
Inhaltsverzeichnis

1	Tag 1: Vorbereitungen und Motivation	1
1.1	Motivation: Warum gerade Java?	1
1.1.1	Programme für Webseiten	2
1.2	Vorteile des Selbststudiums	2
1.3	Installation von Java	3
1.4	Testen wir das Java-System	4
2	Tag 2: Grundlegende Prinzipien der Programmentwicklung	7
2.1	Primitive Datentypen und ihre Wertebereiche	7
2.1.1	Primitive Datentypen in Java	8
2.2	Variablen und Konstanten	9
2.2.1	Deklaration von Variablen	9
2.2.2	Variablen und Konstanten	11
2.3	Primitive Datentypen und ihre Operationen	11
2.3.1	boolean	11
2.3.2	char	14
2.3.3	int	14
2.3.4	byte, short, long	15
2.3.5	float, double	16
2.4	Casting, Typumwandlungen	16
2.4.1	Übersicht zu impliziten Typumwandlungen	18
2.4.2	Die Datentypen sind für die Operation entscheidend	19
2.5	Methoden der Programmerstellung	20
2.5.1	Sequentieller Programmablauf	21
2.5.2	Verzweigungen	21
2.5.3	Sprünge	22
2.5.4	Schleifen	22
2.5.5	Mehrfachverzweigungen	22
2.5.6	Mehrfachschleifen (verschachtelte Schleifen)	22
2.5.7	Parallelität	23
2.5.8	Kombination zu Programmen	23

- 2.6 Programme in Java 24
 - 2.6.1 Erstellen eines Javaprogramms 24
- 2.7 Zusammenfassung und Aufgaben 25
- 3 Tag 3: Programmieren mit einem einfachen Klassenkonzept 29**
 - 3.1 Sequentielle Anweisungen 30
 - 3.2 Verzweigungen 31
 - 3.2.1 if-Verzweigung 31
 - 3.2.2 switch-Verzweigung 32
 - 3.3 Verschiedene Schleifentypen 33
 - 3.3.1 for-Schleife 34
 - 3.3.2 while-Schleife 35
 - 3.3.3 do-while-Schleife 36
 - 3.4 Sprunganweisungen 37
 - 3.4.1 break 37
 - 3.4.2 continue 39
 - 3.5 Klassen 41
 - 3.5.1 Funktionen in Java 41
 - 3.6 Zusammenfassung und Aufgaben 43
- 4 Tag 4: Daten laden und speichern 45**
 - 4.1 Externe Programmeingaben 46
 - 4.2 Daten aus einer Datei einlesen 47
 - 4.3 Daten in eine Datei schreiben 49
 - 4.4 Daten von der Konsole einlesen 49
 - 4.5 Zusammenfassung und Aufgaben 50
- 5 Tag 5: Verwendung einfacher Datenstrukturen 53**
 - 5.1 Arrays und Matrizen 53
 - 5.1.1 Erzeugung eines Arrays 53
 - 5.1.2 Matrizen oder multidimensionale Arrays 55
 - 5.1.3 Conway's Game of Life 56
 - 5.1.3.1 Einfache Implementierung 57
 - 5.1.3.2 Eine kleine Auswahl besonderer Muster 60
 - 5.2 Zusammenfassung und Aufgaben 61
- 6 Tag 6: Debuggen und Fehlerbehandlungen 63**
 - 6.1 Das richtige Konzept 63
 - 6.2 Exceptions in Java 65
 - 6.2.1 Einfache try-catch-Behandlung 66
 - 6.2.2 Mehrfache try-catch-Behandlung 67
 - 6.3 Fehlerhafte Berechnungen aufspüren 68
 - 6.3.1 Berechnung der Zahl pi 68
 - 6.3.2 Zeilenweises Debuggen und Breakpoints 71
 - 6.4 Zusammenfassung und Aufgaben 71

