



*David J. Barnes • Michael Kölling*

# Java lernen mit BlueJ

Objects first – Eine Einführung in Java

*6., aktualisierte Auflage*

 Pearson

 EXTRAS  
ONLINE

## Java lernen mit BlueJ

# Java lernen mit BlueJ

## Inhaltsverzeichnis

Java lernen mit BlueJ

Impressum

Inhaltsverzeichnis

Vorwort von James Gosling, Erfinder von Java 13

Vorwort für den Lehrenden 14

Danksagungen 25

Projekte, die in diesem Buch detailliert besprochen werden 26

Teil I - Objekte und Klassen

Kapitel 1 - Objekte und Klassen 31

1.1 Objekte und Klassen 31

1.2 Instanzen erzeugen 32

1.3 Methoden aufrufen 33

1.4 Parameter 35

1.5 Datentypen 36

1.6 Eine Klasse, viele Instanzen 37

1.7 Zustand 38

1.8 Das Innenleben eines Objekts 38

1.9 Java-Code 40

1.10 Objektinteraktion 41

1.11 Quelltext 42

1.12 Ein weiteres Beispiel 44

1.13 Aufrufergebnisse 44

1.14 Objekte als Parameter 45

Kapitel 2 - Klassendefinitionen 51

2.1 Ticketautomaten 51

2.2 Eine Klassendefinition untersuchen 53

2.3 Der Kopf der Klasse 55

2.4 Datenfelder, Konstruktoren und Methoden 56

# Inhaltsverzeichnis

- 2.5 Datenübergabe mit Parametern 62
- 2.6 Zuweisungen 64
- 2.7 Methoden 65
- 2.8 Sondierende und verändernde Methoden 67
- 2.9 Ausgaben in Methoden 70
- 2.10 Zusammenfassung der Methoden 72
- 2.11 Zusammenfassung des naiven Ticketautomaten 72
- 2.12 Bewertung des Entwurfs des naiven Ticketautomaten 73
- 2.13 Entscheidungen treffen: die bedingte Anweisung 76
- 2.14 Ein weiteres Beispiel für eine bedingte Anweisung 78
- 2.15 Hervorhebung von Sichtbarkeitsbereichen 79
- 2.16 Lokale Variablen 80
- 2.17 Datenfelder, Parameter und lokale Variablen 82
- 2.18 Zusammenfassung des besseren Ticketautomaten 84
- 2.19 Übungen zur Selbstüberprüfung 84
- 2.20 Vertrautes neu betrachtet 86
- 2.21 Methoden aufrufen 88
- 2.22 Ausdrücke testen: die Direkteingabe 89

## Kapitel 3 - Objektinteraktion 97

- 3.1 Das Uhren-Beispiel 97
- 3.2 Abstraktion und Modularisierung 98
- 3.3 Abstraktion in Software 99
- 3.4 Modularisierung im Uhren-Beispiel 100
- 3.5 Implementierung der Uhrenanzeige 100
- 3.6 Klassendiagramme und Objektdiagramme 101
- 3.7 Primitive Typen und Objekttypen 103
- 3.8 Die Klasse Nummernanzeige 103
- 3.9 Die Klasse Uhrenanzeige 110
- 3.10 Objekte erzeugen Objekte 112
- 3.11 Mehrere Konstruktoren 114
- 3.12 Methodenaufrufe 114
- 3.13 Ein weiteres Beispiel für Objektinteraktion 118
- 3.14 Die Benutzung eines Debuggers 122
- 3.15 Mehr zu Methodenaufrufen 126

## Kapitel 4 - Objektsammlungen 131

# Inhaltsverzeichnis

- 4.1 Themen aus Kapitel 3 vertiefen 131
- 4.2 Die Abstraktion Sammlung 132
- 4.3 Ein Verwaltungssystem für Musikdateien 133
- 4.4 Eine Bibliotheksklasse verwenden 135
- 4.5 Objektstrukturen mit Sammlungen 138
- 4.6 Generische Klassen 139
- 4.7 Nummerierung in Sammlungen 140
- 4.8 Musikdateien abspielen 144
- 4.9 Komplette Sammlungen verarbeiten 146
- 4.10 Unbestimmte Iteration 152
- 4.11 Verbesserung der Struktur die Klasse Track 159
- 4.12 Der Typ Iterator 163
- 4.13 Zusammenfassung des Musiksammlung-Beispiels 167
- 4.14 Ein weiteres Beispiel: ein Auktionssystem 169

## **Kapitel 5 - Funktionale Verarbeitung von Sammlungen (fortgeschrittene Konzepte) 183**

- 5.1 Ein neuer Blick auf die Themen von Kapitel 4 183
- 5.2 Monitoring von Tierpopulationen 185
- 5.3 Lambda-Ausdrücke ein erster Blick 188
- 5.4 Die forEach-Methode von Sammlungen 190
- 5.5 Streams 193

## **Kapitel 6 - Bibliotheksklassen nutzen 207**

- 6.1 Die Dokumentation der Bibliotheksklassen 208
- 6.2 Das Kundendienstsystem 209
- 6.3 Die Klassendokumentation lesen 215
- 6.4 Zufälliges Verhalten einbringen 221
- 6.5 Pakete und Importe 226
- 6.6 Benutzung von Map-Klassen für Abbildungen 227
- 6.7 Der Umgang mit Mengen 232
- 6.8 Zeichenketten zerlegen 233
- 6.9 Abschluss des Kundendienstsystems 234
- 6.10 Autoboxing und Wrapper-Klassen 236
- 6.11 Die Klassendokumentation schreiben 239
- 6.12 Öffentliche und private Eigenschaften 242
- 6.13 Klassen über ihre Schnittstelle verstehen 245

