisbehandlung *siehe* Eventbehandlung
 Ereignisse *siehe* Events
 Error 47, 545
 Erscheinungsbild 301, 315
 Erweitern *siehe* Klasse -abgeleitet
 Event-Adapter 209
 Eventbehandlung 209, 212, 303, 308, 387, 465, 466
 eventDispatched() 704
 Event-Dispatch-Thread 707
 EventListener 209, 265, 266, 388, 390, 708
 EventListenerList 303, 304, 464
 EventObject 209, 266, 388, 390, 604, 708
 Events auf höherer Ebene *siehe* Komponenten-Events
 Events auf niedriger Ebene *siehe* Fester-Events
 Exception 47, 545, 551
 exit() 47
 exportAsDrag() 602
 exportDone() 605
 extends-Klausel 105, 669

F

Farbauswahldialog 391
 Farbe *siehe* Color
 Farbverlauf *siehe* GradientPaint
 Felddeklarationen 56
 Felder 31
 -geerbte 105, 107, 150
 -selbstreferenzierende 42, 436
 -verdeckte 112
 Fenster 191, 695
 Fenster-Events
 Low-Level-Events 209
 Field 457, 461
 File 47, 550, 553
 fill() 264
 fillOval() 195
 finalize() 153, 159
 finally-Block 545, 548, 549
 fireTreeNodeInserted() 308
 Float 50, 145, 437
 floatValue() 50, 146
 FlowLayout 199, 202, 244, 247, 318
 FocusEvent 209
 Focus-Rectangle 306, 310, 311, 313
 Font 194, 348, 383
 FontMetrics 196
 font-Parameter 192

FontUIResource 316, 554
 for-each-Schleife 55, 444, 448
 forName() 453, 598, 691
 for-Schleife 54
 Fragezeichen-Operator 245
 Frame 188, 195, 201, 695

G

Garbage Collector 153, 159, 709
 GenericArrayType 456, 462
 GenericDeclaration 456, 460
 Generics 433
 Generische Arrays 442, 444, 458
 Generizität *siehe* Generics
 get() 315, 448, 457, 458, 549
 getActionCommand() 599
 getActualTypeArguments() 460, 462
 getAscent() 196
 getBounds() 188, 263, 461
 getChild() 465
 getChildCount() 464
 getClass() 153, 443, 454, 679
 getClipBounds() 197
 getColor() 316, 317, 319
 getColumnClass() 595
 getComponent() 314
 getComponentZOrder() 254, 255, 700
 getConstructor() 458, 694
 getContentPane() 697
 getContents() 603
 getCrossPlatformLookAndFeelClassName() 318
 getDeclaredField() 457, 458, 459
 getDeclaredFields() 461
 getDefault() 588
 getDefaults() 316, 317, 451
 getDefaultToolkit() 194, 195
 getDescent() 196
 getDisplayName() 589
 getDisplayNames() 589
 getDocument() 302
 getDropLocation() 608, 610
 getEnv() 47
 getException() 553
 getField() 458
 getFocus() 311
 getFontMetrics() 194
 getGenericDeclaration() 461
 getGenericInterfaces() 459
 getGenericSuperclass() 459
 getGenericType() 461
 getGraphics() 192, 261, 265, 390
 getImage() 196
 getIndex() 608, 610
 getInsets() 313
 getInstalledLookAndFeelFeels() 318
 getInstance() 588
 getInt() 53

getInterfaces() 459
 getItems() 304
 getJLayeredPane() 255
 getKeyChar() 212
 getKeyCode() 212
 getKeyModifiersText() 212
 getKeyText() 212
 getLastPathComponent() 607, 610
 getLastSelectedPathComponent() 607
 getListeners() 304, 464
 getLogger() 587
 getLowerBounds() 462
 getMaximumSize() 190, 309
 getMessage() 545, 554
 getMimeType() 603
 getMinimumSize() 190, 309, 311
 getModel() 302, 306, 608
 getModifiers() 212
 getName() 459, 461
 getNumericValue() 51
 getOwnerType() 461
 getPackage() 459
 getParameterTypes() 455
 getPath() 610
 getPreferredSize() 190, 197, 202, 208, 309, 311,
 312, 313
 getRawType() 460, 461
 getReturnType() 455
 getRGBColorComponents() 317
 getRoot() 607
 getRootPane() 253
 getRuntime() 47, 591
 getScreenSize() 194, 195
 getSelectionPath() 607
 getShortMonths() 589
 getSourceActions() 605
 getSuperclass() 459
 getSystemClipboard() 599
 getSystemLookAndFeelClassName() 318
 getText() 319, 600
 getTransferable() 604, 608
 getTransferData() 599, 603
 getTypeParameters() 456, 461
 getTypParameters() 455
 getUI() 310
 getUpperBounds() 462
 getValue() 438, 597
 getX() 188
 getY() 188
 GlassPane 253, 254, 257, 258
 GradientPaint 313, 696, 698
 Graphics 191, 194, 197, 258, 262, 263
 Graphics2D 191, 263, 264, 386, 695
 GraphicsDevice 695, 697
 GraphicsEnvironment 695
 Graphikkontext 192, 261, 265, 390, 591
 GregorianCalendar 58, 588, 589
 GridBagConstraints 200, 203

GridBagLayout 203
 GridLayout 199, 202, 212, 438
 GUI
 Graphical-User-Interface 187

H

hasFocus() 555
 HashMap 445
 hashMoreElements() 450
 hashNext() 450
 Hashtable 315, 445
 hat ein-Beziehung 110, 115
 headMap() 549
 Heap 159
 Heap-Pollution 673
 Hintergrund 198
 Hüllenklassen *siehe* Wrapper-Klassen
 HW-Komponente 188, 190, 202, 204, 205, 699

I

Icon 320
 if-Anweisung 51
 IllegalAccessException 547, 554, 690
 IllegalArgumentException 547, 550
 Image 196, 346
 ImageIcon 198, 257, 320, 695
 implements-Klausel 386, 670
 import-Anweisung 43, 44, 45, 46, 107
 importData() 605, 607, 608, 610
 include() 596
 incremental painting
 weiterführendes Zeichnen 192, 261
 IndexOutOfBoundsException 553
 initCause() 553
 initClassDefaults() 316, 321
 initComponentsDefaults() 316, 321
 InputStream 46
 insert() 308
 insertNodeInto() 463, 465, 607, 610
 Insets 188, 317
 installDefaults() 314
 installUI() 309, 319, 707
 instanceof-Operator 156, 456, 461, 659, 666
 InstantiationException 547, 690
 Instanz *siehe* Objekt
 Instanz-Block-Initialisierer 56, 57
 Instanzfelder 31, 32, 35
 -selbsterferenzierende 57
 Instanz-Feld-Initialisierer 56, 57
 Instanzmethoden 31, 32, 35, 42, 48, 109, 150
 -dynamisches Binden 108, 109, 116
 -überschreiben 108, 112
 Instanzvariable 33
 Integer 50, 145, 436, 437, 687
 Interface 148, 149
 -ableiten 151
 -generisches 441, 450

- Member 385, 386
- Static-Member 382, 436
- Interne Fenster *siehe* JInternalFrame
- intValue() 50, 51, 146, 678
- InvocationTargetException 547, 691
- invokeLater() 707
- IOException 603
- isArmed() 306
- isDataFlavorSupported() 608
- isDirectory() 550
- isDrop() 608
- isEmpty() 587
- isFile() 550
- isFlavorTextType() 604
- isOptimizedDrawingEnabled() 190
- isPressed() 555
- ist ein-Beziehung 109, 115
- isWindowTranslucencySupported() 695, 698
- ItemEvent 209, 266, 389
- ItemListener 266
- Iterable 444, 445, 450
- Iterationsvariablen 55
- Iterator 450
- iterator() 450

