

## Modellierung und Programmierung 1 – Übungsserie 5

Abgabetermin: 31.01.2021, 22:00 Uhr

Abgabeformat: 1 ZIP-Datei

Max. Punkte: 18

### Rekursion und Interfaces

#### Operationen auf Bäumen (18 Punkte)

Als selbstständige Programmiererin bzw. selbstständiger Programmierer wurden Sie von einer Softwareschmiede angeheuert, um bei der Implementierung von Operationen auf Bäumen zu helfen. In der Theorie besteht ein Baum aus einer Menge von Knoten und einer Menge von Kanten zwischen den Knoten. Jeder Knoten speichert eine Zahl als Wert sowie eine Liste der Knoten, die nach ihm folgen – die Kinder des Knoten. Die in den Knoten gespeicherten Informationen sollen mit Hilfe verschiedener Operationen zu einem Ergebnis verbunden werden. Der Startpunkt eines Baums ist der sogenannte Wurzelknoten. Dieser ist der oberste Knoten und selbst nicht Kind eines anderen Knoten. Im folgenden Beispielbaum ist es der Knoten mit dem Wert 7,0.

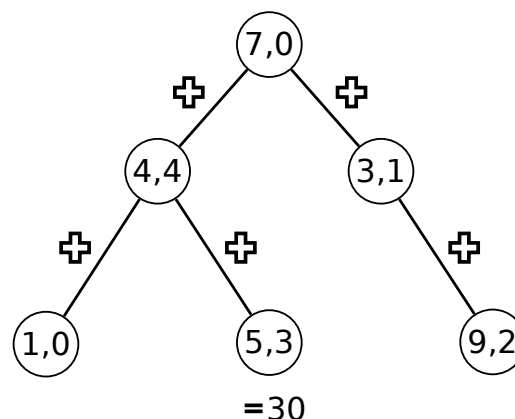


Abbildung 1: Ein Beispielbaum. Der Operator „Summe“ wurde beispielhaft auf den Baum angewendet (Plus an den Kanten) und ergibt 30.

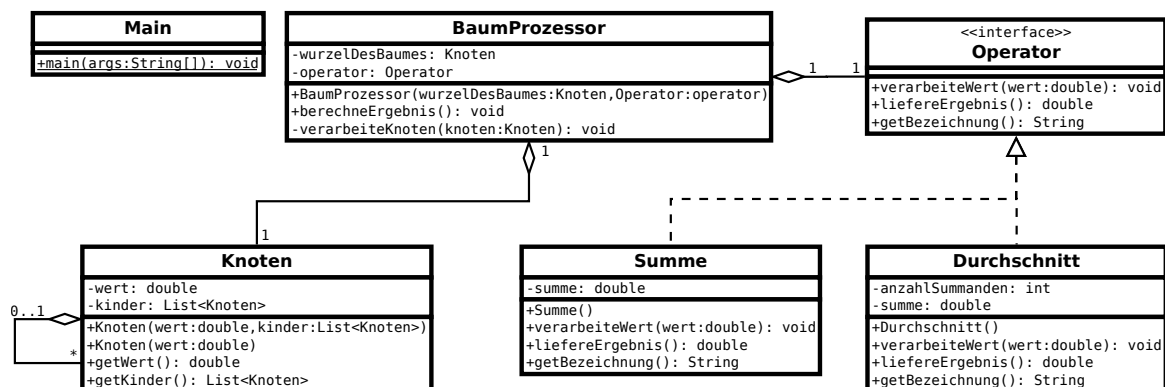


Abbildung 2: UML-Diagramm des Programmes.

Die Softwareschmiede liefert Ihnen dazu bereits die Implementierung der Klasse **Knoten** mit, welche aber nicht verändert werden darf!