# Inhaltsverzeichnis

6.14 Klassenvariablen und Konstanten 250

6.15 Klassenmethoden 253

6.16 Programmausführung ohne BlueJ 254

6.17 Weitere fortgeschrittene Konzepte 255

## Kapitel 7 - Sammlungen mit fester Größe Arrays 263

7.1 Sammlungen fester Größe 263

7.2 Arrays 263

7.3 Die Analyse einer Logdatei 264

7.4 Die for-Schleife 270

7.5 Das Projekt Automat 276

7.6 Arrays mit mehr als einer Dimension (fortgeschritten) 285

7.7 Arrays und Streams (fortgeschritten) 291

## Kapitel 8 - Klassenentwurf 293

8.1 Einführung 294

8.2 Die Welt von Zuul 296

8.3 Kopplung und Kohäsion 297

8.4 Code-Duplizierung 298

8.5 Erweiterungen für Zuul 301

8.6 Kopplung 303

8.7 Entwurf nach Zuständigkeiten 308

8.8 Änderungen lokal halten 311

8.9 Implizite Kopplung 311

8.10 Vorausdenken 315

8.11 Kohäsion 316

8.12 Refactoring 320

8.13 Refactoring für Sprachunabhängigkeit 324

8.14 Entwurfsregeln 330

## Kapitel 9 - Fehler vermeiden 335

9.1 Einführung 335

9.2 Testen und Fehlerbeseitigung 336

9.3 Modultests in BlueJ 337

9.4 Tests automatisieren 343

9.5 Refactoring bei Streams einsetzen (fortgeschritten) 351

9.6 Debugging 351

9.7 Kommentierung und Programmierstil 353

# Inhaltsverzeichnis

- 9.8 Manuelle Ausführung 354
- 9.9 Ausgabeanweisungen 360
- 9.10 Debugger 363
- 9.11 Das Debugging von Streams (fortgeschritten) 365
- 9.12 Die Wahl der richtigen Teststrategie 366
- 9.13 Techniken umsetzen 367

## Teil II - Anwendungsstrukturen

### Kapitel 10 - Bessere Struktur durch Vererbung 371

- 10.1 Das Beispiel Netzwerk 372
- 10.2 Einsatz von Vererbung 382
- 10.3 Vererbungshierarchien 384
- 10.4 Vererbung in Java 385
- 10.5 Weitere Einsendungen für Netzwerk 389
- 10.6 Vorteile durch Vererbung (bis hierher) 391
- 10.7 Subtyping 391
- 10.8 Die Klasse Object 397
- 10.9 Die Hierarchie der Sammlungstypen 398

### Kapitel 11 - Mehr über Vererbung 403

- 11.1 Das Problem: die Methode zum Anzeigen 403
- 11.2 Statischer und dynamischer Typ 405
- 11.3 Überschreiben von Methoden 408
- 11.4 Dynamische Methodensuche 410
- 11.5 super-Aufrufe in Methoden 413
- 11.6 Methoden-Polymorphie 414
- 11.7 Methoden aus Object: toString 414
- 11.8 Objektgleichheit: equals und hashCode 417
- 11.9 Der Zugriff über protected 420
- 11.10 Der Operator instanceof 422
- 11.11 Ein weiteres Beispiel für Vererbung mit Überschreiben 423

### Kapitel 12 - Weitere Techniken zur Abstraktion 429

- 12.1 Simulationen 429
- 12.2 Die Fuchse-und-Hasen-Simulation 431
- 12.3 Abstrakte Klassen 444
- 12.4 Weitere abstrakte Methoden 451

# Inhaltsverzeichnis

- 12.5 Multiple Vererbung 453
- 12.6 Interfaces 457
- 12.7 Ein weiteres Beispiel für ein Interface 465
- 12.8 Die Klasse Class 467
- 12.9 Abstrakte Klasse oder Interface? 468
- 12.10 Ereignisgesteuerte Simulationen 468
- 12.11 Zusammenfassung der Vererbung 470

## **Kapitel 13 - Grafische Benutzungsoberflächen 475**

- 13.1 Einführung 475
- 13.2 Komponenten, Layout und Ereignisbehandlung 476
- 13.3 AWT und Swing 477
- 13.4 Das Beispiel: ein Bildbetrachter 478
- 13.5 Bildbetrachter 1.0: die erste komplette Version 490
- 13.6 Bildbetrachter 2.0: die Programmstruktur verbessern 504
- 13.7 Bildbetrachter 3.0: weitere GUI-Komponenten 509
- 13.8 Innere Klassen 513
- 13.9 Zusätzliche Erweiterungen 518
- 13.10 Ein weiteres Beispiel: der Musikplayer 520

## **Kapitel 14 - Fehlerbehandlung 527**

- 14.1 Das Adressbuch-Projekt 528
- 14.2 Defensive Programmierung 531
- 14.3 Fehlermeldungen durch den Dienstleister 534
- 14.4 Prinzipien der Exception-Behandlung 539
- 14.5 Die Behandlung von Exceptions 545
- 14.6 Neue Exception-Klassen definieren 552
- 14.7 Die Verwendung von Zusicherungen 553
- 14.8 Wiederaufsetzen und Fehlervermeidung 557
- 14.9 Dateibasierte Ein- und Ausgabe 560

## **Kapitel 15 - Entwurf von Anwendungen 573**

- 15.1 Analyse und Entwurf 573
- 15.2 Klassenentwurf 581
- 15.3 Dokumentation 583
- 15.4 Kooperation 584
- 15.5 Prototyping 585
- 15.6 Softwarewachstum 586

# Inhaltsverzeichnis

15.7 Der Einsatz von Entwurfsmustern 588

## Kapitel 16 - Eine Fallstudie 597

16.1 Die Fallstudie 597

16.2 Analyse und Entwurf 599

16.3 Klassenentwurf 602

16.4 Iterative Entwicklung 608

16.5 Ein weiteres Beispiel 616

16.6 Ein Blick nach vorn 616

## Anhänge

### Anhang A - Arbeiten mit BlueJ-Projekten

A.1 BlueJ installieren 617

A.2 Ein Projekt öffnen 617

A.3 Der Debugger in BlueJ 617

A.4 BlueJ konfigurieren 618

A.5 Auf deutsche Schnittstelle umstellen 618

A.6 Einbinden einer lokalen API-Dokumentation 619

A.7 Vorlagen für neue Klassen ändern 619

### Anhang B - Datentypen in Java

B.1 Primitive Typen 621

B.2 Cast-Operator für primitive Typen 622

B.3 Objekttypen 623

B.4 Wrapper-Klassen 623

B.5 Cast-Operator für Objekttypen 624

### Anhang C - Operatoren 625

C.1 Arithmetische Ausdrücke 625

C.2 Boolesche Ausdrücke 626

C.3 Abkürzungsoperatoren 627

### Anhang D - Kontrollstrukturen in Java 629

D.1 Kontrollstrukturen 629

D.2 Auswahlanweisungen 629

D.3 Schleifen 632

D.4 Exceptions 634

D.5 Zusicherungen 635

### Anhang E - Java ohne BlueJ 637

# Inhaltsverzeichnis

- E.1 Java ohne BlueJ ausführen 637
- E.2 Konsolenanwendungen und die Problematik der Umlaute 639
- E.3 Ausführbare jar -Dateien erzeugen 641
- E.4 Entwickeln ohne BlueJ 642

