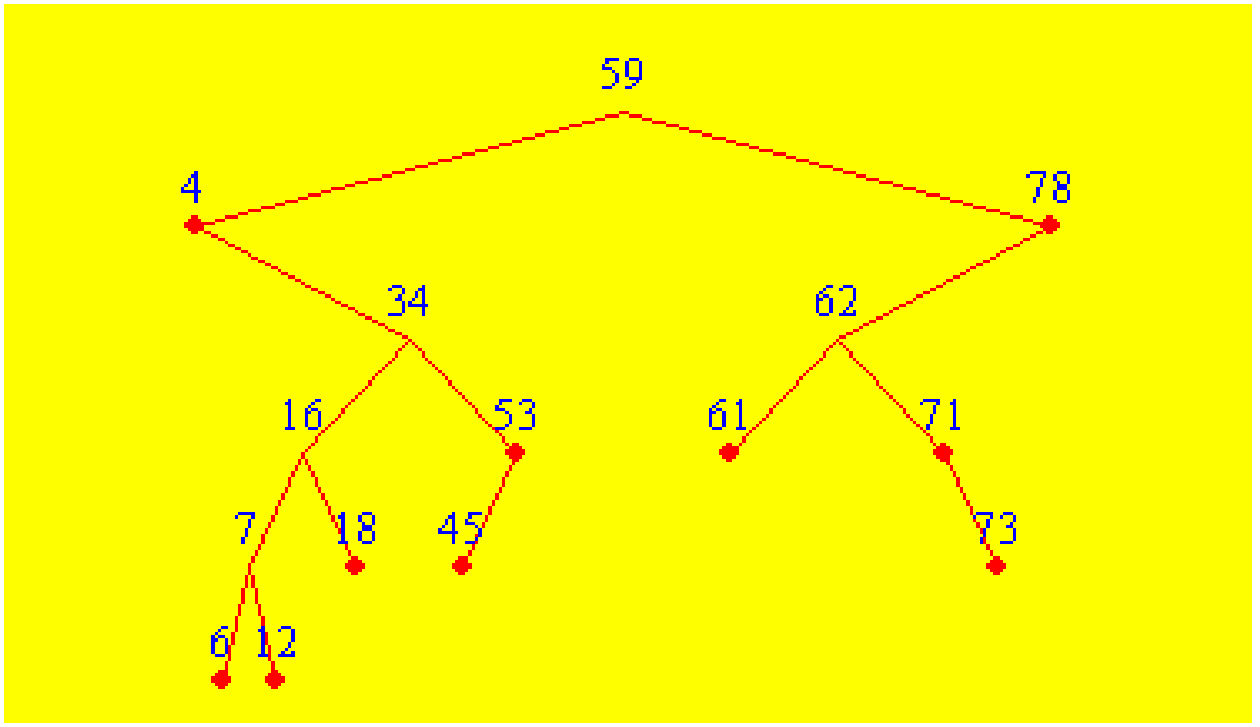
Algorithmen auf Bäumen implementieren



- Beispiel:
  - Bestimme die Anzahl der Knoten

Vorgehensweise

- *Beschreibung der Methoden in „Pseudocode“* (=umgangssprachlich)
- Danach: Implementierung in Java

Bestimme die Anzahl der Knoten

- Man definiert eine Methode
  private int knotenzahl(BinaryTree pTree)
- Die Methode misst die Anzahl der Knoten des Baumes, indem Sie...
  - die Anzahl der Knoten des linken Teilbaumes misst (= knotenLinks)
  - die Anzahl der Knoten des rechten Teilbaumes misst (= knotenRechts)
  - Daraus die Anzahl der Knoten des Gesamtbaumes berechnet (= knotenLinks + knotenRechts + 1) und diese Zahl als Ergebnis zurückgibt.

Wie programmiert man das?

- REKURSIV!!!
  - *Zu Anfang: Abbruchbedingung!*

Wie man vorgeht...

