

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufbau der Kapitel	1
1.2	Grundgerüst des Eclipse-Projekts	3
1.3	Grundgerüst für die Unit Tests	4
1.4	Anmerkung zum Programmierstil	5
1.5	Ausprobieren der Beispiele und Lösungen	9
I Grundlagen		11
2	Mathematische Aufgaben	13
2.1	Einführung	13
2.1.1	Römische Zahlen	17
2.1.2	Zahlenspielerien	18
2.2	Aufgaben	21
2.2.1	Aufgabe 1: Grundrechenarten (★☆☆☆☆)	21
2.2.2	Aufgabe 2: Zahl als Text (★★☆☆☆)	22
2.2.3	Aufgabe 3: Vollkommene Zahlen (★★☆☆☆)	22
2.2.4	Aufgabe 4: Primzahlen (★★☆☆☆)	23
2.2.5	Aufgabe 5: Primzahlpaare (★★☆☆☆)	23
2.2.6	Aufgabe 6: Prüfsumme (★★☆☆☆)	23
2.2.7	Aufgabe 7: Römische Zahlen (★★★★☆)	24
2.2.8	Aufgabe 8: Kombinatorik (★★☆☆☆)	24
2.2.9	Aufgabe 9: Armstrong-Zahlen (★★☆☆☆)	25
2.2.10	Aufgabe 10: Max Change Calculator (★★★★☆)	25
2.2.11	Aufgabe 11: Befreundete Zahlen (★★☆☆☆)	26
2.2.12	Aufgabe 12: Primfaktorzerlegung (★★★★☆)	26
2.3	Lösungen	27
2.3.1	Lösung 1: Grundrechenarten (★☆☆☆☆)	27
2.3.2	Lösung 2: Zahl als Text (★★☆☆☆)	30
2.3.3	Lösung 3: Vollkommene Zahlen (★★☆☆☆)	32
2.3.4	Lösung 4: Primzahlen (★★☆☆☆)	34
2.3.5	Lösung 5: Primzahlpaare (★★☆☆☆)	36

2.3.6	Lösung 6: Prüfsumme (★★☆☆☆)	40
2.3.7	Lösung 7: Römische Zahlen (★★★★☆)	41
2.3.8	Lösung 8: Kombinatorik (★★☆☆☆)	45
2.3.9	Lösung 9: Armstrong-Zahlen (★★☆☆☆)	48
2.3.10	Lösung 10: Max Change Calculator (★★★★☆)	51
2.3.11	Lösung 11: Befreundete Zahlen (★★☆☆☆)	53
2.3.12	Lösung 12: Primfaktorzerlegung (★★★☆☆)	54
3	Rekursion	57
3.1	Einführung	57
3.1.1	Mathematische Beispiele	57
3.1.2	Algorithmische Beispiele	61
3.1.3	Ablauf beim Multiplizieren der Ziffern einer Zahl	65
3.1.4	Typische Probleme	66
3.2	Aufgaben	69
3.2.1	Aufgabe 1: Fibonacci (★★☆☆☆)	69
3.2.2	Aufgabe 2: Ziffern verarbeiten (★★☆☆☆)	69
3.2.3	Aufgabe 3: ggT / GCD (★★☆☆☆)	70
3.2.4	Aufgabe 4: Reverse String (★★☆☆☆)	71
3.2.5	Aufgabe 5: Array Sum (★★☆☆☆)	71
3.2.6	Aufgabe 6: Array Min (★★☆☆☆)	71
3.2.7	Aufgabe 7: Konvertierungen (★★☆☆☆)	72
3.2.8	Aufgabe 8: Exponentialfunktion (★★☆☆☆)	73
3.2.9	Aufgabe 9: Pascal'sches Dreieck (★★☆☆☆)	74
3.2.10	Aufgabe 10: Zahlenpalindrome (★★★★☆)	74
3.2.11	Aufgabe 11: Permutationen (★★★★☆)	75
3.2.12	Aufgabe 12: Count Substrings (★★☆☆☆)	75
3.2.13	Aufgabe 13: Lineal (★★☆☆☆)	76
3.3	Lösungen	77
3.3.1	Lösung 1: Fibonacci (★★☆☆☆)	77
3.3.2	Lösung 2: Ziffern verarbeiten (★★☆☆☆)	79
3.3.3	Lösung 3: ggT / GCD (★★☆☆☆)	80
3.3.4	Lösung 4: Reverse String (★★☆☆☆)	83