## Anhang F - Benutzung des Debuggers 643

- F.1 Haltepunkte 644
- F.2 Die Kontrollknöpfe 644
- F.3 Anzeige der Variablen 645
- F.4 Die Anzeige der Aufruffolge 646
- F.5 Die Thread-Anzeige 646

## Anhang G - Testwerkzeuge für Modultests mit JUnit 647

- G.1 Aktivieren der Test-Funktionalität 647
- G.2 Eine Testklasse erzeugen 647
- G.3 Eine Testmethode erzeugen 647
- G.4 Zusicherungen bei Tests 648
- G.5 Tests ausführen 648
- G.6 Testgerüste 648

## Anhang H - Werkzeuge für die Teamarbeit 649

- H.1 Server-Einrichtung 649
- H.2 Teamarbeit-Funktionalität aktivieren 649
- H.3 Ein Projekt zur gemeinsamen Nutzung einrichten 649
- H.4 An einem Projekt gemeinsam arbeiten 650
- H.5 Aktualisieren und Abgeben 650
- H.6 Weitere Informationen 650

## Anhang I - Javadoc 651

- I.1 Dokumentationskommentare 651
- I.2 Unterstützung für Javadoc in BlueJ 654

## Anhang J - Quelltextkonventionen 655

- J.1 Benennung 655
- J.2 Layout 656
- J.3 Dokumentation 657
- J.4 Restriktionen bei der Sprachbenutzung 658
- J.5 Programmiermuster 659

## Anhang K - Wichtige Bibliotheksklassen 661

# Inhaltsverzeichnis

- K.1 Das Paket java.lang 661
- K.2 Das Paket java.util 662
- K.3 Die Pakete java.io und java.nio.file 664
- K.4 Das Paket java.util.function 665
- K.5 Das Paket java.net 666
- K.6 Weitere wichtige Pakete 666

Register 667

Widmung

Vorwort von James Gosling, Erfinder von Java

Vorwort für den Lehrenden

Neues in der sechsten Auflage

Danksagungen

Projekte, die in diesem Buch detailliert besprochen werden

Teil I - Objekte und Klassen

Kapitel 1 - Objekte und Klassen

- 1.1 Objekte und Klassen
- 1.2 Instanzen erzeugen
- 1.3 Methoden aufrufen
- 1.4 Parameter
- 1.5 Datentypen
- 1.6 Eine Klasse, viele Instanzen
- 1.7 Zustand
- 1.8 Das Innenleben eines Objekts
- 1.9 Java-Code
- 1.10 Objektinteraktion
- 1.11 Quelltext
- 1.12 Ein weiteres Beispiel
- 1.13 Aufrufergebnisse
- 1.14 Objekte als Parameter

Kapitel 2 - Klassendefinitionen

# Inhaltsverzeichnis

- 2.1 Ticketautomaten
- 2.2 Eine Klassendefinition untersuchen
- 2.3 Der Kopf der Klasse
- 2.4 Datenfelder, Konstruktoren und Methoden
- 2.5 Datenübergabe mit Parametern
- 2.6 Zuweisungen
- 2.7 Methoden
- 2.8 Sondierende und verändernde Methoden
- 2.9 Ausgaben in Methoden
- 2.10 Zusammenfassung der Methoden
- 2.11 Zusammenfassung des naiven Ticketautomaten
- 2.12 Bewertung des Entwurfs des naiven Ticketautomaten
- 2.13 Entscheidungen treffen: die bedingte Anweisung
- 2.14 Ein weiteres Beispiel für eine bedingte Anweisung
- 2.15 Hervorhebung von Sichtbarkeitsbereichen
- 2.16 Lokale Variablen
- 2.17 Datenfelder, Parameter und lokale Variablen
- 2.18 Zusammenfassung des besseren Ticketautomaten
- 2.19 Übungen zur Selbstüberprüfung
- 2.20 Vertrautes neu betrachtet
- 2.21 Methoden aufrufen
- 2.22 Ausdrücke testen: die Direkteingabe

## Kapitel 3 - Objektinteraktion

- 3.1 Das Uhren-Beispiel
- 3.2 Abstraktion und Modularisierung
- 3.3 Abstraktion in Software
- 3.4 Modularisierung im Uhren-Beispiel
- 3.5 Implementierung der Uhrenanzeige
- 3.6 Klassendiagramme und Objektdiagramme
- 3.7 Primitive Typen und Objekttypen
- 3.8 Die Klasse Nummernanzeige

# Inhaltsverzeichnis

- 3.9 Die Klasse Uhrenanzeige
- 3.10 Objekte erzeugen Objekte
- 3.11 Mehrere Konstruktoren
- 3.12 Methodenaufrufe
- 3.13 Ein weiteres Beispiel für Objektinteraktion
- 3.14 Die Benutzung eines Debuggers
- 3.15 Mehr zu Methodenaufrufen

## Kapitel 4 - Objektsammlungen

- 4.1 Themen aus Kapitel 3 vertiefen
- 4.2 Die Abstraktion Sammlung
- 4.3 Ein Verwaltungssystem für Musikdateien
- 4.4 Eine Bibliotheksklasse verwenden
- 4.5 Objektstrukturen mit Sammlungen
- 4.6 Generische Klassen
- 4.7 Nummerierung in Sammlungen
- 4.8 Musikdateien abspielen
- 4.9 Komplette Sammlungen verarbeiten
- 4.10 Unbestimmte Iteration
- 4.11 Verbesserung der Struktur die Klasse Track
- 4.12 Der Typ Iterator
- 4.13 Zusammenfassung des Musiksammlung-Beispiels
- 4.14 Ein weiteres Beispiel: ein Auktionssystem

## Kapitel 5 - Funktionale Verarbeitung von Sammlungen (fortgeschrittene Konzepte)

- 5.1 Ein neuer Blick auf die Themen von Kapitel 4
- 5.2 Monitoring von Tierpopulationen
- 5.3 Lambda-Ausdrücke ein erster Blick
- 5.4 Die forEach-Methode von Sammlungen
- 5.5 Streams

