

# JAVA-ÜBUNGSAUFGABEN POLYMORPHIE

Bemerkungen:

B1) Jede der folgenden Aufgabenteile enthält bzw. ergänzt die vorhergehenden, bzw. baut auf diesen auf. Erstellen (bzw. ergänzen) Sie zu jedem Aufgabenteil ein UML-Diagramm.

B2) Aufgaben ohne Interfaces lösen !

1)

Eine **Henne** bzw. eine **Kuh** ist ein **Nutztier**. Jede Henne bzw. Kuh, hat einen Tierwert, der wie folgt bestimmt wird:

Tierwert einer Kuh = 100 \* Milchleistung

Tierwert einer Henne = 2 \* Legeleistung

Implementieren Sie dies mit Hilfe einer polymorphen Methode.

2)

Wenn man ein Konsolenprogramm entwickelt wird (im Gegensatz zu einer grafischen Oberfläche) die Ausgabe auf der Konsole ausgegeben.

Wenn man also mit einem Konsolenprogramm arbeitet, ist es - zu Testzwecken - sinnvoll die Werte aller Attribute einer Klasse auf der Konsole auszugeben (arbeitet man mit einer grafischen Oberfläche ist dies sinnlos, da es dort keine Konsole gibt).

Dies wird durch die Methode **printAllAttributs(...)** gemacht, die in jeder Klasse implementiert (auch in der Klasse Auto, die hier später noch benötigt wird) werden muß

Erstellen Sie dazu vorher die gemeinsame Oberklasse "**DruckObjekt**".

3)

a) Erstellen Sie die Klasse "**Bildschirm**" mit der Klassenmethode **printAll(...)**, die von einem Array von beliebigen Objekten (wie z.B. Kuh, Henne, Auto, usw.) die Werte der jeweiligen Attribute auf dem Bildschirm ausgibt.

b) Erstellen Sie in main() einige Kühe, Hennen und Autos.

Geben Sie mit Hilfe der Methode printAll(...) die Attribute dieser Objekte auf dem Bildschirm aus.

4)

a) Erstellen Sie die Klasse "**Felder**" mit den folgenden Methoden (Klasse "Vergleicher" wird gleich weiter unten erklärt):

```
public static Vergleich berechneMaximum(Vergleicher[] feld){...}
```

```
public static int berechneIndexMaximum (Vergleicher[] feld){...}
```

die von einem Array von Objekten des gleichen Datentyps das Maximum (bzw. den Index des Maximums) bzgl. eines bestimmten Kriteriums berechnet.

Beispiele für eventuelle Aufrufe:

```
berechneMaximum(hennen);
```

```
berechneMaximum(kuehe);
```

```
berechneMaximum(autos);
```

Dieses Kriterium erhält man dadurch, daß in der jeweiligen Klasse (bzgl. der das Maximum berechnet werden soll), die Methode vergleiche(...) implementiert werden muss, mit der z.B. 2 Exemplare (wie z.B. jeweils 2 Hennen, jeweils 2 Kühe, usw.) verglichen werden können und die man dazu benutzt das Maximum zu berechnen. Das erreicht man z.B. dadurch, dass diesen Klassen (die die Methode vergleiche(...) enthalten), die gemeinsame Oberklasse "**Vergleicher**" bekommen.

b) Erstellen Sie in main() einige Autos.

b1) Berechnen Sie mit Hilfe der Methode berechneMaximum(autos) bzw. berechneIndexMaximum(autos) das Objekt mit dem Maximum bzw. den Index dieses Objekts mit dem Maximum.

b2) Geben Sie mit Hilfe der Methode System.out.println(...) die Attribute dieses Objekte auf dem Bildschirm aus.

Geben Sie mit Hilfe der Methode berechneMaximum(...) das Maximum dieser Autos auf dem Bildschirm aus.

5)

Beurteilen Sie das Design Ihrer Lösung (Interfaces dürfen nicht verwendet werden!).

5.1) Aufgabe 2 (ohne Interfaces lösen!)

Ergänzen Sie die vorige Aufgabe mit eigenen Ideen (zur Polymorphie) oder erstellen Sie eine eigene Aufgabe, in der die Polymorphie verwendet wird.

Weitere Ideen:

Gemeinsamkeit von

Felder, Dateien und Tabelle einer DB durch Polymorphie modellieren !