J

- JApplet 189
- Java
 - 2D-API 191, 263
 - Anwendung 47
 - API 50
 - Applet 34
 - Applikation 34, 318
 - Event-Listener-API 210
 - Logging-API 587
 - Programme 33
 - Programmiersprache 50
 - Servlet 34
- Java 5 47, 50, 51, 53, 54, 55, 111, 116, 145, 153, 154, 155, 188, 195, 206, 255, 433, 444, 448, 450, 451, 452
- Java 6 587, 589, 590, 591, 592, 594, 598, 607
- Java 7 657, 661, 671, 672, 688, 690, 704, 708, 711
- Java Collection Framework 445
- java.awt 190, 695
- java.awt.dnd 604
- java.awt.geom 263, 698
- java.awt.List 302, 303, 304, 465
- java.awt.peer 187
- java.io 46
- java.lang 46
- java.lang.reflect 53, 452, 458, 692
- java.math 146
- java.text 589
- java.util 46, 53, 445
- java.util.logging 587
- java.util.Vector 466
- javax.swing 190
- javax.swing.table 594
- javax.swing.tree 307
- JButton 255, 257, 264, 302, 321, 388, 438, 462, 699, 702, 706
- JColorChooser 390, 391
- JComboBox 302, 303, 305, 321, 436, 702
- JComponent 224, 247, 309, 319, 600
- JDesktopPane 189, 190, 259, 702
- JDialog 189, 196, 200, 253, 390, 593
- JEditorPane 301, 306
- JFileChooser 390
- JFrame 189, 194, 195, 200, 253, 593, 695
- JInternalFrame 189, 253, 255, 258, 259, 702
- JLabel 202, 204, 211, 212, 247, 257, 264, 389, 392, 552, 602
- JLayer 704, 706, 707
- JLayeredPane 189, 190, 253, 255, 256, 258, 259, 270, 392
- JList 302, 462, 465, 466, 468, 599, 605, 608
- JList.DropLocation 608
- JMenu 255, 393, 602, 608
- JMenuBar 253, 255, 258, 393, 602
- JMenuItem 302, 393
- JOptionPane 390, 391
- JPanel 189, 193, 199, 202, 208, 310, 321, 388, 390, 392, 698, 700, 703
- JPasswordField 599
- JRadioButton 266, 321, 389
- JRadioButtonMenuItem 602
- JRootPane 189, 253
- JScrollBar 702
- JScrollPane 189, 307, 314, 595
- JSlider 321, 598, 702
- JSplitPane 189
- JTabbedPane 189, 597, 598, 599
- JTable 594, 595, 599
- JTextArea 301, 306, 314, 599, 601, 602
- JTextComponent 301, 302, 310
- JTextField 211, 266, 301, 306, 321, 388, 702, 706
- JToggleButton 302, 706
- JToolBar 264, 391, 465, 593, 696, 697
- JTree 302, 307, 308, 462, 465, 555, 599, 606, 609
- JTree.DropLocation 610
- JVC
 - Java Foundation Classes 187
- JViewport 189, 190, 314
- JVM
 - Virtuelle Java Maschine 33, 34, 159, 434, 709
- JWindow 189, 196, 253

K

- KeyEvent 209, 211, 212
- KeyListener 211, 212

- keyPressed() 212
- keyReleased() 212
- keys() 450, 451, 463
- keySet() 448
- keyTyped() 212
- Klasse 31
 - abgeleitete 105, 109, 110, 112
 - abstrakte 145, 149, 588
 - anonyme 381, 385, 386, 388, 393, 597, 705
 - generische 116, 435
 - innere 210, 381, 387
 - lokale 381, 384, 597
 - Member 381, 382, 383, 436
 - selbstreferenzierende 42
 - Static-Member 381, 383, 384
 - umgebende 382, 383, 384
- Klassen-Block-Initialisierer 56, 57, 317
- Klassendatei 43, 453
- Klassendefinitionen
 - AbstractMapObjekteundExceptions 549, 561
 - AndersAnordnenTest 116, 141
 - AnordnenTest 115, 138
 - AnzeigevonLayoutManager 201, 223
 - Apfel 681, 729
 - ArraysKlonen 155, 176
 - ArrayTest1 53, 90
 - ArrayTest2 55, 95
 - AWTContainer 205, 235
 - AWTLabel 201, 226
 - AWTPaintComponent 208, 244
 - AWTPaintKreis 205, 234
 - BaumStruktur 607, 648
 - BenutzerBorder 317, 361
 - BenutzerButtonUI 318, 361
 - Birne 681, 730
 - Blumen 117, 142
 - BlumenTest 117, 143
 - BrueckenMethoden 441, 481
 - Buch 42, 76
 - BuchalsLayeredPane 392, 429
 - Buchstabe 155, 176
 - BuchTest 43, 78
 - CalendarundDate 589, 612
 - CanvasmitMouseEvent 265, 291
 - ColorIcon 320, 367
 - ComponentmitPaint 262, 284
 - ComponentmitUpdate 262, 285
 - ContainerfuerComponent 208, 243
 - CustomArea 314, 355
 - CustomAreaUI 314, 355
 - CustomBasicButtonUI 554, 576
 - CustomBorder 313, 352
 - CustomButton 554, 575
 - CustomButton1 554, 576
 - CustomButtonUI 314, 352
 - CustomCaret 314, 356
 - CustomField 321, 375
 - CustomFieldLookAndFeel 321, 377
 - CustomFieldUI 321, 376
 - CustomLookAndFeel 554, 577
 - CustomRowFilter 596, 627
 - CustomTableModel 597, 627
 - DefinitionvonFarben 198, 221
 - Destruktor 159, 185
 - DestruktorTest 159, 185
 - DialogmitFrameOwner 197, 220
 - Dialogmitrepaint 198, 220
 - DiamondOperator 677, 718
 - Dodekaeder 152, 174
 - Drachen 112, 133
 - DreieckArten 150, 167
 - DreieckArtenTest 150, 169
 - Enumerationen 449, 492
 - EnumerationenmitMethoden 449, 493
 - EnumerationIteratorIterable 450, 496
 - ExceptionKetten 553, 572
 - ExceptionsundFinally 548, 559
 - FarbenAuswahl 151, 171
 - FarbenTest 152, 171
 - FeldZugriffe 36, 59
 - FeldZugriffeTest 36, 60
 - FensterDekorieren 696, 765
 - FontIcon 320, 367
 - FrameMaske 201, 225
 - FramemitActionListener 211, 249
 - FramemitContainerundPaintKreis 205, 236
 - FramemitListenerKlasse 212, 251
 - Framemitpaint 195, 213
 - FramemitPaintKreis 205, 234
 - Framemitrepaint 260, 279
 - FramemitUpdate 262, 285
 - FramemitWindowAdapter 211, 248
 - FramemitWindowAdapterundAnonymeKlasse 388, 412
 - FramemitWindowAdapterundMemberKlasse 387, 411
 - FramemitWindowListener 210, 247
 - FramemitZOrder 206, 237
 - FullStackException 552, 570
 - GanzeZahlen 41, 71
 - GenericArray 442, 482
 - GenericArrayTypen 685, 746
 - GenericArrayundGenericMethoden 444, 485
 - GenericClass 456, 500
 - GenericClassmitGenericArray 458, 505
 - GenericClassmitGenericConstructor 457, 502
 - GenericFields 460, 511
 - GenericMapObjekte 447, 489
 - GenericMapObjektemitForEach 448, 490
 - GenericMethoden 443, 483
 - GenericPunkt 440, 477
 - GenericPunktmitBounds 678, 723

