

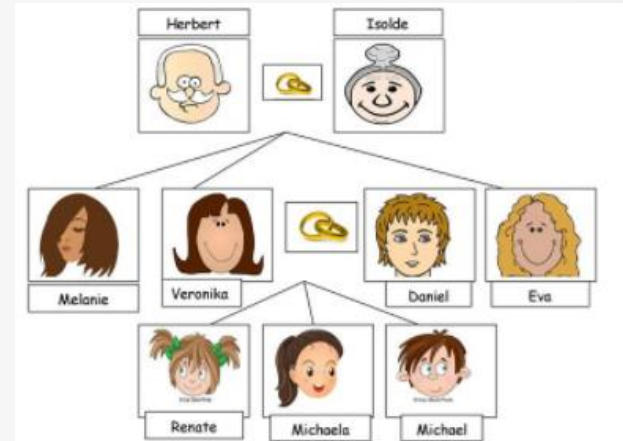
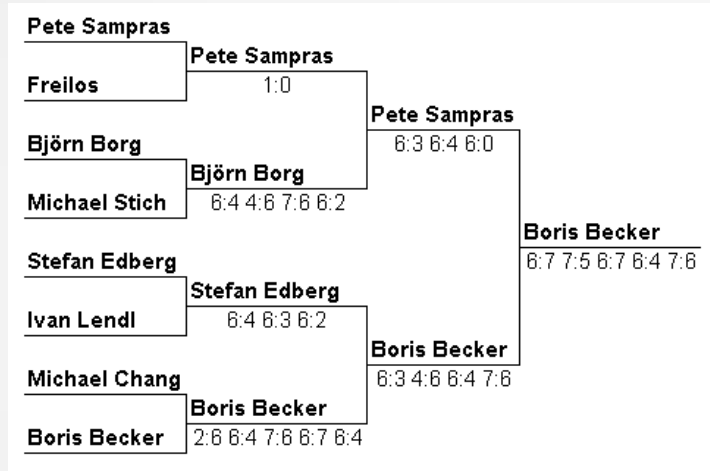
Binäre Bäume



Definition

- ✓ **Bäume** (engl. trees) Erweiterung von Listen: In Bäumen hat jeder Knoten mehrere Nachfolger
- ✓ Nachfolgerknoten heißt **Kind** (engl. child)
- ✓ Vorgängerknoten heißt **Eltern** (engl. parent)
- ✓ Knoten ohne Kinder heißt **Blatt** (engl. leaf)
- ✓ Knoten mit Kindern heißt **innerer Knoten** (engl. internal node/vertex)
- ✓ Baum enthält nur einen Knoten ohne Eltern; Knoten heißt **Wurzel** (engl. root)
- ✓ **Tiefe** eines Knotens ist die Länge des Pfades bis zur Wurzel
- ✓ **Höhe** eines Baumes ist die maximale Tiefe eines seiner Knoten

Anwendungen

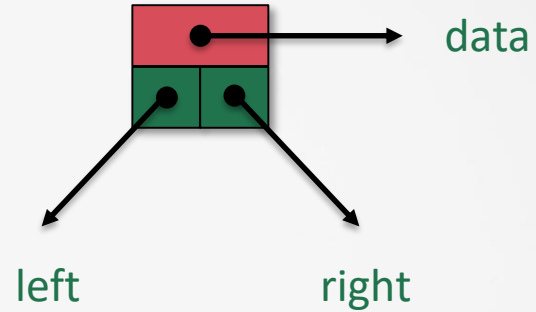


K.O.-System bei Turnieren

Stammbaum

Implementierung eines Knotens

```
class Node {  
    Object data;  
    Node left;  
    Node right;  
  
    Node(Object obj) {  
        data=obj;  
        left=right=null;  
    }  
}
```

