

Musterlösungen zu Kapitel 3

aus H.Mössenböck: Sprechen Sie Java?

Aufgabe 3: Vereinfachung Boolescher Ausdrücke

gegebener Ausdruck	vereinfachter Ausdruck
$!(x < y \ \&\& \ y < z)$	$x \geq y \ \ y \geq z$
$(x \neq y) \ \ !(y == z \ \&\& \ y == x)$	$x \neq y \ \ y \neq z \ \ y \neq x$ $\Rightarrow x \neq y \ \ y \neq z$
$!(x \geq -3 \ \&\& \ x \leq 0) \ \&\& \ 5 < x$	$(x < -3 \ \ x > 0) \ \&\& \ 5 < x$ $\Rightarrow x > 5$

Aufgabe 4: Werte von a, b, c bei gegebenem Programmstück

x	a	b	c
-1	false	true	false
0	false	true	false
5	true	false	false
10	true	true	false

Aufgabe 5: Assertionen

```
if (i > p) {
    if (i < q) /* p < i < q */ ... else /* i > p && i >= q */ ...
} else {
    if (i > r) /* r < i <= p */ ... else /* i <= p && i <= r */ ...
}
```

Aufgabe 6: Kurzschlußauswertung

```
if (a && b) x = 0;
```

ist äquivalent zu

```
if (a) if (b) x = 0;
```

Aufgabe 12: Codevereinfachung

Erstes Codestück: Durch Einsetzen von Assertionen erhält man:

```
if (b == 0)
    // b == 0
    a = 2 * c;
else if (c != 0)
    // b != 0 && c != 0
    a = a * b + 2 * c;
else
    // b != 0 && c == 0
    a = a * b;
```

Man sieht, dass der Term $a * b$ wegfällt, wenn $b == 0$ ist. Ebenso fällt der Term $2 * c$ weg, wenn $c == 0$ ist. Daher erhält man das gleiche Ergebnis, wenn man einfach nur schreibt:

```
a = a * b + 2 * c;
```

Zweites Codestück: Wir setzen zuerst nach jeder Abfrage eine Assertion ein und erhalten:

```
if (x < 0 && y < 0)
    // x < 0 && y < 0
    a = x * y;
else if (x < 0)
    // x < 0 && y >= 0
    a = x * (-y);
else if (y > 0)
    // x >= 0 && y > 0
    a = (-x) * (-y);
else
    // x >= 0 && y <= 0
    a = x * (-y);
```

Man sieht, daß in jedem Zweig der Absolutwert von $x * y$ berechnet wird. Das Codestück kann also ersetzt werden durch:

```
a = Math.abs(x * y);
```

oder durch

```
a = x * y;
if (a < 0) a = -a;
```