

# Mathematik

## Zettel 1

### 1.1 Operatorpräzedenz

Sowohl in der Mathematik, wie auch in der Programmierung werden auswertbare Ausdrücke als Kombination von Operatoren (+, -, \*, /, %) und Operanden (A, b, x, i, 8, etc.) dargestellt. Die als Operationen bezeichneten Kombinationen, unterliegen strikten Reihenfolgen. Diese Ordnung wird als Operatorpräzedenz bezeichnet. Während die Operatorpräzedenz in der klassischen Algebra hinreichend bekannt sein sollte, können derartige Sachverhalte in der Programmierung schnell zu Verwirrung führen. Sauberes Klammern von Termen kann bei der Analyse behilflich sein

a. Gegeben sei folgender Ausdruck:

$$\frac{3(x+2)^2 - (-2)2^{2^2}}{3x+6} = \frac{-2^{3^2} + (2^3)^2 x}{(2^2)^3}$$

a.1 Setzen Sie bei jeder Rechenoperation in dieser Gleichung - entbehrliche - zusätzliche Klammerpaare um den Vorrang der *wichtigsten* Operation deutlich zu machen

a.2 Lösen Sie nach x.

a.3 Machen Sie die Probe, indem Sie  $x = -34/15$  einsetzen!

b. In einer willkürlich gewählten Programmiersprache gilt folgende Präzedenz:

1. Funktionsausdrücke (etwa  $\sin(x)$ ,  $\text{sqrt}(x)$ , usw.)
2. Geklammerte Rechenausdrücke
3. Potenzbildungen
4. Multiplikationen oder Divisionen
5. Additionen oder Subtraktionen
6. Vergleichsoperationen (<, >, etc.)
7. Logische Negation (*nicht*)
8. Logische Konjunktion (*und*)
9. Logische Disjunktion (*oder*)

b.1 Lassen Sie nach diesen Prioritätsregeln alle entbehrlichen Klammern im folgenden Ausdruck weg:

$$(((2 * (\sqrt{x})) - ((\sin(a+b))/(x-y))) < (100 * ((k+m)^2))) \text{und} (\text{nicht}(x > 2))) \text{oder} ((\sqrt{(x^2 + 1)}) > 100)$$

Der Übersetzer einer Programmiersprache ahndet unnachsichtig alle Verstöße gegen die Syntax von Ausdrücken, insbesondere fehlerhafte Klammersetzung!