

Programmierkurs Java

Dr.-Ing. Dietrich Boles

Aufgaben zu UE 4 - Datentypen

(Stand 22.03.2019)

Aufgabe 1:

Schreiben Sie ein Programm zur Umrechnung von Celsius-Temperaturwerten in Fahrenheit-Temperaturwerte. Der Nutzer soll einen Celsius-Temperaturwert eingeben und das Programm den berechneten Fahrenheit-Temperaturwert ausgeben. Die zugrunde liegende Formel lautet

$$\text{Fahrenheit} = (\text{Celsius} * 9) / 5 + 32$$

Beispielablauf des Programms (Benutzereingaben in <>):

```
Celsius: <18>  
Fahrenheit = 64.4
```

Formen Sie die Formel um in

$$\text{Fahrenheit} = 9 / 5 * \text{Celsius} + 32$$

Was passiert? Worauf müssen Sie aufpassen?

Vergleichen Sie auch bspw. mit <http://de.weather.yahoo.com/temp.html>

Aufgabe 2:

Schreiben Sie ein Programm, das auf den Bildschirm eine Tabelle ausgibt, bei der zwischen -20 Grad Celsius und +30 Grad Celsius in Abständen von 1 Grad Celsius allen Celsius-Werten die entsprechenden Fahrenheitwerte gegenübergestellt werden.

Aufgabe 3:

Der *Body-Mass-Index (BMI)* ist eine Maßzahl für die Bewertung des Körpergewichts eines Menschen. Der BMI berechnet sich aus dem Körpergewicht [kg] dividiert durch das Quadrat der Körpergröße [m²]. Die Formel lautet: $\text{BMI} = \text{Körpergewicht} / (\text{Körpergröße in m})^2$. Die Einheit des BMI ist demnach kg/m². Implementieren Sie einen BMI-Rechner.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Bitte geben Sie Ihr Gewicht ein (in kg): <75>  
Bitte geben Sie ihre Größe ein (in m): <1.81>  
BMI = 22.89307408198773
```

Vergleichen Sie bspw. mit <http://www.sueddeutsche.de/app/gesundheit/bmirechner/>

Aufgabe 4:

Der "wünschenswerte" BMI hängt vom Alter ab. Folgende Tabelle zeigt BMI-Werte für verschiedene Altersgruppen (für die Aufgabe leicht abgeändert):

Alter	BMI
< 25 Jahre	19-24
25-64 Jahre	22-27
> 64 Jahre	24-29

Liegt der BMI unterhalb der angegebenen Werte hat die Person Untergewicht, darüber Übergewicht. Schreiben Sie ein Programm, das eine Person nach den erforderlichen Werten fragt und ausgibt, ob sie Unter-, Normal- oder Übergewicht hat.

Beispielablauf des Programms (Benutzereingaben in <>):

```
Bitte geben Sie Ihr Gewicht ein (in kg): <73>
Bitte geben Sie ihre Größe ein (in m): <1.81>
Bitte geben Sie ihr Alter ein (in Jahren): <45>
Sie haben Idealgewicht!
```

Aufgabe 5:

Liegt eine Amputation vor, so muss man vor der Berechnung des BMI das theoretische Körpergewicht berechnen und dieses anschließend in die Formel einsetzen. Hierfür werden folgende Werte herangezogen:

Hand: 0,8
Unterarm: 2,2
Oberarm: 3,5
Fuß: 1,8
Unterschenkel: 5,3
Oberschenkel: 11,6

Die theoretische Masse berechnet sich dann wie folgt:

$$\text{Theoretisches Gewicht} = (\text{Gewicht} * 100) / (100 - \text{Summe Korrekturwerte})$$

Ergänzen Sie den BMI-Rechner aus Aufgabe 3 um diese Korrekturformel.

Aufgabe 6:

(a) Berechnen Sie den double-Wert

$$(1 / 107751) * (1682xy^4 + 3x^3 + 29xy^2 - 2x^5 + 832)$$

für $x = 192119201$ und $y = 35675640$. Verwenden Sie dabei nur die Grundoperationen $+$, $-$, $*$ und $/$ und stellen Sie Ausdrücke wie x^2 bzw. x^4 als $x*x$ bzw. $x*x*x*x$ dar.

(b) Durch eine algebraische Umformung lässt sich eine äquivalente Darstellung für den Ausdruck finden:

$$(xy^2 / 107751) * (1682y^2 + 29) + (x^3 / 107751) * (3 - 2x^2) + (832 / 107751)$$

Vergleichen Sie das Ergebnis für die Auswertung dieser Darstellung mit dem zuvor berechneten. Der richtige Wert des Ausdrucks ist übrigens 1783.

Aufgabe 7:

Schreiben Sie ein Programm, das Sie auffordert, Namen und Alter einzugeben. Das Programm soll Sie danach mit Ihrem Namen begrüßen und Ihr Alter in Tagen ausgeben (ohne Berücksichtigung von Schaltjahren).

Aufgabe 8:

Welche Ausgabe erzeugt folgende Anweisung?

