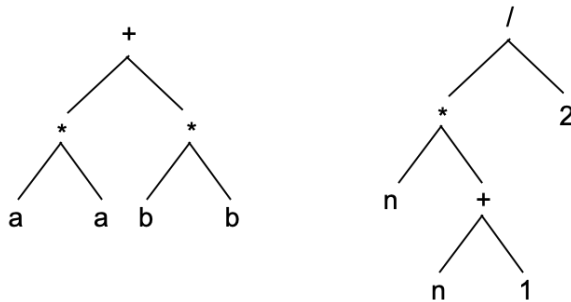
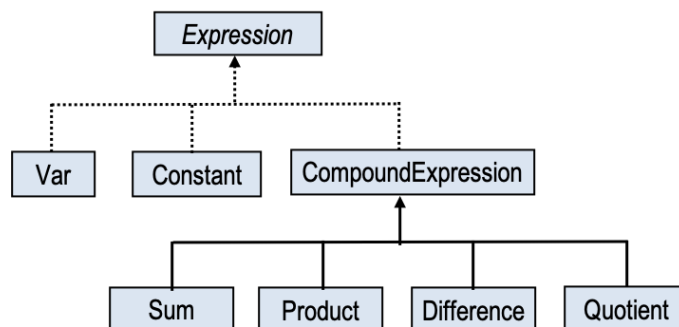


## Aufgabenblatt 7 – Arithmetische Ausdrücke als Bäume

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



Entwickeln Sie in Java einen Typ `Expression`, um arithmetische Ausdrücke definieren zu können. Ausdrücke bestehen aus Variablen (Strings), Konstanten (double) und Operatoren wie `+`, `-`, `*` und `/`. Schauen Sie sich das Entwurfsmuster Kompositum in Kap. 15 an. Das Muster legt folgende Typstruktur nahe:



Der Typ `Expression` bietet drei Methoden an:

- Eine Methode `eval`, um einen Ausdruck wie `e1 = a*a + b*b` auswerten zu können. Dazu wird eine `Map` (`java.util.Map`) übergeben, die jeder Variablen einen `double`-Wert zuordnet. Z.B. wird mit der Variablenbelegung `a = 3.0` und `b = 4.0` der Ausdruck `e1` zu `25.0` ausgewertet.
- Eine Methode `getVars`, die alle Variablen eines Ausdrucks als `Set` (`java.util.Set`) zurückliefert. Z.B. ergibt sich beim Ausdruck `a*a + b*b` die Variablenmenge `{a, b}`.
- Überschreiben Sie die Methode `toString`, so dass vollständig geklammerte Ausdrücke als Strings erzeugt werden. Bei `e1` bzw. `e2` ergeben sich die Strings `"((a * b)) + (b * b)"` bzw. `"((n * (n + 1)) / 2)"`.

Auf der Webseite finden Sie ein Hauptprogramm, das beispielhaft die Funktionalität des Typs `Expression` zeigt. Ein paar Zeilen müssen ergänzt werden. Sie dürfen das Hauptprogramm gerne an die Schnittstelle Ihrer Typen anpassen.