

## Bereich: Vererbung (1)

### Fahrzeuge, Wettrennen

**Package:**

`de.dhbwka.java.exercise.classes.vehicles`

**Klasse:** `Race`

**Aufgabenstellung:**

Für die Simulation eines Wettrennens sollen verschiedene Fahrzeugarten objektorientiert modelliert werden.

Da alle Fahrzeugtypen gemeinsame Eigenschaften haben, definieren wir uns zunächst eine Basisklasse `Vehicle`.

Ein `Vehicle` hat folgende allgemeinen Eigenschaften bzw. Verhalten:

- Die Anzahl der Räder. In der Klasse `Vehicle` soll diese auf 0 voreingestellt sein.
- Die Maximalgeschwindigkeit  $v_{max}$ .  
Für ein Objekt der Klasse `Vehicle` soll sie auf 0 km/h voreingestellt sein.
- Die aktuelle Position (in km und der Einfachheit halber in nur einer Dimension).
- Die aktuelle Geschwindigkeit (in km/h).
- Man kann die Geschwindigkeit per Methode `setSpeed(double speed)` setzen.  
Die Geschwindigkeit darf die Maximalgeschwindigkeit nicht überschreiten, ggf. wird sie auf die Maximalgeschwindigkeit reduziert.
- Ein `Vehicle` kann bewegt werden mit der Methode `drive(double minutes)`.  
Diese Methode wird mit einem `double`-Parameter aufgerufen, der die Anzahl der Minuten angibt, die sich das Fahrzeug mit der aktuellen Geschwindigkeit vorwärts bewegt. Der Methodenaufruf ändert natürlich die Position des Fahrzeugs, wenn es mit einer von 0 verschiedenen Geschwindigkeit bewegt wird.
- Die Methode `toString()` gibt in Form einer Zeichenkette umfassend Auskunft über ein Fahrzeug (inkl. über dessen aktuelle Position).

a) Schreiben Sie die Klasse `Vehicle`, indem Sie geeignete Attribute, Konstruktoren und Methoden definieren!

b) Anschließend sollen einige konkrete Fahrzeuge definiert werden, indem entsprechende Klassen von `Vehicle` abgeleitet werden:

- Ein Fahrrad (`Bicycle`) ist ein Fahrzeug mit 2 Rädern und  $v_{max}$  30 km/h.
- Ein Auto (`Car`) ist ein Fahrzeug mit 4 Rädern und  $v_{max}$  140 km/h.
- Ein Rennwagen (`RacingCar`) ist ein Auto mit  $v_{max}$  220 km/h.
- Ein Krankenwagen (`Ambulance`) ist ein Auto mit einem zusätzlichen Blaulicht, das ein- oder ausgeschaltet sein kann (neues Attribut!). Außerdem muss der Krankenwagen Methoden zum Ein- bzw. Ausschalten des Blaulichts anbieten.

Definieren Sie diese Klassen und nutzen Sie dabei so weit wie möglich die Vererbung von Eigenschaften und Verhalten aus!

*(Fortsetzung s. nächste Seite)*

c) Nun soll die eigentliche Simulation des Wettrennens erfolgen. Die Geschwindigkeiten sollen wie folgt gesetzt werden:

- Fahrrad: 20 km/h
- Auto: 150 km/h
- Rennwagen: 200 km/h
- Krankenwagen: 80 km/h

Dann sollen sich die Fahrzeuge bewegen.

Das Fahrrad erhält einen Vorsprung von 4 Stunden.

Danach sollen alle Fahrzeuge eine Stunde lang mit unveränderter Geschwindigkeit vorwärts fahren. Abschließend sollen die aktuellen Positionen ausgegeben werden.

Implementieren Sie hierzu eine Klasse `Race`, in der exemplarisch je ein Fahrzeug jedes Typs mit o.g. Geschwindigkeit erzeugt sowie die Fahrzeuge wie angegeben bewegt werden. Die Klasse `Race` soll zur Simulation des o.g. Wettrennens folgende `main`-Methode haben:

```
public static void main(String[] args) {
    Vehicle[] vehicles = new Vehicle[4];
    vehicles[0] = new Bicycle(20.0);
    vehicles[1] = new Car(150.0);
    vehicles[2] = new RacingCar(200.0);
    vehicles[3] = new Ambulance(80.0, true);
    // 4 hours lead for the bike
    vehicles[0].drive(240.0);
    // 1 hour of driving for everyone
    for (int i = 0; i < vehicles.length; i++) {
        vehicles[i].drive(60);
    }
    // Output Race
    for (int i = 0; i < vehicles.length; i++) {
        System.out.println(vehicles[i].toString());
    }
}
```

Ausgabe etwa folgendermaßen:

Bicycle at position 100.0 km with 2 wheels at speed 20.0 km/h of max. 30.0 km/h.

Car at position 140.0 km with 4 wheels at speed 140.0 km/h of max. 140.0 km/h.

RacingCar at position 200.0 km with 4 wheels at speed 200.0 km/h of max. 220.0 km/h.

Ambulance at position 80.0 km with 4 wheels at speed 80.0 km/h of max. 140.0 km/h. Signal on.

*Hinweis:*

Den Namen der Klasse eines Objekts kann man folgendermaßen ermitteln:

```
this.getClass().getSimpleName()
```

*Danke an Herrn Dr. Mark Minas von der Uni Erlangen für die Anregung zu dieser Aufgabenstellung! ☺*