

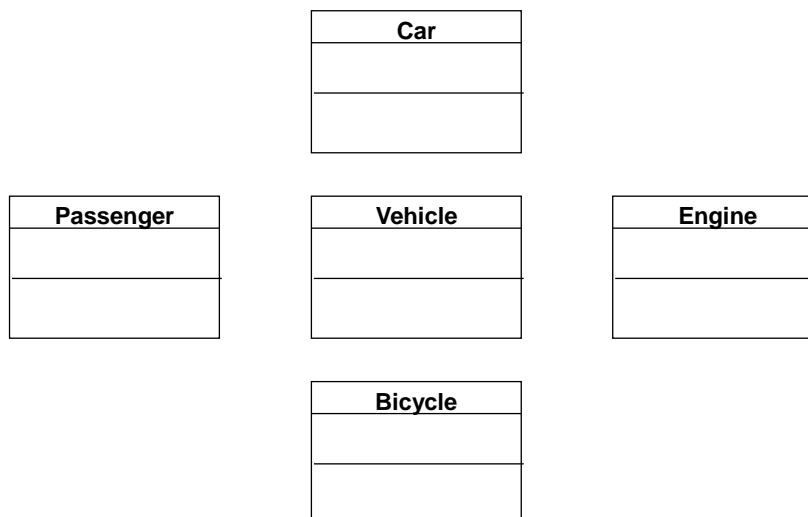
Arbeitsblatt (24.11.2015)

Klassendiagramme

Aufgabe 1: Code „Vehicles“

Bilden Sie 2er Gruppen, 10 Minuten Zeit.

1. Betrachten Sie den Code auf Seite 3 und das folgende unvollständige Klassendiagramm.



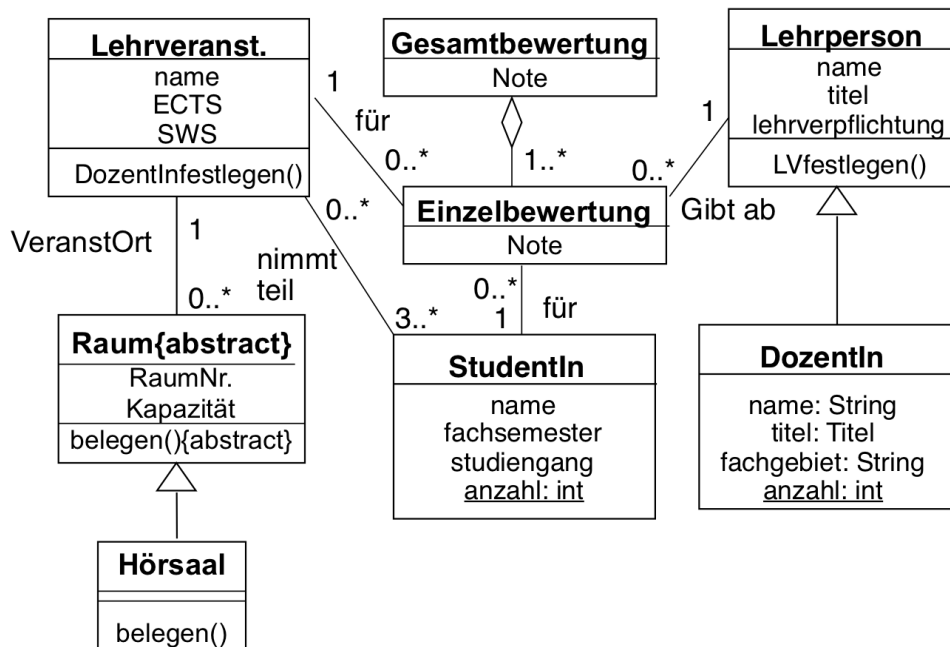
Ergänzen Sie entsprechend zum Code Attribute und Operationen für die Klasse Vehicle

2. Ergänzen Sie entsprechend zum Code so viele Beziehungen wie möglich und begründen Sie, warum Sie jeweils Assoziation/Aggregation oder Komposition gewählt haben.

