

Programmierkurs Java

Dr.-Ing. Dietrich Boles

Aufgaben zu UE 3 - Einführung

(Stand 22.03.2019)

Aufgabe 1:

Programmieren Sie ein Ratespiel. Das Programm generiert anfangs eine Zufallszahl zwischen 0 und 100. Anschließend soll der Benutzer versuchen, diese Zahl zu erraten. Das Programm gibt jeweils aus, ob die vom Benutzer eingegebene Zahl zu groß, zu klein oder korrekt ist. In letzterem Fall soll das Programm stoppen und die Anzahl an Rateversuchen ausgeben. Nutzen Sie zur Zufallszahlenberechnung folgende Anweisung:

```
int zufallszahl = (int) (Math.random() * 101);
```

Beispielablauf des Programms (Benutzereingaben in <>):

```
Ich habe eine Zufallszahl zwischen 0 und 100 generiert! Bitte errate sie!  
Zahl: <50>  
Zufallszahl ist größer!  
Zahl: <70>  
Zufallszahl ist größer!  
Zahl: <90>  
Zufallszahl ist größer!  
Zahl: <95>  
Zufallszahl ist kleiner!  
Zahl: <92>  
Zufallszahl ist größer!  
Zahl: <93>  
Geschafft! Anzahl an Versuchen = 6
```

Aufgabe 2:

Schreiben Sie ein Programm, das die Quersumme einer eingegebenen nicht negativen Zahl berechnet und ausgibt.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Zahl (>= 0): <38469>  
Quersumme = 30
```

Aufgabe 3:

Schreiben Sie ein Programm, bei dem der Nutzer eine nicht negative Zahl eingibt und das Programm die Spiegelzahl dieser Zahl berechnet, bei der alle Ziffern umgedreht sind, z.B. 5678 → 8765 oder 1020 → 201. Führende Nullen fallen weg.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Zahl (>= 0): <384056>
```

Spiegelzahl = 650483

Aufgabe 4:

Schreiben Sie ein Java-Programm, das zunächst die Eingabe einer positiven Zahl *max* und einer positiven Zahl *div* über die Tastatur erwartet. Anschließend soll das Programm alle Zahlen zwischen 0 und *max*, die sich ohne Rest durch *div* dividieren lassen, auf den Bildschirm ausgeben.

Beispielablauf des Programms (Benutzereingaben in <>):

```
max (>0): <30>
div (>0): <7>
0
7
14
21
28
```

Aufgabe 5:

Ein Musik-Anbieter im Internet habe folgendes Geschäftsmodell: Jeder Nutzer kann sich zunächst 5 Musikstücke kostenlos downloaden. Für die nächsten 30 Stücke muss er jeweils 20 Cent bezahlen und für jeden weiteren Download 15 Cent.

Schreiben Sie ein Java-Programm, das es einem Nutzer ermöglicht, die Anzahl an insgesamt gewünschten Downloads anzugeben und das daraufhin die insgesamt entstehenden Gebühren berechnet und diese auf den Bildschirm ausgibt.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Anzahl Downloads (>=0): <41>
Gebuehren = 6,90 EURO
```

Aufgabe 6:

Schreiben Sie ein Java-Programm, das zunächst die Eingabe einer positiven Zahl *anzahl* über die Tastatur erwartet. Anschließend soll das Programm *anzahl*-mal ein Sternchen (*) auf den Bildschirm ausgeben.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Anzahl (>0): <8>
*****
```

Aufgabe 7:

Schreiben Sie ein Java-Programm, das nach Eingabe einer ungeraden Zahl *basislaenge* ein Sterndreieck auf den Bildschirm ausgibt. Die unterste Reihe soll dabei aus *basislaenge* Sternchen bestehen.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Basislaenge (>0 und ungerade): <9>
 *
 ***
 *****
 *****
```

Aufgabe 8:

In dieser Aufgabe geht es darum, Treppen der folgenden Form auf den Bildschirm zu zeichnen:

```
+--+
| |
+-----+
|       |
+-----+
|       |
+-----+
|       |
```

Schreiben Sie ein Java-Programm, das zunächst vom Benutzer eine positive Zahl erfragt. Dieser gibt die Höhe der Treppe an (in dem Beispiel 4). Das Programm soll daraufhin eine Treppe in der oben skizzierten Form auf den Bildschirm ausgeben.

Aufgabe 9:

Schreiben Sie ein Java-Programm, das die Anzahl der Überträge beim Addieren zweier Zahlen berechnet.

Algorithmus: Zunächst sollen vom Benutzer zwei nicht negative Zahlen mit jeweils weniger als 10 Ziffern eingelesen werden. Anschließend soll die Anzahl an Überträgen beim ziffernweisen Addieren der beiden Zahlen von rechts nach links berechnet werden. Zum Schluss soll die Anzahl an Überträgen auf den Bildschirm ausgegeben werden.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Zahl 1: <123>
Zahl 2: <594>
Anzahl Uebertraege = 1
```

Aufgabe 10:

Ziel 100 ist als ein Strategiespiel für zwei Spieler. Begonnen wird mit einer zufälligen Zahl kleiner 30. Die Spieler addieren abwechselnd eine selbst gewählte Zahl zwischen 1 und 10 zu dieser Zahl. Gewonnen hat der Spieler, der als erster 100 oder mehr erreicht.

