

Programmieren und Software-Engineering I

Übung 10

Name: _____ Klasse: _____ Datum: _____

Lernziele:

- Erstellen von einfachen Java-Programmen (mit Verzweigungen und Schleifen)
- Testdokumentation
- Geschachtelte Schleifen
- Zufallszahlen

Aufgabe 1: Anrede (Ex_10_01_Address)

Erstelle ein Java-Programm, das abhängig von der Uhrzeit (Stunde) und des Geschlechts (m, w, d) eine korrekte Anrede formuliert. Die Anrede beginnt je nach Tageszeit mit „Guten Morgen“ (0 – 10 Uhr), „Guten Tag“ (11 – 17 Uhr) oder „Guten Abend“ (18 – 23) Uhr).

Beispiel: Guten Morgen Herr Toni Maier!
 Guten Tag Frau Susi Sonnenschein!
 Guten Abend Caroli Corallo!

Code-Beispiel: int hour = 19; // Tageszeit 0-23 (ansonsten ungültig)
 char sex = 'm'; // Geschlecht
 String name = "Toni Maier"; // Vorname und Nachname

Aufgabe 2: Abrechnung (Ex_10_02_Salary)

Herr und Frau Clever sind Politiker. Sie haben je ein Grundeinkommen von 100000 Euro. Dazu kommt eine Abgeordnetenpauschale von 40000 Euro für Herrn Clever und eine von 50000 für Frau Clever. Zusätzlich bekommen sie für jede Rede 500 Euro (ab der 6. Rede 700 Euro) und für jede Stunde Sekretariatsarbeit 200 Euro. Schreibe ein Java-Programm, das aus den ausgeführten Reden und Arbeiten das Einkommen von Herrn und Frau Clever berechnet.

Beispiel: speechesMr = 7;
 hSecMr = 10;
 speechesMrs = 4;
 hSecMrs = 15;

Ausgabe: Das gemeinsame Gehalt von

Testdokumentation

Beschreibung wie das Programm getestet wurde.

Testfall	Mr	sec	Mrs	sec	Ausgabe erwartet	Ausgabe tatsächlich

Aufgabe 3: Würfel (Ex_10_03_Dice)

Erstelle ein Java-Programm, welches das das Würfeln simuliert. Es wird dabei solange gewürfelt, bis zweimal hintereinander!!! dieselbe Augenzahl erscheint.

Beispiel: 2 4 1 3 1 4 1 5 6 3 1 1
 Es wurde 12x gewürfelt.

Programmieren und Software-Engineering I

Übung 10

Aufgabe 4: Zahl erraten (Ex_10_04_GuessNumber)

Erstelle ein Java-Programm für folgendes einfache Spiel:

Zu Beginn wird eine Zufallszahl zwischen 1 und 100 festgelegt. Der Benutzer versucht nun, diese Zufallszahl mit maximal 7 Versuchen zu erraten. Nach jedem Tipp bekommt er einen Hinweis, ob die gesuchte Zahl größer oder kleiner ist.

Falls er die Zahl erraten kann, bekommt er eine entsprechende Gewinnmeldung mit der Anzahl der benötigten Versuche. Im anderen Fall bekommt er die Meldung, dass er verloren hat und wie die richtige Zahl gelaute hätte.

<u>Beispiel:</u>	Tipp: 63	Ausgabe: Die gesuchte Zahl ist größer.
	Tipp: 80	Ausgabe: Die gesuchte Zahl ist kleiner.
	Tipp: 74	Ausgabe: Die gesuchte Zahl ist kleiner.
	Tipp: 70	Ausgabe: Die gesuchte Zahl ist größer.
	Tipp: 72	Ausgabe: Gratuliere, Du hast die Zahl mit 5 Versuchen erraten!!!
	Tipp: 40	Ausgabe: Die gesuchte Zahl ist größer.
	Tipp: 60	Ausgabe: Die gesuchte Zahl ist größer.
	...	
	Tipp: 92	Ausgabe: Leider auch beim 7. Versuch falsch. Die gesuchte Zahl wäre 94 gewesen.

Aufgabe 5: Monsun (Ex_10_05_Monsoon)

Klimaforscher haben festgestellt, dass sich in den letzten 15 Jahren auch der Monsunregen wesentlich verändert hat. Waren früher Regenmengen und -tage eher jedes Jahr gleich, so unterscheiden sie sich jetzt deutlich. In Jahren mit wenigen Regentagen regnet es dafür deutlich stärker, so dass es oft kurzzeitig zu Überschwemmungen kommt. In Jahren mit vielen Regentagen ist die Regenmenge dagegen so gering, dass keine wirkliche Durchfeuchtung des Bodens erfolgt.

Erstelle ein Java-Programm, das aufgrund der nachstehenden Tabelle ausgibt, wie die Regenmenge gewesen ist.

Regentage:	Ausgabe
<30	Starke Ueberschwemmungen
30 – 40	Ueberschwemmungsgefahr
41 – 50	Starkregen
51 – 60	Gute Bodendurchfeuchtung
61 – 70	Ausreichende Bodenfeuchte
71 – 90	Regen ohne ausreichende Bodenfeuchte

Beachte auch, dass eine Tagesanzahl < 0 oder > 90 nicht korrekt ist. Entsprechend soll eine Fehlermeldung („Falsche Tagesanzahl“) ausgegeben werden.

WICHTIG

Alle Programme müssen einen Programmkopf (=Beschreibung) enthalten.

z.B.:

```
/**
 * Name: Max Mustermann
 * Hü: 4
 * Bsp: 2
 * Datum: 20.10.2015
 * Dateiname: HUE_04_02_Dreieck.java
 * Beschreibung: Es wird für gegebene Seitenlängen a, b und c geprüft, ob es ein gleichseitiges, ein
 * gleichschenkeliges, ein rechtwinkeliges, ein sonstiges gültiges oder ein ungültiges
 * Dreieck ist.
 */
```

```
public class HUE_04_02_Dreieck {
```