Übung 4: Set

Abgabetermin: 17.04.2012

Name: _____ **Matrikelnummer:** _____

Gruppe: □ **G1** Di 10:15 □ **G2** Di 12:45

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	24	□	Java-Programm Testfälle und Ergebnisse	Java-Programm	□	

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

Implementieren Sie eine sortierte Menge für Zeichen in der Klasse *BinaryTreeSet.* Erlaubt sind alle Zeichen. Die Schnittstelle ist durch die abstrakte Klasse *Set* gegeben: *add* fügt ein Zeichen ein, *get* prüft ob ein Zeichen enthalten ist, *remove* entfernt ein Zeichen, *size* liefert die Anzahl der Zeichen und *iterator* liefert einen Iterator (mit Schnittstelle *Iterator*) mit dem die Menge sortiert durchlaufen werden kann. Die Methode *union* liefert die Vereinigungsmenge zweier Mengen, *intersect* liefert die Schnittmenge zweier Mengen, *diff* liefert die Mengendifferenz von *this* minus *set*, und *subset* liefert eine Teilmenge mit den Zeichen die im gegebenen Intervall liegen.

```
package at.jku.ssw;

public abstract class Set {
  public abstract void add(char value);
  public abstract boolean get(char value);
  public abstract boolean remove(char value);
  public abstract int size();
  public abstract Iterator iterator();
  public abstract Set union(Set other);
  public abstract Set intersect(Set other);
  public abstract Set diff(Set other);
  public abstract Set subset(char from, char to);
}
```

```
public abstract class Iterator {
  public abstract boolean hasNext();
  public abstract char next();
}
```

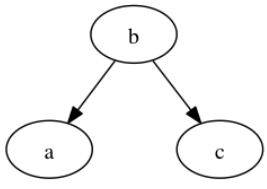
Implementieren Sie die Klassen *BinaryTreeSet* und *BinaryTreeIterator* im Paket *at.jku.students*.

```
package at.jku.students;
public class BinaryTreeSet extends Set {
  TreeNode root;
  public String makeDot() {
    return DotMaker.makeDotForBinaryTree(root);
  }
  ...
}
public class BinaryTreeIterator extends Iterator {
  ...
}
```

```
Set s = new BinaryTreeSet();
s.add('b'); s.add('c'); s.add('c');
s.add('b'); s.add('a');
Iterator it = s.iterator();
while (it.hasNext()) {
  Out.print(" " + it.next());
} // Ausgabe: a b c
Out.open("test.dot");
Out.print(
  ((BinaryTreeSet) s).makeDot());
Out.close();
```

Implementierungshinweise:

- Verwenden Sie in der Klasse *BinaryTreeSet* einen binären Suchbaum um die Elemente zu speichern. Verwenden Sie die Klasse *at.jku.ssw.TreeNode* für die Knoten das Binärbaums.
- Implementieren Sie die Methoden *add*, *get* und *size* mit rekursiven Algorithmen.
- Verwenden Sie die Vorgabeklasse *DotMaker* um Bilder der in Ihren Testfällen erzeugten Binärbäume zu erstellen.



test.dot

Abzugeben ist: Java-Programm, Testfälle