

Übungsblatt 15: Software-Entwicklung 1 (WS 2017/18)

Ausgabe: 13.02.18

Abgabe: keine

Aufgabe 1 Terminierung

Geben Sie für die folgenden Prozeduren jeweils eine sinnvolle Vorbedingung an, so dass die Prozedur immer ohne Fehler terminiert, wenn die Vorbedingung erfüllt ist.

Beweisen Sie dann die Terminierung der Prozedur für alle erlaubten Werte. Verwenden Sie dazu das Verfahren aus der Vorlesung.

a) Binärsuche in Java:

```
1 public class DataSet {
2     int key;
3     String data;
4
5     DataSet (int key, String data) {
6         this.key = key;
7         this.data = data;
8     }
9     /* Für bessere Übersichtlichkeit im Skript verzichten wir hier
10    auf Getter/Setter-Methoden */
11 }
12
13 public static DataSet search (int x, DataSet[] f, int lo, int hi) {
14     // Element nicht gefunden
15     if (hi <= lo) {
16         return null;
17     }
18     int mid = lo + (hi-lo) /2;
19     if (x < f[mid].key) {
20         // Suche im Intervall [lo, mid)
21         return search (x, f, lo, mid);
22     }
23     if (x > f[mid].key) {
24         // Suche im Intervall [mid+1, hi)
25         return search (x, f, mid+1, hi);
26     }
27     // Element gefunden
28     return f[mid];
29 }
```

b) Hier ist `TreeNode` wie in der Vorlesung definiert. Sie können davon ausgehen, dass `totalSize(TreeNode t)` für die Anzahl der Knoten im Baum und `height(TreeNode t)` für die Höhe des Baums definiert sind.

```
public TreeNode mirror(TreeNode t) {
    if (t == null) {
        return null;
    }
    return new TreeNode(t.getMark(), mirror(t.getRight()), mirror(t.getLeft()));
}
```