

Inhalt

Geleitwort des Fachgutachters	17
Vorwort	19

TEIL I: GRUNDLAGEN

1 Einführung 25

1.1 Was ist ein Algorithmus?	25
1.2 Was haben Sie mit Algorithmen zu tun?	27
1.3 Beispiele für Algorithmen	27
1.4 Wie arbeitet ein Computer?	29
1.5 Überblick über Java und den Entwicklungsprozess	31
1.6 Ist Java der Weisheit letzter Schluss?	32
1.7 Was bedeutet Objektorientierung?	33
1.8 Das Wichtigste auf einen Blick	35
1.9 Übungen	36

2 Installation und Konfiguration von Java und BlueJ 37

2.1 Java installieren	37
2.2 BlueJ installieren	39
2.3 BlueJ konfigurieren	40
2.4 Erste Schritte mit BlueJ	40
2.4.1 Das erste Projekt anlegen	40
2.4.2 Die erste Klasse anlegen	41
2.4.3 Quelltext einer Klasse bearbeiten	42
2.5 Weitere Features von BlueJ	45
2.5.1 Projekte dokumentieren	45
2.5.2 Projekte drucken	46
2.5.3 Programme weitergeben	46
2.6 Übungen	47

3 Erste Schritte in Java 49

3.1 Der Rahmen einer Klasse	49
3.2 Klassen in der Objektorientierung	51
3.2.1 Attribute	51
3.2.2 Methoden	52

3.2.3	Optional: Parameter und Rückgabewert	54
3.2.4	Abschließendes zu Methoden	55
3.2.5	Blöcke	56
3.3	Ein Objekt erzeugen	57
3.4	Kommentare	59
3.5	Das Wichtigste auf einen Blick	62
3.6	Übungen	63

4 Variablen, Datentypen und Operatoren 65

4.1	Variablen deklarieren	65
4.1.1	Bezeichner	66
4.1.2	Datentypen	68
4.2	Arten von Variablen	80
4.2.1	Finale und variable Variablen	81
4.2.2	Möglichkeiten der Deklaration	82
4.3	Operatoren	88
4.4	Arithmetische Operatoren	89
4.4.1	Priorität und Assoziativität	89
4.4.2	Anweisungen	91
4.4.3	Ein Beispiel zur Wiederholung	92
4.4.4	Codezeilen testen	93
4.4.5	Inkrement und Dekrement	94
4.4.6	Division und Rest	97
4.5	Datentypen umwandeln	101
4.6	Das Wichtigste auf einen Blick	102
4.7	Übungen	103

5 Bedingungen und Vergleiche 107

5.1	Blöcke	107
5.2	Die if-Anweisung	108
5.3	Vergleichsoperatoren	109
5.4	Mögliche Operanden bei Vergleichen	110
5.5	Die Handlungsalternative else	111
5.6	Logische Operatoren	112
5.6.1	Die Und-Verknüpfung	113
5.6.2	Die Oder-Verknüpfung	115
5.6.3	Logische Negation	116
5.7	Der ternäre Operator	117
5.8	Die switch-Anweisung	117

5.9	Beispiel: Bodymaßindex (BMI) berechnen	120
5.10	Objekte vergleichen	122
5.10.1	Für welche Objekte gelten die Aussagen?	122
5.10.2	Warum können Objekte nicht mit dem Vergleichsoperator verglichen werden?	125
5.10.3	Wie man Objekte stattdessen vergleicht	127
5.11	Das Wichtigste auf einen Blick	128
5.12	Übungen	129

6 Schleifen und Wiederholungen 133

6.1	Kopfgesteuerte Schleifen – while	133
6.2	Fußgesteuerte Schleifen – do ... while	134
6.3	For-Schleifen	134
6.3.1	Schleifen unterbrechen – break und continue	137
6.3.2	Ein größeres Beispiel für for-Schleifen	138
6.3.3	Schachteln von for-Schleifen	141
6.3.4	Erweiterung – die for-each-Schleife	146
6.4	Das Wichtigste auf einen Blick	150
6.5	Übungen	151