3.3.5	Lösung 5: Array Sum (★★☆☆☆)	84
3.3.6	Lösung 6: Array Min (★★☆☆☆)	85
3.3.7	Lösung 7: Konvertierungen (★★☆☆☆)	86
3.3.8	Lösung 8: Exponentialfunktion (★★☆☆☆)	90
3.3.9	Lösung 9: Pascal'sches Dreieck (★★☆☆☆)	93
3.3.10	Lösung 10: Zahlenpalindrome (★★★★☆)	96
3.3.11	Lösung 11: Permutationen (★★★★☆)	99
3.3.12	Lösung 12: Count Substrings (★★☆☆☆)	102
3.3.13	Lösung 13: Lineal (★★☆☆☆)	106

4	Strings	109
4.1	Einführung	109
4.1.1	Die Klasse <code>String</code>	110
4.1.2	Die Klassen <code>StringBuffer</code> und <code>StringBuilder</code>	111
4.1.3	Die Klasse <code>Character</code>	112
4.1.4	Beispiele zu <code>Character</code> und <code>String</code>	113
4.2	Aufgaben	116
4.2.1	Aufgabe 1: Zahlenumwandlungen (★★☆☆☆)	116
4.2.2	Aufgabe 2: Joiner (★☆☆☆☆)	116
4.2.3	Aufgabe 3: Reverse String (★★☆☆☆)	117
4.2.4	Aufgabe 4: Palindrom (★★★☆☆)	117
4.2.5	Aufgabe 5: No Duplicate Chars (★★★☆☆)	118
4.2.6	Aufgabe 6: Doppelte Buchstaben entfernen (★★★☆☆)	118
4.2.7	Aufgabe 7: Capitalize (★★☆☆☆)	119
4.2.8	Aufgabe 8: Rotation (★★☆☆☆)	120
4.2.9	Aufgabe 9: Wohlgeformte Klammern (★☆☆☆☆)	120
4.2.10	Aufgabe 10: Anagramm (★☆☆☆☆)	121
4.2.11	Aufgabe 11: Morse Code (★☆☆☆☆)	121
4.2.12	Aufgabe 12: Pattern Checker (★☆☆☆☆)	122
4.2.13	Aufgabe 13: Tennis-Punktstand (★★★☆☆)	122
4.2.14	Aufgabe 14: Versionsnummern (★★☆☆☆)	123
4.2.15	Aufgabe 15: Konvertierung <code>strToLong</code> (★★☆☆☆)	123
4.2.16	Aufgabe 16: Print Tower (★★★☆☆)	124
4.3	Lösungen	125
4.3.1	Lösung 1: Zahlenumwandlungen (★★☆☆☆)	125
4.3.2	Lösung 2: Joiner (★☆☆☆☆)	128
4.3.3	Lösung 3: Reverse String (★★☆☆☆)	130
4.3.4	Lösung 4: Palindrom (★★★☆☆)	132
4.3.5	Lösung 5: No Duplicate Chars (★★★☆☆)	135
4.3.6	Lösung 6: Doppelte Buchstaben entfernen (★★★☆☆)	136
4.3.7	Lösung 7: Capitalize (★★☆☆☆)	138
4.3.8	Lösung 8: Rotation (★★☆☆☆)	142
4.3.9	Lösung 9: Wohlgeformte Klammern (★☆☆☆☆)	143
4.3.10	Lösung 10: Anagramm (★☆☆☆☆)	145
4.3.11	Lösung 11: Morse Code (★☆☆☆☆)	146
4.3.12	Lösung 12: Pattern Checker (★☆☆☆☆)	148
4.3.13	Lösung 13: Tennis-Punktstand (★★★☆☆)	150
4.3.14	Lösung 14: Versionsnummern (★★☆☆☆)	154
4.3.15	Lösung 15: Konvertierung <code>strToLong</code> (★★☆☆☆)	156
4.3.16	Lösung 16: Print Tower (★★★☆☆)	159

5	Arrays	163
5.1	Einführung	163
5.1.1	Eindimensionale Arrays	164
5.1.2	Mehrdimensionale Arrays	174
5.1.3	Typische Fehler	181
5.2	Aufgaben	182
5.2.1	Aufgabe 1: Gerade vor ungeraden Zahlen (★★☆☆☆)	182
