

Vorlesung Modellierung und Programmierung I

Inhaltsverzeichnis

Teil I Imperative Programmierung

0	Maschinenzahlen.....	2
0.1	Additionssysteme	2
0.2	Positionssysteme	2
0.3	Dezimal- und Dualsystem	3
0.3.1	Dezimalsystem.....	3
0.3.2	Dualsystem.....	4
0.4	Weitere Beispiele für Positionssysteme	6
0.5	Zusammenfassung Zahlendarstellung	7
0.6	Maschinenzahlen.....	8
0.6.1	Interne Darstellung von ganzen Zahlen $z \in \mathbb{Z}$	8
0.6.2	Interne Darstellung von rationalen Zahlen $z \in \mathbb{Q}$	9
0.7	IEEE - Standard.....	11
1	Algorithmen und Programme	1-2
1.1	Algorithmus.....	1-3
1.2	Programm.....	1-4
1.2.1	Hauptkomponenten des Aufbaus eines Rechners	1-4
1.2.2	Ein Programm für einen Algorithmus.....	1-5
1.2.3	Rekursion	1-7
1.2.4	Codierung.....	1-8
1.3	Programmierparadigmen.....	1-9
1.4	Anhang zum Modellrechner.....	1-11
1.4.1	Backus-Naur-Form (BNF)	1-11
1.4.2	Syntaxdiagramm	1-12
2	Einstieg in das Programmieren mit Java.....	2-2
2.1	Geschichte	2-2
2.2	Konzept	2-3
2.2.1	Compiler und Interpreter.....	2-3
2.2.2	Bytecode und virtuelle Maschinen.....	2-4
2.3	Grundlagen der Java-Programmierung	2-6
2.3.1	Installation von Java	2-6
2.3.2	Java Application.....	2-6
2.3.3	Java Applet.....	2-7
2.3.4	Java Script.....	2-9
2.4	Ein- und Ausgaben.....	2-12
2.4.1	Ausgabe.....	2-12
2.4.2	Eingabe	2-12
2.4.3	Installation des Pakets <code>Tools</code>	2-14
2.4.4	Methoden der Klasse <code>IOTools</code>	2-14

3	Grundelemente der Java-Programmierung	3-2
3.1	Alphabet	3-2
3.2	Bezeichner	3-3
3.3	Kommentare	3-3
3.4	Elementardatentypen	3-5
3.5	Konstanten (Literele).....	3-7
3.5.1	Unbenannte Konstanten	3-7
3.5.2	Benannte Konstanten	3-8
3.6	Variablen	3-9
3.7	Ausdrücke.....	3-11
3.8	Zusammenfassung.....	3-13
4	Anweisungen.....	4-2
4.1	Strukturierte Programmierung.....	4-2
4.1.1	Geschichte.....	4-2
4.1.2	Strukturierung im Kleinen	4-2
4.2	Einige Beispielanwendungen	4-4
4.2.1	Addierer (do-Schleife)	4-4
4.2.2	Ein- Mal- Eins (for-Schleife, if-Anweisung)	4-5
4.2.3	Linaere Gleichung (if-else-Anweisung).....	4-6
4.2.4	Einfacher Rechner (switch-Anweisung)	4-8
4.2.5	Summieren (break, continue)	4-10
5	Referenzdatentypen - Felder	5-2
5.1	Eindimensionale Felder - Vektoren.....	5-3
5.1.1	Vereinbarung.....	5-3
5.1.2	Referenzen sind keine Felder.....	5-6
5.1.3	Kopieren eindimensionaler Felder	5-7
5.2	Zweidimensionale Felder - Matrizen	5-8
5.2.1	Vereinbarung.....	5-8
5.2.2	Kopieren zweidimensionaler Felder	5-9
5.2.3	Beispiel „Tic Tac Toe“	5-9
6	Referenzdatentypen - Klassen.....	6-2
6.1	Deklaration und Instanziierung	6-3
6.1.1	Festlegen eines Strukturtyps als Klasse	6-3
6.1.2	Festlegen einer Struktur als Objekt einer Klasse	6-4
6.1.3	Kopieren von Objekten	6-7
6.2	Felder von Klassen	6-8
6.3	Klassen von Klassen.....	6-11