- GenericStack 436, 470
- GenericStackfuerInversWort 551, 568
- GenericStdKlsundInt 446, 487
- GenericSubTyp 441, 481
- GenericTyp 441, 481
- GenericTyp1 435, 468
- GenericTyp2 435, 469
- GenericTyp3 676, 714
- GenericTypmitExtend 437, 473
- GenericTypmitSuper 438, 475
- GenericTypmitWildcard 437, 472
- GeometrischeKoerper 146, 161
- GeometrischeKoerpermitGenericStdKls 446, 486
- GeometrischeKoerperTest 146, 162
- GeomFigMemberInterface 386, 407
- GleichschenkligesDreieck 150, 168
- GleichseitigesDreieck 150, 168
- GlobaleReferenzen 40, 69
- GrossBuchstaben 306, 331
- GrundMaskefuerAWT 201, 225
- GrundMaskefuerSwing 202, 227
- IconButton 306, 331
- Ikosaeder 152, 173
- InitialisierungInstanzfelder 57, 98
- InitialisierungKlassenfelder 57, 100
- InnereKlassemitmainMethode 391, 427
- InstanzZaehlerTest 108, 122
- JButtonmitCustomButtonUI 313, 353
- JButtonmitIcon 311, 345
- JButtonmitUeberschrift 311, 347
- JButtonmitZeichnung 310, 344
- JComponentmitpaintundclipBounds 197, 218
- JComponentmitpaintundohneclipBounds 196, 217
- JDialogmitpaint 196, 216
- JFramefuerGeomFigMemberInterface 386, 408
- JFramefuerJListTransferHandlermitDropLocation 608, 651
- JFramefuerJTreeTransferHandlermitDropLocation 610, 655
- JFramefuerKomponentenmitZahlen 389, 416
- JFramefuerMemberKlassen 383, 396
- JFramefuerMemberKlassenStatic 384, 401
- JFramefuerRadioButtonEventundListener 389, 420
- JFramefuerUIDefaultsTreeModelmitTreeModelListener 463, 525
- JFramefuerUserEventundListener 266, 298
- JFramefuerUserKomponente 390, 423
- JFrameMaske 202, 227
- JFramemitActionListenerundAnonymeKlasse 388, 412
- JFramemitBenutzerButtonUI 318, 362
- JFramemitBoxLayout 202, 230
- JFramemitClipboard 602, 637
- JFramemitContainerundPaintKreis 207, 241
- JFramemitDefaultComboBoxModel 302, 322
- JFramemitDefaultTreeModel 307, 335
- JFramemitDialogFenster 391, 425
- JFramemitDialogModalityBeispielen 593, 618
- JFramemitDnDfuerJLabel 602, 636
- JFramemitDnDfuerJList 601, 634
- JFramemitDnDfuerJTextArea 601, 635
- JFramemitDropTargetListener 603, 640
- JFramemitGenericStackfuerInversWort 552, 570
- JFramemitgetGraphics 261, 281
- JFramemitgetLayeredPane 255, 267
- JFramemitGridBagLayout 203, 231
- JFramemitInternenFenster 259, 277
- JFramemitJInternalFrame 259, 276
- JFramemitJListTransferHandler 606, 644
- JFramemitJTabbedPane 598, 630
- JFramemitJTabbedPaneV6 598, 631
- JFramemitJTableCustomFiltering 596, 626
- JFramemitJTableStandardFiltering 595, 623
- JFramemitJTableStandardSorting 594, 621
- JFramemitJTableUserSorting 595, 622
- JFramemitJTableUserSortingundFiltering 597, 628
- JFramemitJTree 307, 334
- JFramemitJTreeTransferHandler 607, 648
- JFramemitKeyListener 211, 250
- JFramemitKreiseMalenAnonymeKlasse 385, 404
- JFramemitKreiseMalenLokaleKlasse 384, 402
- JFramemitKreiseMalenLokaleKlasseundFinal 385
- JFramemitKreiseMalenMemberInterface 385, 405
- JFramemitKreiseMalenMemberInterfaceundMemberKlasse 387, 409
- JFramemitKreiseMalenMemberKlasse 382, 393
- JFramemitKreiseMalenStaticMemberKlasse 383, 398
- JFramemitLayeredPane 257, 270, 700, 774
- JFramemitLayeredPaneundGlassPane 257, 272
- JFramemitLookAndFeelAuswahl 320, 368

- JFramemitLookAndFeelundJTree 555, 579
- JFramemitLookAndFeelundJTreeAnders 555, 583
- JFramemitLookAndFeelundStdButtonUI 321
- JFramemitLookAndFeelundTextFieldUI 322, 378
- JFramemitMatrix2D 264, 288
- JFramemitMatrix2DAnders 264, 289
- JFramemitMyButtonUI 319, 365
- JFramemitnewLayeredPane 256, 268
- JFramemitNimbusLookAndFeel 702, 777
- JFramemitNullLayout 204, 233
- JFramemitpaint 195, 214
- JFramemitPaintKreis 206, 239
- JFramemitpaintundgetGraphics 261, 283
- JFramemitRectangle2DMemberInterface 386, 406
- JFramemitrepaint 260, 280
- JFramemitSplash 591, 616
- JFramemitSplashScreen 591, 617
- JFramemitUIDefaultsHashTable 451, 499
- JFramemitUIDefaultsLesen 316, 358
- JFramemitUIDefaultsSetzen 317, 360
- JFramemitUIDefaultsTreeModel 463, 520
- JFramemitUIDefaultsTreeModelundLookAndFeel 465, 528
- JFramemitUserButtonUI 313, 351
- JFramemitUserComboBoxModel 305, 329
- JFramemitUserDefaultTreeModel 308, 339
- JFramemitUserListModel 468, 541
- JFramemitUserListModelmitEventBehandlung 467, 537
- JFramemitUserListModelohneEventBehandlung 466, 532
- JFramemitUserModell 306, 332
- JFramemitVektor2D 263, 286
- JLayerundLayerUIInstanzen 707, 784
- JListTransferHandler 605, 643
- JListTransferHandlermitDropLocation 608, 650
- JPanelmitLayeredPaneundGlassPane 258, 274
- JPanelmitpaint 310, 343
- JTreeTransferHandler 606, 646
- JTreeTransferHandlermitDropLocation 609, 653
- JWindowmitpaint 196, 215
- Kegel 146, 162
- Klasse 45, 78
- Klasse1 45, 79, 106, 118
- Klasse2 45, 79, 107, 118
- Klasse3 107, 119
- Klasse4 107, 120
- KlasseA 108, 121
- KlasseB 108, 121
- KlasseC 109, 123
- KlasseD 109, 123
- KlassenDefinition 34, 59
- KlassenInformationen 459, 506
- KlassenTest 45, 79
- KlonenTest 155, 176
- Kombinationen 114, 136
- KombinationenausVariationen 115, 137
- KombinationenhabenPermutationen 115, 140
- KomponenteDekorieren 705, 782
- KomponenteDekorierenmitEventHandling 707, 787
- KomponentenEvent 266, 295
- KomponentenmitZahlen 388, 413
- KomponentenUeberlappen 314, 357
- KongruenteDreiecke 48, 80
- KongruenteDreieckeTest 49, 82
- KopievonReferenzTypen 158, 184
- KovarianteRueckgabeTypen 111, 129
- Kreis 49, 83
- KreisComponent 208, 243
- KreisContainer 207, 240
- KreishatPunkt 110, 125
- KreishatPunktTest 110, 111, 126
- KreisistPunkt 111, 127
- KreisistPunktTest 111, 128
- Kugel 146, 161
- LabelmitMouseEvent 265, 292
- Lehrer 147, 163
- LeseProdukt 457, 501
- ListenerKlasse 213, 251
- LiteraleJava7 658, 711
- LWundHWButtonMix 699, 773
- MalenmitderMaus 264, 293
- MemberKlassen 383, 395
- MemberKlassenStatic 384, 399
- Menge 149, 165
- MengenDuplizieren 158, 181
- MengenDuplizierenTest 158, 183
- MengeTest 149, 166
- MethodenAufrufer 36, 61
- MethodenAufrufe2 37, 61
- MethodenAufrufeTest 37, 62
- MethodenmitVararg 56, 96
- MethodenParameter 40, 67
- MouseAdapterfuergetGraphics 261, 282
- MultipleExceptionTypen 691, 759
- MyButton 319, 362
- MyButtonUI 319, 363
- MyException 551, 566
- MyException1 553, 572
- MyLayerUI1 706, 785
- MyLayerUI2 706, 785
- MyLayerUI3 706, 786
- MyLookAndFeel 321, 371

- MyStdButtonUI 320, 371
- Name 457, 504
- NashiBirne 683, 730
- NavigableMapsundSets 590, 614
- Nelke 117, 143
- NewInstanceException 547, 558
- Objekt 156, 178
- Objekt1 154, 175
- Objekt2 155, 177
- Objekte 111, 128
- ObjektInstanziierung 35, 59
- Obst 681, 729
- Oktaeder 152, 173
- PackageTest 44, 78
- PackageTestr 45, 80
- PaintKreis 206, 238
- Parallelogramm 112, 132
- Permutationen 114, 135
- PerPixelTranslucentFrame 697, 770
- Person 147, 162
- PersonTest 147, 163
- PersonTyp 147, 163
- PersonTypTest 148, 164
- PointEqual 441, 480
- PrivaterKonstruktor 58
- PrivaterKonstruktorTest 58
- Produkt 459, 506
- ProgrammArgumentTest 550, 565
- ProgrammArgumentTestmitMyExcepti-
on 551, 567
- Punkt 38, 64, 110, 125
- PunktNeu 110, 127
- PunktStreckeKreisTest 50, 84
- PunktTest 38, 64
- Quadrat 112, 131
- QuadratDefinition 37, 63
- QuadratDefinitionTest 37, 63
- RadioButtonEventundListener 389, 418
- RationaleZahlen 41, 73
- RawTyp 440, 476
- Rechteck 112, 131
- ReflectionmitTypeInterface 461, 513
- RegelmaessigeGeometrischeKoerper 152,
172
- RegelmaessigeGeometrischeKoerperTest
152, 174
- RethrowingExceptionsmitJava7 693, 761
- Rose 117, 142
- SchreibProdukt 457, 501
- Schueler 147, 163
- ShapedFrame 697, 772
- SprachundAusnahmeErweiterungen 587,
610
- Strasse 457, 504
- Strecke 49, 82
- StringTest 54, 93
- SubTypingfuerReferenzTypen 689, 752
- SwingComponent 208, 245
- SwingContainer 208, 246
- SwingLabel 202, 228
- SwingPaintComponent 208, 246
- Tage 449, 493, 494, 564
- Tetraeder 152, 172
- TransparenteFenster 697, 767
- Tulpe 117, 142
- TypLehrer 148, 164
- TypSchueler 148, 164
- TypUmwandlungen 145, 160
- TypvariablenundParametrisierteTypen
679, 725
- UIDefaultsTreeModel 462, 518
- UIDefaultsTreeModelmitTreeModelListe-
ner 463, 522
- UIDefaultsTreeModelundLookAndFeel
464, 526
- UngepruefteExceptionsmitBehandlung
547, 557
- UngepruefteExceptionsohneBehandlung
546, 556
- UniformTranslucentFrame 697, 768
- UnterklassenvonWindowAdapter 211, 249
- UserButton 266, 297
- UserButtonUI 312, 349
- UserButtonUI1 312, 349
- UserButtonUI2 312, 350
- UserComboBoxModel 305, 326
- UserComboBoxModelohneEventBehand-
lung 303, 324
- UserComboBoxModelohneEventBehand-
lungTest 304, 325
- UserComboBoxModelTest 305, 328
- UserDefaultTreeModel 309, 338
- UserEventundListener 390, 422
- UserGlassPane 258, 274
- UserKomponente 389, 421
- UserListModel 468, 539
- UserListModelmitEventBehandlung 466,
534
- UserListModelmitEventBehandlungTest
467, 536
- UserListModelohneEventBehandlung
465, 530
- UserListModelohneEventBehandlung-
Test 465, 531
- UserRadioButton 266, 297
- UserTableModel 595, 623
- UserTextField 266, 296
- Variationen 114, 136
- VariationenhabenPermutationen 115, 139
- Vektor 39, 65
- VektorKopieren 157, 179
- VektorKopierenTest 157, 180
- VektorTest 39, 66
- ViereckeTest 113, 134