## Kapitel 6 - Bibliotheksklassen nutzen

- 6.1 Die Dokumentation der Bibliotheksklassen

# Inhaltsverzeichnis

- 6.2 Das Kundendienstsystem
- 6.3 Die Klassendokumentation lesen
- 6.4 Zufälliges Verhalten einbringen
- 6.5 Pakete und Importe
- 6.6 Benutzung von Map-Klassen für Abbildungen
- 6.7 Der Umgang mit Mengen
- 6.8 Zeichenketten zerlegen
- 6.9 Abschluss des Kundendienstsystems
- 6.10 Autoboxing und Wrapper-Klassen
- 6.11 Die Klassendokumentation schreiben
- 6.12 Öffentliche und private Eigenschaften
- 6.13 Klassen über ihre Schnittstelle verstehen
- 6.14 Klassenvariablen und Konstanten
- 6.15 Klassenmethoden
- 6.16 Programmausführung ohne BlueJ
- 6.17 Weitere fortgeschrittene Konzepte

## Kapitel 7 - Sammlungen mit fester Größe Arrays

- 7.1 Sammlungen fester Größe
- 7.2 Arrays
- 7.3 Die Analyse einer Logdatei
- 7.4 Die for-Schleife
- 7.5 Das Projekt Automat
- 7.6 Arrays mit mehr als einer Dimension (fortgeschritten)
- 7.7 Arrays und Streams (fortgeschritten)

## Kapitel 8 - Klassenentwurf

- 8.1 Einführung
- 8.2 Die Welt von Zuul
- 8.3 Kopplung und Kohäsion
- 8.4 Code-Duplizierung
- 8.5 Erweiterungen für Zuul
- 8.6 Kopplung

# Inhaltsverzeichnis

- 8.7 Entwurf nach Zuständigkeiten
- 8.8 Änderungen lokal halten
- 8.9 Implizite Kopplung
- 8.10 Vorausdenken
- 8.11 Kohäsion
- 8.12 Refactoring
- 8.13 Refactoring für Sprachunabhängigkeit
- 8.14 Entwurfsregeln

## Kapitel 9 - Fehler vermeiden

- 9.1 Einführung
- 9.2 Testen und Fehlerbeseitigung
- 9.3 Modultests in BlueJ
- 9.4 Tests automatisieren
- 9.5 Refactoring bei Streams einsetzen (fortgeschritten)
- 9.6 Debugging
- 9.7 Kommentierung und Programmierstil
- 9.8 Manuelle Ausführung
- 9.9 Ausgabeanweisungen
- 9.10 Debugger
- 9.11 Das Debugging von Streams (fortgeschritten)
- 9.12 Die Wahl der richtigen Teststrategie
- 9.13 Techniken umsetzen

## Teil II - Anwendungsstrukturen

### Kapitel 10 - Bessere Struktur durch Vererbung

- 10.1 Das Beispiel Netzwerk
- 10.2 Einsatz von Vererbung
- 10.3 Vererbungshierarchien
- 10.4 Vererbung in Java
- 10.5 Weitere Einsendungen für Netzwerk
- 10.6 Vorteile durch Vererbung (bis hierher)
- 10.7 Subtyping

# Inhaltsverzeichnis

10.8 Die Klasse Object

10.9 Die Hierarchie der Sammlungstypen

## Kapitel 11 - Mehr über Vererbung

11.1 Das Problem: die Methode zum Anzeigen

11.2 Statischer und dynamischer Typ

11.3 Überschreiben von Methoden

11.4 Dynamische Methodensuche

11.5 super-Aufrufe in Methoden

11.6 Methoden-Polymorphie

11.7 Methoden aus Object: toString

11.8 Objektgleichheit: equals und hashCode

11.9 Der Zugriff über protected

11.10 Der Operator instanceof

11.11 Ein weiteres Beispiel für Vererbung mit Überschreiben

## Kapitel 12 - Weitere Techniken zur Abstraktion

12.1 Simulationen

12.2 Die Fuechse-und-Hasen-Simulation

12.3 Abstrakte Klassen

12.4 Weitere abstrakte Methoden

12.5 Multiple Vererbung

12.6 Interfaces

12.7 Ein weiteres Beispiel für ein Interface

12.8 Die Klasse Class

12.9 Abstrakte Klasse oder Interface?

12.10 Ereignisgesteuerte Simulationen

12.11 Zusammenfassung der Vererbung

## Kapitel 13 - Grafische Benutzungsoberflächen

13.1 Einführung

13.2 Komponenten, Layout und Ereignisbehandlung

13.3 AWT und Swing

13.4 Das Beispiel: ein Bildbetrachter

# Inhaltsverzeichnis

- 13.5 Bildbetrachter 1.0: die erste komplette Version
- 13.6 Bildbetrachter 2.0: die Programmstruktur verbessern
- 13.7 Bildbetrachter 3.0: weitere GUI-Komponenten
- 13.8 Innere Klassen
- 13.9 Zusätzliche Erweiterungen
- 13.10 Ein weiteres Beispiel: der Musikplayer

## Kapitel 14 - Fehlerbehandlung

- 14.1 Das Adressbuch-Projekt
- 14.2 Defensive Programmierung
- 14.3 Fehlermeldungen durch den Dienstleister
- 14.4 Prinzipien der Exception-Behandlung
- 14.5 Die Behandlung von Exceptions
- 14.6 Neue Exception-Klassen definieren
- 14.7 Die Verwendung von Zusicherungen
- 14.8 Wiederaufsetzen und Fehlervermeidung
- 14.9 Dateibasierte Ein- und Ausgabe

## Kapitel 15 - Entwurf von Anwendungen

- 15.1 Analyse und Entwurf
- 15.2 Klassenentwurf
- 15.3 Dokumentation
- 15.4 Kooperation
- 15.5 Prototyping
- 15.6 Softwarewachstum
- 15.7 Der Einsatz von Entwurfsmustern

## Kapitel 16 - Eine Fallstudie

- 16.1 Die Fallstudie
- 16.2 Analyse und Entwurf
- 16.3 Klassenentwurf
- 16.4 Iterative Entwicklung
- 16.5 Ein weiteres Beispiel
- 16.6 Ein Blick nach vorn

# Inhaltsverzeichnis

## Anhänge

### Anhang A - Arbeiten mit BlueJ-Projekten

- A.1 BlueJ installieren
- A.2 Ein Projekt öffnen
- A.3 Der Debugger in BlueJ
- A.4 BlueJ konfigurieren
- A.5 Auf deutsche Schnittstelle umstellen
- A.6 Einbinden einer lokalen API-Dokumentation
- A.7 Vorlagen für neue Klassen ändern

### Anhang B - Datentypen in Java

- B.1 Primitive Typen
- B.2 Cast-Operator für primitive Typen
- B.3 Objekttypen
- B.4 Wrapper-Klassen
- B.5 Cast-Operator für Objekttypen

