

Übungsaufgabe BinaryTree

In dieser Aufgabe soll der Umgang mit Binärbäumen und Rekursionen geübt werden. Ähnlich wie in Listen erfolgt der Zugriff auf die Baumelemente über eine Verkettung. Im Fall des Binärbaums hat jedes Baumelement (`TreeNode`) maximal zwei Nachfolger vom gleichen Typ (`smaller`, `larger`). Ein Nachfolger mit dem Wert `null` bedeutet, dass der Baum an der Stelle nicht weitergeht. Der Binärbaum dieser Aufgabe soll eine sortierte Zahlenmenge enthalten. Für jeden Knoten im Baum soll gelten, dass seine Zahl grösser ist als die Zahlen aller seiner unter `smaller` zu findenden Nachfolger und kleiner als alle seiner unter `larger` zu findenden Nachfolger. Der Baum soll duplikatfrei sein, d.h. keine Zahl soll in dem Baum mehrfach vorkommen.

Für alle hier zu implementierenden Methoden bietet sich die Verwendung von Rekursionen an.

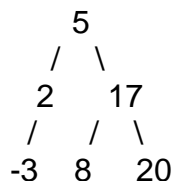
HINWEIS:

Für die Implementierung der Methoden kann es notwendig sein Hilfsmethoden zu verwenden. Sie dürfen für die Lösung dieser Aufgabe beliebige Hilfsmethoden erstellen.

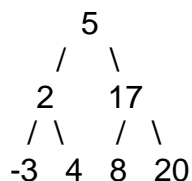
Aufgabe 1 insert

Mit dieser Methode soll ein neuer Knoten mit der durch den Parameter `number` gegebenen Zahl in den Baum eingefügt werden. Die vorhandene Struktur des Baumes soll dabei nicht geändert werden und die neue Zahl als Blatt (Knoten ohne Nachfolger) im Baum zu finden sein. Ist die Zahl bereits im Baum vorhanden, soll nichts geschehen.

Beispiel:



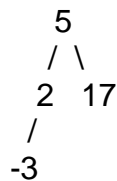
Aufruf von `Tree.insert(4)`



Aufgabe 2 maxDepth

Diese Methode soll die maximale Tiefe eines Baumes zurückgeben. Die maximale Tiefe ist die Länge des weitesten Weges den man von der Wurzel des Baumes zu einem Blatt zurücklegen kann.

Beispiel:

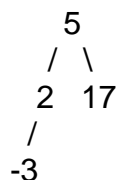


Die maximale Tiefe wird in diesem Baum entlang des Pfades 5 → 2 → -3 erreicht. Der Wert beträgt 3. Falls der Baum kein Element enthält und damit leer ist, soll die zurückgegebene Tiefe 0 sein.

Aufgabe 3 sumOfElements

Mit dieser Methode soll die Summe über alle Elemente des Baumes zurückgegeben werden. Ist der Baum leer, soll 0 zurückgegeben werden.

Beispiel:



Die Summe über diesen Baum wäre $5 + 2 + (-3) + 17 = 21$

Aufgabe 4 deleteNumber

Mit dieser Methode soll die durch `number` gegebene Zahl aus dem Baum entfernt werden. Ist diese Zahl nicht im Baum vorhanden, soll nichts geschehen. Dabei ist zu beachten, dass der Baum nach dem Entfernen eines Elements wieder der oben geforderten Sortierung entspricht.