5.2.2	Aufgabe 2: Flip (★★☆☆☆)	182
5.2.3	Aufgabe 3: Palindrom (★★☆☆☆)	182
5.2.4	Aufgabe 4: Inplace Rotate (★★★☆☆)	183
5.2.5	Aufgabe 5: Jewels Board Init (★★★☆☆)	183
5.2.6	Aufgabe 6: Jewels Board Erase Diamonds (★★★★☆)	185
5.2.7	Aufgabe 7: Spiral-Traversal (★★★★☆)	186
5.2.8	Aufgabe 8: Add One to Array As Number (★★☆☆☆)	186
5.2.9	Aufgabe 9: Sudoku-Checker (★★★☆☆)	187
5.2.10	Aufgabe 10: Flood-Fill (★★☆☆☆)	188
5.2.11	Aufgabe 11: Array Merge (★★☆☆☆)	189
5.2.12	Aufgabe 12: Array Min und Max (★★☆☆☆)	189
5.2.13	Aufgabe 13: Array Split (★★★☆☆)	190
5.2.14	Aufgabe 14: Minesweeper Board (★★★★☆)	191
5.3	Lösungen	193
5.3.1	Lösung 1: Gerade vor ungerade Zahlen (★★☆☆☆)	193
5.3.2	Lösung 2: Flip (★★☆☆☆)	197
5.3.3	Lösung 3: Palindrom (★★☆☆☆)	201
5.3.4	Lösung 4: Inplace Rotate (★★★☆☆)	203
5.3.5	Lösung 5: Jewels Board Init (★★★☆☆)	207
5.3.6	Lösung 6: Jewels Board Erase Diamonds (★★★★☆)	214
5.3.7	Lösung 7: Spiral-Traversal (★★★★☆)	223
5.3.8	Lösung 8: Add One to Array As Number (★★☆☆☆)	228
5.3.9	Lösung 9: Sudoku-Checker (★★★☆☆)	230
5.3.10	Lösung 10: Flood-Fill (★★☆☆☆)	236
5.3.11	Lösung 11: Array Merge (★★☆☆☆)	240
5.3.12	Lösung 12: Array Min und Max (★★☆☆☆)	244
5.3.13	Lösung 13: Array Split (★★★☆☆)	247
5.3.14	Lösung 14: Minesweeper Board (★★★★☆)	252

6	Datumsverarbeitung	259
6.1	Einführung	259
6.1.1	Die Aufzählungen <code>DayOfWeek</code> und <code>Month</code>	259
6.1.2	Die Klassen <code>LocalDate</code> , <code>LocalTime</code> und <code>LocalDateTime</code>	260
6.1.3	Die Klasse <code>ZonedDateTime</code>	262
6.1.4	Die Klasse <code>ZoneId</code>	263
6.1.5	Die Klasse <code>Duration</code>	264
6.1.6	Die Klasse <code>Period</code>	265
6.1.7	Datumsarithmetik	266
6.1.8	Formatierung und Parsing	268
6.2	Aufgaben	270
6.2.1	Aufgabe 1: Schaltjahre (★☆☆☆☆)	270
6.2.2	Aufgabe 2: Basiswissen Date-API (★★☆☆☆)	270
6.2.3	Aufgabe 3: Monatslänge (★★☆☆☆)	271
6.2.4	Aufgabe 4: Zeitzonen (★★☆☆☆)	271
6.2.5	Aufgabe 5: Zeitzonenberechnung (★★☆☆☆)	271
6.2.6	Aufgabe 6: Berechnungen mit <code>LocalDate</code> (★★☆☆☆)	272
6.2.7	Aufgabe 7: Kalenderausgabe (★★★☆☆)	272
6.2.8	Aufgabe 8: Wochentage (★☆☆☆☆)	273
6.2.9	Aufgabe 9: Sonntage und Schaltjahre (★★☆☆☆)	274
6.2.10	Aufgabe 10: <code>TemporalAdjuster</code> (★★★☆☆)	274
6.2.11	Aufgabe 11: <code>NthWeekdayAdjuster</code> (★★★☆☆)	275
6.2.12	Aufgabe 12: <code>PaydayTemporalAdjuster</code> (★★★☆☆)	275
6.2.13	Aufgabe 13: Formatting and Parsing (★★☆☆☆)	276