7	Tag 7: Erweitertes Klassenkonzept	73
7.1	Entwicklung eines einfachen Fußballmanagers	73
7.1.1	Spieler und Trainer	73
7.1.1.1	Generalisierung und Spezialisierung	74
7.1.1.2	Klassen und Vererbung	75
7.1.1.3	Modifizierer public und private	76
7.1.1.4	Objekte und Instanzen	77
7.1.1.5	Konstruktoren in Java	78
7.1.2	Die Mannschaft	81
7.1.3	Turniere, Freundschaftsspiele	82
7.1.3.1	Ein Interface festlegen	82
7.1.3.2	Freundschaftsspiel Deutschland–Brasilien der WM-Mannschaften von 2006	85
7.1.3.3	Interface versus abstrakte Klasse	88
7.2	Aufarbeitung der vorhergehenden Kapitel	89
7.2.1	Referenzvariablen	89
7.2.2	Zugriff auf Attribute und Methoden durch Punktnotation	90
7.2.3	Die Referenzvariable this	91
7.2.4	Prinzip des Überladens	92
7.2.5	Überladung von Konstruktoren	92
7.2.5.1	Der Copy-Konstruktor	93
7.2.6	Garbage Collector	94
7.2.7	Statische Attribute und Methoden	94
7.2.8	Primitive Datentypen und ihre Wrapperklassen	95
7.2.9	Die Klasse String	96
7.2.9.1	Vergleich von Zeichenketten	97
7.3	Zusammenfassung und Aufgaben	98
8	Tag 8: Verwendung von Bibliotheken	101
8.1	Standardbibliotheken	101
8.2	Mathematik-Bibliothek	103
8.3	Zufallszahlen in Java	104
8.3.1	Ganzzahlige Zufallszahlen vom Typ int und long	105
8.3.2	Zufallszahlen vom Typ float und double	105
8.3.3	Weitere nützliche Funktionen der Klasse Random	105
8.4	Das Spiel Black Jack	106
8.4.1	Spielkarten	106
8.4.2	Wertigkeiten der Karten	106
8.4.3	Der Spielverlauf	107
8.4.4	Spieler, Karten und Kartenspiel	108
8.4.4.1	Verwendungsbeispiel für Datenstruktur Vector	108
8.4.4.2	Implementierung der Klassen Spieler, Karten und Kartenspiel	109
8.4.5	Die Spielklasse Blackjack	112
8.5	JAMA - Lineare Algebra	118

8.6	Eine eigene Bibliothek bauen	120
8.7	Zusammenfassung und Aufgaben	121
9	Tag 9: Grafische Benutzeroberflächen	123
9.1	Fenstermanagement unter AWT	123
9.1.1	Ein Fenster erzeugen	123
9.1.2	Das Fenster zentrieren	124
9.2	Zeichenfunktionen innerhalb des Fensters verwenden	125
9.2.1	Textausgaben	126
9.2.2	Zeichenelemente	126
9.2.3	Die Klasse Color verwenden	127
9.2.4	Bilder laden und anzeigen	128
9.3	Auf Fensterereignisse reagieren und sie behandeln	130
9.3.1	Fenster mit dem Interface WindowListener schließen	130
9.3.2	GUI-Elemente und ihre Ereignisse	132
9.3.2.1	Layout Manager	133
9.3.2.2	Die Komponenten Label und Button	133
9.3.2.3	Die Komponente TextField	134
9.4	Auf Mausereignisse reagieren	136
9.5	Zusammenfassung und Aufgaben	137
10	Tag 10: Appletprogrammierung	139
10.1	Kurzeinführung in HTML	139
10.2	Applets im Internet	140
10.3	Bauen eines kleinen Applets	141
10.4	Verwendung des Appletviewers	141
10.5	Eine Applikation zum Applet umbauen	143
10.5.1	Konstruktor zu init	143
10.5.2	paint-Methoden anpassen	144
10.5.3	TextField-Beispiel zum Applet umbauen	144
10.6	Flackernde Applets vermeiden	146
10.6.1	Die Ghosttechnik anwenden	147
10.6.2	Die paint-Methode überschreiben	149
10.7	Ein Beispiel mit mouseDragged	150
10.8	Zusammenfassung und Aufgaben	151
11	Tag 11: Techniken der Programmentwicklung	153
11.1	Der Begriff Algorithmus	153
11.2	Entwurfs-Techniken	154
11.2.1	Prinzip der Rekursion	154
11.2.2	Brute Force	156
11.2.3	Greedy	157
11.2.4	Dynamische Programmierung, Memoisierung	158
11.2.5	Divide and Conquer	160
11.3	Algorithmen miteinander vergleichen	160