7 Anwendungsbeispiel: Geburtstagskalender 153

7.1	Der erste Ansatz	153
7.2	Der zweite Ansatz	154
7.2.1	Die Datenbasis erstellen	155
7.2.2	Die Datenbasis initialisieren	156
7.2.3	Mehrere Geburtstagskinder an einem Tag	157
7.2.4	Kalendereinträge ausgeben	158
7.3	Fazit	161

TEIL II: OBJEKTORIENTIERTE PROGRAMMIERUNG

8 Grundlegende Aspekte der OOP 167

8.1	Mit Objekten eigener Klassen arbeiten	167
8.2	Inhalt einer Klasse – der Klassenentwurf	170
8.3	Bestandteile einer Klasse	173
8.3.1	Datenfelder und Referenzdatentypen	173
8.3.2	Methoden	177
8.3.3	Konstruktoren	194

8.4	Das Konto-Projekt zum Abschluss bringen	199
8.5	Pakete importieren und statische Importe	201
8.5.1	Aufgabe von Paketen	201
8.5.2	Klassen importieren	202
8.5.3	Statische Importe	204
8.5.4	Eine Eingabemöglichkeit schaffen	208
8.6	Enumerations	208
8.6.1	Lösungsansatz 1	209
8.6.2	Lösungsansatz 2	210
8.6.3	Lösungsansatz 3	212
8.7	UML und die Klassenkarte	215
8.8	Annotations	217
8.9	Das Wichtigste auf einen Blick	218
8.10	Übungen	220

9 Vererbung und Polymorphie 223

9.1	Ihre Aufgabe in diesem Kapitel	223
9.1.1	Der erste Entwurf	223
9.1.2	Bewertung des Lösungsansatzes	225
9.2	Mit Vererbung arbeiten	225
9.2.1	Eine Superklasse entwerfen	225
9.2.2	Vererbung testen	227
9.2.3	Beziehungen zwischen Klassen	229
9.3	Erweiterung der Aufgabe	231
9.3.1	Eine neue Superklasse erstellen	231
9.3.2	Die neue Superklasse testen	233
9.4	Typen zuweisen	234
9.4.1	Statischer und dynamischer Typ von Referenzvariablen	235
9.4.2	Der Cast-Operator	237
9.4.3	Prüfung der Typen	238
9.5	Sichtbarkeit	238
9.6	Konstruktoren	240
9.7	Methoden überschreiben	243
9.7.1	Wie Methoden überschrieben werden	243
9.7.2	Wirkung des Überschreibens	244
9.7.3	Unterschiedliche Typen	245
9.7.4	Beispiele für Überschreiben und Überladen ...	246
9.7.5	Zugriff auf Methoden der Superklasse	248
9.8	Die letzte Erweiterung der Aufgabe	249

9.9	Alle Klassen erben von der Klasse »Object«	254
9.9.1	Die Methode »getClass()«	254
9.9.2	Die Methode »toString()«	255
9.9.3	Die Methode »equals()«	257
9.9.4	Die Methode »hashCode()«	262
9.10	Das Wichtigste auf einen Blick	264
9.11	Übungen	265