```
System.out.println("4 + 7 = " + 4 + 7);
```

Aufgabe 9:

Welche Ausgaben erzeugen folgende Anweisungen?

```
System.out.println(5.0 / 2.0 + 5 / 2);  
System.out.println(5.0 / 2.0 + 5 / 2.);  
System.out.println(2.3 + 7 / 2 * 4.1);
```

Aufgabe 10:

Sie wollen Konservendosen herstellen und müssen dazu bestimmte Werte der Dosen berechnen. Sei u der Umfang einer Dose in Zentimetern, h die Höhe einer Dose in Zentimetern und PI die Konstante mit dem Wert 3.141592. Dann berechnet sich

- Der Durchmesser des Dosenbodens $d_{\text{boden}} = u/PI$
- Die Fläche des Dosenbodens $f_{\text{boden}} = PI * (d_{\text{boden}}/2)^2$
- Die Mantelfläche der Dose $f_{\text{mantel}} = u * h$
- Die Gesamtfläche der Dose $f_{\text{gesamt}} = 2 * f_{\text{boden}} + f_{\text{mantel}}$
- Das Volumen der Dose $v = f_{\text{boden}} * h$

Schreiben Sie ein Programm, das nach Eingabe von u und h all diese Werte berechnet und auf den Bildschirm ausgibt.

Aufgabe 11:

Suchen Sie im Internet nach Formeln aus der Mathematik, Physik, Chemie, BWL, etc. und schreiben Sie Programme zur Berechnung der Formeln (z. B. <http://de.wikipedia.org/wiki/Formelsammlung>)

Aufgabe 12:

Entwickeln Sie ein Java-Programm, das zunächst zwei char-Zeichen (Kleinbuchstaben) einliest und anschließend die Differenz (positiver Wert!) bezüglich ihrer ASCII-Repräsentation inklusive einem passenden Begleittext ausgibt.

Beispiel 1: Eingabe: erstes Zeichen = 'd', zweites Zeichen = 'a',
Ausgabe: Differenz zwischen d und a ist 3

Beispiel 2: Eingabe: erstes Zeichen = 'a', zweites Zeichen = 'd',
Ausgabe: Differenz zwischen a und d ist 3

Aufgabe 13:

Programmieren Sie ein Ratespiel. Das Programm generiert anfangs eine `int`-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 die Funktion `Math.random`, die `double`-Zufallszahlen zwischen 0.0 (einschließlich) und 1.0 (ausschließlich) generiert:

```
double zufallszahl = Math.random();  
int z = (int)(zufallszahl * 101); // [0..100]
```

Beispiel für einen Programmablauf (Eingaben in <>):

```
Zahl: <50>  
Zufallszahl ist kleiner!  
Zahl: <25>  
Zufallszahl ist größer!  
Zahl: <37>  
Zufallszahl ist größer!  
Zahl: <45>  
Zufallszahl ist größer!  
Zahl: <48>  
Geschafft! Zufallszahl war 48  
Anzahl an Versuchen = 5
```

Aufgabe 14:

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

Aufgabe 15:

Für ein Geschenk haben 15 Schülerinnen 52,50 € gesammelt. Wie viel kommt zusammen, wenn n Schülerinnen sich nicht mehr beteiligen wollen? Schreiben Sie ein Programm, das die Summe berechnet und ausgibt.

Aufgabe 16:

In 3 Stunden legt ein Fahrzeug bei konstanter Geschwindigkeit 258 km zurück, wie weit kommt es in n Stunden? Schreiben Sie ein Programm, das die Anzahl an Stunden berechnet und ausgibt.

Aufgabe 17:

2 Kühe fressen zusammen an einem Tag 60 kg Gras. Wie viel kg Gras fressen n Kühe in m Stunden? Implementieren Sie ein Java-Programm, bei dem der Benutzer nach n und m gefragt und anschließend das Ergebnis berechnet und präsentiert wird.

Beispiel für einen Programmablauf (Eingaben in <>):

