

# Algorithmen auf Bäumen

In diesem Programmierauftrag sollen die einfachen Baumalgorithmen implementiert werden: Bestimmen der Knotenanzahl, Bestimmen der Tiefe und Suchen nach einem gewünschten Wert.

Arbeite mit folgendem BlueJ-Projekt: <https://codeberg.org/qg-info-unterricht/binaerbaum-algorithmen>

Die Klasse *Binaerbaum* ist aus der Einführung übernommen, ergänzt wird die durch die Klassen *Baumalgorithmen* und *Testbaeume*.

Die Klasse *Baumalgorithmen* erhält bei der Instanziierung den Wurzelknoten eines Baums und speichert diesen im Attribut *wurzel*. Davon ausgehend können dann innerhalb der Klasse *Baumalgorithmen* die erwähnten Algorithmen implementiert werden, so wird beim Aufruf z.B. von *anzahl()* die interne Methode *anzahl(Binaerbaum b)* mit der Wurzel als Parameter aufgerufen.

Um die Algorithmen zu testen, kannst du einen von 100 Testbäumen aus der Klasse *Testbaeume* laden: Klicke nach dem Übersetzen mit der rechten Maustaste auf die Klasse *Testbaeume* und wähle die Methode *getTestBaum* mit einem Parameter zwischen 0 und 99.

Klicke im dann erscheinenden Fenster auf *hole*, um den Baum auf die Objektleiste zu holen. Jetzt kannst du ein *Baumalgorithmen*-Objekt erstellen, dem du den Namen des Testbaums als Parameter übergibst.

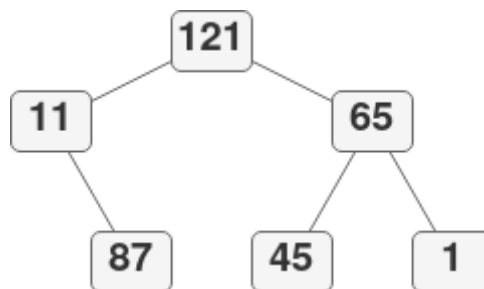
Du kannst dir den Baum mit der Methode *baumAnzeigen()* anzeigen lassen. Mit den Tests der Testklasse *BaumTester* aus, kannst du überprüfen, ob deine Implementationen korrekt arbeitet.

[Screencast öffnen](#)



**(A1)**

Erzeuge mit den zur Verfügung stehenden Konstruktoren den folgenden Binärbaum im BlueJ Projekt:



Lasse dir den Baum anzeigen und überprüfe, ob er korrekt erzeugt wurde. Dazu musst du ein Objekt Baumalgorithmen erzeugen, dem du als Wurzelknoten das zuvor erzeugte Knotenobjekt übergibst. Das Baumalgorithmen-Objekt stellt dann die Methode baumAnzeigen zur Verfügung.

**(A2)**

Implementiere die Methoden `anzahl()`, `tiefe()` und `enthaelt()`.

**Tipps**

Pseudocode "anzahl()"

```
wenn der baum leer ist:  
    gib 0 zurück  
sonst  
    gib 1 + anzahl(links) + anzahl(rechts) zurück
```

Pseudocode "tiefe()"

```
wenn der baum leer ist:  
    gib 0 zurück  
sonst  
    gib 1 + maximum(tiefe(links), tiefe(rechts)) zurück
```

Pseudocode "enthaelt()"

```
wenn der baum leer ist:  
    gib false zurück  
  
wenn daten == wert  
    gib true zurück  
sonst  
    gib (enthaelt(links) || enthaelt(rechts)) zurück
```

**Material**

<a href="#">02_algos.odp</a>	67.4 KiB 07.02.2022 18:23
<a href="#">02_algos.pdf</a>	69.7 KiB 07.02.2022 18:23
<a href="#">aa2.drawio.png</a>	6.7 KiB 07.02.2022 17:27
<a href="#">testbaeume-2024-05-02_09.34.55.gif</a>	3.0 MiB 02.05.2024 07:38

From:  
<https://info-bw.de/> -

Permanent link:  
<https://info-bw.de/faecher:informatik:oberstufe:adt:baeume:baumalgorithmen:start?rev=1714635575>

Last update: **02.05.2024 07:39**

