

Programmieren und Software-Engineering I

Übung 9

Name: _____ Klasse: _____ Datum: _____

Lernziele:

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

Aufgabe 1: Kino (Ex_09_01_Cinema)

Erstelle ein Java-Programm, welches einen Kinosaal darstellt (Sitzpositionen). Verwende dazu 2 ineinander geschachtelte Schleifen (Die äußere Schleife ist die Reihe, die innere Schleife der Sitz).

Beispiel:

5 Reihen mit je 4 Sitzen

reihe = 1 / sitz = 1	reihe = 1 / sitz = 2	reihe = 1 / sitz = 3	reihe = 1 / sitz = 4
reihe = 2 / sitz = 1	reihe = 2 / sitz = 2	reihe = 2 / sitz = 3	reihe = 2 / sitz = 4
reihe = 3 / sitz = 1	reihe = 3 / sitz = 2	reihe = 3 / sitz = 3	reihe = 3 / sitz = 4
reihe = 4 / sitz = 1	reihe = 4 / sitz = 2	reihe = 4 / sitz = 3	reihe = 4 / sitz = 4
reihe = 5 / sitz = 1	reihe = 5 / sitz = 2	reihe = 5 / sitz = 3	reihe = 5 / sitz = 4

Aufgabe 2: Abrechnung (Ex_09_02_Cable)

Herr Clever ist Kabelverleger. Am Ende jeder Woche muss er eine Abrechnung durchführen. Dazu gibt er seiner Firma bekannt, wie viele Meter Kabel er verlegt und wie viele Stunden er dafür benötigt hat.

Die Firma errechnet daraus die angefallenen Kosten wie folgt: Eine Kabeltrommel hat 500m Kabel und kostet 2 Euro pro Meter. Angefangene Kabeltrommeln kosten 3 Euro pro Meter.

Herr Clever hat einen Stundensatz von 50 Euro pro Stunde. Hat Herr Clever mehr als 40 Stunden gearbeitet, so gebührt ihm für jede Überstunde der doppelte Stundensatz.

Erstelle ein Java-Programm, welches eine Abrechnung durchführt und in übersichtlicher Form ausdruckt.

Beispiel:

General Output	General Output
<pre> -----Configuratio Verlegte Kabelmeter: 3200 Arbeitszeit in Std.: 47 Kosten: 6 ganze Kabelrollen: 6000 200 angefangene Meter: 600 Arbeitszeit: 2000 Ueberstunden: 700 Gesamtkosten: 9300 Process completed.</pre>	<pre> -----Configuratio Verlegte Kabelmeter: 2757 Arbeitszeit in Std.: 42 Kosten: 5 ganze Kabelrollen: 5000 257 angefangene Meter: 771 Arbeitszeit: 2000 Ueberstunden: 200 Gesamtkosten: 7971 Process completed.</pre>

Testdokumentation

Beschreibung wie das Programm getestet wurde.

Testfall	Kabel	Std	Ausgabe erwartet	Ausgabe tatsächlich

Programmieren und Software-Engineering I

Übung 9

Aufgabe 3: Ausgabe von Zahlen (Ex_09_03_PrintHTO)

Erstelle ein Java-Programm, um dreistellige Zahlen in einem beliebigen Bereich in folgender Form auszugeben:

Beispielausgabe für den Bereich 274 bis 356

```
274: HHZZZZZZZEEEE
275: HHZZZZZZZEEEE
276: HHZZZZZZZEEEE
...
356: HHHZZZZZEEEE
```

Aufgabe 4: Staffelrechnung (HUE_09_04_Tower)

Erstelle ein Java-Programm, welches eine Staffelrechnung durchführt. Eine Staffelrechnung besteht aus einer gleich großen Anzahl von Multiplikationen und Divisionen, wobei die Ausgangszahl, sowie die Anzahl der Berechnungen eingegeben werden. Achte dabei auf den Wertebereich (Datentyp long).

Beispiel:

Ausgangszahl:	7
Berechnungen:	5
7*2=	14
14*3=	42
42*4=	168
168*5=	840
840*6=	5040
5040/2=	2520
2520/3=	840
840/4=	210
210/5=	42
42/6=	7

Aufgabe 5: Perfekte Zahl (HUE_09_05_PerfectNumber)

Erstelle ein Java-Programm, welches alle perfekten Zahlen, bis zu einer Obergrenze findet. Eine Zahl n heißt perfekt, wenn die Summe ihrer echten Teiler gleich n ist.

Beispielausgabe:

6	// 1 + 2 + 3
28	// 1 + 2 + 4 + 7 + 14
...	

Aufgabe 6: Multiplikationstabelle (HUE_09_06_Multiply)

Erstelle ein Java-Programm, welches das kleine 1x1 in Tabellenform ausgibt.

Ausgabe:

*	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	...				
2	2	4	6	8	10	...				
3	3	6	9	12	15	...				
4										
...										

Zusatz:

Ergänze Deine Tabelle unterhalb durch eine Additionstabelle.

```
WICHTIG
Alle Programme müssen einen Programmkopf (=Beschreibung) enthalten.
z.B.:
/*****
*      Name:      Max Mustermann
*      Hüt:       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 {
```