

## Übung 3: Binärer Suchbaum

Abgabetermin: 11.04.2016

Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

Gruppe:  G1 Di 10:15-11:00  G2 Di 11:00-11:45  G3 Di 12:45-13:30

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	24		Java-Programm Testfälle und Ergebnisse	Java-Programm (*.java Files)		

### Aufgabe 1: Binärer Suchbaum für sortierte Menge von Zahlen (24 Punkte)

Implementieren Sie eine sortierte Menge für Zahlen als binären Suchbaum in der Klasse *BinaryTreeSet*. Die Schnittstelle ist durch die abstrakte Klasse *Set* gegeben (Details in JavaDoc<sup>1</sup>).

```
package at.jku.pi.set;

public abstract class Set {
    public abstract void add(int value);
    public abstract boolean contains(int value);
    public abstract boolean remove(int value);
    public abstract int size();
    public abstract IntIterator iterator();
    public abstract Set union(Set other);
    public abstract Set intersect(Set other);
    public abstract Set diff(Set other);
    public abstract Set subtract(Set other);
}

public abstract class IntIterator {
    public abstract boolean hasNext();
    public abstract int next();
}
```

Implementieren Sie die Klasse *BinaryTreeSet* im Paket *at.jku.pi.set.tree.ue03*:

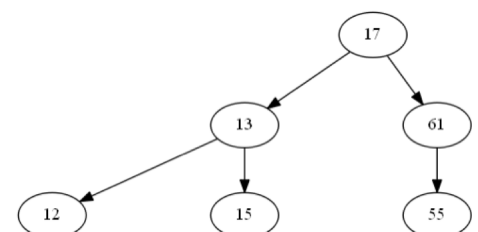
```
package at.jku.students;

class BinaryTreeSet extends Set {
    private TreeNode root;
    public void add(int value) { TODO }
    // ...
    public void union(Set set) { TODO }
    // ...
    public String makeDot() {
        return DotMaker.makeDotForBinaryTree(root);
    }
}
```

Die Methoden **add**, **contains**, **remove** und **size** sind wie in der Vorlesung besprochen zu implementieren. Die Methoden **union**, **intersect**, **diff**, **subtract** sind Mengenoperationen (siehe Vorlesungsstoff zu *Sets*). Diese Methoden führen Mengenoperationen mit einem zweiten Set durch und geben ein drittes Set als Ergebnis zurück, ohne die anderen Sets zu verändern. Verwenden Sie für die Implementierung dieser Methoden den *IntIterator* den jedes *Set* zur Verfügung stellt (die Implementierung ist vorgegeben, *IntIterator iterator()*; Details siehe JavaDoc<sup>1</sup>).

#### Implementierungshinweise:

- Verwenden Sie die Vorgabedateien *pi2\_ue03.jar* und die zugehörige Java-Dokumentation von der LVA-Website<sup>1</sup>.
- Definieren Sie für alle Klassen, Methoden und Felder die geeignete Sichtbarkeit (*private*, *protected*, *package*, *public*).
- Implementieren Sie die Methoden **add**, **contains** und **size** mit rekursiven Algorithmen.
- Testen Sie Ihre Implementierung mit der Vorgabedatei *BinaryTreeSetTest.java*. Diese Tests erstellen GraphViz Files. Verwenden Sie diesen Output um Ihre Implementierung zu überprüfen.



test.d

Abzugeben ist: Java-Programm, Testergebnisse

<sup>1</sup> [http://ssw.jku.at/Teaching/Lectures/PI2/2016/Download/ue03\\_doc/](http://ssw.jku.at/Teaching/Lectures/PI2/2016/Download/ue03_doc/)