

## Übungen zu Einführung in die Informatik I

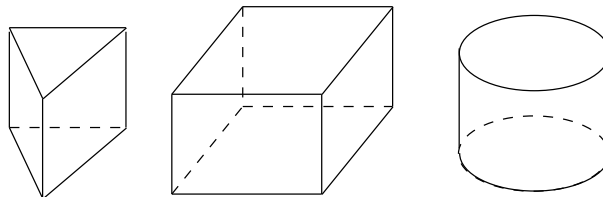
### Aufgabe 44      **Doppelt verkettete Liste in Java**

Doppelt verkettete Listen haben gegenüber normalen Listen den Vorteil, dass jedes Element zusätzlich zum Nachfolger auch seinen Vorgänger kennt.

- a) Erstellen sie in UML einen Entwurf für eine doppelt verkettete Liste. Diese soll folgende Anforderungen erfüllen:
  - Die Liste soll Elemente vom Typ `Object` aufnehmen können.
  - Die Liste soll folgende Operationen effizient (konstanter Zugriff) unterstützen: `isEmpty`, `addFirst`, `addLast`, `getFirst`, `getLast`, `removeFirst`, `removeLast`, `getSize`
- b) Implementieren Sie das UML-Model in Java.

### Aufgabe 45      **Vererbung**

*Gerade Prismen* oder *gerade Säulen* sind geometrische Körper, die durch senkrechte Parallelverschiebung einer ebenen Fläche im Raum entstehen. Die folgende Abbildung zeigt als Beispiele für gerade Prismen ein gerades Dreiecksprisma, einen Quader und einen geraden Zylinder, die durch Verschiebung eines Dreiecks, eines Rechteckes bzw. eines Kreises entstehen:



Ein *Würfel* ist bereits durch die Angabe seiner Höhe eindeutig bestimmt, Prismen über regelmäßigen (d. h. gleichseitigen) Vielecken wie das *regelmäßige Dreiecksprisma*, der *quadratische Quader* oder das *regelmäßige Sechsecksprisma* sind durch die Angabe ihrer Höhe und der Seitenlänge ihrer jeweiligen Grundfläche bestimmt, ein *Quader* durch die Angabe seiner Höhe sowie der Länge und Breite seiner Grundfläche und ein *Zylinder* durch die Angabe seiner Höhe und des Radius seiner Grundfläche.

In dieser Aufgabe soll eine Klassenhierarchie für diese verschiedenartigen geraden Prismen in UML modelliert und in Java implementiert werden. Wir werden von jetzt an immer nur kurz von Prismen sprechen, aber immer gerade Prismen meinen.

- a) An Operationen soll ein Prisma die Berechnung von Umfang und Flächeninhalt seiner Grundfläche, die Berechnung seiner Mantelfläche, seiner Oberfläche und seines Volumens zur Verfügung stellen, sowie den Vergleich seines Volumens mit dem eines anderen Prismas. Geben Sie eine Klassenhierarchie zur Modellierung aller oben genannten Prismen in UML an. Dabei sollen jeweils gleichartige Attribute in einer geeigneten Oberklasse zusammengefasst werden.

- b) An welcher Stelle Ihrer Klassenhierarchie müssen die in a) spezifizierten Operationen implementiert werden, damit möglichst viele der Implementationen in Unterklassen wiederverwendet werden können?
- c) Implementieren Sie Ihr Klassenmodell in Java und überlegen Sie, wie man geeignete Testfälle konstruieren kann.

**Hinweise:** Die Fläche eines gleichseitigen Dreiecks mit Seitenlänge  $a$  ist  $\frac{a^2}{4}\sqrt{3}$ , die Fläche eines regelmäßigen Sechsecks mit Seitenlänge  $a$  ist  $\frac{3a^2}{2}\sqrt{3}$ . Die Java-Klasse `Math` stellt in der Klassenvariablen `Math.PI` einen Wert für  $\pi$  zur Verfügung und enthält eine Klassenmethode `Math.sqrt()`.

#### **Aufgabe 46 (H) Vererbung in Java: Auswertung arithmetischer Ausdrücke (20 Punkte)**

In dieser Aufgabe soll das Konzept der Vererbung aus Java zur Auswertung arithmetischer Ausdrücke herangezogen werden. Der Einfachheit halber sollen nur die vier Grundrechenarten und ganze Zahlen betrachtet werden, wobei die beliebige Schachtelung geklammerter Ausdrücke möglich sein soll. Die Basis der Klassenhierarchie soll die abstrakte Oberklasse `Expression` mit ihren zwei Methoden `print()` und `evaluate()` bilden, jeweils für die Ausgabe und die Auswertung eines (Teil-)Ausdrucks. Davon werden Konstanten (`Constant`) und Operatoren (`Operator`) abgeleitet.

##### **Aufgabenstellung:**

- Entwerfen Sie zwei Lösungsansätze:
  - a) Die Klasse `Operator` soll für die Auswertung aller Operatoren (+, −, \*, /) zuständig sein, indem sie zusätzlich zum linken und rechten Teilausdruck auch noch die Art der Verknüpfung übergeben bekommt.
  - b) Leiten Sie von einer Basis-Operator-Klasse für die jeweiligen Operatoren eigene Klassen ab.
- Erstellen sie zur Veranschaulichung Ihres Design jeweils ein UML Diagramm.
- Testen Sie Ihr Programm indem Sie folgende zwei Ausdrücke auswerten:

```
Expression a = new Operator('*',  
    new Operator('+', new Constant(5), new Constant(2)),  
    new Operator('-', new Constant(6), new Constant(9)));
```

```
Expression b = new MultiplyOperator(  
    new PlusOperator(new Constant(5), new Constant(2)),  
    new MinusOperator(new Constant(6), new Constant(9)));
```