

Inhaltsverzeichnis

1 EINFÜHRUNG	7
1.1 Literatur.....	8
1.2 Objektorientierte Programmierung (OOP).....	9
1.3 Ein Miniprogramm.....	12
1.4 Programmentwicklungsablauf.....	13
1.5 Ein erstes nützliches Programm: Brutto.....	14
1.6 Wissenswertes aus dem Übersetzerbau.....	16
1.7 Qualitätskriterium Redundanzfreiheit.....	20
1.8 Applet.....	23
2 PROZEDURALE PROGRAMMIERUNG	25
2.1 Elementare Datentypen.....	25
2.2 Ablaufsteuerung.....	30
2.2.1 Sequenz.....	30
2.2.2 Alternative.....	32
2.2.2.1 If-Anweisung.....	32
2.2.2.2 switch-Anweisung.....	41
2.2.3 Iteration.....	44
2.2.3.1 Vorprüfende Schleife.....	44
2.2.3.2 Mittelprüfende Schleife.....	48
2.2.3.3 Schleifenübungen.....	50
2.2.3.4 Aufgabe: Zahlen interaktiv aufsummieren.....	51
2.2.3.5 Geschachtelte Schleifen.....	53
2.2.4 Automatische und individuelle Ausnahmebehandlung.....	55
2.3 Objekt benutzen.....	66
2.4 Einführung in die Entwicklungsumgebung Eclipse.....	71
2.4.1 Installation und Einrichten eines Projekts.....	72
2.4.2 Java-Quelldateien anlegen, übersetzen und ausführen.....	76
2.4.3 JUnit einrichten und Programmtest ausführen.....	77
2.4.4 Ein Programm unter Steuerung des Debuggers ausführen.....	80
2.5 Methode (Unterprogramm).....	81
2.5.1 Definition und Aufruf.....	81
2.5.2 Überladen eines Methodennamens.....	88
2.5.3 Aufgaben zur Methodendefinition.....	90
2.5.4 Rekursion.....	94
2.6 Programmierprinzipien.....	98
2.6.1 Lokalität.....	98
2.6.2 Initialisiertheit.....	99
2.6.3 Konstantheit.....	99
2.6.4 Schachtelungsorientierte Einrückung.....	100
2.6.5 Wiederverwendbarkeit.....	100
2.7 Feld.....	101
2.7.1 Eindimensionales Feld.....	101
2.7.2 For-each-Schleife zum Durchlaufen einer Sammlung von Elementen.....	103
2.7.3 Referenzsemantik.....	105
2.7.4 Initialisierung eines Feldes.....	108
2.7.5 Ein neues Feldobjekt erzeugen.....	109
2.7.8 Behandlung von Indexfehlern.....	110
2.7.9 Suche in Feld.....	110
2.7.9.1 Lineare Suche.....	111
2.7.9.2 Binäre/Logarithmische Suche.....	112
2.7.10 Sortieren.....	114
2.7.10.1 Simple-Sort.....	114
2.7.10.2 Mehrdimensionales Feld.....	117
2.8 Operatoren.....	119
2.9 Deklaration und Auslösen einer Ausnahme.....	122
3 OBJEKTORIENTIERTE PROGRAMMIERUNG	126
3.1 Zusammengesetzter Datentyp und Referenzsemantik.....	126
3.2 Datenabstraktion.....	129
3.2.1 Der Punkt als Abstrakter Datentyp.....	129
3.2.2 Der Abstrakte Datentyp Stack.....	136
3.2.3 Wiederholbare Tests mit JUnit 4.....	144
3.2.4 Der Abstrakte Datentyp Queue.....	150

