

e) Grundlagen sind der Klassenentwurf im Entwurfsdiagramm und dessen Überführung in ein Implementationsdiagramm. Dabei müssen Datentypen programmiersprachenspezifisch geändert, Rückgaben und Parameter eingeführt, get- und set-Methoden gesetzt und der Konstruktor entwickelt werden. Parallel muss eine Klassendokumentation erstellt werden. Hilfreich für die Implementation von Objektkommunikation sind Sequenzdiagramme.

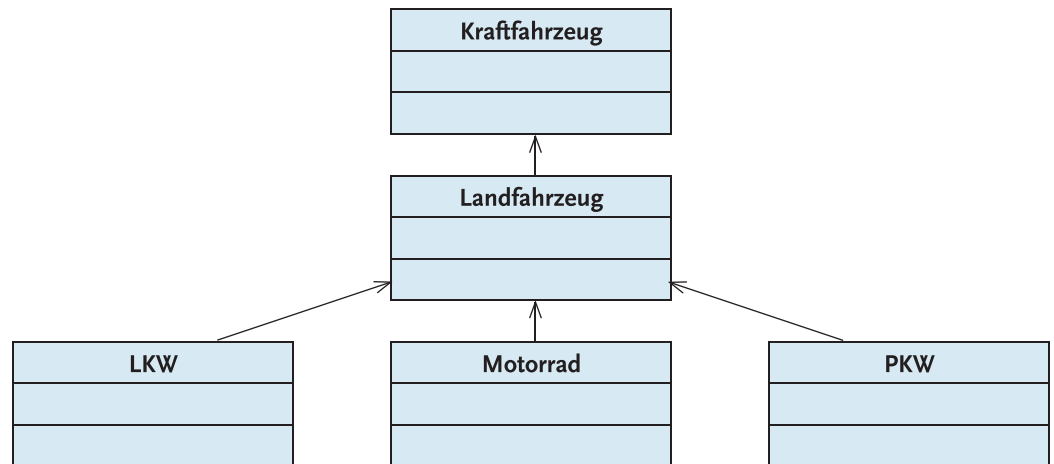
f) Siehe die Programmierung im Anhang.

## 5.5 Vererbung

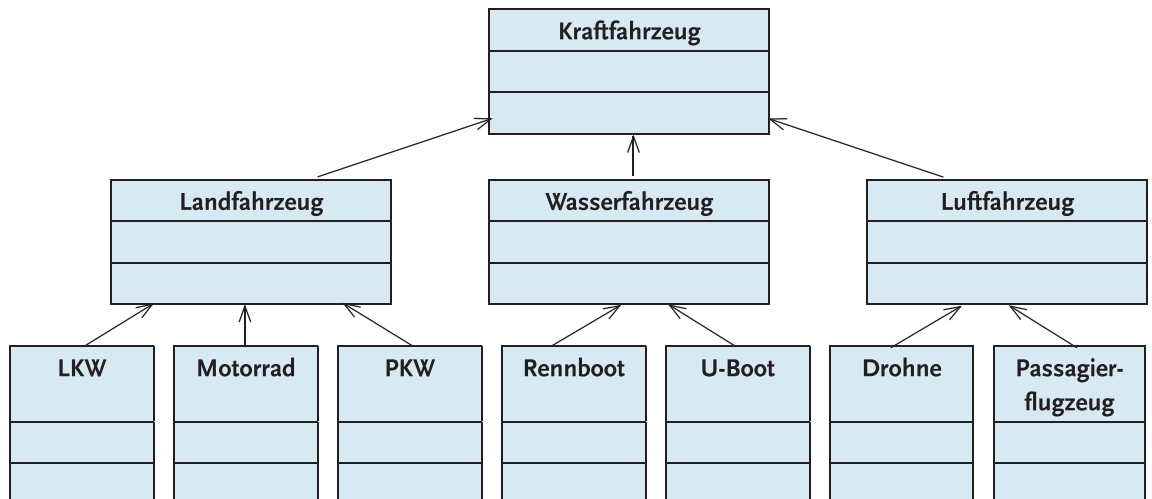
SB S. 122

1. a) Bei der Vererbung werden mindestens zwei Klassen in eine hierarchische Beziehung gesetzt. Es gibt eine Oberklasse und eine oder mehrere Unterklassen, die wiederum Unterklassen besitzen können. Betrachtet man die Vererbungsstruktur von der Oberklasse zu den Unterklassen, spricht man davon, dass die Unterklassen die Oberklasse spezialisieren. Betrachtet man die Vererbungsstruktur von den Unterklassen zur Oberklasse, so bedeutet dies, dass die Oberklasse ihre gemeinsamen Unterklassen generalisiert. Das Besondere an der Vererbung ist, dass alle Unterklassen (auch über mehrere Ebenen hinweg) die Attribute und Methoden der Oberklasse besitzen, ohne dass diese erneut modelliert oder implementiert werden müssen.

- b) i) Je nach Interpretation der vorgegebenen Klassen und Anwendungsfälle können auch andere Vererbungshierarchien sinnvoll sein. Sowohl bei i) als auch bei ii) können auch Attribute und Methoden eingefügt werden.



- ii) Es sind sowohl mehrere Klassen auf jeder Ebene als auch noch weitere Ebenen möglich.



- iii) Eine Vererbungshierarchie ist in Gänze übersichtlicher als die Modellierung in separaten, voneinander unabhängigen Klassen. Die Zuordnung von Klassen zu Oberklassen ermöglicht die Übernahme von Attributen und Methoden, ohne dass diese erneut modelliert werden müssen. Vererbungshierarchien sind i. d. R. zudem gut für Erweiterungen geeignet, wie sie von i) zu ii) stattfindet.

iv)

```

public class Luftfahrzeug {
    int maxHoehe;

    public Luftfahrzeug(int pMaxH) {
        maxHoehe = pMaxH;
    }

    public void fliegen()
    { ... }
}
  
```

```
public class Drohne extends Luftfahrzeug {
    double reichweite;

    public Drohne(int pMaxH, double pReichw) {
        super(pMaxH);
        reichweite = pReichw;
    }
}
```

```
public class Passagierflugzeug extends Luftfahrzeug {
    int plaetze;

    public Drohne(int pMaxH, int pPlaetze) {
        super(pMaxH);
        plaetze = pPlaetze;
    }
    public void durchsage(String pMessage)
    { ... }
}
```

2. siehe Anhang

## 5.6 Prüfungsvorbereitung

SB S. 123

### Aufgabe 1

a)

Elefant
koerperhoehe : Zahl
koerperlaenge : Zahl
ruessellaenge : Zahl
ohrengroesse : Zahl
farbe : Text
beinlaenge : Zahl
gewicht : Zahl
alter : Zahl
...

Für die jeweiligen Attribute lassen sich anschauliche Exemplare finden.

Weitere Attribute sind möglich.

b) Im Prinzip müssen alle Attribute per Parameter initialisiert werden. Es können jedoch Attribute – wie *anzahl Beine* u. Ä. – vorkommen, deren Wert dann durch eine Zuweisung eines festen Werts belegt würde.