6.2.14	Aufgabe 14: Fault Tolerant Parsing (★★☆☆☆)	276
6.3	Lösungen	277
6.3.1	Lösung 1: Schaltjahre (★☆☆☆☆)	277
6.3.2	Lösung 2: Basiswissen Date-API (★★☆☆☆)	278
6.3.3	Lösung 3: Monatslänge (★★☆☆☆)	279
6.3.4	Lösung 4: Zeitzonenberechnung (★★☆☆☆)	280
6.3.5	Lösung 5: Zeitzonen (★★☆☆☆)	281
6.3.6	Lösung 6: Berechnungen mit <code>LocalDate</code> (★★☆☆☆)	282
6.3.7	Lösung 7: Kalenderausgabe (★★★☆☆)	284
6.3.8	Lösung 8: Wochentage (★☆☆☆☆)	287
6.3.9	Lösung 9: Sonntage und Schaltjahre (★★☆☆☆)	290
6.3.10	Lösung 10: <code>TemporalAdjuster</code> (★★★☆☆)	292
6.3.11	Lösung 11: <code>NthWeekdayAdjuster</code> (★★★☆☆)	293
6.3.12	Lösung 12: <code>PaydayTemporalAdjuster</code> (★★★☆☆)	295
6.3.13	Lösung 13: Formatting and Parsing (★★☆☆☆)	299
6.3.14	Lösung 14: Fault Tolerant Parsing (★★☆☆☆)	300

7	Basisdatenstrukturen: Listen, Sets und Maps	303
7.1	Einführung	303
7.1.1	Das Interface <code>Collection</code>	303
7.1.2	Listen und das Interface <code>List<E></code>	304
7.1.3	Mengen und das Interface <code>Set</code>	305
7.1.4	Schlüssel-Wert-Abbildungen und das Interface <code>Map</code>	305
7.1.5	Der Stack als LIFO-Datenstruktur	306
7.1.6	Die Queue als FIFO-Datenstruktur	307
7.2	Aufgaben	309
7.2.1	Aufgabe 1: Mengenoperationen (★★☆☆☆)	309
7.2.2	Aufgabe 2: List Reverse (★★☆☆☆)	309
7.2.3	Aufgabe 3: Duplikate entfernen (★★☆☆☆)	310
7.2.4	Aufgabe 4: Maximaler Gewinn (★★★☆☆)	310
7.2.5	Aufgabe 5: Längstes Teilstück (★★★☆☆)	311
7.2.6	Aufgabe 6: Eigener Stack (★★☆☆☆)	311
7.2.7	Aufgabe 7: Wohlgeformte Klammern (★★☆☆☆)	311
7.2.8	Aufgabe 8: Check Magic Triangle (★★★☆☆)	312
7.2.9	Aufgabe 9: Pascal'sches Dreieck (★★★☆☆)	312
7.2.10	Aufgabe 10: Häufigste Elemente (★★☆☆☆)	313
7.2.11	Aufgabe 11: Addition von Ziffern (★★★☆☆)	313
7.2.12	Aufgabe 12: Compound Key (★★☆☆☆)	314
7.2.13	Aufgabe 13: List Merge (★★☆☆☆)	314
7.2.14	Aufgabe 14: Excel Magic Select (★★☆☆☆)	315
7.3	Lösungen	316
7.3.1	Lösung 1: Mengenoperationen (★★☆☆☆)	316
7.3.2	Lösung 2: List Reverse (★★☆☆☆)	321
7.3.3	Lösung 3: Duplikate entfernen (★★☆☆☆)	324
7.3.4	Lösung 4: Maximaler Gewinn (★★★☆☆)	325
7.3.5	Lösung 5: Längstes Teilstück (★★★☆☆)	328
7.3.6	Lösung 6: Eigener Stack (★★☆☆☆)	331
7.3.7	Lösung 7: Wohlgeformte Klammern (★★☆☆☆)	333
7.3.8	Lösung 8: Check Magic Triangle (★★★☆☆)	338
7.3.9	Lösung 9: Pascal'sches Dreieck (★★★☆☆)	342
7.3.10	Lösung 10: Häufigste Elemente (★★☆☆☆)	344
7.3.11	Lösung 11: Addition von Ziffern (★★★☆☆)	346
