

Programmieren und Software-Engineering I

Übung 12

Name: _____ Klasse: _____ Datum: _____

Aufgabe 0: Tutorial

Schau Dir die wichtigsten Kapitel des Java-Tutorials an:

https://www.youtube.com/watch?v=rv_lmAy6-HE

Aufgabe 1: Figur (L)

Erstelle ein Java-Programm, welches die unten dargestellte Figur ausgibt. Die Größe wird dabei zu Beginn festgelegt. Verwende dabei verschiedene Schleifen.

Beispielausgabe: 1
23
456
7890
12345
67...

Aufgabe 2: Zahlen (L)

Erstelle ein Java-Programm, welches alle durch 3 oder 4 oder 5 teilbaren Zahlen, nicht jedoch jene, die durch 12 oder 15 oder 20 teilbar sind, ausgibt. Die Obergrenze wird zu Beginn festgelegt. Es werden immer genau 8 Zahlen pro Zeile ausgegeben.

Beispielausgabe: 3 4 5 6 8 9 10 16
18 21 ...

Aufgabe 3: Minimum-Maximum (L)

Erstelle ein Java-Programm, welches das Minimum und Maximum, sowie den Durchschnitt einer Reihe von Zufallszahlen (zwischen 1 und 100) ausgibt.

Beispielausgabe: Zufallszahlen: 56 17 87 23 11
Minimum: 11
Maximum: 87
Durchschnitt: 46.375

Aufgabe 4: Zufallszahlen (L)

Erstelle ein Java-Programm, welches solange Zufallszahlen (zwischen 1 und 100) ausgibt, solange die Zahlenfolge aufsteigend ist.

Beispielausgabe: Zufallszahlen: 48 67 73 26
Es wurden genau aufsteigende 3 Zahlen erzeugt.

Aufgabe 5: Figur (L)

Erstelle ein Java-Programm, welches die unten stehende Figur ausgibt. Die Größe wird zu Beginn festgelegt.

Beispielausgabe: *
**

Programmieren und Software-Engineering I

Übung 12

Aufgabe 6: Minimum-Minimum2 (M)

Erstelle ein Java-Programm, welches die kleinste und die zweitkleinste Zahl aus einer Reihe von Zufallszahlen (zwischen 10 und 50) ausgibt.

Beispielausgabe: Zufallszahlen: 34 45 22 15
Kleinste Zahl: 12
Zweitkleinste Zahl: 15

Aufgabe 7: Zahlen (M)

Erstelle ein Java-Programm, welches alle dreistelligen Zahlen ausgibt für die gilt: $\text{zahl} = h! + z! + e!$

Beispielausgabe: 145 (weil $1! + 4! + 5! = 1 + 1*2*3*4 + 1*2*3*4*5 = 1 + 24 + 120 = 145$)

Aufgabe 8: Zahlen (M)

Erstelle ein Java-Programm, welches alle dreistelligen Zahlen ausgibt, die durch ihre Ziffernsumme teilbar sind. Es werden immer genau 10 Zahlen pro Zeile ausgegeben.

Beispielausgabe: 100 102 108 110 111 112 114 117 120 126
132 133 135 140 144 ...

Aufgabe 9: Zahlen (M)

Erstelle ein Java-Programm, welches alle dreistelligen Zahlen ausgibt für die gilt $\text{zahl} = h^3 + z^3 + e^3$ (z.B. $153 = 1^3 + 5^3 + 3^3$). Die Zahlen werden untereinander ausgegeben.

Beispielausgabe: 153
370
371
407

Aufgabe 10: Zahlen (M)

Erstelle ein Java-Programm, alle zweistelligen Zahlen die größer als das 4fache ihrer Ziffernsumme sind. Es werden immer genau 6 Zahlen pro Zeile ausgegeben.

Beispielausgabe: 10 11 20 21 22 23
30 31 32 33...

Aufgabe 11: Fakultät (M)

Erstelle ein Java-Programm, welches die Fakultät für die ersten 15 natürlichen Zahlen berechnet und ausgibt.

Beispielausgabe: $1! = 1 = 1$
 $2! = 1*2 = 2$
 $3! = 1*2*3 = 6$
 $4! = 1*2*3*4 = 24$
...
 $15! = 1*2*3*4*...$

Aufgabe 12: Ziffernsumme (M)

Erstelle ein Java-Programm, welches die Ziffernsumme aller Zahlen von 10 bis 99 berechnet und untereinander ausgibt.

Beispielausgabe: Die Ziffernsumme von 10 = 1
Die Ziffernsumme von 11 = 2
...
Die Ziffernsumme von 99 = 18

Programmieren und Software-Engineering I

Übung 12

Aufgabe 13: Spiegelzahlen (M)

Erstelle ein Java-Programm, welches alle dreiziffrigen Zahlen ausgibt, die von vorne und hinten gelesen gleich sind. Dabei werden immer genau 8 Zahlen in entsprechender Form pro Zeile ausgegeben.

Beispiel:

101	111	121	131	141	151	161	171
181	191	202	212	...			