7	Methoden	7-2
7.1	Prozedurorientierte Programmierung	7-2
7.2	Klassenmethoden	7-5
7.2.1	Methodenvereinbarung	7-7
7.2.2	Methodenaufruf.....	7-9
7.2.3	Referenzdatentypen in der Schnittstelle einer Methode	7-10
7.2.4	Beispiel „Polynomwertberechnung mittels Horner Schema“	7-10
7.2.5	Überladen von Methoden.....	7-14
7.3	Die Kommandozeile.....	7-15
7.4	Rekursionen.....	7-16
7.4.1	Beispiel „Eine unendliche Geschichte“	7-16
7.4.2	Beispiel „Türme von Hanoi“	7-17
7.4.3	Iteration oder Rekursion	7-20
7.5	Konstanten und Methoden der Klasse <code>java.lang.Math</code>	7-22

Teil II Objektorientierte Programmierung

8	Das objektorientierte Programmierparadigma	8-2
8.1	Strukturierung im Großen - Modularisierung	8-2
8.2	Modularisierung	8-5
8.2.1	Objekte zur Kapselung von Attributen und Methoden	8-5
8.2.2	Klassen zur Datenabstraktion	8-6
8.3	Vererbung, Generalisierung und Spezialisierung.....	8-9
8.3.1	Abgeleitete Klassen - Spezialisierung	8-9
8.3.2	Überschreiben von Methoden	8-10
8.3.3	Abstrakte Klassen - Generalisierung	8-11
8.3.4	Einfach- und Mehrfachvererbung	8-13
8.4	Polymorphismus und dynamisches Binden.....	8-15
8.5	Zusammenfassung - Konzepte der OOP	8-17
8.6	Methoden der Klasse <code>java.lang.String</code>	8-18
9	Klassen.....	9-2
9.1	Instanzvariablen und Instanzmethoden	9-2
9.1.1	Vereinbarung.....	9-2
9.1.2	<code>this</code>	9-5
9.1.3	<code>toString()</code>	9-5
9.2	Klassenvariablen und Klassenmethoden.....	9-7
9.2.1	Vereinbarung.....	9-7
9.2.2	<code>super</code>	9-10
9.3	Klassenkonstanten.....	9-11
9.4	Instanziierung und Initialisierung.....	9-13
9.4.1	Konstruktoren	9-13
9.4.2	Beispiel „Boot“	9-15
9.4.3	Der Mechanismus der Objekterzeugung.....	9-22
9.5	Pakete	9-23
9.5.1	Erstellen von Paketen.....	9-23
9.5.2	Installation komprimierter Pakete.....	9-24
9.6	Zugriffsrechte	9-25

10	Modellierung.....	10-2
10.1	Entwurfs- und Zerlegungstechniken.....	10-3
10.1.1	Zielgerichteter Entwurf.....	10-3
10.1.2	Kompositioneller Entwurf.....	10-3
10.1.3	Objektorientierter Entwurf.....	10-3
10.2	Klassenbeziehungen.....	10-5
10.2.1	UML (Unified Modeling Language).....	10-5
10.2.2	Assoziation, Aggregation und Komposition.....	10-5
10.2.3	Entwurfsmuster.....	10-6
10.3	Modulbildung.....	10-7
10.4	Beispiel „Milchladen“.....	10-9
10.5	Vor- und Nachteile objektorientierter Programmierung.....	10-18
11	Dynamische Verwaltung großer Datenmengen.....	11-2
11.1	Einige spezielle Klassen des Paketes <code>java.lang.*</code>	11-2
11.1.1	Klasse <code>Throwable</code> , <code>Exception</code> und <code>Error</code>	11-2
11.1.2	Klasse <code>Object</code>	11-7
11.1.3	Wrapper-Klassen.....	11-7
11.2	Abstrakte Klassen und Interface.....	11-9
11.2.1	Abstrakte Klassen.....	11-9
11.2.2	Interface.....	11-10
11.3	Collection-Klassen <code>java.util.*</code>	11-12
11.3.1	Generische Datentypen - Template.....	11-13
11.3.2	Interface <code>Collection<E></code>	11-13
11.3.3	Interface <code>Iterator<E></code>	11-14
11.3.4	Interface <code>Set<E></code>	11-15
11.3.5	Interface <code>List<E></code>	11-16
11.4	Klassen <code>Arrays</code> und <code>Collections</code>	11-22
11.5	Zusammenfassung.....	11-22
12	Grafische Benutzerschnittstellen.....	12-2
12.1	Komponenten.....	12-2
12.1.1	AWT und Swing.....	12-2
12.1.2	Beispiel „Hallo Welt!“.....	12-3
12.2	Ereignisverarbeitung.....	12-6
12.3	Grafik.....	12-10
12.3.1	Die Methoden <code>paint</code> und <code>repaint</code>	12-10
12.3.2	Die abstrakte Klasse <code>java.awt.Graphics</code>	12-10
12.3.3	Beispiel „Simple Grafik“.....	12-11

