S. 11 Mitte *31.10.11*	FALSCH: wie wir sehen werden, gibt es zusätzliche Schlüsselwörter in C++ wie z. B. `class` , `public` oder `pivate`. RICHTIG: wie wir sehen werden, gibt es zusätzliche Schlüsselwörter in C++ wie z. B. `class` , `public` oder `private`.

S. 28 oben
S. 29 oben
 	
2.1.2016
	STATT:
	`vector<T>::size_type i;`
	
	BESSER:
	`typename vector<T>::size_type i;`
	
	BEMERKUNG:
	Wird von manchen Compilern zwar übersetzt, ist aber erst durch das Schlüsselwort `typename` standardkonform.

S.32 Mitte *2.1.2016*	FALSCH: Wenn die Variable `v` vom Datentyp `vector[double]` ist, … RICHTIG: Wenn die Variable `v` vom Datentyp `vector<double>` ist, …

S. 46 oben
 	
20.4.11
	FALSCH:
	`struct BKnoten`
	`{`
	`  struct Knoten *left;  // Zeiger auf linken Nachfolger`
	`  struct Knoten *right;  // Zeiger auf rechten Nachfolger`
	`  int id;  // Benutzer-ID`
	`  short priority;`
	`  char name[STR_SIZE];`
	`};`
	
	RICHTIG:
	`struct BKnoten`
	`{`
	`  struct BKnoten *left;  // Zeiger auf linken Nachfolger`
	`  struct BKnoten *right;  // Zeiger auf rechten Nachfolger`
	`  int id;  // Benutzer-ID`
	`  short priority;`
	`  char name[STR_SIZE];`
	`};`

S. 49 Mitte oben *2.1.2016*	FALSCH: ***Induktionsschritt*:** Es sei nun $k_0 \geq 0$. RICHTIG: ***Induktionsschritt*:** Es sei nun $k_0 \geq 1$.
S. 57 Mitte *31.10.11*	FALSCH: hat die Wurzel den Wert *w* und gilt $w < s$, dann durchsuche den linken Teilbaum der Wurzel; RICHTIG: hat die Wurzel den Wert *w* und gilt $s < w$, dann durchsuche den linken Teilbaum der Wurzel;

<table>
<tr><td>S. 97 oben
31.10.11</td><td>FALSCH:
#include "Stack2 .hpp”

RICHTIG:
#include "stack2.hpp”</td></tr>
<tr><td>S. 104 oben
31.10.11</td><td>FALSCH:
#include "Stack1a .hpp”

RICHTIG:
#include "stack1a.hpp”</td></tr>
<tr><td>S. 105 oben
31.10.11</td><td>FALSCH:
#include "Stack3 .hpp”

RICHTIG:
#include "stack3.hpp”</td></tr>
<tr><td>S. 105 oben
Kommentar zum Quelltext Stack3::pop()
23.11.10</td><td>FALSCH:
// Ist der Stack leer, dann wird -1 zurückgeben

RICHTIG:
// Ist der Stack leer, dann wird <code>fehlerwert</code> zurückgeben</td></tr>
<tr><td>S. 126 unten

Tabelle 4.1: letzte Tabellenzeile

27.1.17</td><td>FALSCH:
<table><tr><td>2^n</td><td>1024</td><td>$1{,}0\cdot10^{6}$</td><td>$1{,}3\cdot10^{6}$</td><td>$1{,}3\cdot10^{30}$</td><td>$1{,}3\cdot10^{301}$</td><td>$1{,}3\cdot10^{301029}$</td></tr></table>
RICHTIG:
<table><tr><td>2^n</td><td>1024</td><td>$1{,}0\cdot10^{6}$</td><td>$1{,}1\cdot10^{15}$</td><td>$1{,}3\cdot10^{30}$</td><td>$1{,}1\cdot10^{301}$</td><td>$9{,}9\cdot10^{301029}$</td></tr></table></td></tr>
<tr><td>S. 129 oben

23.11.10</td><td>FALSCH:
Satz 4.1 Existiert der Grenzwert $\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}$ (und ist endlich), dann folgt $g(n)=O(f(n))$

RICHTIG:
Satz 4.1 Existiert der Grenzwert $\lim\limits_{n\to\infty}\frac{g(n)}{f(n)}$ (und ist endlich), dann folgt $g(n)=O(f(n))$</td></tr>
<tr><td>S. 129 oben

Bemerkung:

23.11.10</td><td>Ebenso müssen im Beweis die Funktionen f und g vertauscht werden:
FALSCH:
Aus $\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}=D<\infty$ folgt sofort aus der Definition des Grenzwerts, dass es eine natürliche Zahl n_0 gibt, sodass $f(n)\le(D+1)\cdot g(n)$ für alle $n\ge n_0$. Also gilt $g(n)=O(f(n))$.

