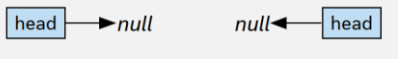
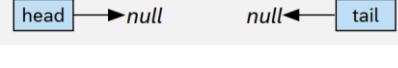
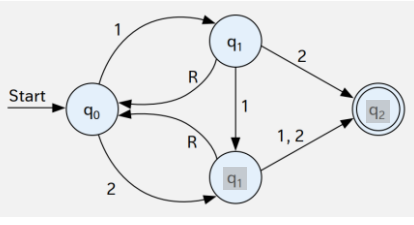
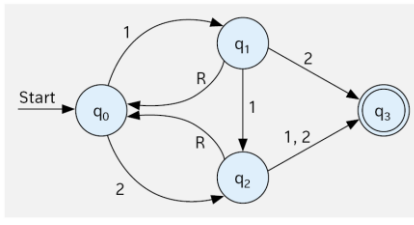
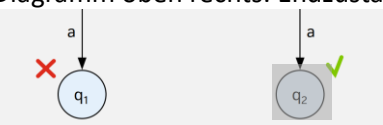
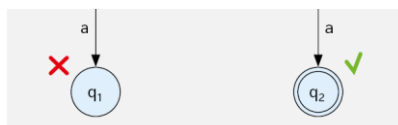
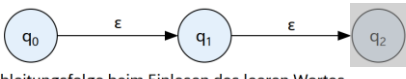
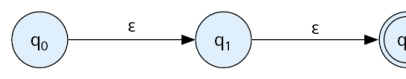
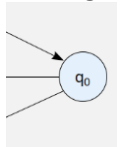
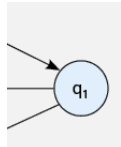
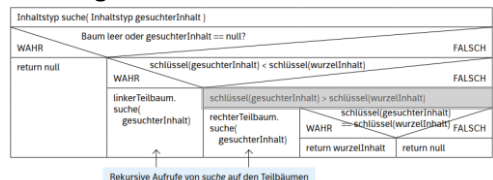
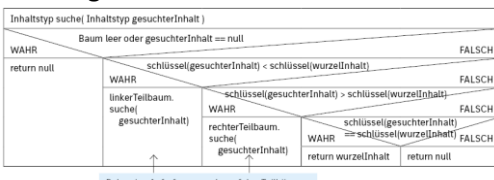
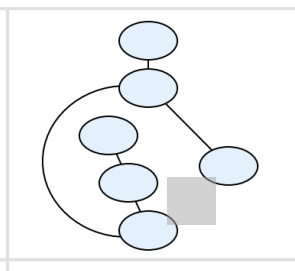
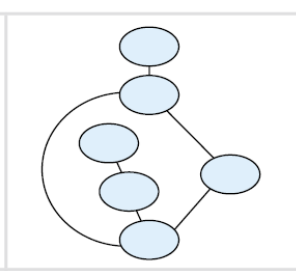


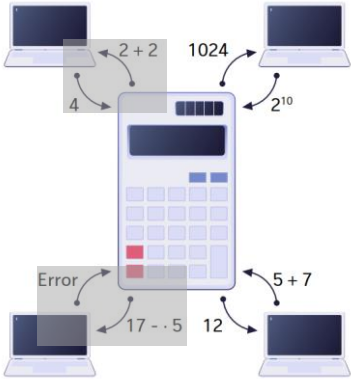
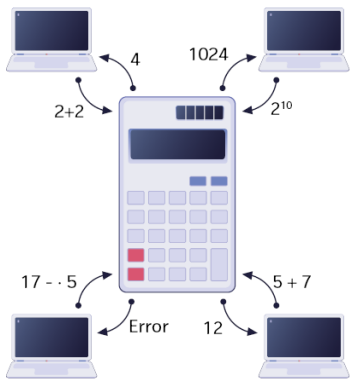


Stand: 26.09.24

Die folgenden Korrekturen sind in der digitalen Fassung des Buches in der Bibox und ab dem Druck A² / Jahr 2024 bereits durchgeführt. Rechtschreibkorrekturen sind hier nicht mit aufgelistet.

