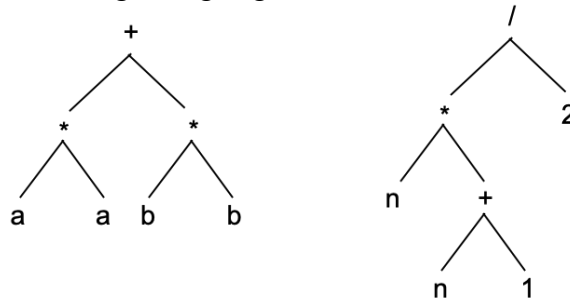
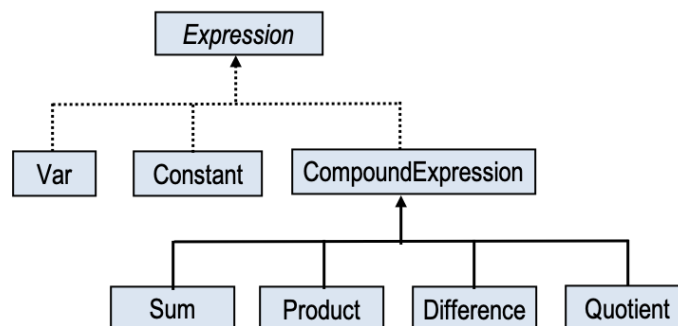


Aufgabenblatt 9 – Arithmetische Ausdrücke als Bäume

Wie in der Vorlesung besprochen lassen sich arithmetische Ausdrücke wie $a*a + b*b$ und $n*(n+1)/2$ als binäre Bäume darstellen. Die Klammerung wird weggelassen, weil durch die Baumstruktur die Auswertungsreihenfolge festgelegt ist.



Entwickeln Sie ein Kompositum (ähnlich zu Aufgabe 6) zur Darstellung arithmetischer Ausdrücke als Bäume. Ausdrücke bestehen aus Variablen (Strings), Konstanten (double) und Operatoren wie $+$, $-$, $*$ und $/$. Das Kompositum soll folgende Struktur haben:



Die Schnittstelle `Expression` ist bereits vorgegeben. Die Knoten eines Ausdrucksbaums haben also alle den Typ `Expression`, die Blätter entweder den Typ `Var` oder `Constant`, und die inneren Knoten haben den Typ `CompoundExpression`.

Die Schnittstelle `Expression` definiert eine Methode `eval`, um einen Ausdruck wie $e1 = a*a + b*b$ auswerten zu können. Da der Wert eines solchen Ausdrucks von den aktuellen Werten der Variablen abhängt, wird der Methode `eval` eine `Map` (`java.util.Map`) übergeben, die jeder Variablen einem `double`-Wert zuordnet. Diese Zuordnung wird Variablenbelegung genannt. Z.B. wird mit der Variablenbelegung $a = 3.0$ und $b = 4.0$ der Ausdruck $e1$ zu 25.0 ausgewertet. Lassen Sie die Methode `getVars` zunächst auskommentiert.

Implementieren Sie die oben gezeigten Klassen des Kompositums. Überschreiben Sie außerdem die `toString`-Methode geeignet. Vorgegeben ist ebenfalls eine Testklasse `ExpressionTest`, mit der Sie Ihr Kompositum testen sollen. Sie sehen in dieser Klasse außerdem das gewünschte `toString`-Format.

Freiwillige Bonus-Aufgabe

Kommentieren Sie nun die Methode `getVars` der Schnittstelle `Expression` ein und implementieren Sie sie in Ihren Klassen. Diese Methode soll alle Variablen eines Ausdrucks als `Set` (`java.util.Set`) zurückliefern. Z.B. ergibt sich beim Ausdruck $a*a + b*b$ die Variablenmenge $\{a, b\}$.

Kommentieren Sie ebenfalls die entsprechenden Test-Zeilen in der Klasse `ExpressionTest` ein, um diese zusätzliche Methode zu testen.