```
Anzahl Kuehe: <2>
Anzahl Stunden: <24>
In 24 Stunden fressen 2 Kuehe 60.0 kg Gras!
```

Aufgabe 18:

Entwickeln Sie ein Java-Programm zum Codieren von Kleinbuchstaben mit der Caesar-Verschlüsselung. Zunächst wird das zu codierende char-Zeichen und anschließend die Caesar-Verschiebung (Wert ≥ 0) eingelesen. Anschließend wird das codierte Zeichen ausgegeben.

Beispiel 1: Eingabe: 'a' und 2
Ausgabe: 'c'

Beispiel 2: Eingabe: 'z' und 4
Ausgabe: 'd'

Aufgabe 19:

Welche Ausgabe liefert das folgende Programm jeweils bei folgenden Eingaben: -1, 0 und 1.

```
int i = IO.readInt();
System.out.println(i<0 ? (i+1) : ((i==0 ? (i+2) : (i+3))+1));
```

Aufgabe 20:

Schreiben Sie ein Programm zur Umrechnung von DM- in Euro-Beträge und umgekehrt. Der Umrechnungsfaktor ist: 1 Euro = 1.95583 DM.

Aufgabe 21:

Initialisieren Sie eine double-Variable mit 20.0. Ziehen Sie dann in einer Schleife jeweils 0.2 ab und geben den jeweiligen Wert aus. Ab wann ergeben sich Rundungsfehler?

Aufgabe 22:

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

Aufgabe 23:

Schreiben Sie ein Programm, bei dem der Nutzer einen Buchstaben eingibt und das Programm ermittelt, an der wievielten Stelle des Alphabetes sich dieser Buchstabe befindet.

Aufgabe 24:

Schreiben Sie ein Programm, bei dem der Nutzer eine nicht-negative ganze Zahl eingibt und das Programm die Quersumme der Zahl berechnet.

Aufgabe 25:

Implementieren Sie ein Java-Programm, das den Benutzer zunächst auffordert einen Kleinbuchstaben einzugeben und das anschließend den entsprechenden Großbuchstaben auf den Bildschirm ausgibt. Für die Umwandlung ist der Einsatz einer if- oder switch-Anweisung und die explizite Verwendung von ASCII-Codes nicht erlaubt.

Beispiel für einen Programmablauf (Eingaben in <>):

```
Eingabe = ,h\  
Ausgabe = ,H'
```

Aufgabe 26:

Schreiben Sie ein Programm, das den Benutzer auffordert, einen nicht-negativen int-Wert einzugeben und das daraufhin den eingegebenen Wert in die entsprechende

Dualzahl umrechnet und auf die Konsole ausgibt. Hinweise zum Dualsystem siehe <http://de.wikipedia.org/wiki/Dualsystem>.

Die Umrechnung einer nicht-negativen Dezimal- in eine Dualzahl erfolgt so, dass die Dezimalzahl fortwährend bis zur 0 durch 2 dividiert wird. Die jeweiligen Reste der Division ergeben dann in umgekehrter Reihenfolge die Dualzahl. Tipp: Nutzen Sie zur Speicherung der Dualzahl einen String.

Beispiel: $19_{10} = 10011_2$

```
19 / 2 = 9 R 1
 9 / 2 = 4 R 1
 4 / 2 = 2 R 0
 2 / 2 = 1 R 0
 1 / 2 = 0 R 1
```

Aufgabe 27:

Schreiben Sie ein Programm, das den Benutzer zunächst auffordert, zwei Dualzahlen als int-Werte einzugeben. Sie können sich darauf verlassen, dass der Benutzer nur gültige Dualzahlen, also Zahlen mit den Ziffern 0 und 1 eingibt.

Anschließend soll das Programm die Summe der beiden Dualzahlen berechnen und ausgeben zwar durch ziffernweise Addition der beiden Zahlen von rechts nach links im Dualsystem. Dabei gilt bezüglich der Addition von Dualzahlen:

```
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 mit Übertrag 1.
```

Tipp: Hinweise zum Dualsystem siehe <http://de.wikipedia.org/wiki/Dualsystem>.

Nutzen Sie zur Speicherung der Dualzahl einen String.

Beispiel:

```
1011 zahl1
1011 zahl2
10110 uebertrag
-----
10110 Summe
```

Aufgabe 28:

Schreiben Sie ein Programm, das den Benutzer auffordert, einen nicht-negativen int-Wert *zahl* sowie einen Wert für ein Zahlensystem *system* einzugeben ($2 \leq \text{system} \leq 36$). Das Programm soll anschließend *zahl* in die entsprechende Repräsentation des Zahlensystem umrechnen und auf die Konsole ausgeben.

Die Umrechnung einer nicht-negativen Dezimal- in eine Zahl eines bestimmten Zahlensystems erfolgt so, dass die Dezimalzahl fortwährend bis zur 0 durch *system* dividiert wird. Die jeweiligen Reste der Division ergeben dann in umgekehrter Reihenfolge die entsprechende Repräsentation der Zahlen im Zahlensystem *system*. Dabei werden Reste kleiner 10 auf die entsprechenden Ziffern und Reste größer gleich 10 beginnend bei ‚A‘ auf entsprechende Großbuchstaben abgebildet. Tipp: Nutzen Sie zur Speicherung der Ergebniszahl einen String.

Betrachten wir als Beispiel das Hexadezimalsystem. Das ist das 16er-Zahlensystem mit den Divisionsresten 10 = A, 11 = B, 12 = C, 13 = D, 14 = E und 15 = F. Die Dezimalzahl 43 würde hier umgerechnet die Hexadezimalzahl 2B ergeben.

$$43 / 16 = 2 \text{ R } 11 \rightarrow \text{B}$$

$$2 / 16 = 0 \text{ R } 2$$

Aufgabe 29:

Schreiben Sie ein Java-Programm, das zunächst zwei positive 2-stellige int-Werte w1 und w2 von der Tastatur einliest (inkl. Überprüfung, dass es auch tatsächlich 2-stellige int-Werte sind), anschließend das Produkt der zwei Werte w1 und w2 berechnet und dann das Ergebnis auf der Konsole ausgibt. Die Produktberechnung soll dabei nach der folgenden vedischen Methode erfolgen:

- Berechne das Produkt p1 der ersten Ziffer von w1 und der ersten Ziffer von w2
- Berechne das Produkt px der ersten Ziffer von w1 und der zweiten Ziffer von w2
- Berechne das Produkt py der zweiten Ziffer von w1 und der ersten Ziffer von w2
- Berechne die Summe s von px und py.
- Berechne das Produkt p2 der zweiten Ziffer von w1 und der zweiten Ziffer von w2.
- Falls vorhanden streiche aus p2 die erste Ziffer und addiere sie zu s.
- Falls vorhanden streiche aus dem neuen s die erste Ziffer und addieren sie zu p1.
- P1 und s und p2 hintereinandergeschrieben bilden das Ergebnis e ($e = 100 * p1 + 10 * s + p2$)

Beispiel 1: $56 * 23 = 1288$

```

5      6
2      3
-----
p1  s  p2
10 27 18 (27 = 5*3 + 2*6)
10 28  8 (28 = 27 + 1 (Übertrag))
12  8  8 (12 = 10 + 2 (Übertrag))

```

Beispiel 2: $73 * 74 = 5402$

```

7      3

```



```

7      4
-----
p1  s  p2
49 49 12 (49 = 7*4 + 7*3)
49 50  2 (50 = 49 + 1 (Übertrag))
54  0  2 (54 = 49 + 5 (Übertrag))

```

Aufgabe 30:

Um eine positive dreistellige Zahl mit 11 zu multiplizieren, setze die Summe sum1 des ersten Ziffernpaares und die Summe sum2 des zweiten Ziffernpaares zwischen die erste und die letzte Ziffer.

Beispiel 1: $234 * 11 = ?$

sum1: $2 + 3 = 5$

sum2: $3 + 4 = 7$

$234 * 11 = 2574$

Aber beachte: Falls sum2 größer als 9 ist, nehme nur ihre letzte Ziffer und addiere die erste Ziffer zu sum1. Falls dann sum1 größer als 9 ist, nehme auch hier nur ihre letzte Ziffer und addiere die erste Ziffer zur ersten Ziffer des Produktes.

Beispiel 2: $987 * 11 = ?$

sum2: $8 + 7 = 15$ (nehme 5, addiere 1 zu sum1)

sum1: $9 + 8 + 1 = 18$ (nehme 8, addiere 1 zur ersten Ziffer)

Erste Ziffer: $9 + 1 = 10$

$987 * 11 = 10857$

Schreiben Sie ein Java-Programm, das zunächst einen positiven 3-stelligen int-Wert value von der Tastatur einliest (inkl. Überprüfung, dass es auch tatsächlich ein 3-stelliger int-Wert ist), anschließend das Produkt von value mit 11 nach dem obigen Algorithmus berechnet und dann das Ergebnis auf der Konsole ausgibt.

Achtung: Sie dürfen für Ihre Berechnungen nur den Datentyp int, zu keiner Zeit aber den Datentyp String benutzen!