- WildcardTypen 681, 731
- WochenTage 58, 102
- WochenTageTest 58, 103
- Wohnort 457, 504
- Wort 158, 183
- Worte 111, 129
- WrapperKlassen 51, 85
- WrapperKlassenmitAutoBoxing 51, 87
- WrapperKlassenmitAutoBoxingTest 52, 89
- WrapperKlassenTest 51, 86
- Wuerfel 152, 172
- Zahl 158, 183
- Zahlen 111, 129
- ZahlenTest 41, 74
- Klassenfelder 31, 32, 36
 - selbstreferenzierende 57
- Klassen-Feld-Initialisierer 56, 57
- Klassenhierarchie 31, 147
- Klassenlader 33, 43
- Klassenmethoden 31, 32, 35, 42, 49, 109, 150
 - statisches Binden 107, 108
 - verdecken 107
- Klassenobjekt 453, 459, 461
- Klassenparameter 435
- Klassenvariablen 33
- Komponente 187
 - bevorzugte Größe 189
 - Eigenschaften 189
 - Kind 187
 - leichtgewichtige 188
 - maximale Größe 189
 - minimale Größe 189
 - schwergewichtige 188
 - Standard 191
 - Vater 187
- Komponenten-Events 209, 211
- Komposition, Anreihung 110
- Konsole 46
- Konstantendefinition 31, 41, 150
- Konstruktor 31, 32, 51, 449
 - privater 57, 58
- Konstruktordefinitionen 39
- Konvertierung siehe Typumwandlung
- Kovariante Rückgabetypen 111, 440, 441

L

- Label 201, 205
- LayeredPane 253, 254, 255, 257, 393
- LayerUI 704, 706, 707
- Layout-Manager 187, 189
- LayoutManager 199, 201
- LayoutManager2 199
- Legacy code 439
- Level 587
- List 304, 446, 683
- ListDataEvent 305

- ListDataListener 303, 305, 465
- ListModel 303, 465
- Literal 657, 658
- Locale 588, 589
- log() 587
- Logger 587
- Logging 587
- Long 50, 145
- look and feel 301, 464
 - spezifisch 193, 314, 315, 319, 554
- LookAndFeel 315, 316, 320, 321, 554, 697
- LookAndFeel-Defaults 315
- lower bound wildcard
 - untere Schranke 436, 438, 667
- LW-Komponente 188, 190, 198, 204, 205, 699

M

- main() 34, 391, 393
- Map 445, 450, 549, 590
- Marker-Interface 153, 209, 316
- Maskenprototyp 201
- Math 46, 48, 49, 257
- Member 453
- MenuBar 253
- MetalButtonUI 312, 318, 320
- MetalLookAndFeel 366, 700
- MetalScrollButton 438
- Methoden 31, 50, 449
 - erben 105, 127, 150, 311, 314
 - generische 443, 444, 664, 679, 681
 - überladen 37, 56
 - überschreiben 147, 155
- Modalitytyp 592, 593
- Model 189, 321
- Modelle 302, 594
 - ComboBox 303, 305
 - DefaultComboBox 302
 - List 465, 466, 467
 - Tree 308
 - UIDefaults-Tree 462, 464
- Modelle *siehe* Model
- Modellklassen 302
- Modifikator 31, 38, 44, 45, 105, 110
- MotifLookAndFeel 366
- MouseAdapter 261, 389
- mouseClicked() 314
- mouseDragged() 391
- MouseEvent 209, 707
- MouseListener 257, 260, 261, 265, 389
- MouseMotionListener 265, 391, 392
- mousePressed() 260, 265
- moveToBack() 393
- moveToFront() 392
- Multi-catch-Klausel 690
- MVC-Architektur
 - Model-View-Controller-Architektur 189, 301

N

NavigableMap 590
 NavigableSet 590
 newInstance() 53, 442, 443, 458, 547, 598, 672, 691, 693, 694
 new-Operator 31, 32, 33, 53, 57, 150, 159, 199, 382, 383, 433, 442, 456, 671, 705
 next() 450
 nextElement() 450, 451, 463
 NimbusLookAndFeel 700, 702
 NoSuchFieldException 691
 NoSuchMethodException 691
 null-Layout 204
 NullPointerException 588
 Number 145, 146, 437, 678
 NumberFormatException 550

O

Oberinterfaces 151
 Oberklasse, Superklasse 105, 108, 109, 114, 117, 145, 310
 Oberklassenreferenz 116
 Object 46, 109, 110, 149, 153, 154, 434
 Objekt 31
 -aktuelle (aufrufende) 32, 38, 157
 -beseitigen 159
 -Gleichheit 156
 -klonen 154
 Objektzustände 41, 198, 334
 Operatoren 50
 -arithmetische 50
 -logische 50
 ordinal() 448

P

pack() 194, 195, 197, 207
 package-Anweisung 43, 44, 107
 Paint 696
 paint() 192, 194, 195, 196, 197, 198, 205, 208, 260, 261, 262, 281, 309, 313, 314, 319, 386, 555, 704, 705, 708
 paintBorder() 193, 306, 311, 313, 317, 322
 paintButtonPressed() 312
 paintChildren() 193, 204, 208, 314
 paintComponent() 193, 194, 197, 199, 258, 265, 306, 310, 311, 322, 382, 698
 Painter 702
 Pakete 43, 45, 107, 315
 Panel 188
 ParameterizedType 456, 459, 461
 Parametervariable 33
 Parametrisierung 436, 662
 parseDouble() 54
 parseInt() 54, 546, 548
 parseShort() 54
 PlainDocument 301, 306

Point 255, 262, 392, 441
 Polygon 386
 Polymorphismus 116, 146
 -implizite 116, 201, 433
 -parametrisierte 116, 433
 -Subtyp 117, 147
 -überladene 116, 612
 primitive Datentypen 31, 146, 156
 printStackTrace() 545
 PrintStream 34, 46
 Process 46
 ProcessBuilder 46, 47
 ProcessBuilder.start() 47
 processMouseEvent() 708
 Programmdateien 45
 Programmunterbrechung 550
 Prozess 47
 Punkt-Operator 42
 put() 315, 317, 318, 321, 448, 549, 590, 701, 703
 putAll() 549
 putClientProperty() 701
 putDefaults() 321

R

Rand *siehe* Border
 random() 257
 Raw-Typ 439, 440, 660, 665
 Rectangle 386
 Rectangle2D 263
 Rectangle2D.Float 263, 264, 386
 RectangleShape 263
 Referenzen 31
 -globale 40
 -lokale 40
 Referenztyp 31, 116, 146, 148, 155, 158
 Referenzvariable 33, 147, 153, 437
 Reflection 452, 460, 691
 ReflectiveOperationException 690
 regexFilter() 594
 remove() 308
 removeNodeFromParent() 465
 removeTreeModelListener() 463
 repaint() 192, 198, 260, 262
 Resizing 208
 retainAll() 549
 Rethrowing 690, 693
 RGB-Modell 194, 317, 696
 RootLayout 253
 RootPane 253, 393
 RootPaneContainer 253
 RoundRectangle2D 698
 RowFilter.Entry 597
 RowSorter 594
 Runnable 707
 Runtime 46, 47, 591
 Runtime.exec() 47
 RuntimeException 546, 550, 551, 588