Implementieren Sie das Spiel *Ziel 100* in Java, so dass zwei menschliche Spieler gegeneinander antreten können. Beachten und behandeln Sie dabei bitte auch fehlerhafte Benutzereingaben. Nutzen Sie zur Berechnung einer Zufallszahl zwischen 0 und 29 (einschließlich) die folgende Anweisung:

```
int zufallszahl = (int) (Math.random() * 30);
```

Beispielablauf des Programms (Benutzereingaben in <>):

```
Die Zahl ist 29
Spieler 1! Zahl eingeben (>= 1, <= 10):<10>
Die Zahl ist 39
Spieler 2! Zahl eingeben (>= 1, <= 10):<9>
```

```
Die Zahl ist 48
Spieler 1! Zahl eingeben (>= 1, <= 10):<13>
Fehler! Spieler 1! Zahl eingeben (>= 1, <= 10):<8>
Die Zahl ist 56
Spieler 2! Zahl eingeben (>= 1, <= 10):<9>
Die Zahl ist 65
Spieler 1! Zahl eingeben (>= 1, <= 10):<10>
Die Zahl ist 75
Spieler 2! Zahl eingeben (>= 1, <= 10):<4>
Die Zahl ist 79
Spieler 1! Zahl eingeben (>= 1, <= 10):<10>
Die Zahl ist 89
Spieler 2! Zahl eingeben (>= 1, <= 10):<8>
Die Zahl ist 97
Spieler 1! Zahl eingeben (>= 1, <= 10):<3>
Zahl = 100! Spieler 1 hat gewonnen!
```

Aufgabe 11:

Implementieren Sie das folgende kleine Taktikspiel für 2 Spieler.

Anfangs haben beide Spieler je einen Haufen von 100 Kugeln. Gespielt werden 10 Runden. In jeder Runde wählt jeder Spieler (geheim) eine bestimmte Anzahl (≥ 0) an Kugeln aus, die von seinem Haufen entfernt werden. Jeweils der Spieler mit der größeren Anzahl an gewählten Kugeln gewinnt die Runde und einen Punkt. Bei Unentschieden bekommt keiner der Spieler einen Punkt. Das Spiel gewinnt der Spieler, der nach 10 Runden die meisten Punkte hat. Bei Gleichstand endet das Spiel unentschieden.

Orientieren Sie sich bei der Implementierung des Spiels an folgendem Beispielablauf (Benutzereingaben in $\langle \rangle$):

```
Runde 1 von 10 Runden
Spieler 1: Wie viele Kugeln von 100?<10>
Spieler 2: Wie viele Kugeln von 100?<6>
Spieler 1 hat die Runde gewonnen!
Spielstand nach Runde 1:
Spieler 1 hat 1 Punkte und 90 Restkugeln
Spieler 2 hat 0 Punkte und 94 Restkugeln
```

```
Runde 2 von 10 Runden
Spieler 1: Wie viele Kugeln von 90?<20>
Spieler 2: Wie viele Kugeln von 94?<20>
Die Runde endet unentschieden!
Spielstand nach Runde 2:
Spieler 1 hat 1 Punkte und 70 Restkugeln
Spieler 2 hat 0 Punkte und 74 Restkugeln
```

...

```
Runde 9 von 10 Runden
Spieler 1: Wie viele Kugeln von 15?<10>
Spieler 2: Wie viele Kugeln von 30?<9>
Spieler 1 hat die Runde gewonnen!
Spielstand nach Runde 9:
Spieler 1 hat 3 Punkte und 5 Restkugeln
Spieler 2 hat 5 Punkte und 21 Restkugeln
```

```
Runde 10 von 10 Runden
Spieler 1: Wie viele Kugeln von 5?<5>
Spieler 2: Wie viele Kugeln von 21?<21>
```

Spieler 2 hat die Runde gewonnen!
Spielstand nach Runde 10:
Spieler 1 hat 3 Punkte und 0 Restkugeln
Spieler 2 hat 6 Punkte und 0 Restkugeln

Spieler 2 hat gewonnen!