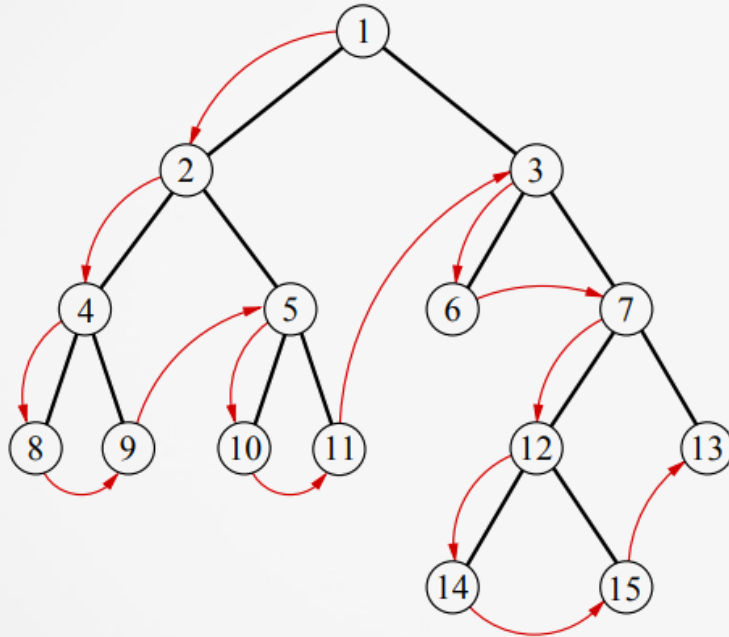


Implementierung eines Knotens II

```
public void setLeft(Node n) { }  
public void setRight(Node n) { }  
public Node getLeft() { }  
public Node getRight() { }  
public void setData(int d) { }  
public int getData() { }
```

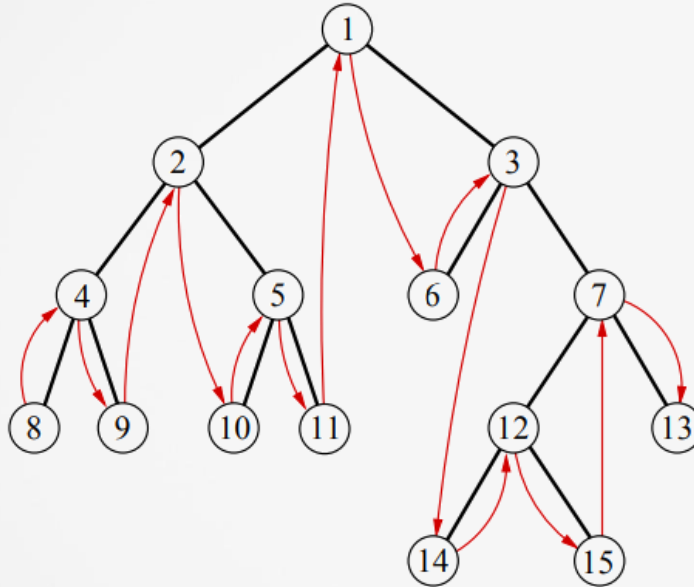
Noch zu implementieren in der Klasse Node!

Rekursive Präorder-Traversierung von binären Bäumen



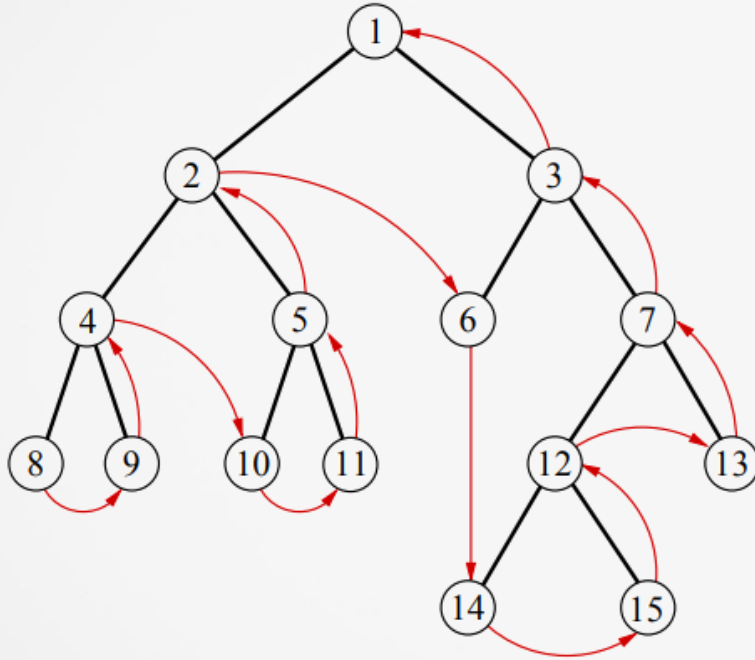
```
p.action(root);  
leftTree.preorder(p);  
rightTree.preorder(p);
```

Rekursive Inorder-Traversierung von binären Bäumen



```
leftTree.inorder(p);  
p.action(root);  
rightTree.inorder(p);
```


Rekursive Postorder-Traversierung von binären Bäumen



```
leftTree.postorder(p);  
rightTree.postorder(p);  
p.action(root);
```