

Algorithmen und Datenstrukturen I

Blatt 6

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik
Hochschule München

Letzte Änderung: 18.03.2018 20:05

Aufgabe 1 — Binärbaum

Das Repository für diese Aufgabe bekommen Sie unter <https://classroom.github.com/a/QmRlxMAM>.

Implementieren Sie den Binärbaum wie im Repository durch die Header-Datei vorgegeben ist:

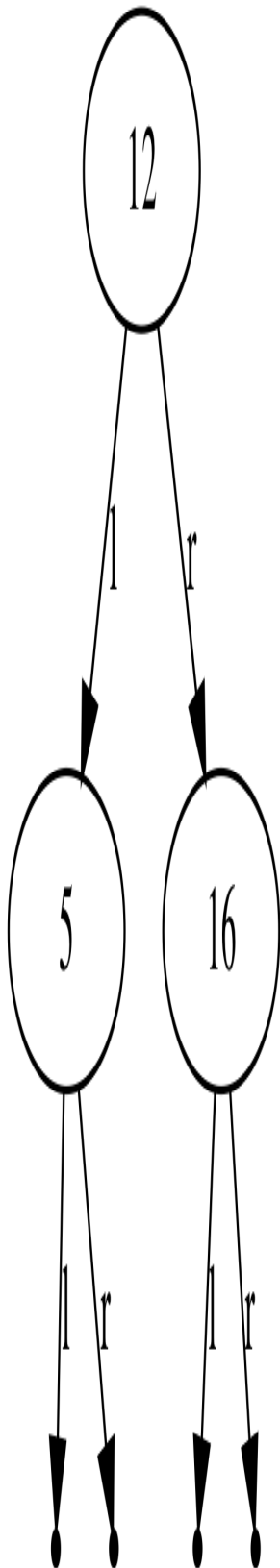
- Die Methoden `preorder`, `inorder` und `postorder` zur Traversierung eines Baumes sind bereits implementiert. Ändern Sie daran nichts.
- Der Operator `<<` ist für den Baum schon überladen. Ändern Sie daran nichts.
- Ändern Sie nichts am öffentlichen Interface der Klasse `BinTree`. Private Komponenten und Friends dürfen Sie hinzufügen.
- Die mitgelieferten Tests sind relativ rudimentär. Erweitern Sie diese bei Bedarf.

Im Repository finden Sie außerdem eine `main.cpp` mit einer interaktiven `main`-Funktion zum Arbeiten mit dem Binärbaum. Die Funktion schreibt den Baum als in der `dot`-Syntax in eine Datei `tree.dot`, z.B.

```
digraph tree {
  12 -> 5 [label="l"];
  null5[shape=point];
  5 -> null15 [label="l"];
  null16[shape=point];
  5 -> null16 [label="r"];
  12 -> 16 [label="r"];
}
```

```
    null7[shape=point];  
    16 -> null7 [label="l"];  
    null8[shape=point];  
    16 -> null8 [label="r"];  
}
```

Wenn Sie [GraphViz](#) installiert haben, wird gleich eine png-Datei erzeugt, z.B.



Getestet habe ich das nur unter UNIX. Ich würde mich über eine Rückmeldung freuen ob und wie das unter Windows funktioniert.

Sie können den Inhalt der dot-Datei auch unter <http://www.webgraphviz.com/> nutzen.