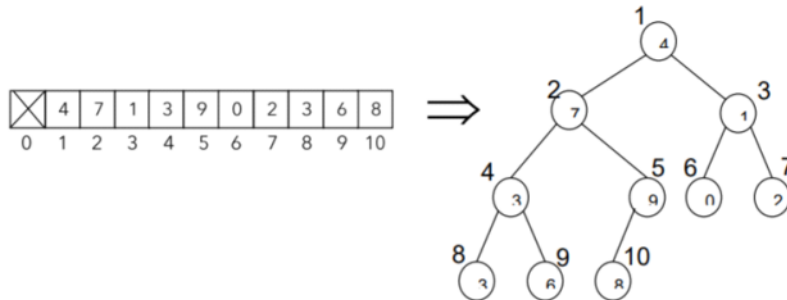


# ITB3, Übung 4, Aufgabe 1

Sebastian Schiener

## 1. Darstellungsformen von Binären Bäumen (4 Punkte)

Gegeben ist ein vollständiger binärer Baum zur Speicherung von Ganzzahlenwerten. Dieser Baum in **sequentieller Darstellung** soll in eine **verkettete Darstellung** umgeformt werden.



Die Daten für die sequentielle Darstellung werden aus einer Datei eingelesen werden. (Geben Sie auch die Testdateien mit ab!)

Überlegen Sie sich den sinnvollen Einsatz einer *Factory*, etc.

## Lösungsidee:

Darstellungsformen von binären Bäumen. Sequentielle Darstellung in TXT-Datei über Scanner einlesen und in verkettete Darstellung umgeformt. Wird dann auf die einzelnen Werte in ein `String[]` aufgeteilt. Dieses wird dann iteriert und daraus der Baum erstellt.

# ITB3, Übung 4, Aufgabe 2

Sebastian Schiener

## 2. Binäre Bäume / Binäre Suchbäume (10 Punkte)

Realisieren Sie eine oder mehrere Klassen, die für Binäre Bäume und Binäre Suchbäume die folgende Funktionalität zur Verfügung stellt.

- Einfügen eines Wertes in einem Baum
- Ausgabe des Baums am in PreOrder, InOrder, PostOrder und Level-By-Level
- Zählen der Blätter des Baums
- Bestimmen der Höhe des Baums
- Umwandeln eines Binären Baumes in einen Binären Suchbaum

### Lösungsidee:

Binäre Bäume und binäre Suchbäume können erstellt werden.

Werte können in beide eingefügt werden. (Beim Suchbaum natürlich nach entsprechendem Schema) Die Werte beider Baumarten können auf 4 verschiedene Traversmethoden ausgegeben werden. (PreOrder, PostOrder, InOrder, LevelByLevel). Die Anzahl der Blätter und die Höhe des Baumes können ermittelt werden.

Ein vorhandener oder erstellter Binärer Baum kann in einen binären Suchbaum umgewandelt werden.

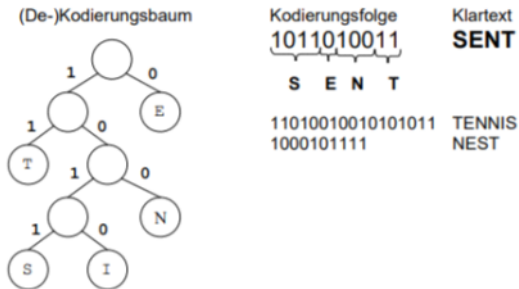
Das Interfaces Tree stellt Methoden für die verschiedenen Bäume zur Verfügung. Über die ENUM-Klasse Order und das Interface TraverseOrder wird die richtige Traverse-Methode in der InnerClass InOrder angesprochen. Die Klasse Node implementiert TreeNode und stellt die Verwendung der Knoten zur Verfügung. Die Klasse DecodeTree ist rein für Aufgabe 3 vorgesehen, die Klasse BinaryTree realisiert den Binären Baum und den Binären Suchbaum und bietet ihnen die gewünschte Funktionalität.

# ITB3, Übung 4, Aufgabe 3

Sebastian Schiener

## 3. Dekodierung ( 6 Punkte )

(De-)Kodierungsverfahren verwenden häufig binäre Bäume (Kodierungsbäumen) zur Abbildung von Kodierungsfolgen. Unter Kodierungsfolgen versteht man eindeutige Folgen von Bits.



Die Dekodierung bei gegebenem Kodierungsbaum erfolgt durch den gezielten Abstieg, wobei ,1' in den linken Unterbaum und ,0' in den rechten Unterbaum führt. Ein Blatt steht für das dekodierte Zeichen.

Entwickeln Sie eine Methode, die als Eingabe eine Kodierungsfolge in Form einer Zeichenkette und den dazu passenden, bereits vorhandenen Kodierungsbaum als Parameter erhält und den dekodierten Text als Rückgabe zurückliefert. Der Kodierungsbaum kann hartkodiert werden.

### Lösungsidee:

Hartkodierter Dekodierungsbaum wird im Konstruktor erstellt. Der Methode decode wird dieser fertige Baum (laut Angabe) übergeben und eine Kodierungsfolge (1er und 0er) anhand den übergebenen Baums wird der Klartext bestimmt und ausgegeben.