

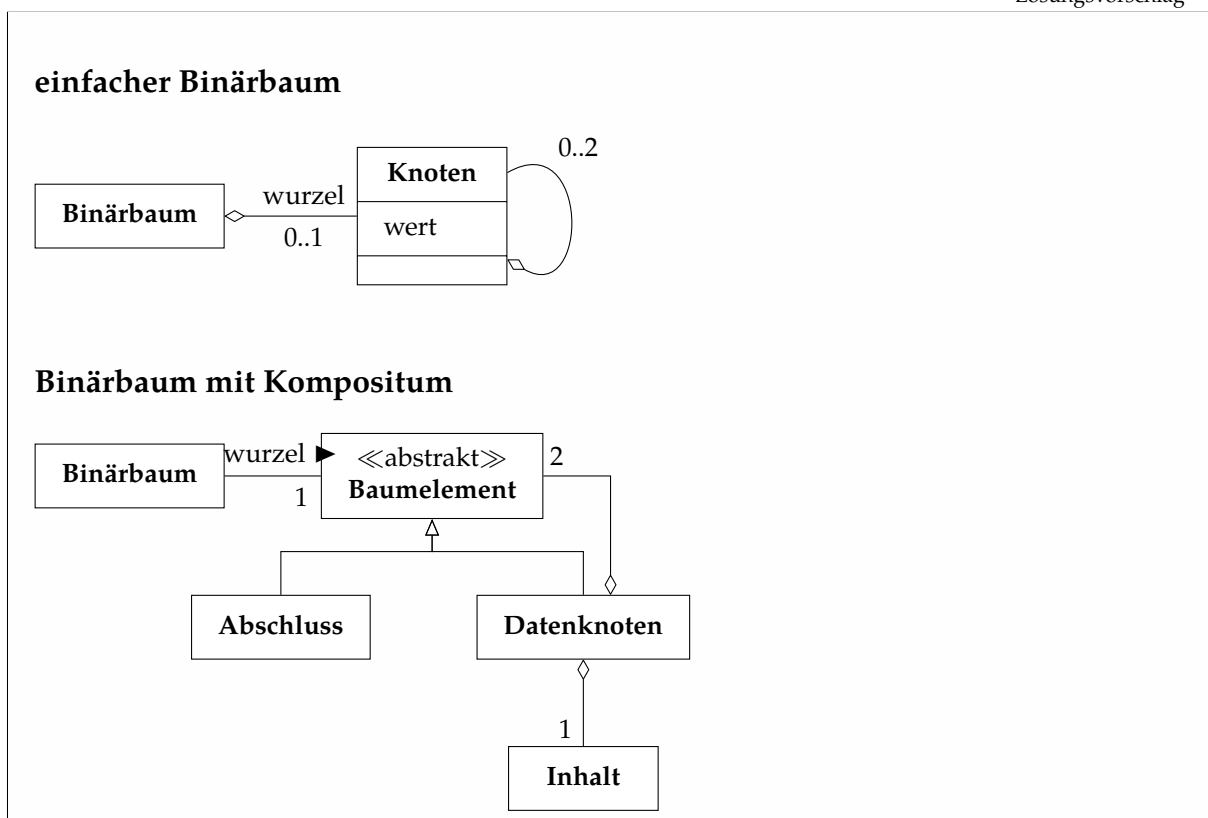
Klassendiagramm und Implementierung *(Binärbaum: Klassendiagramm und Implementierung)*

Stichwörter: Binärbaum, Klassendiagramm, Implementierung in Java

Klassendiagramm und Implementierung

(a) Erstellen Sie ein Klassendiagramm für einen Binärbaum.

Lösungsvorschlag



(b) Entwerfen Sie eine mögliche Implementierung zur Erzeugung eines binären Baumes in Java.

einfacher Binärbaum

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/baum/einfach/Baum.java](https://github.com/bschlangaul/aufgaben/blob/main/java/org/bschlangaul/aufgaben/aud/baum/einfach/Baum.java)

```
public class Knoten {  
  
    private Knoten links;  
    private Knoten rechts;  
    private int wert;  
  
    public Knoten(int wert) {  
        this.wert = wert;  
    }  
}
```

```
public void setzeWert(int w) {
    wert = w;
}

public int gibWert() {
    return wert;
}

public void setzeLinks(Knoten l) {
    links = l;
}

public void setzeRechts(Knoten r) {
    rechts = r;
}

public Knoten gibLinks() {
    return links;
}

public Knoten gibRechts() {
    return rechts;
}

public void fügeEin(int wert) {
    if (wert <= this.gibWert()) {
        if (this.gibLinks() != null) {
            this.gibLinks().fügeEin(wert);
        } else {
            this.setzeLinks(new Knoten(wert));
        }
    } else {
        if (this.gibRechts() != null) {
            this.gibRechts().fügeEin(wert);
        } else {
            this.setzeRechts(new Knoten(wert));
        }
    }
}
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/baum/einfach/Knoten.java](https://github.com/bschlangaul/aufgaben/blob/main/java/org/bschlangaul/aufgaben/aud/baum/einfach/Knoten.java)

Binärbaum mit Kompositum

