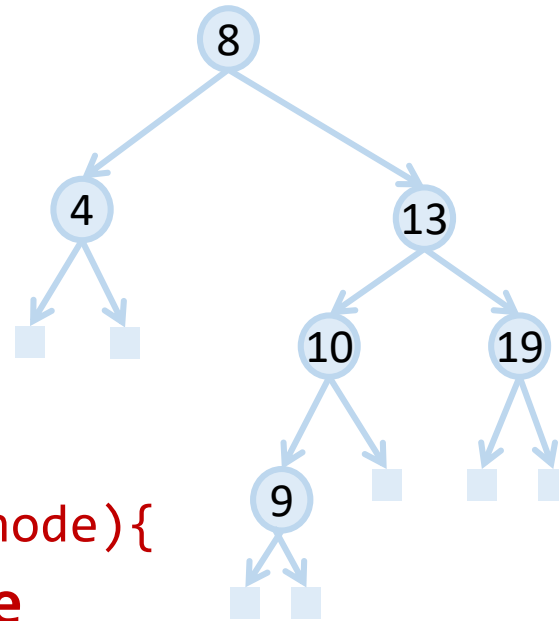


Binäres Banales

```
Node{  
    Node left, right;  
    int key;  
}
```

```
int void PrintKeys(Node node){  
    // insert code here  
}
```



Die Elemente eines Suchbaumes sollen mit **PrintKeys(root)** absteigend ausgegeben werden, also hier:
19, 13, 10, 9, 8, 4

Wie müssen die Statements angeordnet werden?

- (1) a,b,c,d
- (2) b,a,d,c
- (3) d,c,a,b
- (4) c,d,b,a
- (5) c,b,a,d
- (6) c,b,d,a
- (7) d,b,a,c
- (8) Geht nicht so einfach

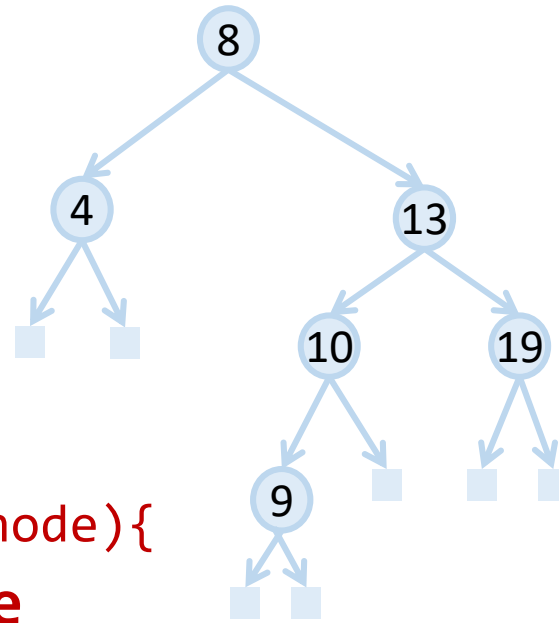
Statements:

- (a) PrintKeys(node.left);
- (b) PrintKeys(node.right);
- (c) if (node == null) return;
- (d) System.out.println(node.key);

Binäres Banales

```
Node{  
    Node left, right;  
    int key;  
}
```

```
int void PrintKeys(Node node){  
    // insert code here  
}
```



Wenn node == null dann sind Zeilen a,b,d ungültig.

Also: Zuerst (c)

Sicher rechts vor links, also: (b) vor (a)

Mitte nach rechts, also (b) vor (d).

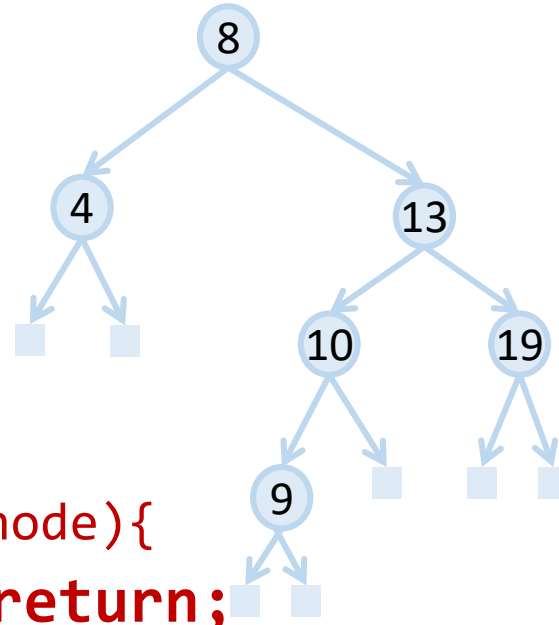
Statements:

```
(a) PrintKeys(node.left);  
(b) PrintKeys(node.right);  
(c) if (node == null) return;  
(d) System.out.println(node.key);
```

Binäres Banales

```
Node{  
    Node left, right;  
    int key;  
}
```

```
int void PrintKeys(Node node){  
    if (node == null) return;  
    PrintKeys(node.right);  
    System.out.println(node.key);  
    PrintKeys(node.left);  
}
```



Wenn node == null dann sind Zeilen a,b,d ungültig.

Also: Zuerst (c)

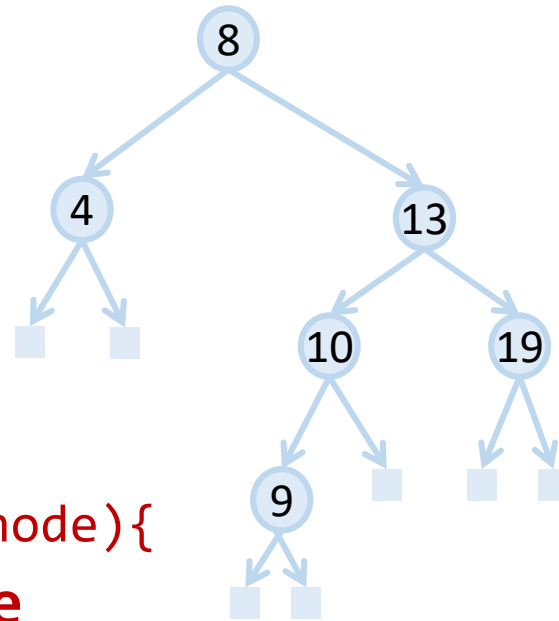
Sicher rechts vor links, also: (b) vor (a)

Mitte nach rechts, also (b) vor (d).

Binäres Banales

```
Node{  
    Node left, right;  
    int key;  
}
```

```
int void PrintKeys(Node node){  
    // insert code here  
}
```



Die Elemente eines Suchbaumes sollen absteigend ausgegeben werden, also hier:

19, 13, 10, 9, 8, 4

Wie müssen die Statements angeordnet werden?

- (1) a,b,c,d
- (2) b,a,d,c
- (3) d,c,a,b
- (4) c,d,b,a
- (5) c,b,a,d
- (6) c,b,d,a**
- (7) d,b,a,c
- (8) Geht nicht so einfach

Statements:

- (a) PrintKeys(node.left);
- (b) PrintKeys(node.right);
- (c) if (node == null) return;
- (d) System.out.println(node.key);