### Anhang C - Operatoren

- C.1 Arithmetische Ausdrücke
- C.2 Boolesche Ausdrücke
- C.3 Abkürzungsoperatoren

### Anhang D - Kontrollstrukturen in Java

- D.1 Kontrollstrukturen
- D.2 Auswahlanweisungen
- D.3 Schleifen
- D.4 Exceptions
- D.5 Zusicherungen

### Anhang E - Java ohne BlueJ

- E.1 Java ohne BlueJ ausführen
- E.2 Konsolenanwendungen und die Problematik der Umlaute
- E.3 Ausführbare jar -Dateien erzeugen
- E.4 Entwickeln ohne BlueJ

# Inhaltsverzeichnis

## Anhang F - Benutzung des Debuggers

- F.1 Haltepunkte
- F.2 Die Kontrollknöpfe
- F.3 Anzeige der Variablen
- F.4 Die Anzeige der Aufruffolge
- F.5 Die Thread-Anzeige

## Anhang G - Testwerkzeuge für Modultests mit JUnit

- G.1 Aktivieren der Test-Funktionalität
- G.2 Eine Testklasse erzeugen
- G.3 Eine Testmethode erzeugen
- G.4 Zusicherungen bei Tests
- G.5 Tests ausführen
- G.6 Testgerüste

## Anhang H - Werkzeuge für die Teamarbeit

- H.1 Server-Einrichtung
- H.2 Teamarbeit-Funktionalität aktivieren
- H.3 Ein Projekt zur gemeinsamen Nutzung einrichten
- H.4 An einem Projekt gemeinsam arbeiten
- H.5 Aktualisieren und Abgeben
- H.6 Weitere Informationen

## Anhang I - Javadoc

- I.1 Dokumentationskommentare
- I.2 Unterstützung für Javadoc in BlueJ

## Anhang J - Quelltextkonventionen

- J.1 Benennung
- J.2 Layout
- J.3 Dokumentation
- J.4 Restriktionen bei der Sprachbenutzung
- J.5 Programmiermuster

## Anhang K - Wichtige Bibliotheksklassen

# Inhaltsverzeichnis

- K.1 Das Paket java.lang
- K.2 Das Paket java.util
- K.3 Die Pakete java.io und java.nio.file
- K.4 Das Paket java.util.function
- K.5 Das Paket java.net
- K.6 Weitere wichtige Pakete

## Register

### Symbols

- !, Operator 106
- &&, Operator 106
- %, Operator 109
- +, Operator 108
  - für Zeichenketten 71
- =, Operator 64
- >=, Operator 77
- ||, Operator 106

