Häufige Fehler (1)

```
// Es wird true oder false zurück gegeben je nachdem
// ob die Tahl gefunden wurde
public boolean get(int val) {
  Node p = head;
  if(p == null) {
    return true;
  }
  if(find(val) == null) {
    return true;
  } else {
    return false;
  }
}
```

Häufige Fehler (2)

```
void set(int val){
  Node p = head;

  if(get(val) == false){
    Node q = new Node(val);
    q.next = head;
    head = q;
  }
}
```

Häufige Fehler (3)

```
public void set(int val) {
  boolean truth = false;
  Node p = head;
  while (p != null) {
    if (p.val==val) {
      truth = true;
    }
    p = p.next;
  }
  if (truth == false) {
    insert(val);
  }
}

public void insert(int val) {
  Node p = head, prev = null;
  Node q = new Node(val);
  while(p != null && p.val < val) {
    prev = p;
    p = p.next;
}
```

```
public void set(int val) {
  if(!get(val))
    insert(val);
}
```

Häufige Fehler (4)

```
public
```

java.lang

Class Object

protected Object | clone()
Creates and returns a copy of this object.

```
Set clone() {
  LinkedListSet result = new LinkedListSet();
  for(Node p=head; p != null; p=p.next) {
    Node q = new Node(p.val);
    q.next = result.head;
    result.head = q;
  }
  return result;
}
```

```
LinkedListSet.java:42: clone() in LinkedListSet cannot
 implement clone() in Set; attempting to assign weaker
 access privileges; was public
        Set clone() {
            ^
1 error
```

Selbsttest – Information hiding

```
public class Tree {
  Node head;
  public Tree() { … }
  public void insert(String name) { … }
  public boolean search(String name) { … }
  public boolean remove(String name) { … }
}
public class Node {
  String name;
  Node left, right;
  Node(String name) { … }
}
```

-Gibt es Verletzungen bzgl. Information hiding? Wenn ja, wo?
-Können einige Variablen/Methoden weniger restriktiv deklariert werden, ohne das Information hiding aufzuweichen?

Selbsttest – Lineare Liste vs. Binärer Suchbaum

Es liegen eine sortierte einfach verkettete lineare Liste und ein vollständiger binärer Suchbaum vor. Beide Datenstrukturen haben dieselbe Anzahl von Stringelementen (n Stück) gespeichert.

- Wie viele Stringvergleiche sind bei beiden Datenstrukturen maximal, minimal und durchschnittlich nötig um ein bestimmtes Element zu finden?
- Angenommen der binäre Suchbaum ist stark (d.h. maximal) entartet, auf welche Anzahl (minimal, maximal, durchschnittlich) von Vergleichen ändert sich die Suche?
- Welche Tiefe hat der Suchbaum im entarteten Fall?

Selbsttest – Einfügen in binären Suchbaum

Ein binärer Suchbaum für Integer-Werte soll aufgebaut werden. Zeichnen Sie jeweils die Bäume auf die sich zwischen den einzelnen Schritten ergeben (Die Suchbaum-Ordnung ist: Elemente des linken Unterbaums < Wurzelelement < Elemente des rechten Unterbaums)

- Einfügen von 6, 12, 4, 13, 9, 7, 8, 5, 3, 1, 20
- Welche max. Tiefe hat der Baum?
- Fügen Sie das Element 7 ein. Geht das? Welche Möglichkeiten gibt es dann?

Selbsttest – Binärer Suchbaum

Welche der folgenden Behauptungen stimmen (nicht)? Und warum?

- Der Grad eines Knoten ist die Anzahl der Knoten, die auf ihn verweisen.
- Die Anzahl der Blätter eines vollständigen Baumes wächst exponentiell mit dessen Höhe.
- Die Höhe eines beliebigen Baumes mit n Knoten ist ungefähr ld(n).
- In einem Binärbaum hat jeder Knoten Grad 2.
- Binäre Suchbäume sind entweder aufsteigend oder absteigend sortiert.
- Von der Wurzel zu einem beliebigen Blatt gibt es immer genau einen Pfad.