

Aufgabenblatt für das 1. Vertiefungstutorium

Im Folgenden wollen wir einen Taschenrechner, zum Beispiel in einer `class Calculator`, implementieren. Um uns dabei auch mit Stringfunktionen vertraut zu machen, wollen wir einen *String* direkt parsen anstatt *Integer.parseInt* zu benutzen.

1. Machen sie sich mit *String.charAt* und *String.length* vertraut (JavaDoc)
2. Schreiben sie eine Funktion `public Integer toInt(String number)`, die eine Ganzzahl einlesen kann
 - Die Funktion soll sowohl positive als auch negative Zahlen einlesen können, wobei bei positiven das Vorzeichen optional sein soll
 - Eine korrekte Zahl bestehe im Moment bis auf das Vorzeichen nur aus beliebig vielen Ziffern 0-9
 - Machen sie sich Gedanken darüber, was passieren soll, wenn *number* keine Zahl ist (Was ist der Unterschied zwischen *Integer* und *int*?)
3. Testen sie ihre Methode `public Integer toInt(String number)`
Schreiben sie dafür eine *main*-Methode. Versuchen sie dabei sowohl manuelles als auch automatisches Testen umzusetzen. Versuchen sie „interaktives“ Testen, das kein neukompilieren benötigt um andere Zahlen zu testen, zu realisieren (Hinweis: *args*).
4. Schreiben sie eine Funktion `public Integer calculate(String operation, Integer operand0, Integer operand1)` die das Ergebniss einer mathematischen Operation der Form *operand0 operation operand1* berechnet und zurückgibt
Implementieren sie die Operationen $+$, $-$
Beachten sie die Fälle wenn die Operation oder ein Operand ungültig ist, verfahren sie in diesem Fall analog zu ungültigen Zahlen
5. Testen sie `public Integer calculate(String operation, Integer operand0, Integer operand1)`
6. Schreiben sie eine Methode `public Integer calculate(String[] calculation)`
calculation enthalte Zahlen und Operationen in der gewohnten Reichenfolge, also zB $5 + 4 - 3$
Verwenden sie `public Integer toInt(String number)` und `public Integer calculate(String operation, Integer operand0, Integer operand1)`
7. Schreiben sie eine *main*-Methode, die `public Integer calculate(String[] calculation)` mit *args* als Parameter aufruft

Fertig!