Einzelprüfung „Theoretische Informatik / Algorithmen (vertieft)“

Einzelprüfungsnummer 66115 / 2014 / Frühjahr

Thema 1 / Aufgabe 2

(Binäre Bäume)

Stichwörter: Binärbaum

Implementieren Sie in einer objekt-orientierten Sprache Ihrer Wahl eine Klasse namens `BinBaum`, deren Instanzen binäre Bäume mit ganzzahligen Datenknoten darstellen, nach folgender Spezifikation:

(a) Beginnen Sie zunächst mit der Datenstruktur selbst:

- Mit Hilfe des Konstruktors soll ein neuer Baum erstellt werden, der aus einem einzelnen Knoten besteht, in dem der dem Konstruktor als Parameter übergebene Wert (ganzzahlig, in Java z.B. `int`) gespeichert ist.

Lösungsvorschlag

```
class Knoten {
  int value;

  Knoten left;
  Knoten right;

  public Knoten(int value) {
    this.value = value;
  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66115/jahr_2014/fruehjahr/BinBaum.java`

- Die Methoden `setLeft(int value)` bzw. `setRight(int value)` sollen den linken bzw. rechten Teilbaum des aktuellen Knotens durch einen jeweils neuen Teilbaum ersetzen, der seinerseits aus einem einzelnen Knoten besteht, in dem der übergebene Wert `value` gespeichert ist. Sie haben keinen Rückgabewert.

Lösungsvorschlag

```
  public void setLeft(Knoten left) {
    this.left = left;
  }

  public void setRight(Knoten right) {
    this.right = right;
  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66115/jahr_2014/fruehjahr/BinBaum.java`

- Die Methoden `getLeft()` bzw. `getRight()` sollen den linken bzw. rechten Teilbaum zurückgeben (bzw. `null`, wenn keiner vorhanden ist)

Lösungsvorschlag

```
public Knoten getLeft() {
  return left;
}

public Knoten getRight() {
  return right;
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66115/jahr_2014/fruehjahr/BinBaum.java`

- Die Methode `int getValue()` soll den Wert zurückgeben, der im aktuellen Wurzelknoten gespeichert ist.

Lösungsvorschlag

```
public int getValue() {
  return value;
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66115/jahr_2014/fruehjahr/BinBaum.java`

(b) Implementieren Sie nun die Methoden `preOrder()` bzw. `postOrder()`. Sie sollen die Knoten des Baumes mit Tiefensuche traversieren, die Werte dabei in pre-order bzw. post-order Reihenfolge in eine Liste (z.B. `List<Integer>`) speichern und diese Ergebnisliste zurückgeben. Die Tiefensuche soll dabei zuerst in den linken und dann in den rechten Teilbaum absteigen.

Lösungsvorschlag

```
void preOrder(Knoten knoten, List<Integer> list) {
  if (knoten != null) {
    list.add(knoten.getValue());
    preOrder(knoten.getLeft(), list);
    preOrder(knoten.getRight(), list);
  }
}

List<Integer> preOrder() {
  List<Integer> list = new ArrayList<>();
  preOrder(head, list);
  return list;
}

void postOrder(Knoten knoten, List<Integer> list) {
  if (knoten != null) {
    postOrder(knoten.getLeft(), list);
    postOrder(knoten.getRight(), list);
    list.add(knoten.getValue());
  }
}

List<Integer> postOrder() {
  List<Integer> list = new ArrayList<>();
  postOrder(head, list);
  return list;
```

```
  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66115/jahr_2014/fruehjahr/BinBaum.java`

(c) Ergänzen Sie schließlich die Methode `isSearchTree()`. Sie soll überprüfen, ob der Binärbaum die Eigenschaften eines binären Suchbaums erfüllt. Beachten Sie, dass die Laufzeit-Komplexität Ihrer Implementierung linear zur Anzahl der Knoten im Baum sein muss.

Lösungsvorschlag

```
boolean isSearchTree(Knoten knoten) {
  if (knoten == null) {
    return true;
  }

  if (knoten.getLeft() != null && knoten.getValue() <
  ↪  knoten.getLeft().getValue()) {
    return false;
  }

  if (knoten.getRight() != null && knoten.getValue() >
  ↪  knoten.getRight().getValue()) {
    return false;
  }

  return isSearchTree(knoten.getLeft()) && isSearchTree(knoten.getRight());
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66115/jahr_2014/fruehjahr/BinBaum.java`

Additum

```
import java.util.ArrayList;
import java.util.List;

public class BinBaum {

  Knoten head;

  public BinBaum(Knoten head) {
    this.head = head;
  }

  class Knoten {
    int value;

    Knoten left;
    Knoten right;

    public Knoten(int value) {
      this.value = value;
    }
```

```
  public void setLeft(Knoten left) {
    this.left = left;
  }

  public void setRight(Knoten right) {
    this.right = right;
  }

  public Knoten getLeft() {
    return left;
  }

  public Knoten getRight() {
    return right;
  }

  public int getValue() {
    return value;
  }
}

void preOrder(Knoten knoten, List<Integer> list) {
  if (knoten != null) {
    list.add(knoten.getValue());
    preOrder(knoten.getLeft(), list);
    preOrder(knoten.getRight(), list);
  }
}

List<Integer> preOrder() {
  List<Integer> list = new ArrayList<>();
  preOrder(head, list);
  return list;
}

void postOrder(Knoten knoten, List<Integer> list) {
  if (knoten != null) {
    postOrder(knoten.getLeft(), list);
    postOrder(knoten.getRight(), list);
    list.add(knoten.getValue());
  }
}

List<Integer> postOrder() {
  List<Integer> list = new ArrayList<>();
  postOrder(head, list);
  return list;
}

boolean isSearchTree(Knoten knoten) {
  if (knoten == null) {
    return true;
  }
```

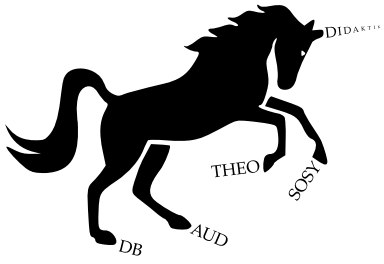
```
    if (knoten.getLeft() != null && knoten.getValue() < knoten.getLeft().getValue()) {
      return false;
    }

    if (knoten.getRight() != null && knoten.getValue() > knoten.getRight().getValue()) {
      return false;
    }

    return isSearchTree(knoten.getLeft()) && isSearchTree(knoten.getRight());
  }

  boolean isSearchTree() {
    return isSearchTree(head);
  }
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66115/jahr_2014/fruehjahr/BinBaum.java`



Die Bschlangaul-Sammlung
Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net.Der TEX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: `https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/66115/2014/03/Thema-1/Aufgabe-2.tex`