11.4	Kleine algorithmische Probleme	161
11.4.1	Identifikation und Erzeugung von Primzahlen mit Brute Force	161
11.4.2	Sortieralgorithmen	162
11.4.2.1	InsertionSort	162
11.4.2.2	BubbleSort	163
11.4.2.3	QuickSort	165
11.4.3	Needleman-Wunsch-Algorithmus	166
11.5	Zusammenfassung und Aufgaben	168
12	Tag 12: Bildverarbeitung	171
12.1	Das RGB-Farbmodell	171
12.2	Grafische Spielerei: Apfelmännchen	173
12.2.1	Mathematischer Hintergrund	174
12.2.2	Fraktale	175
12.2.2.1	Implementierung eines einfachen Apfelmännchens	176
12.2.3	Die Klasse BufferedImage	177
12.2.4	Bilder laden und speichern	179
12.2.5	Bilder bearbeiten	184
12.2.5.1	Ein Bild invertieren	184
12.2.5.2	Erstellung eines Grauwertbildes	185
12.2.5.3	Binarisierung eines Grauwertbildes	186
12.3	Zusammenfassung und Aufgaben	187
13	Tag 13: Methoden der Künstlichen Intelligenz	189
13.1	Mustererkennung	189
13.1.1	k -nn	192
13.1.1.1	Visualisierung	193
13.1.2	k -means	195
13.1.2.1	Expectation-Maximization als Optimierungsverfahren	197
13.1.2.2	Allgemeine Formulierung des k -means Algorithmus	197
13.1.2.3	Implementierung des k -means	198
13.2	Spieltheorie	202
13.2.1	MinMax-Algorithmus	202
13.2.1.1	MinMax-Algorithmus mit unbegrenzter Suchtiefe	204
13.2.1.2	MinMax-Algorithmus mit begrenzter Suchtiefe und Bewertungsfunktion	204
13.2.1.3	Beispiel TicTacToe	205
13.3	Zusammenfassung und Aufgaben	208

- 14 Tag 14: Entwicklung einer größeren Anwendung** 211
 - 14.1 Entwurf eines Konzepts 211
 - 14.1.1 Klassendiagramm 213
 - 14.1.1.1 GUI Klassen 213
 - 14.1.1.2 Spiellogik 214
 - 14.1.1.3 Spieldatenverwaltung 214
 - 14.1.1.4 Komplettes Klassendiagramm 215
 - 14.2 Implementierung 215
 - 14.2.1 Klasse TeeTristBox 215
 - 14.2.2 Klasse TeeTristStein 216
 - 14.2.3 Klasse TeeTristSpielfeld 219
 - 14.2.4 Klasse SpielThread 223
 - 14.2.5 Klasse TeeTristPanel 227
 - 14.2.6 Klasse TeeTrist 228
 - 14.3 Spielen wir ein Spiel TeeTrist 228
 - 14.4 Dokumentation mit javadoc 229
 - 14.5 Zusammenfassung und Aufgaben 230
 - 15 Java – Weiterführende Konzepte** 233
 - 15.1 Professionelle Entwicklungsumgebungen 233
 - 15.2 Das Klassendiagramm als Konzept einer Software 234
 - 15.2.1 Klassendiagramm mit UML 234
 - 15.2.1.1 Klasse 234
 - 15.2.1.2 Vererbung 235
 - 15.2.1.3 Beziehungen zwischen Klassen 235
 - 15.3 Verwendung externer Bibliotheken 236
 - 15.4 Zusammenarbeit in großen Projekten 237
 - Glossar** 239
 - Literatur** 245
 - Sachverzeichnis** 249