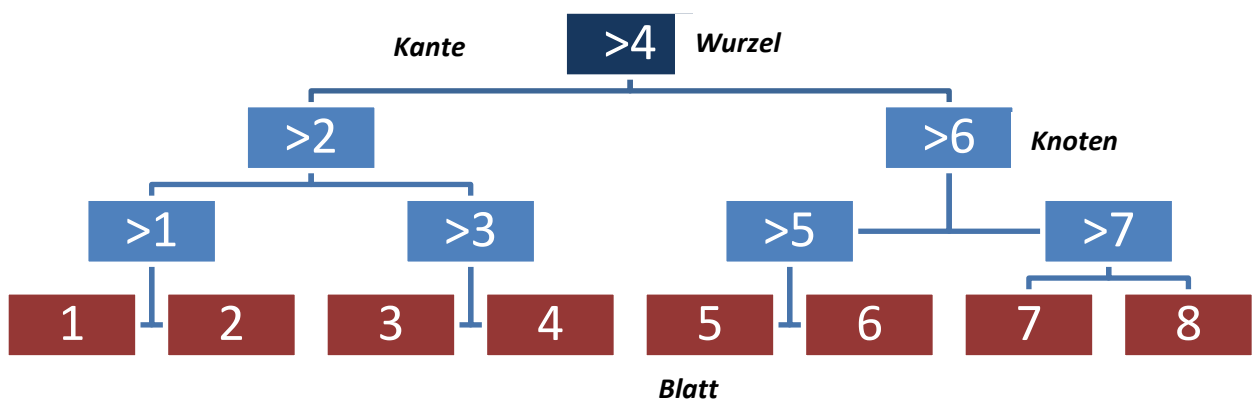


Protokoll der Informatikstunden vom 29.04. – 08.05.2014

In den Stunden haben wir angefangen, uns mit dem Programmieren von Binärbäumen zu beschäftigen. Als Einstieg haben wir überlegt, wie man eine Zahl zwischen 1 und 1000 mit möglichst wenigen Ja/Nein-Fragen herausfinden kann. Es ist am einfachsten, wenn man den Zahlenbereich immer halbiert. Für eine Zahl zwischen 1 und 1000 (bzw. 1024 als nächst größere Zweierpotenz) sind zehn Fragen notwendig.

Binärbäume

Hier wird verdeutlicht wie ein Binärbaum aussieht, der eine Zahl zwischen 1 und 8 ermittelt:



Ein Binärbaum ist

- Entweder leer
- Oder besteht aus einer Wurzel sowie einem linken und einem rechten Teilbaum

Knoten: Verzweigung eines Binärbaums

Wurzel: oberster Knoten eines Binärbaums

Blatt: unterste Knoten eines Binärbaums

Pfad: Weg von Wurzel zum Blatt (Länge = Anzahl der Knoten von der Wurzel bis zum Blatt)

Höhe: Länge des längsten Pfades

Programmierung eines ersten Baums

Die erste Aufgabe bestand darin, beliebige Texteinträge in einem Binärbaum zu speichern. Über zwei Buttons kann man jeweils zum nächst tieferen rechten bzw. linken Knoten navigieren. Durch einen weiteren Button gelangt man wieder zur Wurzel des Baums. Der Texteintrag wird immer an der aktuellen Stelle eingefügt.

Methode zum Einfügen eines Knotens

```
procedure TMain.btNeuKnotenClick(Sender: TObject);

    var
        Text : string;
        TextObjekt: TText;

begin
    if (edNeuKnoten.text = '') then
        lbWarnung.Caption := 'Bitte Text eingeben'
    else
        begin
            lbWarnung.Caption := '';
            Text := edNeuKnoten.text;
            TextObjekt := TText.Create(Text);
            Knoten.SetRootItem(TextObjekt);
            anzeigen;
        end;
    end;
end;
```

Methode , um zum rechten Knoten (Sohn) zu wechseln

```
procedure TMain.btRechtsClick(Sender: TObject);

begin
    if not Knoten.IsEmpty then
        begin
            Knoten := Knoten.getRightTree;
            anzeigen
        end;
    end;
end;
```