S

Schnittstelle siehe Interface

Schnittstellendefinitionen

- Equality 441, 480
- Farben 151, 170
- FarbenDunkel 151, 170
- FarbenHell 151, 170
- FarbeundSchrift 265, 291
- FlaechenInhalt 150, 167
- InKugelRadius 152, 171
- JavaKomponenten 598, 630
- KomponentenListener 266, 296
- LabelFarben 203, 224
- Services 148, 165
- StackInterface 551, 570
- Tabelle 594, 620
- Texte 600, 634
- UmKugelRadius 152, 171

Schrift *siehe* Font

ScrollPane 188

Serializable 687

Set 445, 450, 683

setBackground() 195, 202, 392, 696, 698

setBounds() 187, 188, 196, 200, 204, 254, 256, 272, 392, 699

setColor() 195

setComponentZOrder() 188, 206, 254

setContents() 603

setDefaultCloseOperation() 191, 195

setDefaultLookAndFeelDecorated() 695

setDocument() 302

setDragEnabled() 599

setDropMode() 607

setEventMask() 708

setInt() 53

setJMenuBar() 258, 602

setLayer() 255

setLayeredPane() 256

setLayerEventMask() 707

setLayout() 199, 204

setLocation() 188, 200

setLookAndFeel() 316, 318, 319, 554, 555

setMaximumSize() 189

setMenuBar() 255, 393

setMinimumSize() 189

setModal() 592

setModalExclusionType() 592

setModalityType() 592

setModel() 302, 305, 308, 467

setOpacity() 696, 698

setOpaque() 190, 198, 202, 206

setPreferredSize() 189, 195, 311

setRowSorter() 595

setShape() 696, 698

setSize() 188, 200

setTabComponentAt() 598

setText() 600

setTransferHandler() 600

setUI() 313, 316, 322, 554

setUndecorated() 695

setVisible() 194, 214

shallow cloning

oberflächliches Klonen 157

Shape 263, 264, 319, 386, 698

Short 50, 145

showDialog() 391

Sichtbarkeitsebenen 38, 45, 106

Skinning 702

sort() 683

SortedMap 445, 549

SplashScreen 591

Standardausgabe 46

Standardeingabe 46

Standard-Konstruktor 32

Standard-Layout-Manager 200, 201

Steuerung *siehe* Controller

String 50, 54, 201, 202, 265, 392, 436, 587, 687

StringSelection 599, 603

subMap() 549

substring() 54, 546, 548

Subtyp 111, 116, 155, 205, 434, 660

Subtyping 668, 689

super() 105, 551

super.paint() 215, 217, 281

super.paintChildren() 204, 209

super.paintComponent() 193, 199, 204, 206, 209, 306

Supertyp 434, 660

Swing 187

SwingConstants 593

Swing-Container 187

Swing-Komponente 187, 189, 201, 202, 208, 253, 260, 303, 309

SwingUtilities 707

switch-Anweisung 41, 657, 658, 697, 708

System 34, 46, 545

System.exit() 47

System.gc() 159

System.out.print() 34

System.out.println() 34

System-Defaults 315

Systemeigenschaften 47

T

Tab

Registerkarte 597

TableRowSorter 594

tailMap() 549

Tastatureingaben 209

TextComponent 603

TextListener 266
 TexturePaint 696
 this-Referenz 38
 Throwable 545, 553
 throw-Anweisung 47, 587, 691
 throws 48, 546, 547, 691
 TitledBorder 256
 toArray() 447
 toBinaryString() 50
 toHexString() 50
 toOctalString() 51
 Toolkit 194, 599
 toString() 53, 110, 149, 153, 304, 449, 548
 toUpperCase() 306
 Transferable 599, 600, 603, 606
 TransferHandler 600, 601, 605
 TransferHandler.DropLocation 607
 TransferHandler.TransferSupport 607, 608, 610
 translate() 263
 translation-Parameter 192
 Transparenz-Eigenschaft *siehe* opaque-Eigenschaft
 TreeMap 445, 447
 TreeModel 307, 463
 TreeModelEvent 464
 TreeModelListener 308, 463, 464
 TreeNode 308
 treeNodesChanged() 464
 treeNodesInserted() 464
 treeNodesRemoved() 464
 TreePath 308, 610
 TreeSelectionListener 308
 TreeSelectionMode 307
 try-Block 48, 545, 549, 691
 Typ
 -Disjunction 690, 691
 -dynamischer 661
 -generischer 434, 435, 441, 662
 -Intersection 663, 690
 -parametrisierter 433, 662, 664, 681, 689
 -reifizierbarer 439, 663
 -statischer 461, 659, 661, 674
 -Wildcard-parametrisierter 438, 667, 690
 Typargument 433, 660, 681
 Type 455, 459, 460, 461
 TypeVariable 456, 461
 Typinferenz 662, 670, 671, 677
 Typinvarianz 667, 681
 Typisierung 662
 Typkompatibilität 439
 Typlöschung *siehe* Erasure
 Typparameter 154, 433, 435, 660, 662
 Typprüfung 434, 661, 690
 Typsicherheit 51, 433, 439, 458, 662, 686

Typsystem 659
 Typumwandlung 146, 434
 Typvariable 433, 436, 662

U

Überlappen von Komponenten 190, 204
 Über-Typ 434, 435
 UIDefaults 315, 321, 451, 463, 702
 UIDefaults-Tabelle 316, 317, 318, 319, 320, 321, 451, 554, 555
 UI-Delegate *siehe* UI-Delegationsobjekt
 UI-Delegationsklasse 189, 193, 199, 301, 306, 309, 312, 314, 315, 319, 320, 321, 554, 555, 704
 UI-Delegationsobjekt 189, 190, 199, 317
 UIManager 315, 317, 318, 319, 463
 UIResource 315, 316, 321
 Umgebungsvariable 43
 uninstallUI() 309, 707
 UnsupportedFlavorException 603
 UnsupportedLookAndFeelException 554
 Unterklasse, Subklasse 105, 109, 115, 117, 145
 Unterklassenreferenz 116
 up-casting
 vergrößernde Konvertierung 114, 117, 139, 147
 update() 192, 193, 199, 260, 262, 265, 309
 updateComponentTreeUI() 701
 upper bound wildcard
 obere Schranke 436, 437, 667
 UserListModel 468

V

valueOf() 548, 609
 values() 448, 549, 593, 609
 varargs-Methoden 55, 658
 Variable
 -lokale 33, 40, 56, 384, 391
 Vector 446, 599
 Vererbung 105, 109, 153
 Viewer 189, 302, 305, 309, 321, 466, 467, 594
 Vordergrund 316

W

Web Browser 34, 188
 while-Schleife 43
 Wildcard-Ausdruck *siehe* Wildcardtypen
 Wildcard-Capture 675, 682, 684
 WildcardType 456
 Wildcardtypen 436, 443, 662, 667, 681
 Window 188, 210, 695
 WindowAdapter 210, 387, 388
 WindowEvent 209, 210, 212
 WindowListener 210, 262
 Wrapper-Klassen 50, 51

Z

Zeichenoberfläche 191, 208, 261, 265, 382, 386,
391
Zeichenoperationen 207, 259, 314

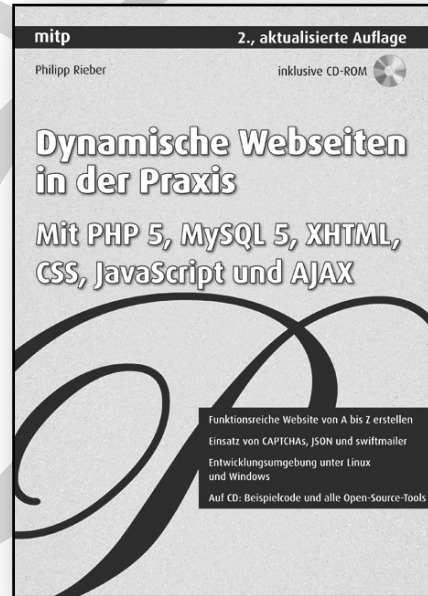
Z-Order 188, 699
-Index 188, 206, 207, 238, 242, 254, 255,
257, 268, 270, 271
Zugriffsmethoden 38, 42, 110, 114, 438
Zwischenablage *siehe* Clipboard
701

Philipp Rieber

Dynamische Webseiten in der Praxis

Mit PHP 5, MySQL 5, XHTML, CSS,
JavaScript und AJAX

- Funktionsreiche Website von A bis Z erstellen
- Einsatz von CAPTCHAs, JSON und swiftmailer
- Entwicklungsumgebung unter Linux und Windows
- Auf CD: Beispielcode und alle Open-Source-Tools



Anhand eines konkreten, groß angelegten Webseiten-Projekts zeigt Ihnen Philipp Rieber, wie Sie eine dynamische, funktionsreiche Website selbstständig aufbauen, programmieren, verwalten und pflegen. Angefangen bei der Konzeption, dem Seitenaufbau, dem Layout und dem Datenbankdesign bis hin zur Erstellung einer Administrationsoberfläche, der Hauptseite und den dynamischen Anwendungen werden Sie Schritt für Schritt den alltäglichen Umgang mit XHTML, CSS, PHP, MySQL, JavaScript und AJAX erlernen und darüber hinaus phpMyAdmin, Apache und Postfix einsetzen. Dabei arbeiten Sie ausschließlich mit Open-Source-Technologien, die frei verfügbar sind und die Sie auf der beiliegenden Buch-CD finden.