Aufgabe 14: Figur (M)

Erstelle ein Java-Programm, welches die untenstehende Figur ausgibt. Die Größe wird zu Beginn vom Benutzer eingegeben.

Beispielausgabe:

```
*****
*****
*****
****
***
*
***
****
*****
*****
*****
```

Aufgabe 15: Maximum (L)

Erstelle ein Java-Programm, welches das Maximum von fünf beliebigen Zahlen ausgibt.

Beispielausgabe: Das Maximum der Zahlen 123, 5, 7, 776 und 551 = 776

Aufgabe 16: Sortierung (M)

Erstelle ein Java-Programm, welches drei beliebige Zahlen aufsteigend sortiert ausgibt.

Beispielausgabe:

Unsortiert:	34, 51, 16
Sortiert:	16, 34, 51

Aufgabe 17: Sortierung (S)

Erstelle ein Java-Programm, welches vier beliebige Zahlen aufsteigend sortiert ausgibt.

Beispielausgabe:

Unsortiert:	24, 11, 17, 13
Sortiert:	11, 13, 17, 24

Aufgabe 18: Zeitdifferenz (S)

Erstelle ein Java-Programm, welches die Differenz von zwei Zeiten berechnet. Die Zeiten werden dabei als int-Werte festgelegt. Achte dabei auf Zeiten vor 10:00:00.

Beispiel:

int t1 = 120558;	// 12:05:58
int t2 = 141610;	// 14:16:10

Beispielausgabe:

Zeit 1 :	12:05:58
Zeit 1 :	14:16:10
Differenz :	02:10:12

Beispiel:

int t1 = 605;	// 00:06:05
int t2 = 32010;	// 03:20:10

Beispielausgabe:

Zeit 1 :	00:06:05
Zeit 1 :	03:20:10
Differenz :	03:14:05

Aufgabe 19: Würfel (S)

Erstelle ein Java-Programm, welches das mehrmalige Würfeln mit einem Würfel simuliert. Anschließend wird eine „grafische Statistik“ ausgegeben.

Programmieren und Software-Engineering I

Übung 12

Beispiel: Gesamtwürfe: 100

1: 16x	2: 18x	3: 19x	4: 19x	5: 20x	6: 18x

Aufgabe 20: Primzahlenzwilling (S)

Erstelle ein Java-Programm, welches die ersten 50 Primzahlenzwillinge ausgibt. Zwei Primzahlen p und q sind Primzahlenzwillinge wenn ihre Differenz 2 ist.

Beispielausgabe:

3 und 5
5 und 7
11 und 13
17 und 19
...

Aufgabe 21: Figur (M)

Erstelle ein Java-Programm, welches ein Dreieck mit Sternen ausgibt.

Beispielausgabe:

```

      *
     * *
    *  *
   *   *
  *    *
 *     *
*****
  
```

Aufgabe 22 - 25:

ULAM-FOLGE

Man schreibe ein Java-Programm, das bis zu einer positiven ganzen Zahl die Ulam-Folge bestimmt. Dabei wird, beginnend mit der gegebenen Zahl die jeweils folgende wie folgt berechnet: Ist die vorhergehende Zahl gerade, dividiere durch 2, ist sie ungerade, multipliziere mit 3 und addiere 1. Bei 1 wird die Folge beendet.

Beispiel:
13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

BEFREUNDETE ZAHLEN

Man schreibe ein Java-Programm, das befreundete Zahlen bis zu einer gewissen Grenze bestimmt. Unter einem Paar befreundeter Zahlen versteht man zwei unterschiedliche natürliche Zahlen, von denen wechselseitig jeweils eine Zahl gleich der Summe der echten Teiler der anderen Zahl ist. Die Teiler brauchen dabei nicht ausgegeben zu werden.

Beispiel:
220: $110 + 55 + 44 + 22 + 20 + 11 + 10 + 5 + 4 + 2 + 1 = 284$
284: $142 + 71 + 4 + 2 + 1 = 220$

KAPREKAR-ZAHLEN

Man schreibe ein Java-Programm, das Kaprekar-Zahlen bis zu einer gewissen Grenze bestimmt. Dabei nehme man eine n-stellige natürliche Zahl, bilde ihr Quadrat und addiere die aus den rechten n Stellen gebildete Zahl des Ergebnisses zu der aus den n oder n-1 linken Stellen gebildeten. Ist dieses Ergebnis gleich der ursprünglichen Zahl, so ist dies eine Kaprekar-Zahl.

Beispiel:
 $297^2 = 88209$, $88 + 209 = 297$

ARMSTRONG-ZAHLEN

Man schreibe ein Java-Programm, das Armstrong-Zahlen bis zu einer gewissen Grenze bestimmt. Eine n-ziffrige Armstrong-Zahl hat die Eigenschaft, dass die Summe der n-ten Potenzen ihrer Ziffern gleich der ursprünglichen Zahl ist.

Beispiel:
 $371 = 3^3 + 7^3 + 1^3$