### A

#### Abbildung

- mit dem Map-Konzept 228

#### Abkürzungsoperator 627

#### abstract

- für Klassen 448
- für Methoden 446

#### Abstrakte Klasse 448

- Beispiel 444
- vs. Interface 468

#### Abstrakte Methode 446

#### Abstrakte Subklasse 449

#### Abstraktion 98

- Flexibilität 455

#### Adressbuch, Projekt 528

# Inhaltsverzeichnis

Adressbuch-Assert, Projekt 554

Adressbuch-IO, Projekt 569

Adressbuch-JUnit, Projekt 557

Adressbuch-V1T, Projekt 530

Adressbuch-V2G, Projekt 534

Adressbuch-V2T, Projekt 534

Adressbuch-V3T, Projekt 548

## Akteur

als abstrakte Klasse 454

als Interface 457

Akzeptanztest 337

Analyse und Entwurf 573

Änderungen lokal halten 311

Anonyme innere Klasse 515

Anonymes Objekt 174

## Anweisung

assert 554

Ausgabeeanweisung 360

bedingte 76

catch 546, 634

finally 634

geschützte 547

if 76

import 136, 226

throw 539

throws 545

try 546, 634

Zuweisung 64

## Array

for-each-Schleife 272

Index 140

length 272

mehrdimensionales 285

# Inhaltsverzeichnis

Objekt benutzen 268

Objekt erzeugen 267

Variable deklarieren 266

ArrayIndexOutOfBoundsException Exception 269

Array-Initialisierer 284

ArrayList 133

Beispiel für Benutzung 133, 173, 223

Elemente entfernen 142

generische Klasse 139

implementiert das Interface List 462

Indexzugriff 165

iterator 163

Iteratorzugriff 165

remove 166

Schlüsselmethode 137

assert-Anweisung 554

Attribut

synonym zu Datenfeld 38

Aufzählungstyp 326

Auktionssystem, Projekt 169

Ausdruck 64

testen 89

Ausgabenweisung 360

Auswahlenweisung 629

Autoboxing 237

Automat Projekt 276

Automat, zellulärer 276

AWT (Java-Bibliothek) 477

## B

Baelle, Projekt 249

Bedingte Anweisung 76

Benutzungsoberfläche, grafische 475

# Inhaltsverzeichnis

Besserer-Ticketautomat, Projekt 73, 129

Bestimmte Iteration 152

Bibliothek

AWT 477

importieren 136

Paket 135

Swing 477

Bibliotheksklasse 661

Bildbetrachter0-2, Projekt 486

Bildbetrachter0-4, Projekt 492

Bildbetrachter1-0, Projekt 478, 490

Bildbetrachter2-0, Projekt 504

Bildbetrachter3-0, Projekt 509

Bildformat 478

Block 66

BlueJ

auf Deutsch umstellen 618

installieren 617

konfigurieren 618

Boolescher Ausdruck 77

BorderLayout 494

BoxLayout 496

Brain Projekt 285

Buch-Aufgabe, Projekt 94

Bug 119

## C

Cast-Operator 396

für Objekttypen 624

für primitive Typen 622

catch-Klausel 546, 634

Class (Klasse) 467

Closure 184

# Inhaltsverzeichnis

Code-Duplizierung 298, 382

vermeiden für Klienten 396

Code-Vervollständigung 248

Combo-Boxen 522

Compiler 43

Computermodell

Modell eines Ausschnitts der realen Welt 31

CRC-Karten 576

## D

Datenfeld 57

automatische Initialisierung 62

öffentlich 244

vs. lokale Variable und formaler Parameter 82

Zustand halten 38

Debugger 119, 363, 643

anhalten 644

Aufruffolge 646

Einzelausführung 124

Haltepunkt setzen 123, 644

Kontrollknöpfe 644

Schritt hinein 126

Defensive Programmierung 531

Die Welt von Zuul, Projekt 296, 423

Direkteingabe 40, 89

Dokumentation

generische Klassen 225

Java-Klassenbibliothek 215

Klasse 239

Konstruktor 240

Methode 240

Dynamische Methodensuche 410

Dynamische Sicht 102

Dynamischer Typ 405

# Inhaltsverzeichnis

Dynamisches Binden 410

## E

Ein-/Ausgabe

CSV-Format 567

dateibasierte 560

Eingabe parsen 567

Scanner 567

Text ausgeben 562

Text einlesen 565

Einzeiliger Kommentar 58

Einzelausführung 124

Eliza 210

Enge Kopplung

durch öffentliche Datenfelder 311

Entkopplung 311

Entwurf

CRC-Karten 576

Dokumentation 583

Klassen 293

Klassen identifizieren 575

Kooperation 584

nach Zuständigkeiten 160, 308

Prototyping 585

Softwarewachstum 586

Szenarios 577

Verb/Substantiv-Methode 574

von Benutzungsschnittstellen 583

von Klassenschnittstellen 582

Wasserfallmodell 586

Entwurfsmuster 588

Beobachter 592

Dekorierer 589

Fabrikmethode 591

Singleton 590

# Inhaltsverzeichnis

Entwurfsregeln 330

Enumeration siehe Aufzählungstyp

Ereignisverarbeitung 484

Ergebnistyp

    nicht bei Konstruktoren 66

    void 68

Ersetzbarkeit 394

Event-Listener 484

Exception 634

    ArrayIndexOutOfBoundsException Exception 269

    Auswirkungen 542

    behandeln 545

    fangen 546

    geprüfte 540, 545

    Handler 546

    Hierarchie 541

    IndexOutOfBoundsException 141

    mehrere Exceptions werfen und fangen 548

    neue Klassen definieren 552

    Prinzipien der Behandlung 539

    propagieren 550

    ungeprüfte 540, 543

    werfen 539

extends zwischen Klassen 383

Externer Methodenaufruf 115

Extreme Programming 584

## F

Fallstudie 597

    Analyse und Entwurf 599

    CRC-Karten 600

    iterative Entwicklung 608

    Klassen identifizieren 599

    Partnerklassen 603

# Inhaltsverzeichnis

Szenarios 600

testen 607

## Fehler

vermeiden 558

vom Dienstleister gemeldet 534

Wiederaufsetzen 557

## Fehlerbeseitigung 336

Ausgabeeinweisung 360

Debugger 363

manuelle Ausführung 354

mündliche Ausführung 360

## Figuren, Projekt 32

## File (Klasse) 561

## filter (Funktion) 195

## Filter-Map-Reduce 195

## finally-Klausel 551, 634

## FlowLayout 494

## forEach-Methode 190

## for-each-Schleife 147, 151, 271

## Formaler Parameter 63

vs. lokale Variable und Datenfeld 82

## for-Schleife 270

## Frame (GUI) 480

## Fuechse-und-Hasen, Projekt 432

## Funktion, anonyme 190

## G

## Geheimnisprinzip 243

## Generische Klasse 136, 139, 225

## Generischer Typ 163

## Geprüfte Exception 540, 545

## Geschäftsfall 577

## Grafische Benutzungsoberfläche 475

# Inhaltsverzeichnis

GridLayout 495

GUI 475

GUI-Klassen

JButton 510

JComboBox 520

JFrame 479

JLabel 481

JList 522

JMenu 483, 499

JMenuBar 483

JMenuItem 499

JPanel 496

JScrollPane 522

## H

Haltepunkt setzen 123, 644

HashMap, Beispiel für Benutzung 229, 306

Haus, Projekt 41

Hohe Kohäsion 312

## I

Implementierung

einer Klasse 216

vs. Schnittstelle 216

implements (Schlüsselwort) 458

implements-Klausel 458

import-Anweisung 136, 226

Index in einer Sammlung 140

Index-Grenzverletzung 141

IndexOutOfBoundsException 141

Indexvariable 154

Indezzugriff 165

Inhaltsfläche (GUI) 480

Initialisierung 60

# Inhaltsverzeichnis

instanceof-Operator 422

Instanz 33

einer bestimmten Klasse 33

interaktiv erzeugen 32

synonym zu Objekt 32

Instanzvariable, synonym zu Datenfeld 57

Interaktion zwischen Klient und Dienstleister 531

Interface

als Spezifikation 462

als Typ 461

Event-Listener 484

multiple Vererbung 459

Path 561

Serializable 569

vs. abstrakte Klasse 468

Interner Methodenaufruf 114

Ist-ein-Beziehung 383

Iteration

bestimmte 152

mit Indexvariable 154

unbestimmte 152

Iterative Entwicklung 608

Iterator 163

## J

Jäger-Beute-Modellierung 431

jar-Dateien 641

Java Archive Format 641

Java-Bibliothek 208

Java-Datentypen 621

javadoc 651

Klasse 240

Konstruktorkommentar 240

Markierungsabschnitt 652

# Inhaltsverzeichnis

Methodenkommentar 240

Schlüsselwörter 241

JButton 510

JComboBox 520

JFrame 479

JLabel 481

JList 522

JMenu 483, 499

JMenuBar 483

JMenuItem 499

JPanel 496

JScrollPane 522

JUnit 344, 647

JUnit-Test 556

## K

Kapselung 304

Kinobuchungssystem, Projekt 574

Klasse

abstrakte 448

anonyme innere 515

ArrayList 134

Autoboxing 237

beschreibt Objekte 31

Class 467

definiert einen Typ 101

entwerfen 293

File 561

generische 136, 139, 225

identifizieren 575

importieren 136

Kohäsion 317

neue Exception-Klasse 552

Object 397

# Inhaltsverzeichnis

Paket	135
parametrisierte	225
Random	221
Scanner	567
Unboxing	237
verbessern	159
Wrapper	236, 623
Klassenbibliothek	
importieren	136
Paket	135
Klassendiagramm	101
Klassendokumentation	
lesen	215
schreiben	239
Klassenentwurf	293
Klassenmethode	253
Einschränkung	254
main	254
vs. Instanzmethode	253
Klassenvariable	250
Klausel	
catch	546
finally	551
try	546
Klient-Dienstleister-Interaktion	531
Knopf (GUI)	509
Kohäsion	
Beispiele	480, 484
für bessere Lesbarkeit	318
für Wiederverwendung	319
hohe	312
von Klassen	317
von Methoden	316
Kommentar	58