```
class Binaerbaum {
    private Bauelement wurzel;

    public Binaerbaum() {
        wurzel = new Abschluss();
    }
}
```

```
}

public void setzeWurzel(Bauelement wurzel) {
    this.wurzel = wurzel;
}

public Bauelement gibWurzel() {
    return wurzel;
}

public int gibAnzahl() {
    return wurzel.gibAnzahl();
}

public void gibAus() {
    wurzel.gibAus();
}
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Binaerbaum.java](https://github.com/bschlangaul/aufgaben/aud/baum/kompositum/Binaerbaum.java)

```
abstract class Bauelement {
    public abstract void setztleLinks(Bauelement nl);

    public abstract void setzeRechts(Bauelement nr);

    public abstract Bauelement gibLinks();

    public abstract Bauelement gibRechts();

    public abstract Datenelement gibInhalt();

    public abstract int gibAnzahl();

    public abstract void gibAus();
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Bauelement.java](https://github.com/bschlangaul/aufgaben/aud/baum/kompositum/Bauelement.java)

```
class Abschluss extends Bauelement {

    public void setztleLinks(Bauelement links) {
        System.out.println("Ein Abschluss hat kein linkes Element!");
    }

    public void setzeRechts(Bauelement rechts) {
        System.out.println("Ein Abschluss hat kein rechts Element!");
    }

    public Bauelement gibLinks() {
        System.out.println("Linkes Element nicht bekannt!");
        return this;
    }
}
```

```
public Bauelement gibRechts() {
    System.out.println("Linkes Element nicht bekannt!");
    return this;
}

public Datenelement gibInhalt() {
    return null;
}

public int gibAnzahl() {
    return 0;
}

public void gibAus() {
}
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Abschluss.java](https://github.com/orgs/bschlangaul/repositories?q=aufgaben%2Faud%2Fbaum%2Fkompositum%2FAbschluss.java)

```
class Datenknoten extends Bauelement {
    private Bauelement links, rechts;
    private Datenelement inhalt;

    public Datenknoten(Bauelement links, Bauelement rechts, Datenelement inhalt) {
        this.links = links;
        this.rechts = rechts;
        this.inhalt = inhalt;
    }

    public void setztleLinks(Bauelement links) {
        this.links = links;
    }

    public void setzeRechts(Bauelement rechts) {
        this.rechts = rechts;
    }

    public void inhaltSetzen(Datenelement inhalt) {
        this.inhalt = inhalt;
    }

    public Bauelement gibLinks() {
        return links;
    }

    public Bauelement gibRechts() {
        return rechts;
    }

    public Datenelement gibInhalt() {
        return inhalt;
    }

    public int gibAnzahl() {
```

```
        return 1 + links.gibAnzahl() + rechts.gibAnzahl();
    }

    public void gibAus() {
        System.out.print(" [");
        links.gibAus();
        inhalt.gibAus();
        rechts.gibAus();
        System.out.print("] ");
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Datenknoten.java](#)

```
abstract class Datenelement {
    public abstract void gibAus();
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Datenelement.java](#)

```
class Inhalt extends Datenelement {
    private String inhalt;

    public Inhalt(String inhalt) {
        this.inhalt = inhalt;
    }

    public String gibInhalt() {
        return inhalt;
    }

    public void gibAus() {
        System.out.print(inhalt);
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Inhalt.java](#)

```
import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class BinaerbaumTest {

    @Test
    public void teste(){
        Binaerbaum baum = new Binaerbaum();
        Inhalt[] inhalte = new Inhalt[16];
        Datenknoten[] datenknoten = new Datenknoten[16];
        inhalte[0] = new Inhalt("Inhalt 1");

        inhalte[1] = new Inhalt("Inhalt 2");
        inhalte[2] = new Inhalt("Inhalt 3");
    }
}
```

```
inhalte[3] = new Inhalt("Inhalt 4");
inhalte[4] = new Inhalt("Inhalt 5");

for (int i = 0; i < 5; i++) {
    datenknoten[i] = new Datenknoten(new Abschluss(), new Abschluss(),
    ↪    inhalt[i]);
}
baum.setzeWurzel(datenknoten[0]);

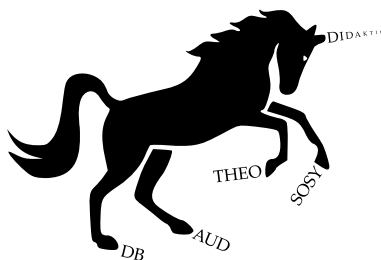
datenknoten[0].setzteLinks(datenknoten[1]);
datenknoten[0].setzeRechts(datenknoten[2]);

datenknoten[1].setzteLinks(datenknoten[3]);
datenknoten[1].setzeRechts(datenknoten[4]);

assertEquals(5, baum.gibAnzahl());

Inhalt inhalt = (Inhalt) baum.gibWurzel().gibLinks().gibLinks().gibInhalt();
assertEquals("Inhalt 4", inhalt.gibInhalt());
}
}
```

Code-Beispiel auf Github ansehen: [src/test/java/org/bschlangaul/aufgaben/aud/baum/kompositum/BinaerbaumTest.java](https://github.com/bschlangaul/aufgaben/blob/main/aud/baum/kompositum/BinaerbaumTest.java)



Die Bschlangaul-Sammlung

Hermine Bschlangauland Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/30_AUD/80_Baeume/20_Binaerer-Suchbaum/Aufgabe_Klassendiagramm-Implementierung.tex