Das Besondere an diesem Buch ist sein Praxisbezug: Der Autor erstellt mit Ihnen von Anfang bis Ende eine komplette, funktionsreiche Website, für die Sie den gesamten Code auf der Buch-CD erhalten. Diese Website können Sie sich darüber hinaus im Internet im Live-Einsatz ansehen. Sie können den Code für Ihre eigene Website einsetzen, weiterentwickeln oder Code-Bausteine daraus verwenden.

Das Buch wendet sich an Einsteiger und Fortgeschrittene in der Webprogrammierung. Für Fortgeschrittene steigt der Autor direkt in die Programmierung der Website ein. Hierbei können Sie Ihr Wissen ausbauen, in der Praxis umsetzen und professionelle Techniken erlernen. Für Einsteiger bietet der zweite Teil des Buches einen Crashkurs, der sie auf alle benötigten Techniken des Projekts vorbereitet.

Alle benötigten Werkzeuge und den vollständigen Code der Website und der Beispiele finden Sie auf der beiliegenden Buch-CD.

Auf der CD:

Kompletter Quellcode des Beispiel-Projekts, alle Beispielskripte des Crashkurses, Apache, PHP, MySQL, phpMyAdmin, Postfix, swiftmailer, JUpload

Leserstimme zur Voraufgabe:

»Sehr gutes Buch, besonders durch den hohen Praxisbezug!«

Wertung bei amazon.de: ★★★★★

Probekapitel und Infos erhalten Sie unter:
www.mitp.de/1782

ISBN 978-3-8266-1782-9

Uwe Rozanski

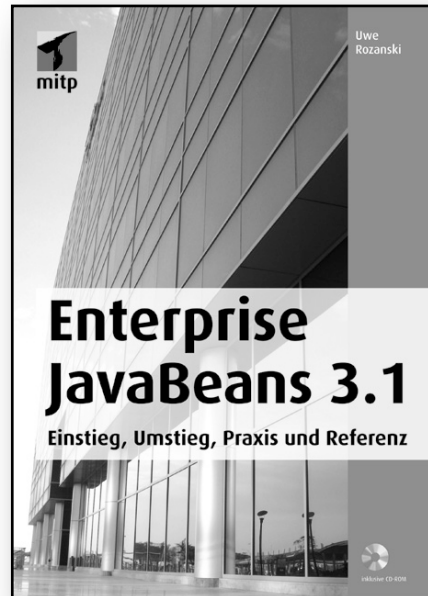
Enterprise JavaBeans 3.1

Einstieg, Umstieg, Praxis und Referenz

- **Aufbau, Anwendung und Einsatz aller Beanarten**
- **Java Persistence API, Criteria API, XML Deployment Descriptor, Transaktionsschutz, Security, Webservices**
- **Praxisbeispiele und Übungen mit Musterlösungen zu jedem Kapitel, vollständiges Beispiel eines Webshops**

Der Einsatz von Enterprise JavaBeans ist bei der Programmierung von serverseitigen Unternehmensanwendungen Standard. Dieses Buch behandelt den neuesten Entwicklungsstand der Enterprise JavaBeans und die Java Persistence API. Es bietet eine übersichtliche Darstellung der komplexen EJB-Welt und zeigt anhand zahlreicher praktischer Beispiele, Übungsaufgaben und Musterlösungen, wie Sie effizient und erfolgreich EJBs programmieren. Gleich zu Anfang wird eine Entwicklungsumgebung mit Eclipse und JBoss aufgebaut, damit die Beispiele in der Praxis nachvollzogen werden können.

Uwe Rozanski erläutert ausführlich den Aufbau und die Anwendung jeder einzelnen Beanart. Jedes Kapitel enthält zahlreiche Beispiele, die den konkreten Einsatz der Beans anhand von Sourcecode aufzeigen. In einem großen Teil des Buches wird eine Kundenverwaltung aufgebaut und weiterentwickelt, um die praktische Verwendung der einzelnen Beanarten konkret aufzuzeigen.



Zusätzlich werden auch so wichtige Themen behandelt wie der Transaktionsschutz, die Verwendung von XML Deployment Descriptoren, Security und Webservices. Des Weiteren geht der Autor in einzelnen Abschnitten auf Unterschiede zu EJB 2.1 und EJB 3.0 ein. Ein vollständiges Beispiel eines Webshops rundet das Buch ab. Hier finden Sie auch wichtige Hinweise für die Erstellung größerer Projekte.

Mit diesem Buch wird es Ihnen gelingen, die kompliziert wirkende EJB-Welt zu verstehen und EJBs effektiv für Ihre Anwendungen einzusetzen. Das Buch ist so aufgebaut, dass es sich sowohl zum Selbststudium als auch als Seminarunterlage eignet. Durch den strukturierten Aufbau ist es ebenfalls langfristig als Referenz einsetzbar.

Über den Autor:

Uwe Rozanski hält seit 1988 Seminare zu Großrechner- und Client/Server-Systemen, hat in zahlreichen IT-Projekten umfangreiche Erfahrungen gesammelt und bereits Bücher zu Cobol, Silverlight und EJB 3.0 geschrieben.

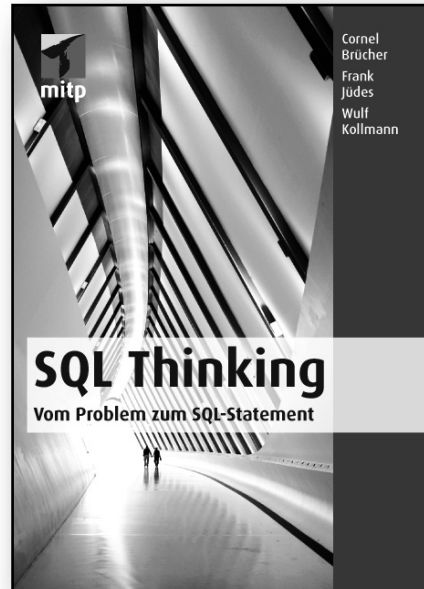
Probekapitel und Infos erhalten Sie unter:
www.mitp.de/9066

ISBN 978-3-8266-9066-2

Cornel Brücher
Frank Jüdes
Wulf Kollmann

SQL Thinking

Vom Problem zum
SQL-Statement



Dies ist ein Buch mit einzigartiger Herangehensweise: von der rein kaufmännischen Fragestellung hin zum SQL-Statement.

Wer mit Oracle-Datenbanken arbeitet, dort Daten anhand von fachlichen Anfragen auswertet und beispielsweise eine Tabellenkalkulation verwendet, um kaskadierende Summen zu bilden oder eine Pivottabelle zu erstellen, der braucht dieses Buch. Hier lernt man, aus einer Frage heraus ein SQL-Statement zu formulieren, mit Ergebnismengen zu jonglieren und diese kreuz und quer zu verbinden.

Cornel Brücher befasst sich seit 1980 beruflich und privat mit Computern und arbeitete schon 1989 mit relationalen Datenbanken. Auf Oracle-Technologien spezialisierte er sich Ende der 90er Jahre bei Oracle Consulting in Deutschland. Seit 2000 ist er für verschiedene Unternehmen als Berater im Einsatz, seit 2007 im Data Warehouse-Umfeld.

Frank Jüdes befasst sich seit 1981 beruflich und privat mit Computern und hat die Entwicklung der relationalen Datenbanken von dBase III über Paradox(e) Clipper zu MySQL und Oracle in voller Länge miterleben dürfen. Seine Oracle-Erfahrungen sammelte er ab 1999 bei Oracle Consulting in Deutschland, seit 2007 arbeitet er als Security-Spezialist für Oracle America Inc. in den USA.