- public int knotenzahl(BinaryTree theTree)
  - Abbruchbedingung:
    Wenn theTree leer ist,
    dann gib 0 zurück. (Ein leerer Baum hat keine Knoten)
  - Rekursiver Aufruf:
    - *knotenLinks = knotenzahl(linkerTeilbaum);*
    - *knotenRechts = knotenzahl(rechterTeilbaum);*
    - *ergebnis = knotenLinks + knotenRechts + 1;*
    - *gib ergebnis zurück.*

Java Code

```
public int knotenzahl (BinaryTree pTree){
   int ergebnis = 0;
   if(pTree.isEmpty()){
      return ergebnis;
   }
   int knotenLinks =
         knotenzahl(pTree.getLeftTree());
   int knotenRechts =
         knotenzahl(pTree.getRightTree());
   int ergebnis =
         knotenLinks+knotenRechts+1;
      return ergebnis;
    }
```

Aufgaben

- public int summe(BinaryTree<Integer> pTree)
- public int groessteZahl(BinaryTree<Integer> pTree)
- public int tiefe(BinaryTree<Integer> pTree)
- Für den Strukturbaum:
- public double berechnen(BinaryTree pTree)
  - *Zahlen kommen nur in Blättern vor!*

```
double zahl = Double.parseDouble((String)b.getObject());
```

  - *Rechenzeichen kommen nur in „inneren“ Knoten vor!*

```
String rechenzeichen = (String)b.getObject();
```

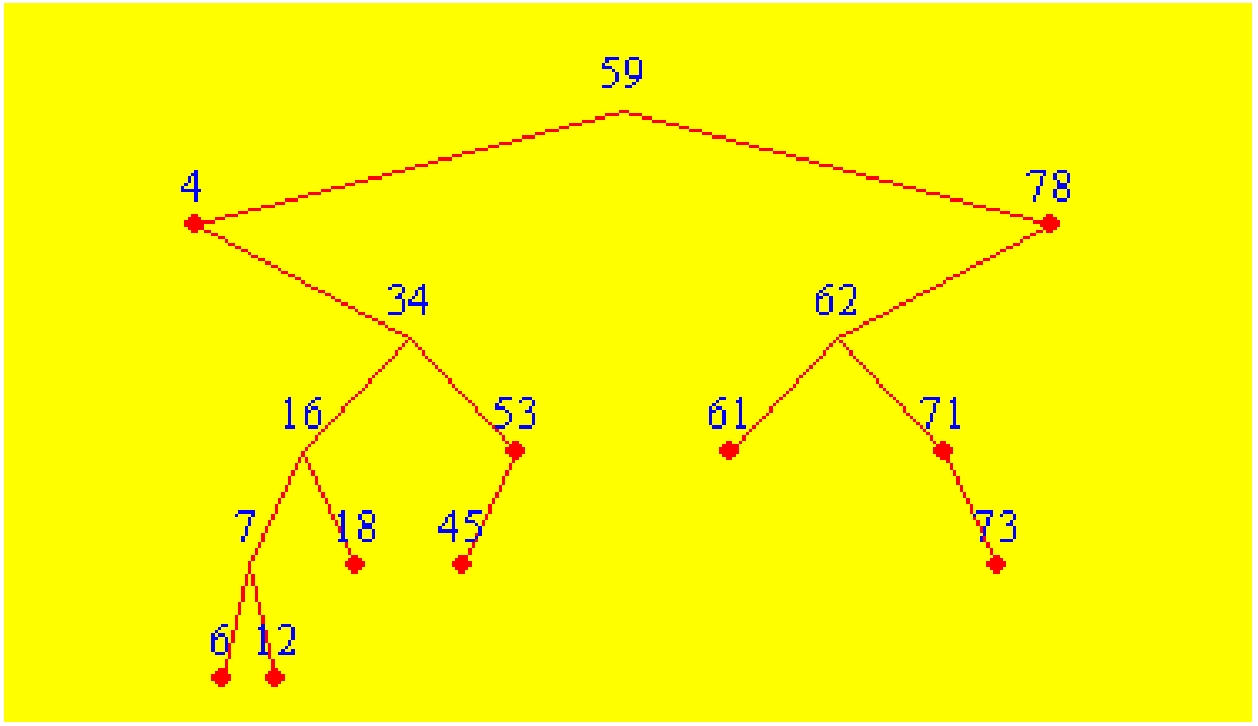
Hausaufgabe

- public int anzahlKleiner50(BinaryTree b)
  - Pseudocode
  - Java-Quellcode

Durchlauftechniken

- Preorder (die Wurzel am Anfang)
  - Besuche die Wurzel
  - Besuche den linken Teilbaum
  - Besuche den rechten Teilbaum
- Inorder (die Wurzel in der Mitte)
  - Besuche den linken Teilbaum
  - Besuche die Wurzel
  - Besuche den rechten Teilbaum
- Postorder (die Wurzel am Ende)
  - Besuche den linken Teilbaum
  - Besuche den rechten Teilbaum
  - Besuche die Wurzel