# Inhaltsverzeichnis

- Komponente (GUI) 476
  - Konditionaloperator 279
  - Konsistenzüberprüfung, interne 553
  - Konstante 252
  - Konstruktor 60
    - der Superklasse aufrufen 388
    - Dokumentation 240
    - initialisiert ein Objekt 60
    - javadoc-Kommentar 240
    - Überladung 114
    - und Vererbung 387
  - Kontrollstruktur 629
    - Auswahanweisung 629
    - Schleifen 632
  - Konvention 655
    - Dokumentation 657
    - Layout 656
    - Restriktion 658
  - Kopf einer Methode 66
  - Kopplung 243, 303
    - enge 311
    - entkoppeln 311
    - implizite 311
    - Kapselung 308
    - lose 303
    - Pfeile im Klassendiagramm 314
  - Kritzeln, Projekt 245
- L
- Laborkurse, Projekt 44, 86, 102, 401
  - Lambda-Ausdruck 184, 188
  - Layout (GUI) 476
  - Lebensdauer
    - einer lokalen Variable 81

# Inhaltsverzeichnis

- eines Datenfelds 63
- eines formalen Parameters 63
- LinkedList implementiert das Interface List 462
- Liste 522
- Logischer Fehler 335
- Logischer Operator 106
- Lokale Variable 80
  - Lebensdauer 81
  - Sichtbarkeit 81
  - vs. Datenfeld und formaler Parameter 82
- Lokalität von Änderungen 311
- Lose Kopplung 303
- Lotka-Volterra-Modell 431

## M

- Mail-System, Projekt 120
- main-Methode 254
- Manifest-Datei 642
- Manuelle Ausführung 354
- map (Funktion) 195
- Map, Konzept 228
- Map-Klasse 227
- Maschinencode 43
- Mehrzeiliger Kommentar 58
- Menge, Konzept 232
- Menüzeile (GUI) 480
- Methode 35
  - abstrakte 446
  - Aufrufe verketteten 175
  - aufrufen 42, 88
  - Dokumentation 240
  - dynamische Methodensuche 410
  - Ergebnisse liefern 45

# Inhaltsverzeichnis

extern aufrufen 115  
implementiert Verhalten 56  
interaktiv aufrufen 33  
intern aufrufen 114  
javadoc-Kommentar 240  
Kohäsion 316  
Kopf und Rumpf 66  
main 254  
private 244  
protected 420  
sondierende 67  
super-Aufruf 413  
überschreiben 408  
verändernde 68

Methodenauswahl 410

Methoden-Polymorphie 414

Methodensuche 410

Mock-Up 353

Modell-Ansicht-Trennung 521

Modularisierung 99

Modulo-Operator % 109

Modultest 337, 647

Moore-Nachbarschaft 286

Multiple Vererbung 453

    für Interfaces 459

Musikplayer, Projekt 521

Musiksammlung, Projekt 133

## N

Naiver Ticketautomat, Projekt 51

Negatives Testen 343

Netzwerk, Projekt 372

Netzwerk-V1, Projekt 381

Netzwerk-V2, Projekt 386

# Inhaltsverzeichnis

Netzwerk-V3, Projekt 409

new, Operator 113

null, Schlüsselwort 171

## O

Object (Klasse) 397

    equals 418

    hashCode 419

    toString 398, 414

Objekt 31

    als Parameter 46

    anonymes 174

    einer bestimmten Klasse 31

    erzeugen mit new 112

    erzeugt Objekte 112

    Erzeugung verhindern 544

    Interaktion 41

    Iterator 163

    Schnittstelle 118

    Serialisierung 569

    unveränderliches 218

Objektdiagramm 101

Objekterzeugung 112

Objektinspektor 38, 340

Objektleiste in BlueJ 33

Objektreferenz 102

Objektyp 623

    vs. primitiver Typ 103

Öffentliches Datenfeld 244

Onlineshop, Projekt 337

OnlineShop-JUnit, Projekt 344

Operation

    intermediäre 198

    terminale 198

# Inhaltsverzeichnis

## Operator

- + zur Verkettung 108
- Abkürzungsoperatoren 627
- arithmetische Ausdrücke 625
- boolesche Ausdrücke 626
- Cast- 396, 622, 624
- instanceof 422
- Konditional- 279
- Modulo 109
- new 113
- ternärer 279

## P

Paar-Programmierung 584

### Pakete 135

- in der Standardbibliothek 226
- java.io 664
- java.lang 661
- java.net 666
- java.nio 664
- java.util 136, 662
- java.util.function 665

### Parameter

- aktueller 63
- formaler 63
- interaktiv übergeben 35
- prüfen 533

Parameterübergabe 62

- Typregeln 65

Parametrisierte Klasse 225

Partnerklasse 603

Pipeline 197

Polymorphe Variable 395

Positives Testen 343

# Inhaltsverzeichnis

- Prädikat (Lambda) 199
- Primitiver Typ vs. Objekttyp 103
- private (Zugriffsmodifikator) 242
- Programmausführung
  - im Debugger 124
  - manuell 354
  - mündlich 360
  - ohne BlueJ 637
- Programmiermuster 659
- Programmierstil
  - funktionaler 184, 189
  - imperativer 183
- Programmierung, defensive 531
- Projekte
  - Adressbuch 528
  - Adressbuch-Assert 554
  - Adressbuch-IO 569
  - Adressbuch-JUnit 557
  - Adressbuch-V1T 530
  - Adressbuch-V2G 534
  - Adressbuch-V2T 534
  - Adressbuch-V3T 548
  - Auktionssystem 169
  - Automat 276
  - Automat-v1 277
  - Automat-v2 281
  - Automat-v3 282
  - Automat-v4 284
  - Baelle 249
  - Besserer-Ticketautomat 73
  - Bildbetrachter0-2 486
  - Bildbetrachter0-4 492
  - Bildbetrachter1-0 478, 490
  - Bildbetrachter2-0 504

# Inhaltsverzeichnis

Bildbetrachter3-0	509
Brain	285
Buch-Aufgabe	94
Die Welt von Zuul	296, 423
Fallstudie	597
Figuren	32
Fuechse-und-Hasen	432
Haus	41
Kinobuchungssystem	574
Kritzeln	245
Laborkurse	44, 86, 102, 401
Mail-System	120
Musikplayer	521
Musiksammlung	133
Naiver Ticketautomat	51
Netzwerk	372
Netzwerk-V1	381
Netzwerk-V2	386
Netzwerk-V3	409
Onlineshop	337
OnlineShop-JUnit	344
Recheneinheit	353, 364
Recheneinheit-mit-Ausgaben	361
Rechner-GUI	364
Technischer-Kundendienst1	209
Technischer-Kundendienst2	227
Technischer-Kundendienst-IO	566
Technischer-Kundendienst- komplett	209, 568
Tiermonitoring-v1	185
Weblog-Auswertung	264
Zeitanzeige	103
Ziegelsteine	367
Zuul-besser	315
Zuul-schlecht	295
protected (Zugriffsmodifikator)	420

