Übungsserie Nr. 3

1.) Sortieren von Strings

(a) siehe SortString.java

(b) Sortierung abhängig von Gross-/Kleinschreibung, um eine unabhängige Sortierung zu erhalten ist die Methode compareToIgnoreCase anstelle von compareTo zu verwenden.

2.) Objekte, Strings und Referenzen

(a)

```
Java ist auch eine
Java ist auch eine Insel
```

(b)

Markierung 1:
(1, StringObject, '')
(2, String, 'Java ')
(3, StringBuffer, 'Java ')

(ref, 1)
(mystring, 2)
(mybuffer, 3)

Markierung 2:
(1, StringObject, '')
(2, String, 'Java ')
(3, StringBuffer, 'Java ist auch eine ')
(4, String, 'Java ist auch eine ')

(ref, 1)
(mystring, 4)
(mybuffer, 3)

Markierung 3:
(1, StringObject, '')
(2, String, 'Java ')
(3, StringBuffer, 'Java ist auch eine')
(4, String, 'Java ist auch eine ')
(5, String, 'Java ist auch eine ')
(6, String, 'String, 'Insel')
(7, String, 'Java ist auch eine Insel')

(ref, 1)
(mystring, 4)
(mybuffer, 3)
(s1, 7)
(s2, 6)

Markierung 3:
(1, StringObject, '')
(2, String, 'Java ')
(3, StringBuffer, 'Java ist auch eine Insel')
(4, String, 'Java ist auch eine ')
(5, String, 'Java ist auch eine ')
(6, String, 'String, 'Insel')
(7, String, 'Java ist auch eine Insel')

(ref, 1)
(mystring, 4)
(mybuffer, 3)

Zusätzlich wäre auch noch eine Referenz auf die Argumente (args) vorhanden, da diese im Programm nicht angesprochen werden, wurde hier darauf verzichtet.

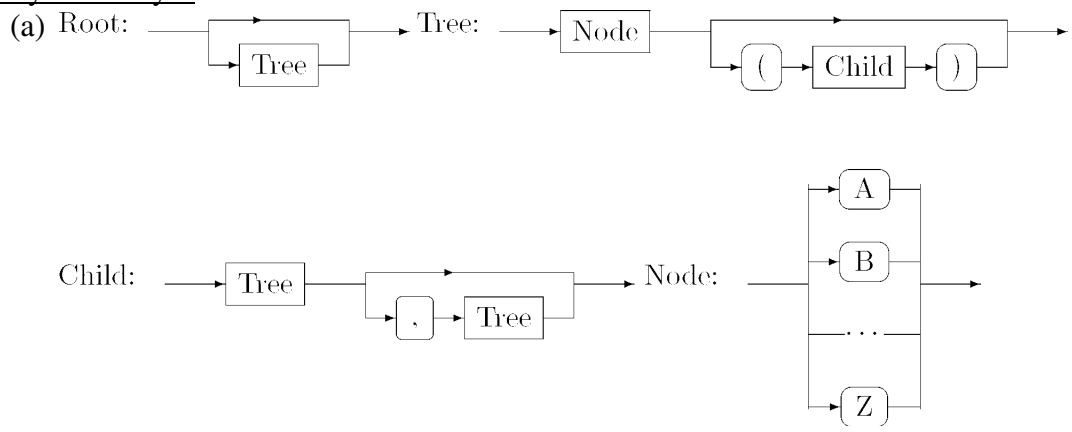
3.) Syntaxdiagramm

(a) X_1 (nicht erzeugbar)
$(X_1 OR \sim X_2)$ (erzeugbar)
$\sim (X_1 OR \sim X_2)$ (nicht erzeugbar)
$(X_2 OR X_2)$ (erzeugbar)

(b) $(\sim X_1 AND X_2)$ (nicht erzeugbar)
$(X_1 OR X_1) AND (\sim X_1)$ (erzeugbar)

4.) Syntaxanalyse

(a)



(b) siehe LKD.java