10 Erweiterte Konzepte der OOP 267

10.1	Abstrakte Klassen	267
10.1.1	Abstrakte Klassen erzeugen	268
10.1.2	Abstrakte Methoden in abstrakten Klassen	269
10.1.3	Ein Beispiel für abstrakte Klassen: Zeit- und Datumsfunktionen	272
10.2	Schnittstellen	275
10.2.1	Definition des Begriffes	275
10.2.2	Ein Beispiel für die Arbeit mit Interfaces	277
10.3	Wann nehmen Sie was?	286
10.4	Entwurfsmuster	286
10.4.1	Beispiel 1: das Observer-Pattern	287
10.4.2	Beispiel 2: das Singleton-Pattern	290
10.4.3	Patterns generieren lassen – der PatternCoder	291
10.4.4	Sinn von Entwurfsmustern	294
10.5	Innere Klassen	295
10.5.1	Der erste Lösungsansatz: mit herkömmlichen Möglichkeiten	295
10.5.2	Der zweite Lösungsansatz: statische innere Klasse	298
10.5.3	Der dritte Lösungsansatz: nicht-statische Memberklasse	301
10.5.4	Der vierte Lösungsansatz: lokale Klasse	304
10.6	Exkurs: Fehler finden mit dem Debugger	310
10.7	Das Wichtigste auf einen Blick	314
10.7.1	Abstrakte Klassen und Schnittstellen	314
10.7.2	Innere Klassen	315
10.7.3	Entwurfsmuster	315
10.7.4	Debugging	316
10.8	Übungen	316

TEIL III: GRAFISCHE BENUTZEROBERFLÄCHEN

11 Einführung in die GUI-Programmierung 319

11.1	Von AWT zu Swing	320
11.2	Die Vererbungshierarchie von AWT und Swing	321
11.3	Das erste Fenster	322
11.3.1	Ein Fenster erzeugen	322
11.3.2	Ein Menü aufbauen	323
11.3.3	Tastenkürzel für das Menü anlegen	325
11.3.4	Ein Menü mit Auswahl von Optionen	327
11.4	Event-Handling	329
11.4.1	Hierarchie der Event-Klassen und Event-Typen	329
11.4.2	Event-Empfänger	333
11.4.3	Event-Quellen	333
11.4.4	Zwei Beispiele für das Zusammenspiel	334
11.4.5	Adapterklassen	338
11.4.6	So sieht es praktisch aus	339
11.5	Das Wichtigste auf einen Blick	344
11.6	Übungen	345

12 Steuerelemente und Layout 347

12.1	Namenskonventionen und ungarische Notation	347
12.2	Top-Level-Container	348
12.2.1	Die Klasse »JFrame«	348
12.2.2	Die Klasse »JDialog«	350
12.2.3	Die Klasse »JWindow«	352
12.2.4	Fazit zu Top-Level-Containern	352
12.3	Weitere Container-Klassen	353
12.3.1	Die Klassen »JComponent« und »JPanel«	354
12.3.2	Die Klasse »JTabbedPane«	356
12.3.3	Die Klasse »JSplitPane«	357
12.3.4	Die Klasse »JScrollPane«	359
12.4	Interaktionskomponenten (Teil 1)	360
12.4.1	Die Klasse »JLabel«	361
12.4.2	Die Klasse »AbstractButton« und ein paar ihrer Subklassen	363
12.4.3	Die Klasse »JList«	368
12.5	Das MVC-Pattern	372
12.5.1	Allgemeine Beschreibung	372

12.5.2	Wie sieht dieses Prinzip in Swing aus?	382
12.5.3	Besonderheit der Swing-Modelle	389
12.6	Interaktionskomponenten (Teil 2)	390
12.6.1	Die Klasse »JTextField«	390
12.6.2	Die Klasse »JComboBox«	392
12.6.3	Die Klasse »JPasswordField«	394
12.6.4	Die Klasse »JTextArea«	395
12.6.5	Die Klasse »JEditorPane«	396
12.6.6	Die Klasse »JSlider«	397
12.6.7	Die Klasse »JSpinner«	399
12.6.8	Die Klasse »JOptionPane«	401
12.7	Das Wichtigste auf einen Blick	404
12.8	Übungen	405