Aufgabe

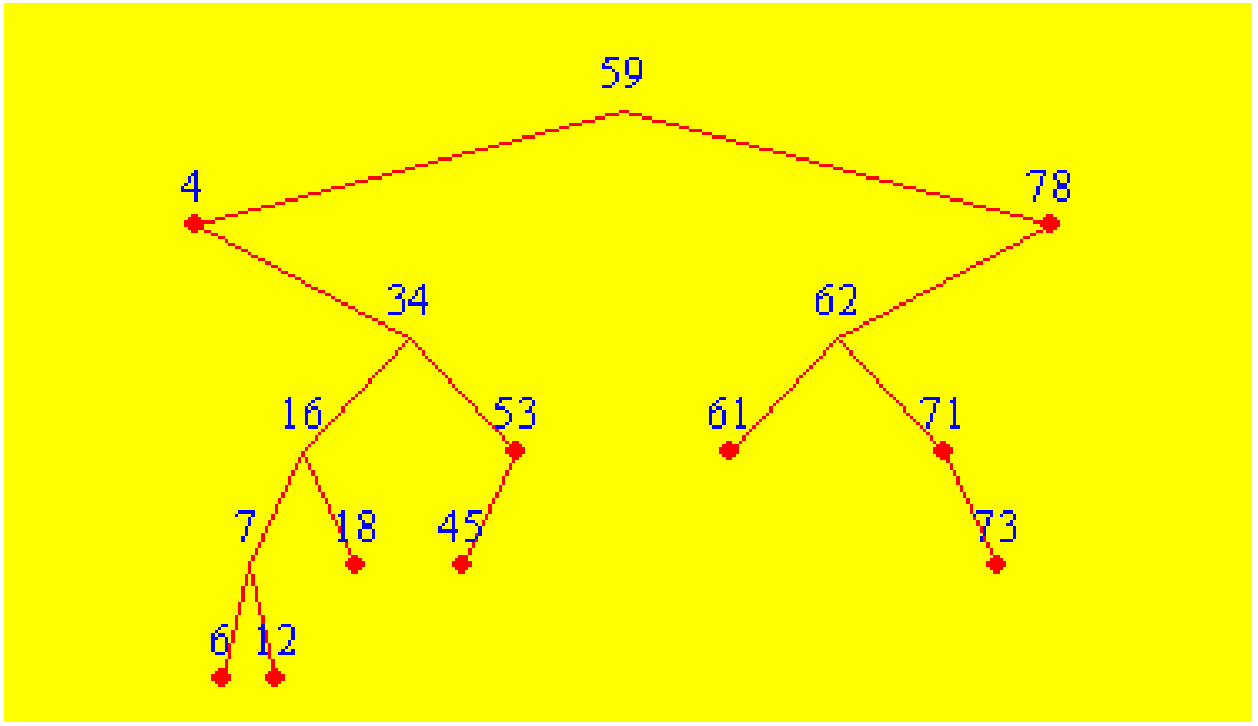


- Durchlaufe den Suchbaum in Preorder
- Durchlaufe den Suchbaum in Inorder
- Durchlaufe den Suchbaum in Postorder

Hausaufgabe

- Schreiben Sie für einen binären Suchbaum den Pseudocode für die folgenden Methoden:
  - public boolean enthaelt(BinTree theTree, int zahl)
    - Die Methode überprüft, ob theTree die zahl enthält.
  - public void preOrder(BinTree theTree)
    - Die Methode gibt die Knoten von theTree in Preorder aus.

Algorithmen auf Suchbäumen



- public boolean enthaelt(BinTree theTree, int zahl)
  - überprüft, ob in dem Teilbaum, der von theNode abhängt, zahl enthalten ist
- public void einfuegen(BinTree theTree, int zahl)
  - Fügt in den Teilbaum, der von theNode abhängt, zahl ein.