Seite	Bisheriger Text	Korrektur
22		Im Implementationsdiagramm fehlt jeweils ein Minus vor den Attributen
31	Im Implementationsdiagramm „Ware“: ~ name : String ~ preis : double	# name : String # preis : double
31	Im Quellcode unten: protected int temperatur; protected int mhd;	private int temperatur; private int mhd;
32	Im Implementationsdiagramm „Ware“: - name: String - preis : double	# name : String # preis : double
34	Im Implementationsdiagramm „Ware {abstract}“: - name : String - preis : double + gibInformation() : String Im Implementationsdiagramm „Kuehlware {abstract}“: - temperatur : int - mhd : int	# name : String # preis : double + <i>gibInformation()</i> : String [abstrakte Methode kursiv schreiben] # temperatur : int # mhd : int
35	Im Implementationsdiagramm „Ware {abstract}“: - name : String - preis : double	# name : String # preis : double
35		Methode „+gibInformation()“ in Klasse „Ware (abstract)“ muss kursiv geschrieben werden
35		Im Quellcode ergänzen: public abstract class Ware{ ... public abstract String gibInformation(); ... }
35	Im Quellcode: return ("Der" + geschmack + " " + name + "kosten" + " " preis);	Ersetzen durch: return ("Der/Die " + name + " schmeckt " + geschmack + " und kostet " + preis + ".");
40		In den Implementationsdiagrammen fehlt jeweils ein Minus vor den Attributen
60	Im Implementationsdiagramm „Spieler“: - hand : Karte[]	- deck : Karte[]
62	Aufgabe 2 a, Quellcode: Karte[] runde = new Karte[4]; Karte[2] = k3; Karte[4] = k2; Karte[2] = ax6; Karte[2] = k81; karteZumDeck(k9, 3); Karte[3] = null;	Ersetzen durch: Karte[] runde = new Karte[4]; runde[2] = k3; runde[4] = k2; runde[2] = ax6; runde[2] = k81; karteZumDeck(k9, 3); runde[3] = null;
62	Aufgabe 3a, Quellcode letzte Zeile: return z;	return x;
62	Aufgabe 3b: int mittelwert()	double mittelwert()

63	<p>Aufgabe 8 Quellcode: <code>tausche(t[0][i-1],t[0][i])</code></p> <p>Aufgabenteil a): „Implementieren Sie die Methode <code>void tausche(int pZeile, int pSpalte)</code>, die zwei Inhalte des zweidimensionalen Arrays vertauscht.“</p>	<p>Aufgabe 8 Quellcode: <code>tausche(i-1,i)</code></p> <p>Aufgabenteil a): „Implementieren Sie die Methode <code>void tausche(int pSpalte1, int pSpalte2)</code>, die zwei Inhalte der Zeile [0] vertauscht.“</p>
69	<p>Aufgabe 6 Quellcode: <code>Gleis lGleis = new Gleis();</code> <code>if(pGleis1.gibName() = (pGleis2.gibName())){</code></p>	<code>Stack<Waggon> lGleis = new Stack<>();</code>
71		<p>Bei der graphischen Darstellung muss dreimal „head“ durch „tail“ ersetzt werden:</p> 
72	<p>Quellcode: <code>public kundeKommt(Wartezettel pW)</code></p>	<code>public void kundeKommt(Wartezettel pW)</code>
73	<p>Aufgabe 2a): „...jeweils mit den Methoden enqueue, dequeue und top...“</p>	„...jeweils mit den Methoden enqueue, dequeue und front...“
78	<p>Unten bei „Löschen eines Elements“: „nachher: Das aktuelle Element bleibt gleich, last verweist auf das letzte Element der hinzugefügten Liste.“</p>	„nachher: Die Zeiger first, last und current verweisen auf null.“
79	<p>Oben bei „Fall 2a“: „nachher: Das Attribut head verweist auf den Nachfolger des bisher ersten Elements. Der Verweis vom Attribut tail ändert sich nicht.“</p>	„nachher: Das Attribut last verweist auf den Vorgänger des bisher letzten Elements, current verweist nun auf null. Der Verweis vom Attribut first ändert sich nicht.“
80	<p>Diverse Male: „Nebenrollenverwaltung“</p>	„Nebenrollenvermittlung“
83	<p>Aufgabe 4 c und d): <code>lichterketteAnhaengen(List pLichterkette)</code></p>	<code>lichterketteAnhaengen(List<LED> pLichterkette)</code>
102	<p>In der Methodendokumentation: „Berechner“</p>	„Bezeichner“
104	<p>Quellcode: <code>int a = (int) (Math.random()*31);</code> <code>int b = (int) (Math.random()*31);</code></p>	<code>int a = (int) (Math.random()*32);</code> <code>int b = (int) (Math.random()*32);</code>
107	<p>Quellcode unten rechts: <code>for(int i=1; i<n; i++)</code></p>	<code>for(int i=1; i<n-1; i++)</code>
119	<p>Quellcode: <code>if (akt > pivot)</code></p>	<code>if (akt < pivot)</code>
127	<p>Quellcode: <code>while (hilfsliste.hasAccess() && akt <= hilfsliste.getContent())</code></p>	<code>while (hilfsliste.hasAccess() && akt > hilfsliste.getContent())</code>
148	<p>Aufgabe 3, Beschriftung der Zustände im Diagramm</p> 	
151	<p>Diagramm oben rechts: Endzustand</p> 	