Aufgabe 2: Klassendiagramm erweitern

Bilden Sie 2er Gruppen, 10 Minuten Zeit.

Betrachten Sie das folgende Klassendiagramm.



- Gegeben sei für jede (nicht abstrakte) Klasse X ein Objekt mit dem Namen ObjektX. Ergänzen Sie für die folgenden Assoziationen

- Lehrveranstaltung – Hörsaal
- Lehrveranstaltung – StudentIn
- Einzelsbewertung - DozentIn

Beziehungen zwischen den Objekten und ggf. auch weitere Objekte und Beziehungen so dass

- die Multiplizitäten erfüllt sind
- die Multiplizitäten verletzt ist.

- Beschreiben Sie für die oben genannten Assoziationen die Möglichkeiten für die Implementierung der Beziehungen zwischen den Klassen in der Programmiersprache Java.
- Führen Sie ÜbungsleiterIn als eine weitere Lehrperson ein.
- Führen Sie ein Abschlusszeugnis und passende Beziehungen ein.

Quellcode für Aufgabe 1)

```

1  package de.hd.uni.isw.vehicles;
2
3  import java.util.ArrayList;
4
5  abstract class Vehicle {
6      private int wheelCount;
7      private int speed;
8
9      public Vehicle(int pWheels, int pSpeed) {
10         this.wheelCount = pWheels;
11         this.speed = pSpeed;
12     }
13
14     public int getWheelCount() {
15         return wheelCount;
16     }
17
18     public int getSpeed() {
19         return speed;
20     }
21 }
22
23 class Passenger {
24     private String name;
25
26     public Passenger(String pName) {
27         this.name = pName;
28     }
29
30     public String getName() {
31         return name;
32     }
33
34     public void setName(String pName) {
35         this.name = pName;
36     }
37 }
38
39 class Engine {
40     private boolean isRunning = false;
41
42     public void startEngine() {
43         isRunning = true;
44     }
45
46     public void stopEngine() {
47         isRunning = false;
48     }
49
50     public boolean isEngineRunning() {
51         return isRunning;
52     }
53 }
54
55
56
57
58 class Car extends Vehicle {
59     private Engine engine;
60     private ArrayList<Passenger> passengers;
61     private int passengerCapacity;
62
63     public Car(int pWheels, int pSpeed,
64         int pCapacity) {
65         super(pWheels, pSpeed);
66         engine = new Engine();
67         passengers = new
68             ArrayList<Passenger>();
69         passengerCapacity = pCapacity;
70     }
71     public int getPassengerCapacity() {
72         return passengerCapacity;
73     }
74
75     public boolean addPassenger(Passenger p)
76     {
77         if (passengers.size() <
78             passengerCapacity) {
79             passengers.add(p);
80             return true;
81         } else {
82             return false;
83         }
84     }
85
86     public boolean removePassenger(
87         Passenger p) {
88         return passengers.remove(p);
89     }
90
91     public void startCar() {
92         engine.startEngine();
93     }
94
95     public void stopCar() {
96         engine.stopEngine();
97     }
98 }
99
100 class Bicycle extends Vehicle {
101     private Passenger p = null;
102
103     public Bicycle(int pWheels, int pSpeed) {
104         super(pWheels, pSpeed);
105     }
106
107     public void setPassenger(Passenger p) {
108         this.p = p;
109     }
110
111     public Passenger getPassenger() {
112         return p;
113     }
114 }
115
116 public class ExampleVehicles {
117     public static void main(String[] args) {
118         Car car = new Car(4, 250, 5);
119         Bicycle bike = new Bicycle(2, 40);
120         Passenger leonard = new Passenger("Leonard");
121         Passenger sheldon = new Passenger("Sheldon");
122         Passenger penny = new Passenger("Penny");
123         car.addPassenger(leonard);
124         car.addPassenger(sheldon);
125         bike.setPassenger(penny);
126     }
127 }

```