

Arbeitsblatt: PSPP

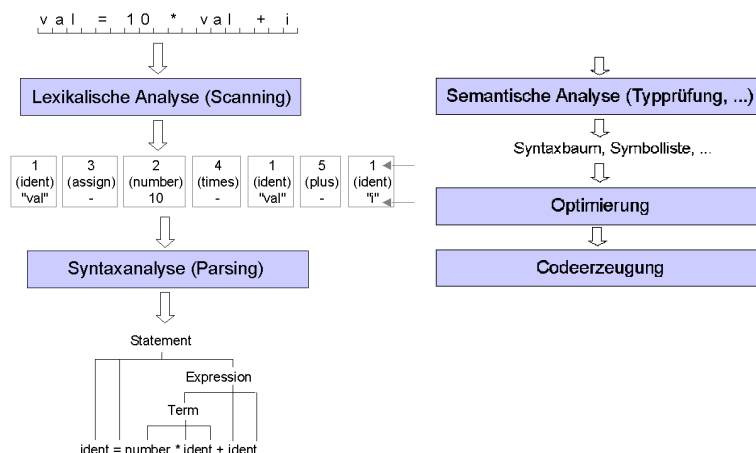
Name:

Kurznamen:

Berechnung von arithmetischen Ausdrücke

In diesem Praktikum werden ein Scanner und ein Parser eingesetzt, um arithmetische Ausdrücke in Fließkommazahlen auszuwerten und zu berechnen, z.B.: $(1 + 3) * 5 = 20$.

Dabei soll die Berechnung direkt im Parser durchgeführt werden. Eine Grundstruktur von Scanner und Parser ist vorgegeben. Der Ablauf vom bei der Verarbeitung stellt folgende Grafik dar:



Der Scanner

Mit dem Scanner wird der arithmetische Ausdruck (Zeichenstrom) in folgende Teile (Tokens) zerlegt: Zahlen, Operatoren und Klammern. Dieser Tokenstrom wird dann dem Parser übergeben.

Der Parser

Der Parser in unserem Beispiel kann Ausdrücke aus der Sprache mit folgender Grammatik parsen:

```

Expression = Term { ("+" | "-") Term }
Term = Faktor { ("*" | "/" ) Faktor }
Faktor = Zahl | "(" Expression ")"
  
```

Ausdruck, Term und Faktor sind sog. Produktionen. Für jede Produktion wird im Parser eine Methode gleichen Namens entwickelt, welche analog der Grammatik diese überprüft und allenfalls rekursiv die anderen Produktionsmethoden aufruft. So ruft z.B. die Methode `Factor` diejenige Methode der Produktion `Ausdruck (expr)` auf.

Aufgabe 1

Der Scanner soll so umgeschrieben werden, dass Fließkommazahlen (vom Typ `Double`) verarbeitet werden können. Dazu muss die Methode `readNumber` im Scanner so angepasst werden, dass Fließkommazahlen erkannt werden können.

Hinweis:

- Überlegen Sie sich den Automaten für Fließkommazahlen mit folgender Grammatik `Ziffer {Ziffer} ['.' {Ziffer}]` und implementieren Sie diesen im Scanner; *optional: auch mit Vorzeichen und Exponent.*

Abgabe:

Praktikum: **PS2.1**

Filename: **Scanner.java**

Aufgabe 2

Der Parser ist noch unvollständig. Ihre Aufgabe ist es den Parser so zu ergänzen, dass er einfache arithmetische Ausdrücke direkt ausrechnen kann. Die Berechnung soll als sog. Stackrechner implementiert werden: Bei einer Zahl wird diese einfach auf den Stack geschoben. Bei einem Operanden wird die Operation mit den beiden obersten Werten auf dem Stack ausgeführt und das Resultat wieder auf dem Stack abgelegt. Die Berechnung soll in der `start()` Methode durchgeführt werden.

Abgabe:

Praktikum: **PS2.2**

Filename: **Calculator.java**

Aufgabe 3

Es sollen auch die beiden Konstanten "PI" und "E" in Ausdrücken verwendet werden können: z.B. `4*PI + E` als Ausdruck. Im Scanner muss dafür die Methode `readName` implementiert werden. Als *kind* soll `IDENT` zurückgegeben werden und in *str* dann "PI" oder "E" stehen.

Abgabe:

Praktikum: **PS2.3**

Filename: **Scanner.java**

Zusätzlich muss noch die `Factor` Methode erweitert werden, damit beide Konstanten erkannt und die entsprechenden Werte eingesetzt werden.

Hinweis:

- Die Konstantenwerte werden (ausschliesslich) im Parser behandelt.

Abgabe:

Praktikum: **PS2.4**

Filename: **Calculator.java**