13 Erweiterte Konzepte von Swing 407

13.1	Layoutmanager	408
13.1.1	Layoutmanager in der Praxis	408
13.1.2	Das FlowLayout	409
13.1.3	Das BorderLayout	410
13.1.4	Das GridLayout	412
13.1.5	Das GridBagLayout	414
13.2	Look and Feel (LaF)	419
13.3	Die Klasse »JTable«	422
13.3.1	Umsetzung mit Arrays	422
13.3.2	Datenbasis mit »TableModel« verwalten	423
13.3.3	Realisierung mit »AbstractTableModel«	428
13.3.4	Realisierung mit »DefaultTableModel«	430
13.3.5	Darstellung der Daten (Rendering)	430
13.3.6	Tabellen sortieren	434
13.3.7	Zellen editieren	437
13.3.8	Mehr über die Arbeit mit »JTable«	441
13.4	Die Klasse »JTree«	442
13.4.1	Ein erster Versuch	442
13.4.2	Auf Ereignisse lauschen	444
13.4.3	Selektionen und die Klasse »TreePath«	446
13.4.4	Ein eigenes Modell entwickeln	448
13.4.5	Die Anzeige ändern – das Rendering	453
13.5	Das Interface »Action«	455
13.6	Das Wichtigste auf einen Blick	457

TEIL IV: FORTGESCHRITTENE THEMEN

14 Ausnahmebehandlung – Exceptions 461

- 14.1 Was bedeutet Fehlerbehandlung? 461
 - 14.1.1 Wie Fehler abgefangen werden können 462
 - 14.1.2 Nachteile des ersten Ansatzes 462
- 14.2 Einführung von Exception-Handling 463
 - 14.2.1 Die Exception-Klassen 464
 - 14.2.2 Der try- und der catch-Block 464
 - 14.2.3 Der finally-Block 466
- 14.3 Fangen oder Weiterreichen (Catch-or-Specify) 467
- 14.4 Differenziert auf Ausnahmesituationen reagieren 468
 - 14.4.1 Eigene Exception-Typen definieren 468
 - 14.4.2 Exceptions differenziert fangen 472
- 14.5 Runtime-Exceptions 475
- 14.6 Error 477
- 14.7 Die Methode »printStackTrace()« 477
- 14.8 Assertions 479
- 14.9 Neu in Java 7 480
- 14.10 Zusammenfassendes Beispiel 481
 - 14.10.1 Die bisherige Implementierung 481
 - 14.10.2 Die neue Implementierung 482
 - 14.10.3 Die neue Implementierung testen 483
- 14.11 Das Wichtigste auf einen Blick 486

15 Collections und Generics 487

- 15.1 Ein eigenes Array entwickeln 487
 - 15.1.1 Die Datenbasis der Klasse »MyArray« 488
 - 15.1.2 Methoden der Klasse »MyArray« 489
 - 15.1.3 Weitere Methoden der Klasse »MyArray« 489
- 15.2 Primitive Datentypen als Objekte – Wrapper-Klassen 491
 - 15.2.1 Autoboxing und Unboxing 491
 - 15.2.2 Hinweis zum Umgang mit Wrapper-Klassen 492
- 15.3 Einführung in Generics – Typsicherheit 493
- 15.4 Die Interfaces »List« und »Set« 496
 - 15.4.1 Das Interface »List« 497
 - 15.4.2 Das Interface »Set« 503

15.4.3	Sammlungen mit »Comparable« und »Comparator« sortieren	508
15.4.4	Verwendung der Klasse »TreeSet«	513
15.5	Das Interface »Map«	516
15.5.1	Eine Map erzeugen	516
15.5.2	Einer Map Werte übergeben	517
15.5.3	Interne Abbildung der Daten und Iteration	518
15.6	Das Wichtigste auf einen Blick	520
15.7	Übungen	521

16 Nebenläufigkeit 523

16.1	Ein erstes Beispiel	523
16.1.1	Die Quellen des Beispiels	524
16.1.2	Auswertung des Beispiels	525
16.2	Zustandsänderungen eines Threads	527
16.2.1	Zustände, die ein Thread einnehmen kann	527
16.2.2	Änderungen des Zustandes	527
16.2.3	Einen Thread beenden	528
16.2.4	Die Methoden »sleep()« und »interrupt()«	530
16.2.5	Prioritäten von Threads festlegen	531
16.2.6	Eine Zeitschaltuhr – die Klasse »Timer«	532
16.3	Synchronisieren von Threads	533
16.3.1	Wo liegt die Fehlerquelle?	533
16.3.2	Wie kann dieses Problem umgangen werden?	537
16.4	Kommunikation zwischen Threads	540
16.5	Das Wichtigste auf einen Blick	542
16.6	Übungen	543