RICHTIG:
Aus $\lim\limits_{n\to\infty}\frac{g(n)}{f(n)}=D<\infty$ folgt sofort aus der Definition des Grenzwerts, dass es eine natürliche Zahl n_0 gibt, sodass $g(n)\le(D+1)\cdot f(n)$ für alle $n\ge n_0$. Also gilt $g(n)=O(f(n))$.</td></tr>
<tr><td>S. 138 unten</td><td>FALSCH:
Im i-ten Aufruf wird $V(n)$ um den Summanden $i-1$ erhöht; insgesamt erhalten wir somit ein quadratisches Worst-Case-Verhalten:
$$V(n)=\sum_{i=2}^{n}(i-1)=\sum_{i=1}^{n-1}i=\frac{n(n-1)}{2}=\Theta(n^2)$$
RICHTIG:
Im i-ten Aufruf wird $V(n)$ um den Summanden $n-i$ erhöht; insgesamt erhalten wir somit ein quadratisches Worst-Case-Verhalten:</td></tr>
</table>

23.11.10	$V(n)=\sum_{i=1}^{n}(n-i)=\sum_{i=1}^{n-1} i=\frac{n(n-1)}{2}=\Theta(n^2)$
S. 146 unten *2.1.2016*	FALSCH: $E:=\sum_{i=1}^{b} t_i$ RICHTIG: $E:=\frac{1}{b}\sum_{i=1}^{b} t_i$
S. 177 unten *2.1.2016*	STATT: Auf dasselbe Ergebnis kommt man auch, wenn zuerst Spalten- und danach Zeilenreduktionen durchgeführt werden. BESSER: Auf dasselbe Ergebnis kommt man auch in diesem Beispiel, wenn zuerst Spalten- und danach Zeilenreduktionen durchgeführt werden. Im Allgemeinen können sich aber unterschiedliche Schranken ergeben.
S. 182 oben *20.4.11*	FALSCH: Die ersten Elemente der Fibonacci-Folge $\{f(n)\}_{n\in\mathbb{N}}$ lauten also 0, 1, 2, 3, 5, 8, 13, 21, 34, … RICHTIG: Die ersten Elemente der Fibonacci-Folge $\{f(n)\}_{n\in\mathbb{N}}$ lauten also 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …
S. 182 unten *20.4.11*	FALSCH: Danach wird mithilfe der Tabellenwerte der Wert *f*(3) berechnet und das Ergebnis in der Tabelle gespeichert. Ebenso wird mit den Werten *f*(4), *f*(5), … , *f*(9) verfahren … RICHTIG: Danach wird mithilfe der Tabellenwerte der Wert *f*(*2*) berechnet und das Ergebnis in der Tabelle gespeichert. Ebenso wird mit den Werten *f*(3), *f*(4), … , *f*(9) verfahren …
S. 196 oben *2. Absatz* *27.1.17*	FALSCH: Innerhalb der Klasse gibt es eine … RICHTIG: Innerhalb der Klasse *NP* gibt es eine …
S. 202 unten *Übungsaufgabe 6.2: vorletzte Zeile der Programmtabelle* *27.1.17*	FALSCH: Z_2 | 1, L, S | 0, R, Z_2 | 1, R, Z_2 RICHTIG: Z_2 | 1, L, Z_1 | 0, R, Z_2 | 1, R, Z_2
S. 211 Mitte	FALSCH: Die Postorder-Ausgabe liefert folgende Werte: **50, 20, 10, 32, 35, 30, 64, 70, 60, 90, 80, 40**

20.4.11	RICHTIG: Die Postorder-Ausgabe liefert folgende Werte: **`50, 20, 10, 32, 35, 30, 64, 70, 60, 90, 80, 14`**
S. 226 Mitte *31.10.11*	FALSCH: #include "Stack3 .hpp” RICHTIG: #include "stack3.hpp”
S. 227 oben *31.10.11*	FALSCH: #include "Stack4 .hpp” RICHTIG: #include "stack4.hpp”
S. 233 Mitte *2.1.2016*	FALSCH: Insgesamt ergibt sich die Schranke 19 + 0 = 19. ~~Allgemein kann man zeigen, dass sich stets dasselbe Ergebnis ergibt, unabhängig davon, ob zuerst die Spalten- oder die Zeilenreduktionen durchgeführt werden.~~ RICHTIG: Insgesamt ergibt sich die Schranke 19 + 0 = 19. BEMERKUNG: Wie das Beispiel der nächsten Übungsaufgabe zeigt, können sich durchaus unterschiedliche Schranken ergeben, wenn die Reihenfolge der Zeilen- oder die Spaltenreduktionen vertauscht wird.
S. 241 oben *Letzter Satz des 2. Absatzes* *27.1.17*	FALSCH: Im Fall A wird vom Zustand Z_3 in den Zustand E und im Fall B wird vom Zustand Z_2 in den Zustand S gewechselt. RICHTIG: Im Fall A wird vom Zustand Z_3 in den Zustand E und im Fall B wird vom Zustand Z_2 in den Zustand Z_1 gewechselt.
S. 241 Mitte *Erster Satz des letzten Absatzes* *27.1.17*	FALSCH: Im Fall B wird also wie beschrieben in den Zustand S gewechselt und … RICHTIG: Im Fall B wird also wie beschrieben in den Zustand Z_1 gewechselt und …