Aufgabe 12:

Die **Russische Bauernmultiplikation** (auch *Ägyptisches Multiplizieren*, *Abessinische Bauernregel* oder *Verdopplungs-Halbierungs-Methode* genannt) ist ein einfaches Verfahren zur Multiplikation zweier natürlicher Zahlen. Am Beispiel $25 \times 13 = 325$ wird im Folgenden das Schema der russischen Bauernmultiplikation dargestellt:

Linke Seite	Rechte Seite	Dies addieren
25	13	13
12	26	-
6	52	-
3	104	104
1	208	208
		Ergebnis: 325

Der Algorithmus besteht aus folgenden Schritten (aus Wikipedia):

- Man schreibt die beiden zu multiplizierenden Zahlen nebeneinander.
- Auf der linken Seite werden die Zahlen jeweils halbiert (Reste abgerundet) und die Ergebnisse untereinander geschrieben, bis man zur 1 gelangt.
- Auf der rechten Seite werden die Zahlen verdoppelt und untereinander geschrieben.
- Die rechts stehenden (verdoppelten) Zahlen werden gestrichen, wenn die links stehende Zahl gerade ist.
- Die Summe der nicht gestrichenen rechts stehenden Zahlen ergibt das gesuchte Produkt.

Aufgabe: Implementieren Sie ein Java-Programm, das zwei Faktoren (positive Zahlen) einliest und deren Produkt nach dem Algorithmus der russischen Bauernmultiplikation berechnet und ausgibt.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Zahl 1 (> 0): <25>  
Zahl 2 (> 0): <13>  
25 * 13 = 325
```

Aufgabe 13:

Schreiben Sie ein Programm, das den Benutzer zunächst nach einer beliebigen Zahl fragen soll. Nach Eingabe der Zahl soll das Programm dann berechnen und ausgeben, ob die Zahl gerade oder ungerade ist.

Beispiel:

Eingabe: 78
Ausgabe: 78 ist eine gerade Zahl

Aufgabe 14:

Schreiben Sie ein Programm, das den Benutzer zunächst zur Eingabe einer beliebigen Zahl *zahl* auffordert. Das Programm soll anschließend alle geraden Zahlen zwischen 0 und *zahl* berechnen und ausgeben.

Beispiel:

Eingabe: 12
Ausgabe: 0 2 4 6 8 10 12

Aufgabe 15:

Schreiben Sie ein Programm, das den Benutzer zunächst zur Eingabe einer beliebigen Zahl *zahl* auffordert. Das Programm soll anschließend überprüfen, ob es sich um eine Zahl mit zwei Ziffern handelt und eine entsprechende Meldung auf den Bildschirm ausgeben.

Beispiel:

Eingabe: 12
Ausgabe: 12 besteht aus 2 Ziffern

Aufgabe 16:

Schreiben Sie ein Programm, das den Benutzer zunächst zur Eingabe zweier beliebiger Zahlen *zahl1* und *zahl2* auffordert. Das Programm soll anschließend die Summe, die Differenz, das Produkt, den Quotienten und den Divisionsrest der Zahlen *zahl1* und *zahl2* auf den Bildschirm ausgeben.

Beispiel:

Eingabe: 12 5
Ausgabe: 12 + 5 = 17
12 - 5 = 7
12 * 5 = 60
12 / 5 = 2
12 % 5 = 2

Aufgabe 17:

Schreiben Sie ein Programm, das den Benutzer zunächst zur Eingabe zweier beliebiger Zahlen *zahl1* und *zahl2* auffordert. Das Programm soll anschließend alle geraden Zahlen zwischen *zahl1* und *zahl2* berechnen und ausgeben.

Beispiel:

Eingabe: 11 und 20

Ausgabe: 12 14 16 18 20

Aufgabe 18:

Schreiben Sie ein Java-Programm, das nach Eingabe einer positiven Zahl *hoehe* eine Treppe der im folgenden Beispielprogramm skizzierten Form auf den Bildschirm ausgibt. Die Treppe soll aus *hoehe*-Zeilen bestehen.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Hoehe (>0): <5>
+
++
+++
++++
+++++
```

Aufgabe 19:

Stellen Sie sich folgende prinzipiell unendliche Zahlenfolge vor:

1 2 1 2 3 2 1 2 3 4 3 2 1 2 3 4 5 4 3 2 1 2 3 4 5 6 ...

Überlegen Sie: Was ist das Bildungsgesetz dieser Zahlenfolge?

Schreiben Sie ein Java-Programm, das zunächst eine Stoppzahl als int-Wert von der Konsole einliest. Anschließend sollen die einzelnen Zahlen der Zahlenfolge auf die Konsole ausgegeben werden, bis zum ersten Mal die Stoppzahl erreicht wurde (also im obigen Beispiel die 6).

Aufgabe 20:

Stellen Sie sich folgende prinzipiell unendliche Zahlenfolge vor:

1 2 3 2 3 4 5 3 4 5 6 7 4 5 6 7 8 ...

Überlegen Sie: Was ist das Bildungsgesetz dieser Zahlenfolge?

Schreiben Sie ein Java-Programm, das zunächst eine Stoppzahl als int-Wert von der Konsole einliest. Anschließend sollen die einzelnen Zahlen der Zahlenfolge auf die Konsole ausgegeben werden, bis zum ersten Mal die Stoppzahl erreicht wurde (also im obigen Beispiel die 8).