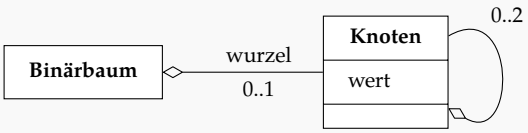
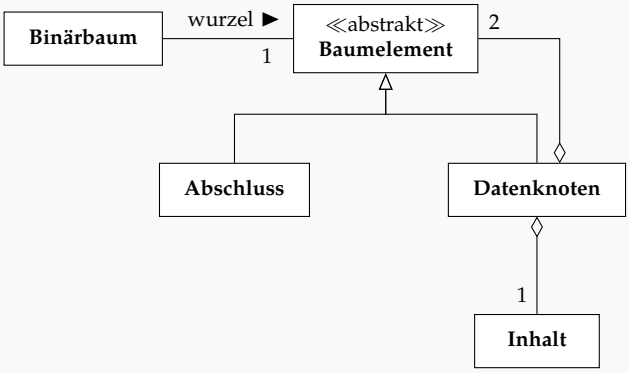
Klassendiagramm und Implementierung

(a) Erstellen Sie ein Klassendiagramm für einen Binärbaum.

einfacher Binärbaum



Binärbaum mit Kompositum



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

einfacher Binärbaum

```java
public class Baum {

  public Knoten wurzel;
  public int anzahl;

  public Baum(int anzahl) {
    this.anzahl = anzahl;

    if (anzahl == 0) {
      return;
    } else {
      wurzel = new Knoten();
      // wurzel.wert = anzahl;
      Baum linkBaum = new Baum(anzahl / 2);
      // Zeiger auf linken Teilbaum
      wurzel.links = linkBaum.wurzel;
      Baum rechtBaum = new Baum(anzahl - 1 - anzahl / 2);
      // Zeiger auf rechten Teilbaum
      wurzel.rechts = rechtBaum.wurzel;
    }
  }
```

```
24
25  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/aud/baum/einfach/Baum.java`

```
3   public class Knoten {
4
5     public Knoten links;
6     public Knoten rechts;
7     public int wert;
8
9     public Knoten() {
10    }
11
12  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/aud/baum/einfach/Knoten.java`

Binärbaum mit Kompositum

```
3   class Binaerbaum {
4     private Baumelement wurzel;
5
6     public Binaerbaum() {
7       wurzel = new Abschluss();
8     }
9
10    public void setzeWurzel(Baumelement wurzel) {
11      this.wurzel = wurzel;
12    }
13
14    public Baumelement gibWurzel() {
15      return wurzel;
16    }
17
18    public int gibAnzahl() {
19      return wurzel.gibAnzahl();
20    }
21
22    public void gibAus() {
23      wurzel.gibAus();
24    }
25  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Binaerbaum.java`

```
3   abstract class Baumelement {
4     public abstract void setzteLinks(Baumelement nl);
5
6     public abstract void setzeRechts(Baumelement nr);
7
8     public abstract Baumelement gibLinks();
9
10    public abstract Baumelement gibRechts();
11
12    public abstract Datenelement gibInhalt();
13
```

```
14    public abstract int gibAnzahl();
15
16    public abstract void gibAus();
17  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Baumelement.java`

```
3   class Abschluss extends Baumelement {
4
5     public void setzteLinks(Baumelement links) {
6       System.out.println("Ein Abschluss hat kein linkes Element!");
7     }
8
9     public void setzeRechts(Baumelement rechts) {
10      System.out.println("Ein Abschluss hat kein rechts Element!");
11    }
12
13    public Baumelement gibLinks() {
14      System.out.println("Linkes Element nicht bekannt!");
15      return this;
16    }
17
18    public Baumelement gibRechts() {
19      System.out.println("Linkes Element nicht bekannt!");
20      return this;
21    }
22
23    public Datenelement gibInhalt() {
24      return null;
25    }
26
27    public int gibAnzahl() {
28      return 0;
29    }
30
31    public void gibAus() {
32    }
33  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Abschluss.java`

```
3   class Datenknoten extends Baumelement {
4     private Baumelement links, rechts;
5     private Datenelement inhalt;
6
7     public Datenknoten(Baumelement links, Baumelement rechts,
      ↪  Datenelement inhalt) {
8       this.links = links;
9       this.rechts = rechts;
10      this.inhalt = inhalt;
11    }
12
13    public void setzteLinks(Baumelement links) {
14      this.links = links;
15    }
16
17    public void setzeRechts(Baumelement rechts) {
18      this.rechts = rechts;
19    }
20
```

```
21    public void inhaltSetzen(Datenelement inhalt) {
22      this.inhalt = inhalt;
23    }
24
25    public Baumelement gibLinks() {
26      return links;
27    }
28
29    public Baumelement gibRechts() {
30      return rechts;
31    }
32
33    public Datenelement gibInhalt() {
34      return inhalt;
35    }
36
37    public int gibAnzahl() {
38      return 1 + links.gibAnzahl() + rechts.gibAnzahl();
39    }
40
41    public void gibAus() {
42      System.out.print(" [");
43      links.gibAus();
44      inhalt.gibAus();
45      rechts.gibAus();
46      System.out.print("] ");
47    }
48  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Datenknoten.java`

```
3   abstract class Datenelement {
4     public abstract void gibAus();
5   }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Datenelement.java`

```
3   class Inhalt extends Datenelement {
4     private String inhalt;
5
6     public Inhalt(String inhalt) {
7       this.inhalt = inhalt;
8     }
9
10    public String gibInhalt() {
11      return inhalt;
12    }
13
14    public void gibAus() {
15      System.out.print(inhalt);
16    }
17  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/aud/baum/kompositum/Inhalt.java`

```
3   import static org.junit.Assert.assertEquals;
4
5   import org.junit.Test;
6
7   public class BinaerbaumTest {
```

```java

    @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(), inhalte[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`