17 Dateien, Datenströme, Serialisieren 545

17.1	Dateibehandlung ab Java 7	545
17.1.1	Einstiegspunkte sind »FileSystem« und »Path«	545
17.1.2	Eine Datei oder ein Verzeichnis bearbeiten	548
17.2	Exkurs: Dateibehandlung bis Java 7	550
17.3	Was sind Ströme und was kann man damit machen?	552
17.3.1	Mit Byte-Streams arbeiten	553
17.3.2	Character-Streams und Brückenklassen	556
17.3.3	BufferedReader und BufferedWriter einsetzen	559

17.4	Kurze Zusammenfassung und Ausblick	562
17.5	Objekte serialisieren und deserialisieren	564
17.5.1	Die Klasse »Adressbuch« schreiben	564
17.5.2	Die Klasse »Freund« schreiben	565
17.5.3	Die Klasse »Verwaltung« schreiben	565
17.5.4	Das Projekt testen	568
17.5.5	Variablen von der Serialisierung ausnehmen	569
17.6	Mit dem FileWhisperer das Dateisystem erkunden	569
17.6.1	Die Oberfläche gestalten	570
17.6.2	Verzeichnisse anzeigen	571
17.6.3	Die Datei-Attribute anzeigen	574
17.7	Das Wichtigste auf einen Blick	580

18 Netzwerke 583

18.1	Rechner verbinden	583
18.2	Verbindung über Sockets	584
18.2.1	IP-Adressen und Domain-Namen	585
18.2.2	Ports	586
18.3	Der Multifunktionsserver	587
18.3.1	Schritt 1 – Grundversion	587
18.3.2	Schritt 2 – mehrere Aufgaben an den Server richten	593
18.3.3	Schritt 3 – Vorbereitung von Multithreading	594
18.3.4	Schritt 4 – Einführung von Multithreading	597
18.3.5	Schritt 5 – eigene Objekte statt Strings	598
18.3.6	Schritt 6 – Einführung von Services	603
18.3.7	Schritt 7 – Funktionen in HashMap speichern	604
18.4	Anregung für eigene Projekte	607
18.5	Das Wichtigste auf einen Blick	608

19 Datenbanken 609

19.1	Grundlagen	609
19.1.1	Aufbau einer Datenbank	610
19.1.2	Wichtige Begriffe	612
19.2	Installation einer Datenbank	612
19.2.1	Die Serversoftware installieren	612
19.2.2	Einen Treiber installieren	614

19.3	Eine Datenbank anlegen	614
19.3.1	Die Verbindung herstellen	615
19.3.2	Befehl an die Datenbank absetzen	616
19.3.3	Überblick über den Quelltext	618
19.4	Tabellen anlegen	620
19.4.1	Treiber laden und Verbindung herstellen	620
19.4.2	Tabellen anlegen und Verbindungen schließen	621
19.5	Spalten anlegen	623
19.5.1	Datentypen von Attributen	624
19.5.2	Die Primärschlüssel der Tabellen anlegen	624
19.6	Daten in die Tabellen eintragen	626
19.7	Daten abfragen, ändern und löschen	628
19.7.1	Daten abfragen	628
19.7.2	Tabellen verbinden	631
19.7.3	Die where-Klausel	632
19.7.4	Daten löschen und Subabfragen definieren	633
19.7.5	Daten ändern	634
19.8	Prepared Statement	635
19.9	Exkurs SQL-Injection	636
19.10	Weitergehende Konzepte	640
19.11	Das Wichtigste auf einen Blick	641
A	Lösungen	643
	Index	649