# Inhaltsverzeichnis

Prototyping 585  
Pseudocode 76  
public (Zugriffsmodifikator) 242  
Punkt-Notation 116

## Q

Quelltext 42  
Quelltextkonventionen 655

## R

Random (Klasse) 221  
Reader 561  
Recheneinheit, Projekt 353, 364  
Recheneinheit-mit-Ausgaben,  
    Projekt 361  
Rechner-GUI, Projekt 364  
Redefinieren, synonym zu Überschreiben 409  
reduce (Funktion) 196  
Refactoring 320  
    Klassen extrahieren 324  
    Methoden extrahieren 324  
    und Testen 320  
Regressionstest 344  
Responsibility-driven Design 308  
Rollbalken 522  
Rückgabeabweisung 66  
Rumpf einer Methode 66  
RuntimeException 552

## S

Sammlung  
    Abstraktion 132  
    ArrayList 133  
    durchsuchen 156  
    Elemente entfernen 142, 165

# Inhaltsverzeichnis

- fester Größe 263
- flexibler Größe 133
- Index 140
- Indexzugriff 165
- Iteratorzugriff 165
- Nummerierung 140
- Objektstrukturen 138
- selektiv verarbeiten 149
- verarbeiten 146

## Scanner (Klasse) 567

### Schleife

- do-while 632
- Elemente entfernen 165
- for 633
- for-each 147, 151, 271
- for-Schleife 270
- Kopf 147
- Schleifenvariable 148
- while 153, 632

## Schleifenvariable 148

### Schlüsselwort 56

- null 171
- static 250

### Schnittstelle 118

- einer Klasse 216
- einer Methode 217
- öffentlicher Teil einer Klasse 242
- vs. Implementierung 216

## Selektive Darstellung 456

## Serializable, Interface 569

### Sichtbarkeit 79

- einer lokalen Variablen 83
- eines Datenfelds 63
- eines formalen Parameters 63, 83

# Inhaltsverzeichnis

- Simulation 429
  - ereignisgesteuert 468
- Softwareabstraktion 99
- Softwarewachstum 586
- Sondierende Methode 67
- static, Schlüsselwort 250
- Statischer Typ 405
- Stream 184, 193, 561
- String (Klasse) 88
  - equals 220
  - length 88, 217
  - startsWith 214
  - substring 88
  - trim 218
- Subklasse 383
  - definiert Subtyp 393
  - Subtyp 393
- Subtyp
  - Parameterübergabe 395
  - Subklasse 393
  - Zuweisung 393
- Subtyping 391
- super in einer Methode 413
- Superklasse 383
  - Object 397
- Swing (Java-Bibliothek) 477
- Syntaxfehler 335
- System.out.println 71
- Szenarios 577

## T

- Teamarbeit 649
- Technischer-Kundendienst1, Projekt 209

# Inhaltsverzeichnis

Technischer-Kundendienst2, Projekt 227

Technischer-Kundendienst-IO, Projekt 566

Technischer-Kundendienst-komplett, Projekt 209, 568

Teile und herrsche 98

## Testen

automatisieren 344

bei Refactorings 320

JUnit 647

JUnit-Tests in BlueJ 556

negatives 343

positives 343

Regressionstest 344

Testklasse erzeugen 647

Testmethode erzeugen 647

von Grenzfälle 341

Testgerüst 350, 648

Tests aufzeichnen 347

Testwerkzeuge 647

this 121

throw-Anweisung 539

throws-Anweisung 545

Tiermonitoring, Projekt 185

toString in BlueJ 414

try-Klausel 546, 634

## Typ

Aufzählung 326

Datentypen in Java 621

dynamischer 405

eines Parameters 36

generischer 163

Interface 461

Iterator 163

Objektypen 623

# Inhaltsverzeichnis

primitiver 103, 621

statischer 405

## U

Überladen 114

Überschreiben von Methoden 408

Unbestimmte Iteration 152

Unboxing 237

Und-Operator && 106

Ungeprüfte Exception 540, 543

Unveränderliches Objekt 218

Use Case 577

## V

Variable

Index 154

lokale 81

polymorphe 395

und Subtypen 393

Verändernde Methode 68

Verb/Substantiv-Methode 574

Vererbung

für einfachere Wartung 391

gegen Code-Duplizierung 382, 391

im Klassendiagramm 382

Konstruktoren 387

multiple 453

Vorteile 391

zur Spezialisierung 383

zur Wiederverwendung 389, 391

zwischen Klassen mit extends 385

Vererbungshierarchie 384

Verkettung von Zeichenketten 108

void als Ergebnistyp 68

# Inhaltsverzeichnis

## W

- Walkthrough 354
- Wasserfallmodell 586
- Weblog-Auswertung, Projekt 264
- Weizenbaum, Joseph 210
- while-Schleife 153
- Wiederaufsetzen nach Fehlern 557
- Wiederholung 147
- Wiederverwendung
  - durch Vererbung 391
- Wolfram-Code 283
- Wrapper-Klasse 236, 623
- Writer 561

## Z

- Zeichenketten
  - auf Gleichheit prüfen 220
  - verketteten 108
  - zerlegen 233
- Zeitanzeige, Projekt 103
- Ziegelsteine, Projekt 367
- Zufälliges Verhalten in Computern 221
- Zufallszahl erzeugen 222
- Zugriffsmodifikator
  - private 242, 244
  - protected 420
  - public 242
- Zugriffsstufen 420
- Zuordnungstabelle 283
- Zusicherung 348, 554, 635
  - in BlueJ 556
  - Richtlinien 555
- Zustand

# Inhaltsverzeichnis

definiert über Datenfelder 38

Zuul-besser, Projekt 315

Zuul-schlecht, Projekt 295

Zuweisung 64

linke und rechte Seite 64

Typregeln 65

Zuweisungsoperator = 64

Ins Internet: Weitere Infos zum Buch, Downloads, etc.

Copyright

# Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als **persönliche Einzelplatz-Lizenz** zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschließlich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs und
- der Veröffentlichung

bedarf der **schriftlichen Genehmigung** des Verlags. Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: [info@pearson.de](mailto:info@pearson.de)

## Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. **Der Rechtsweg ist ausgeschlossen.**

## Hinweis

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website herunterladen:

**<http://ebooks.pearson.de>**