5	3.2.5 Die Klasse StringBuilder.....	151
	3.3 Zentrale OO-Eigenschaften.....	154
	3.3.1 Beerbung.....	154
	3.3.2 Polymorphie.....	160
	3.3.3 Dynamisches Binden.....	165
	3.3.4 Konstruktoren und Beerbung.....	168
	3.3.5 Typverwandschaft und Typumwandlung (type cast).....	172
	3.4 Ausnahmen gruppieren und parametrieren.....	174
	3.5 Abstrakte Basisklasse: Design für Vererbung.....	178
	3.6 Objekt- vs. Klassen-Zugehörigkeit (static).....	187
	3.7 final-Modifizierer.....	189
	3.8 Sichtbarkeit und Programmstrukturierung.....	190
	3.8.1 Innere Klasse (ab Java 1.1).....	191
	3.8.2 Paket.....	195
	3.8.3 Sichtbarkeit.....	196
	3.8.4 Schnittstelle und Implementierung.....	197
	3.9 Abschlussbemerkung.....	200
	4 ALLGEMEINE BASISKLASSEN.....	201
	4.1 Das Collections Framework ab Java 2.....	201
	4.1.1 Polymorphe Collection.....	203
	4.1.2 Generische Collection ab Java 5.....	205
	4.1.3 Scala-Collections.....	209
	4.1.4 Folgen der internen Speicherung als polymorphe Referenzen.....	212
	4.1.5 Fazit.....	214
	4.2 Generische, typsichere Verpackung eines polymorphen APIs.....	215
	5 PROJEKT DIAGRAMMEDITOR.....	218
	5.1 Projektstudie.....	218
	5.2 Analyse und Entwurf.....	219
	5.2.1 Aufstellung der Begriffshierarchie.....	219
	5.2.2 Planung der Benutzungsschnittstelle.....	220
	5.2.3 Entwurf.....	221
6	5.3 2D-Grafik in Java.....	227
	5.4 Oberflächenprogrammierung.....	228
	5.4.1 Oberflächenkomponenten in javax.swing.....	229
	5.4.2 Gruppierung von Komponenten.....	230
	5.4.3 Layoutmanager.....	233
	5.4.3.1 BorderLayout.....	234
	5.4.3.2 FlowLayout.....	236
	5.4.3.3 GridLayout.....	238
	5.4.3.4 GridBagLayout.....	239
	5.4.3.5 Weitere Layouts.....	242
	5.4.3.6 Aufgaben zu Layouts.....	243
	5.4.4 Empfohlener Programmaufbau für Oberflächen.....	244
	5.4.5 Update-Verlauf.....	245
	5.4.6 Ereignisorientierung.....	245
	5.4.6.1 Das Listener-Konzept: Reaktion auf typisierte Ereignisse.....	246
	5.4.6.2 Reaktion auf Listen-Ereignisse.....	249
	5.4.6.3 Reaktion auf Mausereignisse.....	250
	5.4.6.4 Parallele Abläufe („threading issues“) in Swing.....	252
	5.5 Objekterzeugung einer dynamisch bestimmten Klasse mittels Reflection.....	254
	5.6 Ausnahme mit Ursachenkette und Meldungsparametern.....	258
	5.7 Zwischenbilanz Diagrammeditor.....	260
	5.8 Speichern/Laden (Dateizugriffe).....	262
	5.8.1 Byte-Ströme (Binärdaten).....	264
	5.8.1.1 Object-Ströme (Binärdaten).....	266
	5.8.2 Zeichen-Ströme (Texte).....	269
	5.8.2.1 Zerlegung von Zeichenketten in Token (=Lexeme).....	270
	5.8.3 Dateisystemnavigation mit der Klasse java.io.File.....	273
	6 ANHANG.....	275
	6.1 Programmierrichtlinien.....	275
	6.1.1 Namenskonventionen.....	275
	6.1.2 Programminterne Dokumentation.....	277
	6.1.3 Richtlinien zur Typspezialisierung.....	278
	6.1.4 Konstantheit, Lokalität, Redundanzfreiheit, Einrückung.....	278
	6.1.5 Richtlinien zur Fehlerbehandlung.....	278

6.2 Weiterführende Literatur.....	279
6.3 Index.....	281

Die Folien sind für ein klassisches Bildverhältnis von 4:3 optimiert.

