# Übung 1 Auto-Simulation

1. Schreibe eine Klasse **Auto**, die folgende Stammdaten verwaltet:
   1. *Modell*
   2. *Höchtsgeschwindigkeit*
   3. *Kilometerstand*
   4. *Tachometer*

Beim Anlegen einer Instanz der Klasse werden

die Merkmale *Modell* und *Höchtsgeschwindigkeit* mitangegeben.

Die übrigen Attribute sollen nachträglich geändert werden können.

Die Klasse soll eine Methode zur Anzeige der aktuellen Werte besitzen. Diese könnte

etwa so aussehen:

Modell .................. Porsche 911

Höchstgeschwindigkeit ... 210 km/h

Kilometerstand .......... 56207 km

Geschwindigkeit ......... 0 km/h

1. Eine Methode „geschwindigkeitÄndern“ soll die Geschwindigkeit um einen mitangegebenen

Wert verändern, falls dies möglich ist. In der Konsole erscheint eine Bestätigungsmeldung.

Die Geschwindigkeit wurde auf 39 km/h erhöht.

1. Eine Methode „fahren“ welche eine Fahrt für einen gegebenen Zeitraum (in Stunden) mit der aktuellen Geschwindigkeit simuliert.

Das Auto fährt 0.3 h.

Bei der aktuellen Geschwindigkeit von 39 km/h

wird dabei eine Strecke von 13 km zurückgelegt.

Der Tachometerstand beträgt jetzt 56220 km.

# Übung 2 Kermit

**Beschreibung:**

Kermit der Frosch ist der Reportertyp schlechthin. Außerdem beschäftigt

er sich liebend gerne mit Formen und Farben.

Mit BlueJ soll eine Klasse **Kermit**erstellt werden, die die unterschiedlichen

Abfragemöglichkeiten in Java zeigt.

1. einseitige / zweiseitige Abfrage (IF - THEN - ELSE)

if (x<10) {

...

}

else {

...

}

1. Mehrfachabfrage ( CASE )

switch (x) {

case 1 : ... break;

case 2, case 3 : ... break;

default : ...

}

1. Vergleich numerischer Werte und einzelner Zeichen:

== < <= > >= !=

1. Vergleich von Strings:

vorname.equals("Anna")

vorname.equalsIgnoreCase("aNnA")

1. komplexe Vergleiche mit logischen Verknüpfungen

&& (UND)

|| (ODER)

1. Die Klasse Kermit dient Kermit dem Frosch dazu, die Passanten, die er interviewen

möchte, richtig anzusprechen. Er erkennt das Geschlecht und vermutet

ein Alter und leitet daraus eine entsprechende Ansprache ab. Eine ältere ca.

60-jährige Dame könnte er etwa mit Gnädige Dame ansprechen. Schreibe eine

entsprechende Methode gibAnrede(int alter, boolean weiblich).

1. Kermit erhält von seinen Verehrern viele Geschenke. Er möchte ein Programm

schreiben, mit dem er sich bei den Verehrern für das Geschenk bedankt. Am

häufigsten erhält er Murmeln, Hütchen und Badeentchen. In diesem Fall bedankt

er sich ganz besonders herzlich bei seinen Verehrern. Schreibe eine Methode

bedanken, die als Argument das Geschenk (als String) mitbekommt und einen

passenden Dankes-String zurückliefert.

1. Kermit erhält den Auftrag, einen Skelettfund journalistisch zu bearbeiten. Da

er dem Arzt nicht traut, macht er sich sachkundig und findet, dass man aus der

Länge der Knochen auf die Körpergröße schließen kann. Für Männer ergibt sich

als statistischer Mittelwert die Körpergröße gemäß folgender Formel

h = 69.089 + 2.238 \* Oberschenkellänge (in cm).

Ab dem 30. Lebensjahr nimmt die Körpergröße um 0, 06 cm pro Jahr ab.

Am Fundort mißt Kermit die Länge des Oberschenkels, das Alter erhält er vom

Pathologen. Die Methode berechneGroesse liefert ihm schließlich die Körpergröße.

# Übung 3 Graf Zahl

**Beschreibung:**

Bekanntlich ist Graf Zahl der Zahlenfresser in der Sesamstraße.

Der *Count of Count* wie er im englischen heißt, zählt liebend gerne

Dinge vor und rückwärts in allen Variationen.

Mit BlueJsoll eine Klasse *GrafZahl* erstellt werden, die die unterschiedlichen

Wiederholungsanweisungen von Java enthält:

1. Zählende Widerholungsanweisung (FOR-Schleife )

Java-Syntax: for (int i=0; i<10; i++) { ... }

1. vorprüfende Wiederholungsanweisung (WHILE-Schleife)

Java-Syntax: while (i<10) { .. }

1. nachprüfende Wiederholungsanweisung (DO..WHILE-Schleife)

Java-Syntax: do { .. } while (i<10)

Bei der vor- und nachprüfenden Schleife muss darauf geachtet werden, dass der Zähler

**i** im Wiederholungsblock so verändert wird, dass die Abbruchbedingung tatsächlich

auch eintreten kann (Gefahr einer Endlosschleife!).

Implementiere die folgenden Aufgaben und teste jeweils, ob *GrafZahl* die Aufgabe

richtig macht.

1. Erstelle die Klasse GrafZahl und implementiere je eine Methode für die unterschiedliche Zählvariante. (zaehlSchleife(), vorpruefendeSchleife(), nachpruefendeSchleife())

Dabei soll zunächst von 0 bis 9 gezählt werden.

1. Nun soll ein Attribut *max* eingeführt werden, mit dem die Obergrenze, bis zu der hochgezählt wird, erfasst wird. Zusätzlich soll eine Methode *setzeMax()* implementiert werden, mit der das Attribut *max* verändert werden kann.
2. Ergänze den Quelltext so, dass auch der Anfangswert geändert werden kann.
3. Zum Schluss soll die Schrittweite, mit der gezählt wird, veränderbar sein.
4. In Extase gerät Graf Zahl, wenn er bestimmte Zahlenreihen hochzählen kann.

So liebt er insbesondere die Zählung der 4711-er Reihe.

1 x 4711 = 4711

2 x 4711 = 9422

…

Lege hierfür noch eine Methode zaehleReihe() an, welche eine Zahl geliefert bekommt deren Reihe berechnet werden soll.

# Übung 4 Caesar

In der ownCloud findet ihr ein BlueJ-Projekt „Caesar“ hierin befindet sich eine Klasse Services welche unausgefüllte Methodenrümpfe beinhält.

1. Strukturiert die Methode enthaeltVokal so um das sie nur noch ein Return-Statement enthaelt, nutzt auch die Hilfsmethode „istVokal“.
2. Implementiert die Methode „caesarVerschluesseln“ anhand des Methodenkommentars.
3. Erweitert nun die Methode „caesarVerschluesseln“ um einen weiteren Parameter der die Verschiebung im Alphabet als numerischen Wert gegeben bekommt.