

Praktikum Programmier Technik (Technische Informatik)

SS 2015, Hochschule für Angewandte Wissenschaften (HAW), Hamburg
Prof. Dr. Philipp Jenke, Prof. Dr. Axel Schmolitzky, Prof. Dr. Michael Schäfers, Norbert Kasperczyk-Borgmann



Für dieses Aufgabenblatt gelten die folgenden Regeln:

- Für jede Aufgabe muss kompilierender und funktionierender Quellcode sowie ein "Kochrezept/Entwurf" (Text), wie in der Vorlesung besprochen, vorgelegt werden.
- Alle Programme für dieses Aufgabenblatt müssen in dem Package `aufgabenblatt2` liegen.

Aufgabenblatt 2: Zahlen und Variablen

Aufgabe 2.1: Satellitenzeit

Ein Satellit funkt Zeitspannen als "Anzahl Sekunden" zur Erde. Schreiben Sie ein Programm Satellitenzeit, das einen ganzzahligen Sekundenbetrag einliest und die Zeitspanne in der Form

`<d> Tage <h>:<m>:<s>`

wieder ausgibt, wobei gilt:

- `<d>` = Anzahl Tage,
- `<h>` = Anzahl Stunden im Bereich 0 bis 23,
- `<m>` = Anzahl Minuten im Bereich 0 bis 59,
- `<s>` = Anzahl Sekunden im Bereich 0 bis 59.

Erstellen Sie einen Entwurf, bevor Sie den Sourcecode entwickeln.

Testen Sie das Programm beispielsweise mit dem Eingabewert 10000. Die korrekte Ausgabe ist

0 Tage 2:46:40,

denn 0 Tage + 2 Stunden + 46 Minuten + 40 Sekunden = $(0 \cdot 86400 + 2 \cdot 3600 + 46 \cdot 60 + 40)$
Sekunden = 10000 Sekunden

Aufgabe 2.2: Body Mass Index

Schreiben Sie ein Programm BMIIndex, das Ihren Body Mass Index (BMI) berechnet (siehe <http://de.wikipedia.org/wiki/Body-Mass-Index>). Zur Berechnung des BMI lassen Sie den Benutzer die Körpergröße (in Metern) und das Körpergewicht (in Kilogramm) eingeben. Der BMI ist dann

$$\text{Körpergewicht} / \text{Körpergröße}^2$$

Wie erreichen Sie durch explizite Typkonversion (Typcasting) eine Ausgabe des BMI-Wertes ohne Kommastellen?

Aufgabe 2.3: Lottozahlen

Die Ziehung der Lottozahlen soll digitalisiert werden. Schreiben Sie ein Programm Lotto, das sechs zufällige „Lottozahlen“ (6 aus 49) ermittelt und als ganze Zahlen in einer Zeile ausgibt. Anders als im realen Leben dürfen sich bei Ihrem Lottozahlenprogramm Zahlen wiederholen, d.h. für jede Zahl stehen alle Möglichkeiten von 1 bis 49 zur Verfügung! Eine Sortierung ist nicht notwendig.

Tipp: Verwenden Sie die Bibliotheksmethode `Math.random()`!

Aufgabe 2.4: Fibonacci-Zahlen

Die Fibonacci-Reihe ist eine unendliche Folge von ganzen positiven Zahlen (siehe <http://de.wikipedia.org/wiki/Fibonacci-Folge>):

$$f_0, f_1, f_2, \dots$$

Die ersten beiden Zahlen sind:

$$\begin{aligned} f_0 &= 0 \\ f_1 &= 1 \end{aligned}$$

Jede weitere Zahl ist die Summe der beiden Vorgängerzahlen:

$$f_n = f_{n-2} + f_{n-1}, \text{ für } n \geq 2$$

Der Anfang der Fibonacci-Reihe lautet also:

$$0, 1, 1, 2, 3, 5, 8, \dots$$

Schreiben Sie ein Programm Fibonacci-Zahlen, das vom Anwender abfragt, die wievielte Fibonacci-Zahl berechnet werden soll, die zugehörige Zahl berechnet und diese ausgibt. Die Anwender-Eingabe soll in einer Variablen mit Namen `eingabe` vom geeigneten Typ abgelegt werden. Erstellen Sie zunächst wieder einen Entwurf, bevor Sie den Quellcode entwickeln. Gehen Sie davon aus, dass der Anwender nur positive ganzen Zahlen ≥ 2 eingibt.

Beispiel: Anwender-Eingabe: 4, Ausgabe: 3

Hinweis: Erinnern Sie sich an das Beispielprogramm `Summe` aus der Vorlesung.