Aufgaben zur Veranstaltung
Programmierung mit Java, WS 2024/2025

J. Dietel, J. Fuchs FH Aachen, Campus Jülich; IT Center, RWTH Aachen

Präsenzaufgaben 12 08./09.01.2025

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

1 Generische Klasse: BinaryTreeNode

a) In dieser Aufgabe sollen Sie eine generische Klasse implementieren, die einen Teil eines Binärbaums darstellt. Legen Sie dazu eine generische Klasse mit dem Namen `BinaryTreeNode` an. Die Klasse hat drei Attribute: Einen Wert, der dem Generic entspricht, eine Referenz auf einen linken `BinaryTreeNode` und eine Referenz auf einen rechten `BinaryTreeNode`. Um den Aufwand zu reduzieren, soll das Attribut, das diesem Wert entspricht, `public` sein.

Hinweis: Die `BinaryTreeNodes` dürfen unterschiedliche Typen haben und müssen nicht alle das gleiche Generic verwenden.

b) Neben einem Konstruktor, der den Wert erhält, sollen folgende Methoden implementiert werden:

c) `getLeft`, `setLeft`, `getRight`, `setRight`: Getter und Setter für die Attribute, die Referenzen darstellen.

d) `toString`: Soll das Attribut, das dem Wert entspricht, als String liefern.

e) `printSubtree`: Soll rekursiv alle Werte des Knotens und der darunter liegenden Knoten ausgeben. Dabei sollen erst alle Werte im linken Subbaum ausgegeben werden, dann der Wert des aktuellen Knotens und danach alle Werte im rechten Subbaum. Diese Vorgehensweise heißt Inorder-Durchlauf.

Hinweis: Der linke Subbaum eines `BinaryTreeNodes` ist der linke `BinaryTreeNode` zusammen mit allen von hier aus erreichbaren `BinaryTreeNodes`.

f) `findParent`: Bekommt einen `BinaryTreeNode findMyParent` und sucht im rechten und linken Subbaum des Knotens (`this`) nach einem `BinaryTreeNode x`, so dass eine der zwei Referenzen von `x` dem Knoten `findMyParent` entspricht. Wenn der Knoten `x` existiert, soll dieser zurückgegeben werden, sonst `null`.

g) Schreiben Sie eine `main`, um `toString` und `printSubtree` zu testen. (Sie dürfen auch mehr testen.)

h) **Bonusfrage**: Was muss geändert werden, um zu gewährleisten, dass alle Knoten, die in einer Verbindung stehen (die Teil des gleichen Baums sind), das gleiche Generic verwenden?

2 Record: Children

a) Fügen Sie der Klasse einen lokalen Record namens `Children` hinzu, welcher aus zwei `BinaryTreeNodes` besteht mit sinnvollen Namen.

b) Wir möchten der Klasse `BinaryTreeNode` die Methoden `getChildren` und `setChildren` hinzufügen, mit denen wir direkt beide Attribute, die Referenzen entsprechen, erhalten und setzen können.

3 JUnit-Test:BinaryTreeNodeTest

Schreiben Sie einen JUnit-Test für Ihre Klasse. Implementieren Sie folgende Tests:

a) `testBinaryTreeNode`: Testet den Konstruktor.

b) `testSetGetLeft`: Testet `getLeft`, `setLeft`

c) `testSetGetRight`: Testet `getRight`, `setRight`

d) `testFindParent`: Testet `findParent` (Testen Sie auch, ob `null` zurückgegeben wird.)

e) `testSetGetChildren`: Testet `getChildren`, `setChildren`