

Praktikum zu
**Einführung in die Informatik für
LogWilngs und WiMas**
Wintersemester 2017/18

Übungsblatt 9

Besprechung:
18.–22.12.2017
(KW 51)

Vorbereitende Aufgaben

Aufgabe 9.1: Bäume

In dieser Aufgabe sollen Sie sich mit Bäumen beschäftigen.

a) Quizfragen:

- Wie nennt man den Knoten ohne Vorgänger?

- 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?

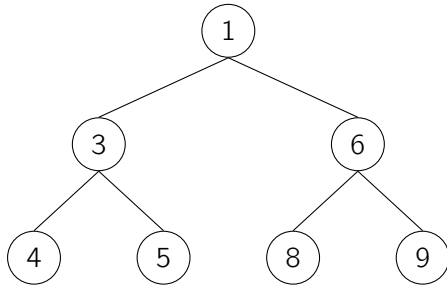
Präsenzaufgaben

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 vollständig, indem Sie, wie in der Vorlesung (Kapitel 5.2) beschrieben, das jeweils 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 Fahrzeug mit Hilfe objektorientierter Programmierung modellieren. Beantworten Sie zuvor die folgende Frage:

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

Gegeben ist die Klasse **Vehicle**. Diese beschreibt ein beliebiges Fahrzeug. 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 Vehicle {
4
5     private int wheels;
6     private String fuel;
7
8     public Vehicle(int inWheels, String inFuel) {
9         wheels = inWheels;
10        /* Restliche Attribute Ergaenzen */
11    }
12
13    public int getWheels() {
14        /* Ergaenzen: Die Anzahl der Reifen zurueckgeben */
15    }
16
17    public void print() {
18        /* Ergaenzen: Den verwendeten Treibstoff und
19         * die Anzahl der Reifen ausgeben */
20    }
21 }
```

Aufgabe 9.4: Klassen selbst definieren

In dieser Aufgabe sollen Sie PKW mit Hilfe objektorientierter Programmierung modellieren.

- Erstellen Sie eine Klasse namens **Car** im Paket **blatt09**. Diese repräsentiert einen PKW.
- Deklarieren Sie die String-Attribute **manufacturer** und **model** sowie ein Integer-Attribut namens **horsePower**, die Hersteller, Modell und die Pferdestärken des Motors repräsentieren sollen.
- Deklarieren Sie folgende Konstante:

```
public static final double WATT_PER_HORSEPOWER = 735.5;
```
- Schreiben Sie einen öffentlichen Konstruktor, der drei Parameter für die drei Attribute entgegen nimmt und diese auf die übergebenen Werte setzt.
- Schreiben Sie eine öffentliche Methode namens **getPower()**, die die Pferdestärken des Autos in Watt umrechnet und als **double** zurückgibt.
- Schreiben Sie eine Methode namens **print()**, die nichts zurückgibt und den Hersteller, das Modell und die Leistung ausgibt. Als Beispiel kann die Ausgabe folgendermaßen aussehen:

```
VW-Bus von Volkswagen mit 18387.5 Watt Leistung
```

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 Klassen eine **main**-Methode geschrieben haben. Entsprechend konnten Sie Ihren Quellcode noch nicht ausführen.

- Schreiben Sie eine Klasse **VehicleTest** im Paket **blatt09** mit einer **main**-Methode.
- Deklarieren und initialisieren Sie in der **main**-Methode folgende Objekte:
 - Drei Fahrzeuge mit den Namen „bike“, „plane“, „spaceship“, der Reifenzahl 2, 3 und 0 sowie der Treibstoffart „feet“, „gasoline“ und „deuterium“
 - Drei Autos, zwei davon vom Hersteller „Ford“ und den Modellen „Fiesta“ und „Model T“ und einen „Toyota Aygo“. Denken Sie sich interessante Werte für die Pferdestärke und die Variablenbezeichner aus.
- Rufen Sie die **print()**-Methoden der Objekte auf.
- Lassen Sie die Leistung der Motoren in Watt berechnen und geben Sie diese aus.

Ergänzende Aufgaben

Aufgabe 9.6: Vergleichende Methoden

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

Ergänzen Sie die beiden Klassen **Vehicle** und **Car** um jeweils eine öffentliche Methoden mit dem Namen **compare** und dem Rückgabety **int**. Die Methoden sollen jeweils ein Objekt des gleichen Typs entgegennehmen und das Objekt, auf dem die **compare**-Methode aufgerufen wurde mit dem übergebenen Objekt vergleichen:

- Das Vergleichskriterium für Vehikel ist ihre Reifenzahl.
- Das Vergleichskriterium für Autos ist ihre Motorleistung.
- Die Methoden sollen einen Wert größer 0 zurückgeben, wenn das aufrufende Objekt mehr Reifen bzw. Leistung hat, als das andere.
- Die Methoden sollen einen Wert kleiner 0 zurückgeben, wenn das aufrufende Objekt weniger Reifen bzw. Leistung hat als das andere.
- Die Methoden sollen 0 zurückgeben, wenn die Objekte gleich viele Reifen bzw. Leistung haben.