Wulf Kollmann studierte Informatik mit BWL und kombiniert seit 1989 beide Fachrichtungen bei Projekten im In- und Ausland. Er entwickelt seit 1994 im eigenen Unternehmen maßgeschneiderte kaufmännische Datenbank-Anwendungen für Kunden aus verschiedensten Branchen.

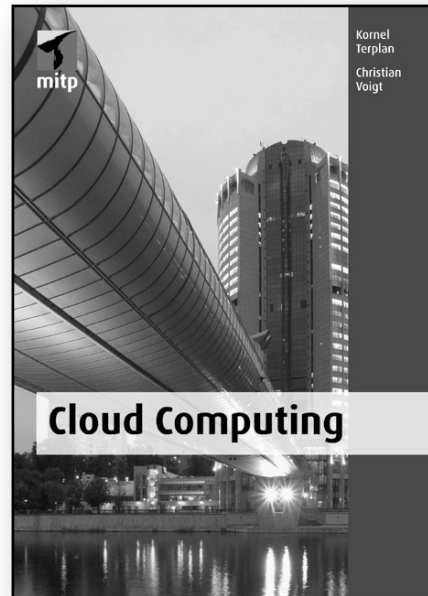
Probekapitel und Infos erhalten Sie unter: www.mitp.de/5505

ISBN 978-3-8266-5505-0

Kornel Terplan, Christian Voigt

Cloud Computing

- **Strategischer Leitfaden für CIOs**
- **Standardisierung, Individualisierung**
- **Sicherheit, Performance Management, Partnermanagement**



Es gibt wohl keinen CIO in mittleren bis großen Unternehmen, der heute nicht vor der Frage steht, wieviel IT er denn in die Cloud auslagern kann, darf oder muss? Die Cloud verspricht eine drastische Reduktion der Kosten und des Administrationsaufwands, aber was bietet die Cloud an Sicherheit, Verfügbarkeit, Kompatibilität, Skalierbarkeit und Performance?

CIOs, die bei diesem Hype-Thema den Überblick und einen klaren Kopf behalten wollen, erhalten mit diesem Buch einen wichtigen und wertvollen Leitfaden und Ratgeber. Sie werden dabei von dem Fachwissen von Autoren profitieren, die über langjährige Projekt- und Trainingserfahrung, über großes technisches Know-how, aber auch über die für dieses Thema notwendige strategische Kompetenz verfügen.

Dr. Kornel Terplan hat sich in seiner langen beruflichen Laufbahn vor allem mit folgenden Themen befasst: System- und Netzwerkmanagement, Business und Operations Support Systems für Telecommunications Service Provider, Intercept-Systeme, Information Lifecycle Management, Benchmarking, Einsetzbarkeit von Open Source sowie Business Intelligence & Analytics.

Christian Voigt war vor allem in folgenden Bereichen tätig: Verkabelungsstrategie, Corporate-IP Netzdesign und Netzbetrieb, Next Generation Network Architecture, Shared Services für Corporate Benutzer, Partner Ready Information Infrastructure zur Ermöglichung einer hochklassigen Collaboration, Umstrukturierung der Rechenzentren in Service Areas sowie Entwurf und Durchführung von Business-Partner-Management-Prozessen.

Probekapitel und Infos erhalten Sie unter:
www.mitp.de/9098

ISBN 978-3-8266-9098-3

Nicholas Carr

The **BIG SWITCH** Der große Wandel

Die Vernetzung der Welt
von Edison bis Google



»Hochinteressant. Carr belegt mit einer umfassenden historischen Analogie, dass Computer-Versorgungsunternehmen die firmeneigenen IT-Abteilungen ersetzen werden so wie die Stromversorger die firmeneigenen Generatoren verdrängt haben ... *The Big Switch* ist eine beeindruckende Diskussion der positiven und negativen Aspekte des kommenden World-Wide-Computer-Zeitalters.«

Thomas P. Hughes, Autor von *Human-Built World* und *American Genesis*

Sein letztes Buch erschütterte die Hightech-Industrie bis in ihre Grundfesten. Jetzt ist Nicholas Carr wieder da. *The Big Switch* gibt einen umfassenden und oft provokanten Blick auf eine neue Computer-Revolution, die Wirtschaft, Gesellschaft und Kultur grundlegend verändern wird.

Carr zieht eine Analogie zu der Revolution, die sich vor knapp hundert Jahren bei der Elektrizität vollzog, als Unternehmen aufhörten, mit Dampfmaschinen und Dynamos ihre eigene Energie zu erzeugen und sich stattdessen dem neu errichteten Stromnetz anschlossen. Diese Entwicklung setzte eine Kettenreaktion wirtschaftlicher und sozialer Transformationen in Gang, durch die unsere moderne Welt entstanden ist. Heute befinden wir uns mitten in einer ähnlichen Revolution. Diesmal ist es Rechenleistung, die zu einem Versorgungsgut wird.

Carr stellt die Prognose auf, dass große Serverfarmen die heute gängigen PCs ablösen werden, so dass so gut wie keine Information und Rechenleistung mehr offline verfügbar sein wird. Dies wird dazu führen, dass neue Wettbewerber, wie etwa Google oder Salesforce.com, alteingesessene Platzhirsche wie Microsoft oder Dell bedrohen werden. Aber die Auswirkungen werden sehr viel weiter reichen. Billige Rechenleistung wird die Gesellschaft letztlich so grundlegend ändern wie die billig gewordene Elektrizität vor hundert Jahren.

Der Wandel hat aber bereits heute große Umwälzungen in der Computerindustrie zur Folge. Vom Software- zum Zeitungsgeschäft, von der Schaffung von Arbeitsplätzen bis zur Bildung von Communities, liefert *The Big Switch* einen panoramaartigen Überblick über eine neue Welt, die durch den »World Wide Computer« entsteht.

Nicholas Carr ist ideal dafür geeignet, diesen historischen Wandel zu erläutern. Er schreibt in einem klaren, engagierten Stil und webt die Entwicklungsstränge von Geschichte, Wirtschaft und Technologie zusammen, um zu beschreiben, wie und warum sich Computer ändern und was das alles für uns bedeutet.

Probekapitel und Infos erhalten Sie
unter: www.mitp.de

ISBN 978-3-8266-5508-1

Patrick
Ditchen

Shell-Skript Programmierung

- Die 100 wichtigsten UNIX-Tools
- Grafiken mit gnuplot
- sh, ksh, bash, csh, tcsh; awk
- Systemadministration und proaktives Systemmanagement



4. Auflage

Das vorliegende Buch ist dreigeteilt. Im ersten Teil, das sind Kapitel 1 bis 3, lernt der Leser Kommandos, Techniken und Konzepte der Shell-Skript-Programmierung kennen. Der zweite Teil – Kapitel 4 und 5 – beschäftigt sich mit awk und über 100 weiteren UNIX-Tools, die in den Shell-Skripten einzubauen und aufzurufen sind. Der dritte Teil - Kapitel 6 bis 8 - zeigt, wie man das erlernte Wissen einsetzt, um die unterschiedlichen Aufgaben aus dem Alltag der System-Administration zu lösen.

Die beiliegende CD enthält alle im Buch vorgestellten Skripte in einer Solaris- und einer Linux-Fassung, sortiert nach Kapitel-Nummer.

Wenn die Skripte auf Dateien mit Messdaten zurückgreifen müssen, sind auch diese hinzugefügt. Der Leser kann also parallel zum Studium des Buches alle Skripte gleich testen.