Die didaktischen Lücken sind durchnummeriert und mit [?] gekennzeichnet, Bsp.: [?]

Die Lösungen werden im Präsenzunterricht diskutiert und an die Tafel geschrieben.

Typographische Auszeichnungen

Normal: Normaler Fließtext

Unterstrichen: Der unterstrichene Begriff wird definiert oder erstmalig eingeführt.

Fett: Der fett gedruckte Begriff wird hier betont und ist daher auch in den Index aufgenommen.

Schreibmaschine: Mit Schreibmaschinenschrift geschriebene Texte sind zur Vearbeitung durch einen Computer bestimmt, z.B. Programmcode, oder die Ausgabe eines Computerprogramms.

Schreibmaschine kursiv: Eingaben des Benutzers in einem Computerdialog.

Schreibmaschine fett: Ein Schlüsselwort der Programmiersprache.

1 EINFÜHRUNG

Das Programmieren ist der zentrale Teil der **Informatik**. Auch in der heutigen Zeit, in der die Schwerpunkte immer mehr weg von der **systemnahen Programmierung** zu den **Computeranwendungen** gehen, bietet das Programmieren die einzigartige Möglichkeit, den Computer gut kennenzulernen und sein Verhalten, auch innerhalb von **Anwendungen**, zu verstehen.

Die Programme, die Sie im Rahmen der Übungen erstellen sollen, sehen sehr einfach aus. Insbesondere gute Lösungen glänzen durch Einfachheit. Das Schwierige daran ist, wie man auf sie kommt. Daher rate ich Ihnen, die Übungen alle selbst anzufertigen (bei Gruppen mit **gleich**-starken Partnern).

Die didaktische Reihenfolge dieser Lehrveranstaltung ist vom Kleinen zum Großen. Dennoch lege ich viel Gewicht auf die Konzepte des „Programmieren im Großen“ (Pakete, Klassen) und versuche das „Programmieren im Kleinen“ nur mit dem Notwendigsten zu behandeln.

1.1 Literatur

Es erscheint mehr und mehr Literatur zu Java. Mittlerweile gibt es auch viele Bücher, die zum Erlernen des Programmierens anhand von Java gedacht sind. Auch in der Beuth-Campusbibliothek finden sich viele Exemplare. Sie sollten sich neben diesem Skript mindestens ein Java-Buch ([RRZN], [Jobst], [JePS] oder [JvKF]) sowie das [JDK] und alle ab [Tut] aufgeführten Internet-Dokumente besorgen.

[RRZN] Thomas KRÖCKERTSKOTHEN, Peter HEUSCH: „Java. Grundlagen und Einführung“. Skript des Regionalen Rechenzentrums für Niedersachsen (RRZN). 10. Auflage 2010, 248 pp., 4,50 €. Ab der 7. Auflage unter dem Titel „Java 6“ verwendbar.

Für Programmieranfänger geeignet, da ausführlich. Besonders preisgünstig, da Direktvertrieb über die Hochschulen, bei uns über das Beuth-CopyCenter.

[Jobst] Fritz JOBST: Programmieren in Java, Hanser-Verlag, 2011, 29,90 €.

Niveaull, deckt PR1 und PR2 ab, als Anfängerlehrbuch nur bedingt geeignet, behandelt auch klassische Informatik (Rekursion, Sortieren, Parallele Prozesse), liegt der Vorlesungsgliederung zugrunde, viele Verweise darauf. Die 2011-er Auflage basiert auf Java 7 mit Swing, Generizität und Annotationen.

[JePS] Joachim GOLL, Cornelia WEIß, Frank MÜLLER: Java als erste Programmiersprache, Teubner-Verlag, 2001, 39,95 €.

Wurde mir für Anfänger empfohlen. Hat einen klassischen, sehr ausführlichen Ansatz.

[JvKF] Kathy SIERRA, Bert BATES, Lars SCHULTEN, Elke BUCHHOLZ: Java von Kopf bis Fuß, O'Reilly, 2006, 49,90 €.