151	Diagramm unten rechts: Endzustand  Ableitungsfolge beim Einlesen des leeren Wortes	 Ableitungsfolge beim Einlesen des leeren Wortes
152	Tabelle bei „Schritt 1“, Eintrag letzte Zeile, letzte Spalte: \emptyset	$\{q_3\}$
161	Aufgabe 1b: „{{{}}}, {{{{}}}}“	„{{{}}}, {{{{}}}}“ (also 5 bzw. 6 Klammerpaare)“
161	Aufgabe 3, Tabelle, Vorletzte Zeile, Spalte „Keller“: A# Letzte Zeile, Spalte „Keller“: [Leer]	A #
163		Bei den Tabellen zu δ und λ sollte jeweils die erste Zeile farblich markiert sein, nicht die zweite.
163	Im Diagramm: Beschriftung 	
164	Aufgabe 3: Popcorn klein, Popcorn groß	Popcorn klein 0,50 €, Popcorn groß 1,00 €
167	Letzter Absatz: „Sie besteht aus den drei Silben <i>la</i> , <i>le</i> und <i>lu</i> , die beliebig kombiniert werden können. Am Ende jedes Wortes muss die Silbe <i>lu</i> stehen.“	„Jedes Wort der Sprache besteht aus genau drei Silben. Die drei Silben sind <i>la</i> , <i>le</i> und <i>lu</i> . Am Ende eines jeden Wortes muss die Silbe <i>lu</i> stehen. Die ersten beiden Silben können aus einer beliebigen Kombination der Silben <i>la</i> , <i>le</i> und <i>lu</i> bestehen.“
175	Aufgabe 11b: „reguläre rechtslineare“	„rechtsreguläre“
187	Aufgabe 2c: $T \rightarrow (VZN.N \mid f,$	$T \rightarrow (VZN.N \mid (f,$
207	Aufgabe 12b: $2^{(h-1)}-1$	$2^{(h+1)}-1$
210	„Da „r“ lexikografisch nach „k“ kommt, wird im linken Teilbaum weitergesucht.“	„Da „r“ lexikografisch nach „k“ kommt, wird im rechten Teilbaum weitergesucht.“
210	Struktogramm:  Rekursive Aufrufe von suche auf den Teilbäumen	Struktogramm:  Rekursive Aufrufe von suche auf den Teilbäumen
211	Mehrmals: „aqua99“ Im Graphen: „alfred43 < alfred43“	„aqua41“ „alfred43 < aqua41“
217	„Beispielsweise stellt „Münster“ – „79“ – „80“ – „Hamm“ – „Lünen“ einen Weg vom Knoten „Münster“ zum Knoten „Lünen“ dar.“	„Beispielsweise stellt „Münster“ – „79“ – „80“ – „Werne“ – „Lünen“ einen Weg vom Knoten „Münster“ zum Knoten „Lünen“ dar.“
218	Aufgabe 5: eine Kante fehlt im rechten Graphen  Nach dem Regensturm	 Nach dem Regensturm

230	„Aus der Liste wird nun der Knoten mit der kürzesten Entfernung zu Haltern herausgesucht und von dort aus weiter erkundet.“	„Aus der Liste wird nun der Knoten mit der kürzesten Entfernung zu „Lüdinghsn“ herausgesucht und von dort aus weiter erkundet.“
239	Tabelle rechts: 01 ADD 13	01 ADD 14
272		Korrekturen an den Implementationsdiagrammen
273		Bei „Aktionen ausgehend vom Server, Aktion 2“: Client und Server vertauscht bei „Methodenaufruf“
274		Bei „Aktionen ausgehend vom Server, Aktion 4“: Client und Server vertauscht bei „Methodenaufruf“
276		Korrekturen an den Implementationsdiagrammen
279	Aufgabe 8, Beschriftung an den Pfeilen 	
286	Rechengesetze Modulo-Operator: Rechengesetze: $(x + y) \bmod b = x \bmod b + y \bmod b$ z. B. $(7 + 16) \bmod 5 = 23 \bmod 5 = 3 = 2 + 1 = 7 \bmod 5 + 16 \bmod 5$ $(x \cdot y) \bmod b = x \bmod b \cdot y \bmod b$ z. B. $(12 \cdot 2) \bmod 5 = 24 \bmod 5 = 4 = 2 \cdot 2 = 12 \bmod 5 \cdot 2 \bmod 5$	Rechengesetze: $(x + y) \bmod b = (x \bmod b + y \bmod b) \bmod b$ z. B. $(7 + 16) \bmod 5 = 23 \bmod 5 = 3 = 2 + 1 = (7 \bmod 5 + 16 \bmod 5) \bmod 5$ $(x \cdot y) \bmod b = (x \bmod b \cdot y \bmod b) \bmod b$ z. B. $(12 \cdot 2) \bmod 5 = 24 \bmod 5 = 4 = 2 \cdot 2 = (12 \bmod 5 \cdot 2 \bmod 5) \bmod 5$
303	Letzte Tabelle, letzte Zeile, Einträge: Verlag.VID Name Ort 456 Westermann Paderborn	Diese Einträge bleiben leer beim Left Join: Verlag.VID Name Ort
306	In „Tabelle Gericht“: Gewinn Text [VARCHAR]	Gewinn Dezimal [DECIMAL]
323	Grafik unten rechts 	n und 1 statt m und n 
355	Im Implementationsdiagramm „Person“: - vorname : String - nachname : String - alter : int	# vorname : String # nachname : String # alter : int