»Wer Sys-Admin ist und wirklich praxisnahe Probleme mit Shell-Skripten lösen möchte/muss, weiss mit diesem Buch auf seinem Schreibtisch ein wunderbar leicht und verständlich geschriebenes Nachschlagewerk zu schätzen. Anstatt drei oder vier »Shell-Script-Bücher« zu wälzen, genügt nun nur noch eines. Nämlich dieses.«

Leserrezension zur ersten Auflage
bei amazon.de

Probekapitel und Infos erhalten Sie unter:
www.mitp.de/9134

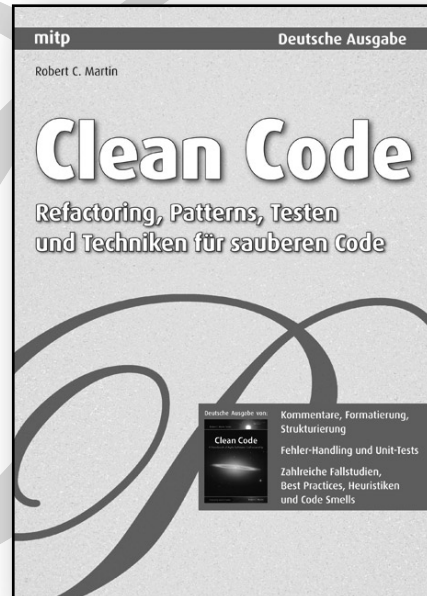
ISBN 978-3-8266-9134-8

Robert C. Martin

Clean Code

Refactoring, Patterns, Testen
und Techniken für sauberen Code

- Kommentare, Formatierung,
Strukturierung
- Fehler-Handling und Unit-Tests
- Zahlreiche Fallstudien,
Best Practices, Heuristiken
und Code Smells



Selbst schlechter Code kann funktionieren. Aber wenn der Code nicht sauber ist, kann er ein Entwicklungsunternehmen in die Knie zwingen. Jedes Jahr gehen unzählige Stunden und beträchtliche Ressourcen verloren, weil Code schlecht geschrieben ist. Aber das muss nicht sein.

Mit Clean Code präsentiert Ihnen der bekannte Software-Experte Robert C. Martin ein revolutionäres Paradigma, mit dem er Ihnen aufzeigt, wie Sie guten Code schreiben und schlechten Code überarbeiten. Zusammen mit seinen Kollegen von Object Mentor destilliert er die besten Praktiken der agilen Entwicklung von sauberem Code zu einem einzigartigen Buch. So können Sie sich die Erfahrungswerte der Meister der Software-Entwicklung aneignen, die aus Ihnen einen besseren Programmierer machen werden – anhand konkreter Fallstudien, die im Buch detailliert durchgearbeitet werden.

Sie werden in diesem Buch sehr viel Code lesen. Und Sie werden aufgefordert, darüber nachzudenken, was an diesem Code richtig und falsch ist. Noch wichtiger: Sie werden

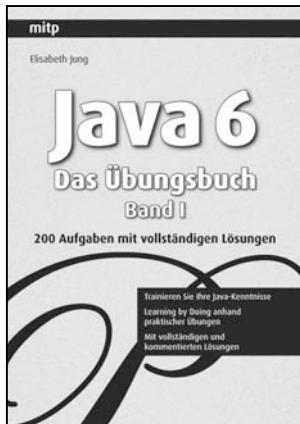
herausgefordert, Ihre professionellen Werte und Ihre Einstellung zu Ihrem Beruf zu überprüfen.

Clean Code besteht aus drei Teilen: Der erste Teil beschreibt die Prinzipien, Patterns und Techniken, die zum Schreiben von sauberem Code benötigt werden. Der zweite Teil besteht aus mehreren, zunehmend komplexeren Fallstudien. An jeder Fallstudie wird aufgezeigt, wie Code gesäubert wird – wie eine mit Problemen behaftete Code-Basis in eine solide und effiziente Form umgewandelt wird. Der dritte Teil enthält den Ertrag und den Lohn der praktischen Arbeit: ein umfangreiches Kapitel mit Best Practices, Heuristiken und Code Smells, die bei der Erstellung der Fallstudien zusammengetragen wurden. Das Ergebnis ist eine Wissensbasis, die beschreibt, wie wir denken, wenn wir Code schreiben, lesen und säubern.

Dieses Buch ist ein Muss für alle Entwickler, Software-Ingenieure, Projektmanager, Team-Leiter oder Systemanalytiker, die daran interessiert sind, besseren Code zu produzieren.

Probekapitel und Infos erhalten Sie unter:
www.mitp.de/5548

ISBN 978-3-8266-5548-7



ISBN 978-3-8266-1780-5

★★★★★ bei Amazon



ISBN 978-3-8266-5956-0



ISBN 978-3-8266-5603-3

- **Trainieren Sie Ihre Java-Kenntnisse**
- **Learning by Doing anhand praktischer Übungen**
- **Mit vollständigen und kommentierten Lösungen**

Diese Bücher sind keine Lehrbücher, sondern reine Übungsbücher und wenden sich an Leser, die ihre Java-Kenntnisse anhand zahlreicher praktischer Übungen durch „Learning by Doing“ vertiefen und festigen möchten. Sie sind ideal, um sich auf Prüfungen vorzubereiten oder das praktische Programmieren mit Java zu üben.

Jedes Kapitel enthält zunächst eine kompakte Zusammenfassung des Stoffes, der in den Übungsaufgaben dieses Kapitels verwendet wird. Anschließend haben Sie die Möglichkeit, zwischen Aufgaben in drei verschiedenen Schwierigkeitsstufen – von einfach bis anspruchsvoll – zu wählen. Anhand dieser Aufgaben können Sie Ihr Wissen praktisch testen. Am Ende des Kapitels finden Sie vollständige und kommentierte Musterlösungen.

Der erste Band behandelt die Themen OOP mit Java, GUI-Programmierung mit AWT und Swing, innere Klassen, Generics, Exceptions und Neuerungen in Java 6.

Der zweite Band behandelt die Themen Dateien und Streams, Multithreading, Java-Applets, Multimedia für Applets und Applikationen, Threads und Animation, Java-Applets und Webbrowser, Netzwerkprogrammierung.

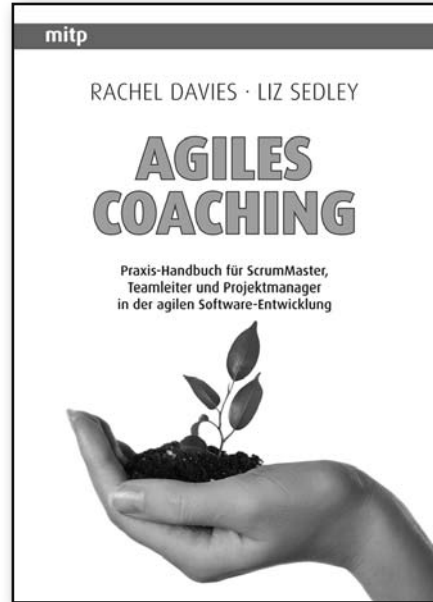
Der dritte Band behandelt die Themen: Java-Bean-Klassen, die Expression Language (EL) und benutzerdefinierte Tags, Tag-Libraries am Beispiel von JSTL, Datenbankbindung von MySQL mit JDBC, die MVC-Architektur von Webapplikationen, Filter und Sicherheitsvorkehrungen in Webapplikationen, das Mapping (Abbilden) von Servlets auf URLs, Webapplikationen zu den Themen Chatrooms und Webshops in der Standard-JSP- und XML-Syntax.

Weitere Infos und Probekapitel unter: www.mitp.de

Rachel Davies · Liz Sedley

AGILES COACHING

**Praxis-Handbuch für ScrumMaster,
Teamleiter und Projektmanager
in der agilen Software-Entwicklung**



Das Coaching agiler Teams ist eine spannende Aufgabe und erfordert Soft Skills, die sich nicht auf der Basis eines Regelwerkes erlernen lassen. Vielmehr zählt hierbei insbesondere die Erfahrung im Umgang mit Teams. Mit diesem Buch können Sie von der Erfahrung zweier langjähriger Coaches lernen und auf diese Weise von Anfang an vieles richtig machen und viele Stolperstellen von vornherein vermeiden.

Rachel und Liz zeigen Ihnen anhand zahlreicher Beispiele, wie Sie als Coach agile Teams bestmöglich bei der Arbeit unterstützen können. In diesem Buch lernen Sie, wie Sie ein agiles Team aufbauen und bei der Arbeit begleiten können, so dass alle Teammitglieder Spaß daran haben, gute Software zu entwickeln. Die Autoren führen Sie durch den gesamten agilen Lebenszyklus und zeigen Ihnen, wie man Teams dazu befähigt, das Beste aus den agilen Methoden herauszuholen.

Sie werden erfahren, was bei der Zusammenarbeit mit Teammitgliedern gut funktioniert und was Sie vermeiden sollten. Anstatt feste Regeln vorzugeben, konzentrieren sich die Autoren auf praktische Ratschläge, Tipps und Erfahrungsberichte, die Ihnen helfen, in eigenen Situationen angemessen zu handeln. In jedem Kapitel listen sie typische Hürden auf mit Lösungsvorschlägen, die Ihnen verdeutlichen, wie Sie Schwierigkeiten meistern können. Sie finden zusätzlich hilfreiche Checklisten, die Ihnen Anregungen geben, worauf Sie achten sollten.

Mit diesem Buch sind Sie für Ihre Tätigkeit als Coach optimal vorbereitet.

Probekapitel und Infos erhalten Sie unter:
www.mitp.de/9046

ISBN 978-3-8266-9046-4