---

1. (1 Punkt) Interface **Operator**

Es sollen verschiedene mathematische Operationen umgesetzt werden. Einem Operator werden nacheinander die einzelnen Werte des Baumes übergeben, welcher diesen Wert dann verarbeitet. Sobald alle Knoten mit dem Operator verarbeitet wurden, kann das Gesamtergebnis aus dem Operator ausgelesen werden. Schreiben Sie daher das Java-Interface **Operator**, welches drei Methoden vorgeben soll:

- **verarbeiteWert**: Eine Methode, welche eine Fließkommazahl annimmt und diese mit dem bisherigen Teilergebnis, welches in dem entsprechenden Operator gespeichert ist, verbindet.
- **liefereErgebnis**: Eine Methode, um das in einem Operator gespeicherte Ergebnis auszu-lesen.
- **getBezeichnung**: Eine Methode zur Ausgabe der Bezeichnung der durchgeführten Operati-on.

2. (3 Punkte) Klasse **Summe**

Als ein einfaches Beispiel wollen Sie zeigen, wie die Summe aller gespeicherten Werte im Baum ermittelt werden kann. Schreiben Sie dazu die Klasse **Summe**, welche das Interface **Operator** umsetzen soll.

- Die Klasse soll die Summe der Werte der bisher bearbeiteten Elemente als Fließkommazahl abspeichern.
- Das Attribut **summe** soll nur über die im Interface deklarierten Methoden verändert und ausgegeben werden.

3. (4 Punkte) Klasse **Durchschnitt**

Als ein weiteres Beispiel wollen Sie die Berechnung des Durchschnitts aller gespeicherten Werte im Baum mitliefern. Schreiben Sie dazu die Klasse **Durchschnitt**, welche das Interface **Operator** umsetzen soll.

- Dieser Operator soll die Summe der Werte der bisher bearbeiteten Elemente als Fließkom-mazahl sowie deren Anzahl abspeichern.
- Die Attribute **anzahlSummanden** und **summe** sollen nur über die im Interface deklarierten Methoden verändert und ausgegeben werden.
- Als Ergebnis soll der Durchschnitt der Werte aller bearbeiteten Elemente von der Methode **liefereErgebnis** zurückgegeben werden.

4. (6 Punkte) Klasse **BaumProzessor**

Um an alle im Baum gespeicherten Werte zu gelangen, entscheiden Sie sich für einen rekursiven Ansatz. Schreiben Sie dazu die Klasse **BaumProzessor**.

- Diese soll den obersten Knoten des Baumes sowie den Operator, welcher ausgeführt werden soll, speichern.
- Die Werte für die Attribute sollen nur bei der Initialisierung festgelegt werden können.

Für die rekursiven Methoden haben Sie sich zwei Pseudocode-Schnipsel aufgeschrieben, die Sie jetzt nur noch in Java-Code konvertieren müssen. Die zweite Methode soll dabei nur von der Klasse **BaumProzessor** genutzt werden können.

Methode 'berechneErgebnis':

Parameter: -

    rufe die Methode 'verarbeiteKnoten' mit der Wurzel des Baumes und dem auszufuehrenden  
    Operator auf

    gebe auf der Konsole aus, welche Operation ausgefuehrt wurde

    gebe auf der Konsole aus, was das Ergebnis dieser Operation ist

Methode 'verarbeiteKnoten':

Parameter: Knoten knoten

    verarbeite den Wert des Knoten k mit dem Attribut operator

    FUER JEDEN Knoten kind in der Liste der Kinder:

        rufe die Methode 'verarbeiteKnoten' mit dem Knoten kind und dem Operator op auf

---

5. (4 Punkte) Klasse `Main`

Um zu zeigen, wie Ihr Programm auf einem Beispielbaum funktioniert, schreiben Sie die Klasse `Main`.

- Erzeugen Sie in der `Main`-Methode den zu Beginn gezeigten Baum.
- Zeigen Sie die Funktionsweise Ihres `BaumProzessors` mithilfe dieses Baumes und den Operatoren „Summe“ und „Durchschnitt“ und geben Sie die Summe und den Durchschnitt der Werte im Baum aus.

**Hinweis:** Um die Liste der Kinder beim Instanziiieren eines Knoten zu initialisieren, bietet sich die statische Methode `Arrays.asList` an. Bei gegebenen `Knoten`-Referenzen `a`, `b` und `c`, erzeugt zum Beispiel der Aufruf `Arrays.asList(a,b,c)` eine Liste mit den `Knoten`-Referenzen.