

Praktikum zu
**Einführung in die Informatik für
LogWilngs und WiMas**
Wintersemester 2015/16

Übungsblatt 9
Besprechung:
11.–15.01.2016
(KW 2)

Vorbereitende Aufgaben

Aufgabe 9.1: Bäume

In dieser Aufgabe sollen Sie sich mit Bäumen beschäftigen.

a) Quizfragen:

- Wie nennt man den Knoten ohne Vater?

- Wie nennt man einen Knoten x , der direkter Nachfolger eines Knoten y ist?

- Wie nennt man einen Knoten ohne Nachfolger?

b) Gegeben sei ein binärer Baum in Form eines Arrays:

	5	3	7	2	4	9	8
--	---	---	---	---	---	---	---

Geben Sie die grafische Repräsentation des Baumes an:

c) Handelt es sich bei diesem Baum um einen Heap? Wenn nein, warum nicht?

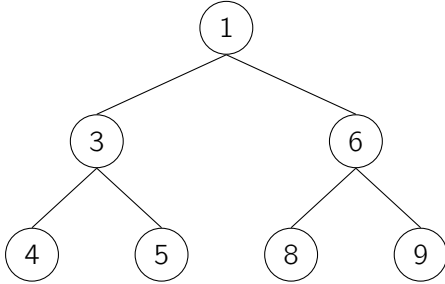
Präsenzaufgaben

Aufgabe 9.2: Heaps

In dieser Aufgabe sollen Sie ihre Kenntnisse über Heaps anwenden, um eine sortierte Folge von Zahlen auszugeben.

Leeren Sie den angegebenen Heap, indem Sie, wie in der Vorlesung (Kapitel 5.2) beschrieben, das kleinste Element entfernen.

Geben Sie eine Baumrepräsentation des Heaps nach jeder Extraktion des Minimums und nach jedem Tausch während der **Heapify**-Operation an:



Aufgabe 9.3: Einführung in die Objektorientierung

In dieser Aufgabe sollen Sie ein geometrisches Objekt mithilfe objektorientierter Programmierung modellieren. Beantworten Sie zuvor die folgende Frage:

Mit welchem Schlüsselwort können neue Objekte mit ihrem Konstruktor instanziiert werden?

Gegeben ist die Klasse **Sphere**. Diese beschreibt eine geometrische Kugel anhand ihres Radius. Ergänzen Sie die vermerkten Stellen im Quellcode der Klasse, indem Sie den Inhalt zuerst in einer eigenen Quelldatei im Paket **blatt09** übernehmen und dann anpassen.

```
1 package blatt09;
2
3 public class Sphere {
4
5     public static double PI = 3.14159265359;
6
7     private double radius;
8
9     public Sphere(double givenRadius) {
10         /* Initialisiere das radius-Attribut
11            * mit dem Parameter des Konstruktors */
12     }
13
14     public double getRadius() {
15         return radius;
16     }
17
18     public double getDiameter() {
19         /* Ergaenzen: Den Durchmesser der Kugel zurueckgeben */
20     }
21
22     public double getVolume() {
23         /* Ergaenzen: Das Volumen der Kugel zurueckgeben */
24     }
25
26     public double getSurfaceArea() {
27         /* Ergaenzen: Die Oberfaeche der Kugel zurueckgeben */
28     }
29 }
```

Aufgabe 9.4: Klassen selbst definieren

In dieser Aufgabe sollen Sie weitere geometrische Objekte mithilfe objektorientierter Programmierung modellieren.

- Erstellen Sie eine Klasse namens **Cuboid** im Paket **blatt09**. Diese repräsentiert einen geometrischen Quader.
- Deklarieren Sie drei **private** Attribute vom Typ **double** mit den Namen **length**, **width** und **height**.
- Deklarieren Sie einen **öffentlichen** Konstruktor für diese Objekte, der drei Parameter für die Maße entgegennimmt. Im Konstruktor sollen Sie die Attribute des Objektes auf die übergebenen Werte setzen.
- Schreiben Sie drei **öffentliche** Methoden namens **getLength**, **getWidth** und **getHeight**, die die drei Attribute zurückgeben.
- Schreiben Sie eine **öffentliche** Methode namens **getVolume**, die das Volumen des Quaders berechnet und zurückgibt.
- Schreiben Sie eine **öffentliche** Methode namens **getSurfaceArea**, die die Oberfläche des Quaders berechnet und zurückgibt.

Aufgabe 9.5: Klassentest

In dieser Aufgabe sollen Sie das Testen und die Verwendung von Klassen in separaten Testklassen erproben.

Bisher sollten Sie in keiner der erstellten Geometrieklassen eine **main**-Methode geschrieben haben. Entsprechend konnten Sie Ihren Quellcode noch nicht ausführen.

- Schreiben Sie eine Klasse **GeometryTest** im Paket **blatt09** mit einer **main**-Methode.
- Deklarieren und initialisieren Sie in der **main**-Methode folgende Objekte:
 - Drei Objekte vom Typ Sphere mit den Radien 1, 3 und 12
 - Drei Objekte vom Typ Cuboid mit den Maßen $(1 \times 1 \times 1)$, $(3\pi \times 3 \times 4)$ und $(2 \times 3 \times 7)$
- Lassen Sie sich einige Testwerte ausgeben und überprüfen Sie sie.
- Testen Sie überdies, ob das Volumen des zweiten Würfels identisch ist mit dem Volumen der zweiten Kugel.

Begründen Sie, weshalb Sie die Gleichheit der Volumina annehmen können.

Warum könnte der Wert dieser Berechnungen unterschiedlich sein?

Ergänzende Aufgaben

Aufgabe 9.6: Überladene Methoden

Achtung: Hier wird auf Stoff der nächsten Vorlesung vorgegriffen. Dennoch sollten Sie diese Aufgabe mit ihrem bisherigen Wissen bearbeiten können.

In dieser Aufgabe sollen Sie Methoden programmieren, die die bisher geschriebenen Objekte miteinander vergleichen können.

Ergänzen Sie die beiden Klassen **Sphere** und **Cuboid** um jeweils zwei öffentliche Methoden mit dem Namen **compare** und dem Rückgabebetyp **int**. Die Methoden sollen sich nur in ihrem Parameter unterscheiden: Die eine soll eine Kugel, die andere einen Quader entgegennehmen.

Die Methoden sollen das aufrufende Objekt mit dem übergebenen Objekt vergleichen:

- Das Vergleichskriterium ist ihr Volumen.
- Die Methoden sollen einen Wert größer 0 zurückgeben, wenn das aufrufende Objekt größer ist, als das andere.
- Die Methoden sollen einen Wert kleiner 0 zurückgeben, wenn das aufrufende Objekt kleiner ist als das andere.
- Die Methoden sollen 0 zurückgeben, wenn die Volumina der beiden Objekte gleich sind.