7.3.12	Lösung 12: Compound Key (★★☆☆☆)	350
7.3.13	Lösung 13: List Merge (★★☆☆☆)	352
7.3.14	Lösung 14: Excel Magic Select (★★☆☆☆)	354

II Fortgeschrittenere und kniffligere Themen 359

8	Rekursion Advanced	361
8.1	Memoization	361
8.1.1	Memoization für Fibonacci-Zahlen	361
8.1.2	Memoization für Pascal'sches Dreieck	363
8.2	Backtracking	366
8.2.1	n-Damen-Problem	366
8.3	Aufgaben	370
8.3.1	Aufgabe 1: Türme von Hanoi (★★★☆☆)	370
8.3.2	Aufgabe 2: Edit Distance (★★★★☆)	371
8.3.3	Aufgabe 3: Longest Common Subsequence (★★★☆☆)	371
8.3.4	Aufgabe 4: Weg aus Labyrinth (★★★☆☆)	372
8.3.5	Aufgabe 5: Sudoku-Solver (★★★★☆)	373
8.3.6	Aufgabe 6: Math Operator Checker (★★★★☆)	374
8.3.7	Aufgabe 7: Wassereimer-Problem (★★★☆☆)	375
8.3.8	Aufgabe 8: Alle Palindrom-Teilstrings (★★★★☆)	376
8.3.9	Aufgabe 9: n-Damen-Problem (★★★☆☆)	376
8.4	Lösungen	377
8.4.1	Lösung 1: Türme von Hanoi (★★★☆☆)	377
8.4.2	Lösung 2: Edit Distance (★★★★☆)	383
8.4.3	Lösung 3: Longest Common Subsequence (★★★☆☆)	389
8.4.4	Lösung 4: Weg aus Labyrinth (★★★☆☆)	392
8.4.5	Lösung 5: Sudoku-Solver (★★★★☆)	395
8.4.6	Lösung 6: Math Operator Checker (★★★★☆)	403
8.4.7	Lösung 7: Wassereimer-Problem (★★★☆☆)	408
8.4.8	Lösung 8: Alle Palindrom-Teilstrings (★★★★☆)	411
8.4.9	Lösung 9: n-Damen-Problem (★★★☆☆)	415
9	Binärbäume	423
9.1	Einführung	423
9.1.1	Aufbau, Begrifflichkeiten und Anwendungsbeispiele	423
9.1.2	Binärbäume	424
9.1.3	Binärbäume mit Ordnung: binäre Suchbäume	425
9.1.4	Traversierungen	427
9.1.5	Balancierte Bäume und weitere Eigenschaften	430
9.1.6	Bäume für die Beispiele und Übungsaufgaben	432
9.2	Aufgaben	434
9.2.1	Aufgabe 1: Tree Traversal (★★☆☆☆)	434
9.2.2	Aufgabe 2: In-, Pre- und Postorder iterativ (★★★★☆)	434
9.2.3	Aufgabe 3: Tree-Höhe berechnen (★★☆☆☆)	434
9.2.4	Aufgabe 4: Kleinster gemeinsamer Vorfahre (★★★☆☆)	435
9.2.5	Aufgabe 5: Breadth-First (★★★☆☆)	435

9.2.6	Aufgabe 6: Level Sum (★★★★☆)	436
9.2.7	Aufgabe 7: Tree Rotate (★★★☆☆)	436
9.2.8	Aufgabe 8: Rekonstruktion (★★★☆☆)	437
9.2.9	Aufgabe 9: Math Evaluation (★★☆☆☆)	437
9.2.10	Aufgabe 10: Symmetrie (★★☆☆☆)	438
9.2.11	Aufgabe 11: Check Binary Search Tree (★★☆☆☆)	439
9.2.12	Aufgabe 12: Vollständigkeit (★★★★★)	439
9.2.13	Aufgabe 13: Tree Printer (★★★★★)	441
9.3	Lösungen	444
9.3.1	Lösung 1: Tree Traversal (★★☆☆☆)	444
9.3.2	Lösung 2: In-, Pre- und Postorder iterativ (★★★★☆)	446
9.3.3	Lösung 3: Tree-Höhe berechnen (★★☆☆☆)	454