13	MVC-Architektur.....	13-2
13.1	Das MVC-Konzept	13-2
13.1.1	Entwurfsmuster	13-2
13.1.2	Mechanismus der MVC-Architektur	13-3
13.1.3	Klassen <code>java.util.Observable</code> und <code>java.util.Observer</code>	13-3
13.2	Beispiel „Zähler modulo 10“	13-10
13.2.1	Design und Implementierung des Models	13-10
13.2.2	Design und Implementierung des Views	13-11
13.2.3	Design und Implementierung des Controllers	13-13
13.2.4	Implementierung des Programmaufbaus	13-15
13.2.5	Beispiel „Zähler modulo 10“ mit grafischer Oberfläche	13-17
13.3	Steuerung mehrerer View und Controller zu einem Model.....	13-20
13.3.1	Abstrakte Klassen	13-21
13.3.2	Abgeleitete Klassen	13-25
13.3.3	Beispiel „Ampelsteuerung“	13-28
13.4	Zusammenfassung.....	13-31
14	Dateiverwaltung, das Stream-Konzept	14-2
14.1	Datenströme	14-2
14.1.1	Datenströme in Java, Paket <code>java.io.*</code>	14-3
14.1.2	Standarddatenströme.....	14-4
14.2	Klasse <code>java.io.File</code>	14-7
14.3	Textdateien.....	14-9
14.3.1	Ungepufferte Reader- und Writer Klassen	14-10
14.3.2	Gepufferte Reader- und Writer Klassen	14-11
14.3.3	Beispiel „TextDatei“	14-14
14.4	Daten- und Objektdateien	14-19
14.4.1	Klassen <code>InputStream</code> und <code>OutputStream</code>	14-19
14.4.2	Dateien für Elementardatentypen	14-22
14.4.3	Dateien für Elementar- und Referenzdatentypen.....	14-25
14.4.4	Beispiel „Obst“	14-28
14.5	Übersicht über häufig verwendete Datenströme	14-33
14.6	Weitere wichtige Klassen	14-35
15	Parallele Programmierung	15-2
15.1	Die Klasse <code>java.lang.Thread</code>	15-2
15.2	Beispiel „0-1-Printer“ als Thread	15-3
15.3	Das Interface <code>java.lang.Runnable</code>	15-4
15.4	Beispiel „0-1-Printer“ mit <code>Runnable</code>	15-5
15.5	Zusammenfassung.....	15-6

16	Rechnerkommunikation – verteilte Systeme	16-2
16.1	Netzwerktechnologie	16-2
16.1.1	Adressen.....	16-2
16.1.2	Ports und Sockets.....	16-3
16.2	Server/Client-Programmierung.....	16-5
16.2.1	Aufbau einer Server/Client-Verbindung.....	16-5
16.2.2	Die Klassen <code>java.net.ServerSocket</code> , <code>java.net.Socket</code>	16-7
16.3	Beispiel „Chatroom“	16-9
16.3.1	Modell.....	16-9
16.3.2	Server	16-10
16.3.3	Client.....	16-15