**Übungen zu Praktische Informatik: Datenstrukturen** **SS 2004**

**Übung 2**

Name: _____ Tutor: _____

Matrikelnummer: _____ Punkte: _____

Gruppe: _____ Abgabe: Di, 30. 3. 2004

---

## Syntaxanalyse und Auswertung von Ausdrücken

Gegeben ist folgende Grammatik:

```
Expr = Expression ";".
Expression = Term { ( "+" | "-" ) Term }.
Term = Factor { ( "*" | "/" ) Factor }.
Factor = number | "(" Expression ")" | "sqrt" "(" Expression ")".
```

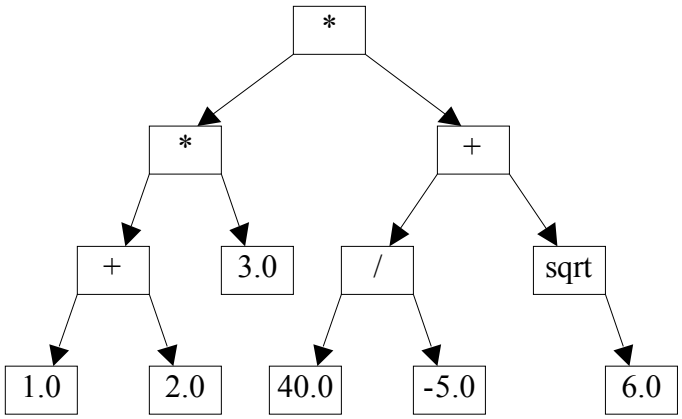
*number* ist eine Fließkommazahl (der Form "1", "1.0", "-1.0", "-1.0E-3", etc.)

*sqrt* bezeichnet eine Funktion für die Quadratwurzel

Die Operatoren (-, +, * und /) werden von links nach rechts ausgewertet.

Schreiben Sie ein Programm, welches eine Expression einliest und in einen Binärbaum (Syntaxbaum) umwandelt. Einen möglichen Vorschlag für einen Baumknoten finden Sie auf der Übungsseite.

Beispiel: (1.0+2.0)*3*(4E1/-5+sqrt(6.0));




das entspricht folgender Klammerung: (((1.0+2.0)*3.0)*((40.0/-5.0)+sqrt(6.0))).

Achten Sie darauf, dass man von außen nicht auf die interne Datenstruktur (d.h. die Knoten) des Baumes zugreifen kann ("Information hiding"). Trennen Sie daher Testprogramm und Baum in jeweils eigene Klassen.

Implementieren Sie eine Methode *evaluate()* zum Auswerten des Ausdrucks.

*Hinweis: Schreiben Sie sich (zum Testen) eine Methode (`void printAll()`) um den aufgebauten Syntaxbaum (vollständig geklammert) ausgeben zu können. So können Sie sehr einfach auf die Richtigkeit der Auswertungsreihenfolge schließen.*

*Hinweis: Verwenden Sie den rekursiven Abstieg zur Analyse des Ausdrucks (siehe Hinweise auf der Übungsseite)*