9.3.4	Lösung 4: Kleinster gemeinsamer Vorfahre (★★★☆☆)	455
9.3.5	Lösung 5: Breadth-First (★★★☆☆)	459
9.3.6	Lösung 6: Level Sum (★★★★☆)	461
9.3.7	Lösung 7: Tree Rotate (★★★☆☆)	465
9.3.8	Lösung 8: Rekonstruktion (★★★☆☆)	468
9.3.9	Lösung 9: Math Evaluation (★★☆☆☆)	474
9.3.10	Lösung 10: Symmetrie (★★☆☆☆)	475
9.3.11	Lösung 11: Check Binary Search Tree (★★☆☆☆)	479
9.3.12	Lösung 12: Vollständigkeit (★★★★★)	481
9.3.13	Lösung 13: Tree Printer (★★★★★)	491
10	Suchen und Sortieren	501
10.1	Einführung Suchen	501
10.1.1	Suchen in Collections und Arrays	501
10.1.2	Binärsuche mit <code>binarySearch()</code>	503
10.2	Einführung Sortieren	504
10.2.1	Insertion Sort	504
10.2.2	Selection Sort	506
10.2.3	Merge Sort	508
10.2.4	Quick Sort	509
10.2.5	Bucket Sort	512
10.2.6	Schlussgedanken	513
10.3	Aufgaben	514
10.3.1	Aufgabe 1: Contains All (★★☆☆☆)	514
10.3.2	Aufgabe 2: Partitionierung (★★★☆☆)	514
10.3.3	Aufgabe 3: Binärsuche (★★☆☆☆)	515
10.3.4	Aufgabe 4: Insertion Sort (★★☆☆☆)	515
10.3.5	Aufgabe 5: Selection Sort (★★☆☆☆)	516
10.3.6	Aufgabe 6: Quick Sort (★★★☆☆)	516
10.3.7	Aufgabe 7: Bucket Sort (★★☆☆☆)	517
10.3.8	Aufgabe 8: Suche in rotierten Daten (★★★★☆)	517

10.4 Lösungen	519
10.4.1 Lösung 1: Contains All (★★☆☆☆)	519
10.4.2 Lösung 2: Partitionierung (★★★☆☆)	520
10.4.3 Lösung 3: Binärsuche (★★☆☆☆)	522
10.4.4 Lösung 4: Insertion Sort (★★☆☆☆)	526
10.4.5 Lösung 5: Selection Sort (★★☆☆☆)	527
10.4.6 Lösung 6: Quick Sort (★★★☆☆)	528
10.4.7 Lösung 7: Bucket Sort (★★☆☆☆)	530
10.4.8 Lösung 8: Suche in rotierten Daten (★★★★☆)	532
11 Schlusswort und ergänzende Literatur	539
11.1 Schlusswort	539
11.1.1 Gelerntes pro Kapitel	539
11.1.2 Bedenkenswertes	541
11.2 Knobelaufgaben	543
11.2.1 Goldsäcke – Fälschung entdecken	543
11.2.2 Pferderennen – schnellste drei Pferde ermitteln	544
11.3 Ergänzende Literatur	547
III Anhang	551
A Schnelleinstieg JShell	553
A.1 Java + REPL => <code>jshell</code>	553
B Kurzeinführung JUnit 5	559
B.1 Schreiben und Ausführen von Tests	559
B.1.1 Beispiel: Ein erster Unit Test	559
B.1.2 Grundlagen zum Schreiben und Ausführen von Tests	560
B.1.3 Behandlung erwarteter Exceptions mit <code>assertThrows()</code> ...	563
B.2 Parametrisierte Tests mit JUnit 5	564
C Schnelleinstieg O-Notation	567
C.1 Abschätzungen mit der O-Notation	567
C.1.1 Komplexitätsklassen	568
C.1.2 Komplexität und Programmlaufzeit	570
Literaturverzeichnis	571
Index	573